Preface

Dr. András Prékopa, full member of the Hungarian Academy of Sciences, Professor Emeritus of the Operations Research Department of Eötvös Loránd University of Sciences and of Rutgers Center of Operations Research, Széchenyi prize winner Hungarian mathematician, passed away in Budapest at the age of 87 on the 18th of September 2016. In addition to being an internationally leading researcher he is truly the Father of operations research (OR) in Hungary. For his major achievements and contributions to the science of OR he was also awarded "Honorary Doctorate" by the Óbuda University. This volume of Acta Polytechnica Hungarica (APH) is devoted to his memory.

Originally, the idea of the APH volume came from András Bakó the founding editor of the journal. He was assigned the position of editor-in-chief of this volume with two co-editors István Maros and Tamás Szántai. While the work was well in progress András Bakó suddenly and unexpectedly passed away. The two co-editors took over the job and did the rest of the work. Herewith we pay tribute to András Bakó for his unforgettable personality, for his original thinking, his contributions to the theory and practice of operations research and also for his care for others.

The volume contains 13 accepted papers that have undergone the strict editorial procedure of APH.

Bakó and Gáspár summarize the development procedure of the optimal maintenance and rehabilitation strategies (models) of roads and bridges in Hungary. In these models the deterioration depending on time and other parameters is given by Markov transition probability matrices. The paper presents the development phases of models concentrating not only on economic aspects but also environmental (sustainability) ones, as well. The Hungarian multi-periodical PMS model was one of the very first models applying total optimization over a ten years' time horizon.

Smidla and Maros study the possibility of improving the accuracy of certain additive floating point operations, especially that of the vector-vector addition and dot-product operations. In stabilized dot product calculations it is customary to use branching which makes it a bottleneck in parallel computations The authors define the „safe add" operation and show how it can be implemented on modern SIMD architecture. They show that the operations can be executed without loss of time and the increased accuracy dot products can be computed without branching. The summarized results of a computational study are also presented.

Böröcz, Tar and Maros perform a thorough comparison of vector operations of open-source linear optimization kernels: CLP, GLPK and Pannon Optimizer (PO). Such kernels are responsible for the speed and accuracy of most optimization algorithms. In PO they introduce a new data type for sparse computing called indexed dense vectors and point out its beneficial properties that make PO more than competitive among the investigated kernels.

Szántai, on the basis of an earlier Hungarian language paper by Prékopa, Szántai and Zsuffa, investigates the water streamflow on probability theoretical bases. It is shown that under some realistic conditions its probability distribution is of gamma type. Then the optimal capacity of a storage reservoir is determined. In a second model optimal water release policy is sought, given that water demands should be met by a prescribed large probability. Finally, in addition to the aforementioned reliability type constraint an upper bound is imposed on the number of days when demands may not be met and the cost of the intake facility is to be minimized.

Szántai, Kovács and Egri are dealing with forecasting of the demand during a sales period. They present two dynamic methodologies for calculating the quantity which has to be placed on the shelves at the beginning of each day such that some constraints expressing lower and upper bounds on the quantities are kept. Both methodologies are new to this field and are useful because of some specific properties of the problem. The new methods use historical data of the demands in previous promotions and the consumptions registered in the previous days. Since the promotion period is relatively short, other methods such as time series analysis can hardly be used.

Fábián, Csizmás, Drenyovszki, van Ackooij, Vajnai, Kovács and Szántai propose a new algorithm for probability maximization under linear constraints by inner approximation. The proposed algorithm has the advantage that it can easily be implemented and is immune to possible noises in gradient computation. They prepared a simple implementation of the proposed new algorithm and show that it is quite reliable and robust.

Illés and Lovrics present a new computational method for the linearly constrained convex multi-objective optimization (LCCMO) problem. They propose some techniques for finding joint decreasing directions for both the unconstrained and the linearly constrained case. Utilizing these techniques, they introduce a method using a subdivision technique to approximate the whole Pareto optimal set of the LCCMO problem.

Izsák and Szeidl are dealing with species abundance models. They suppose the process describing the entering time points of the new species in the system to be Poisson process. In earlier papers the Poisson process was supposed to be homogeneous when the logarithmic distribution played important role in description of the model parameters. In the present paper the authors showed that in the case of inhomogeneous Poisson process the Yule distribution takes over the role of the logarithmic distribution.

Bánhelyi, Csendes, Krisztin and Neumaier give an elementary derivation of a bounding scheme to prove Wright's conjecture on the delay differential equation. Then the elaborated bounding scheme can be applied in a verified computational algorithm for systematic checking the parameter value α in the delay differential equation. Earlier the authors worked out a simpler technique for doing this. By applying this it was possible to prove the truth of Wright's conjecture for $\alpha \in [1.5, 1.5706]$. The main goal of the present paper is to improve the upper limit point of this interval. By applying the described bounding technique the authors continued the computational part of the proof with unchanged theoretical background and they were able to increase the upper limit of the interval up to 1.57065. However the time of computation for this was more than 466 hours (almost 20 days) on a quite strong PC configuration. So the authors conclude that additional theoretical insight should be utilized to achieve a substantial progress in the proven α values.

Abaffy and Galántai present a bisection type global optimization algorithm for continuous real functions over a rectangle. The suggested method combines the branch and bound technique with an always convergent solver of underdetermined nonlinear equations. The paper concludes with a detailed numerical testing of the algorithm.

Dombi, Jónás and Tóth elaborate a new probability distribution which they call epsilon probability distribution. First they introduce the concept of the *n*-th order epsilon differential equation then show that the solution of the 0-th order epsilon differential equation is the exponential function. They solve the 1-st order epsilon differential equation and its solution, which is a power function, they call epsilon function. As an interesting fact they show that this function is in a strong connection with the Dombi operators in continuous logic. Using this new function a new probability distribution is constructed which is called the epsilon probability distribution. It is proved that the epsilon probability distribution is asymptotically equivalent to the exponential probability distribution. The hazard function of the new epsilon probability distribution is determined and its advantages are shown in a practical example.

London, Gera and Bánhelyi examine Markowitz portfolio selection using various estimators of expected returns and filtering techniques for correlation matrices. They use several methods to estimate expected returns. The authors conclude that the James-Stein estimator improves the reliability of the portfolio. It means that the realized risk is closer to the estimated risk in the investigated case.

Fullér, Harmati and Várlaki summarize the measures of dependence between possibility distributions known in the literature. One of them is the measure of possibilistic correlation between marginal possibility distributions of a joint possibility distribution what can be defined as the weighted average of the probabilistic correlations between marginal probability distributions whose joint probability distribution is defined as uniform distribution on the level sets of their joint possibility distribution. Using the averaging technique they discuss three quantities (correlation coefficient, correlation ratio and informational coefficient of correlation) which are used to measure the strength of dependence between two possibility distributions. They also discuss the cases when the level sets of joint possibility distribution are equipped with non-uniform probability distributions.

Budapest, January 2018

István Maros
Tamás Szántai

Development of a Sustainable Optimization Model for the Rehabilitation of Transport Infrastructure

András Bakó[1], **László Gáspár**[2]

[1] Óbuda University, Bécsi út 96/b, 1034 Budapest, Hungary, bako@uni-obuda.hu

[2] Széchenyi University, Egyetem tér 1, 9023 Győr, Hungary, gaspart@sze.hu

Abstract: About twenty years ago, the research activities aiming at the development of the optimal maintenance and rehabilitation strategies (models) of roads and bridges started in several countries, including Hungary. In the first foreign models, the deterioration depending on time and other parameters was given by Markov transition probability matrices. Due to the inaccuracies and inconsistencies of earlier models, a continuous model upgrading could have been carried out by many researchers world-wide. Besides, basically new models appeared in the literature, which are able to describe the actual processes more reliably. The research work of the authors of the paper has concentrated on Pavement Management Systems (PMSs) and Bridge Management Systems (BMSs). Since a common financing of roads and bridges is typical, a combined model of road pavement and bridge managements was developed by the authors increasing considerably the efficient use of available funds.

Keywords: Pavement Management; Bridge Management; Markov deterioration model; maintenance-rehabilitation and operation cost distribution (allocation)

1 Introduction

It was more than two decades ago that a systematic management modelling of transport infrastructure started in Hungary with the collaboration of experts in various fields (transport engineers, mathematicians, economists, meteorologists, etc.). The original goal was to develop cost-efficient systems for development, rehabilitation, maintenance and operation activities in the area. These models can provide effective tools for infrastructure (mainly road) managers to minimize their expenditures if given preconditions are fulfilled. A part of activities was the adaptation of various systems available and used in foreign countries; however, several of these are models based on Hungarian data sets, usually data time series, but every case, the procedure followed was the creation of the first version of a system (model), and long-term monitoring of its operation, then, based on the experiences gained during the monitoring, a new, updated model version is developed.

This paper presents the development phases of such a model that concentrates not only on economic aspects but also environmental (sustainability) ones, as well. (The importance of the problem can be highlighted by the fact that the net value of Hungarian public highway network – some 7,000 billion HUF = 28 billion EUR – exceeds 38% of the Hungarian national wealth). First of all, some basic information on the Road Asset Management System is presented. The main steps of this development process done in Hungary are: single stage network level optimization model, multi-stage model, combined pavement/bridge model and model with climate-dependent parameters.

One of the main development achievements related to multi-periodical model was the total optimization. Since the models available optimize various elements of the model separately. Because of the large size of the model, optimization algorithm was applied. Usually it is not true that the sum of the results of partial optimizations is equal to total optimum. (E.g. all separate elements are optimized in a single model). For this purpose, the optimization model developed in Hungary is more appropriate than the packages developed and traded by various professional software houses. Another significant novelty is the development of combined pavement and bridge management systems. The main advantage of the use of this system comes from the fact that usually the same fund (budget) is used for the management (construction, maintenance, rehabilitation, operation) of both infrastructure elements. The third important innovation is connected with the consideration of the effects of climate change in the long-range model development. The fourth significant research outcome is the inclusion of a parameter related to the change in traffic characteristics in the pavement deterioration model. Then a new algorithm based on the results coming from PMS/BMS model had been developed. This algorithm distributes optimally the available road-bridge funds among the regions (counties). Finally another algorithm has been created for funds distribution in the case of insufficiency of available financial means.

2 Asset Management System

The development, the maintenance and the operation of the high-valued road network can be considered as an extremely important task of the whole country needing a lot of money, human resources, machinery, materials, etc. Several subsystems were developed and being used all over the world to solve the problem mentioned and to allocate economically the necessary resources. However, this task is rather complex and the sum of the best solutions of various subsystems are not identical with the optimum of the operation of the whole system. That is why intensive research activities started in the topic some 20 years ago. It is called

Asset Management System for Road Sector or Road Asset Management System. One of the most significant relevant basic research institutions is US Department of Transport Federal Highway Administration, Office of Asset Management (Asset [2]). Another important effort in this field has been done by an OECD Committee (Asset [3]). There are many definitions for this kind of asset management but each of them refers to a management system, a DSS (Decision Supporting System) and the cost efficiency on road construction, maintenance and operation, besides the model system has both long-term, strategic and short-term, actual elements (What [28]). This case, the term "asset" includes not only its actual gross or net value but also the funds needed for its maintenance throughout service life. The potential users of asset management include decision makers, road users, road proprietors, operators, etc. The Road Asset Management System has several components (Hudson et al. [25]):

- Road pavements
- Pavement structures and connected elements
- Bridges
- Tunnels
- Culverts
- Traffic engineering facilities (traffic signs, road paintings, road lighting)
- Traffic census facilities
- Information and monitoring systems
- Road construction, maintenance and operation machinery
- Road vehicles
- Parking and rest areas
- Roadside building connected with road rehabilitation, maintenance and operation
- Materials used and equipment for their production
- Organisations in the field
- Road staff

The following subsystems are necessary for a working asset management:

- *Information Management Subsystem* collects, systematizes, appraises and archives the basic data of modelling. It utilizes the knowledge on data need, data bases and their operation, archiving, hardware and software need, etc.
- *Assets Valuation Subsystem* deals with a highly important group of basic data needed for the effective operation of the model. It includes also the methodology of the collection and evaluation of technical data, as well as, the maintenance of the system.

- *Condition Evaluation and Performance Modelling Subsystem* concentrates on the actual condition of system elements and the modelling of their expected performance. The subsystem includes the condition parameters of each element, the scaling of the measurement range of condition parameters, as well as, the data storage in close connection with the activity of other subsystems.
- *Deterioration Modelling and Defect Analysis Subsystem* forecasts the worsening of the condition of various system elements, identifies the probable (expected) defect types. The condition of an element can be characterized by various qualifying parameters or a combined index; the deterioration curves are set accordingly.
- *Maintenance, Operation, Rehabilitation and Reconstruction Subsystem* defines the types and the costs of various intervention techniques. It is a very important supporting element for the establishment of the decision strategies.
- *Whole Life Cost and Benefit Subsystem* also has a significant supporting role for the decision process. Here, among others, the discounted values, the inflation rate, the interest rates are taken into consideration.
- *Decision Supporting Models Subsystem* determines the use and the applicability of the whole system. Since there are a high number of elements in a system, a complex model creating total optimum for strategic decisions should be extremely aggregated. So, expert models, methods using basis approach, optimization models can be applied here. The already existing system elements (PMS, BMS, systematic condition survey, etc.) should be also included into the system.
- *Total Quality Management Subsystem* is operational during the whole implementation period of the program. It provides the results and the performance efficiency of the intervention at the end of the period. After feedback, new strategic and tactical objectives are set. When their parameters are set, the whole decision process can be restarted.

Over 20 years ago the systematic management modelling of transport infrastructure started in Hungary with the collaboration of experts in various fields (transport engineers, mathematicians, economists, meteorologists, etc.). The original goal was to develop cost-efficient systems for development, rehabilitation, maintenance and operation activities in the area. These models can provide effective tools for infrastructure (mainly road) managers to minimize their expenditures if given preconditions are fulfilled.

This paper presents the development phases of such a model that concentrates not only on economic aspects but also environmental (sustainability) ones, as well. (Again, the importance of the problem can be highlighted by the fact that the net value of Hungarian public highway network – some 7,000 billion HUF = 28 billion EUR – exceeds 38% of the Hungarian national wealth). The main steps of this development process are: single stage network level optimization model, multi-stage model, combined pavement/bridge model, model with climate-dependent parameters, model with traffic-dependent parameters, towards asset management.

3 Single Stage Network Level Optimization Model

The development of the first Hungarian network level pavement management system was preceded by the creation of an effective, large-scale road data bank (Bakó et al. [4]). It was decided to deal with network level pavement management models before project level ones since the former variants need less previous information on the roads concerned (Bakó et al. [11]). The main aim of a network level management model is to identify the most advantageous maintenance techniques for every road subset with the same surface type, same condition parameters and same traffic category. This type if model is a budget planning tool capable of estimating the total lengths and costs of works required on the network for pavement rehabilitation, resurfacing and routine maintenance. A financial planning type is generally connected with the determination of the funding level needed to maintain the "health" (integrity) of the pavement network at a desirable level. In case of another model type, the available budget is known and the maintenance strategy has to be determined that fulfil the required constraint of pavement conditions, and optimize the total benefit of society (Gáspár et al. [18, 22]).

The first single-stage network level optimization model (MPMS) was developed in Hungary in the late 1980s (Bakó [5], Gáspár [17]). The Hungarian road administration needed quick and practical results which could not be provided by the "too simple" MPMS. That is why the Finnish HIPS model (Männistö [27]) was chosen, because there were already available several-year experiences. The new version, the so-called HUPMS-model was developed using the optimization procedure of MPMS and the model structure of HIPS.

The main features of this model are:

- Several (a maximum of 10) time periods (stages)
- 2 pavement types (asphalt concrete and asphalt macadam)
- 3 traffic categories
- 4 condition parameters (unevenness, bearing capacity, rut depth, surface defects)
- Combined target function
- Max. 8 intervention (rehabilitation) types

In the long-term model, the optimum solution is sought for the distribution of pavement condition in the road network which can be attained after the optimum interventions; it is called the Markov-stable condition. The target function is the minimum of the sum of agency and user costs (i.e. social total optimum).

Possible interventions (rehabilitation strategies) for asphalt concrete roads are: routine maintenance, patching, rut repair, surface dressing, laying thin asphalt course, asphalt overlay, and reconstruction. The interventions for asphalt macadam roads are: routine maintenance, patching, surface dressing, road profile repair, asphalt overlay, reconstruction.

The Markov transition probability matrix for pavement type i, traffic category j, and intervention type k is designated by Q_{ijk}. The matrix size amounts to 135x135, since the total possible number of relevant parameters is 3x3x3x5=135. The number of Markov matrices is 2x3x8=48; thus the number of columns in the model amount to: 48x135=6480.

The unknown vector of pavement type i, traffic category j, and intervention type k should be X_{ijk}, which shows the proportion of road link lengths in 135 condition states for a given i, j, k. The number of vectors is 48, and so the total number of unknown factors reaches 6480.

The unit intervention costs vector for pavement type i, traffic category j, and intervention type k should be C_{ijk}. The road user cost function for pavement type i and traffic category j is designated by K_{ij}. First, the Markov-stable model was formulated.

When the notation above are applied, the model is as follows. Determine the unknown vector series X_{ijk} is sought, which fulfils the Markov stable condition

$$\sum_{i=1}^{2}\sum_{j=1}^{3}\sum_{k=1}^{8}\left(Q_{ijk} - E\ \right)X_{ijk} = 0 \tag{1}$$

and minimises the weighted sum of agency (intervention) and user costs:

$$C = \alpha\sum_{i=1}^{2}\sum_{j=1}^{3}\sum_{k=1}^{8} X_{ijk}C_{ijk} + \beta\sum_{i=1}^{2}\sum_{j=1}^{3}\sum_{k=1}^{8} X_{ijk}K_{ij} \longrightarrow MIN \tag{2}$$

where E unit matrix of size 135x135,

α weighting factor for intervention costs,

β weighting factor for user costs.

Further conditions limiting the amount of intervention costs can be supplied for the model. This case, the Markov stable solution is looked for which fulfils all conditions considered.

4 Multi-Time Period Model

The multi time period, (briefly multiperiod) version of the PMS, was created in 1991 (Csicselyné [13]; Gáspár [19]; Bakó [8]). One of the objectives is to reach a stable model result by means of an approximation over a period of several years. The number of time periods is generally 10, and the model gives the necessary interventions in each period. Let us denote by Y_{ijt} the proportion of the length of the road sections of pavement type i and traffic category j after the interventions carried out during the year t, while b_{ij} is the proportion of the length of the road sections of pavement type i and traffic category j or, initially, at the beginning of the planning period.

This case, the unknown vector has a further index t. Let us denote the unknown vector by X_{ijkt} that belongs to the time period t.

The first mandatory condition is connected with the distribution of pavement condition states during the initial years:

$$\sum_{k=1}^{8} EX_{ijk1} = b_{ij} \quad ,i = 1,2,\ldots,s \; j = 1,2,\ldots,f \tag{3}$$

k=1

where E unit matrix of size 135x135.

The following condition supplies the proportion of road link lengths for the end of the first planning year. So, the proportions of length vectors Y_{ij1} at the end of the first planning year are determined by the following relation:

$$\sum_{k=1}^{8} EX_{ijk1} = b_{ij} \quad \text{, i=1,.2, j=1,2,3} \tag{4}$$

$$\sum_{k=1}^{8} Q_{ijk} X_{ijk1} = y_{ij1} \quad \text{, i=1,.2, j=1,2,3} \tag{5}$$

The following mandatory conditions refer to the later years:

$$\sum_{i=1}^{2} EX_{ijk(t+1)} - Y_{ijt} = 0, \quad \text{j=1, 2, 3, k=1, 2,…8, t=1, 2,…9} \tag{6}$$

This condition means that the proportion of length Y_{ijt} at the end of time period t provides a value for the initial distribution for the period (t+1) that is it is equal to $X_{ijk\,(t+1)}$.

A mandatory boundary is the cost limit, where the total intervention costs can be given for a year or for the planning period. The yearly intervention cost limit is as follows:

$$\sum_{i=1}^{2}\sum_{j=1}^{3}\sum_{k=1}^{8} r^{(l-1)}C_{ijk}X_{ijkt} \le r^{(t-1)}M \text{ , t=1, 2, ...10} \tag{7}$$

where r the discount factor,

M the intervention cost available annually.

The target pavement condition distribution at the end of the planning period can also be specified:

$$\sum_{i=1}^{2}\sum_{j=1}^{3}(Y_{ijT})_v \ge \delta_1 \sum_{i=1}^{2}\sum_{j=1}^{3}(b_{ij})_v,\ v \in \text{G}$$

$$\sum_{i=1}^{2}\sum_{j=1}^{3}(Y_{ijT})_v \le \delta_2 \sum_{i=1}^{2}\sum_{j=1}^{3}(b_{ijv})_v,\ v \in \text{B} \tag{8}$$

$$(\underline{b}_E)_v \le \sum_{i=1}^{2}\sum_{j=1}^{3}(Y_{ijT})_v \le (\bar{b}_E)_v,\ v \in \text{E}$$

where T the number of the planning periods,

G, B, E three sets the pairwise intersection of which is 0, and the sum of these sets is the set of the road segment,

G the set of the road segments which are in good condition,

B set of the road segments which are in bad conditions,

E the set of the road segments which ere in average conditions,

$\underline{b}_E$ the lower bound vector of the other road segment group,

$\bar{b}_E$ the upper bound vector of the other road segment group,

δ_1 and δ_2 constants.

In this case, a combined target function was selected which can be considered as the weighted average of the intervention cost and the user cost target function types:

$$C = \alpha\sum_{i=1}^{2}\sum_{j=1}^{3}\sum_{k=1}^{8}\sum_{t=1}^{10} X_{ijkt}C_{ijk} + \beta\sum_{i=1}^{2}\sum_{j=1}^{3}\sum_{t=1}^{9} Y_{ijt}K_{ij} \longrightarrow MIN\ ! \tag{9}$$

If $\alpha = 0$, only the user costs are considered in the target function, while in the case of $\beta = 0$, only the intervention costs are.

In such a way, these cost function types can be arbitrarily weighted by varying both constants.

5 Combined Pavement/Bridge Management

In the majority of countries – including Hungary – the PMS (Pavement Management System) and BMS (Bridge Management System) operate independently. However, their interdependence is obvious since the bridge surfacing constitutes part of the road pavement. Very often their financial sources are also identical (e.g. Road Funds) contributing to the need for more or less common management. Both PMS and BMS apply the same concept and application of system technology and require a system output function that can be optimised in relation to the benefits and costs.

Several models can be used for solving the BMS problem. It can be a mathematical programming (linear, dynamic, nonlinear, integer, etc.) model. It could be a stochastic model or a fuzzy approach. In all models, the most important and difficult problem is to develop a proper deterioration model.

In Hungary, both the adapted PONTIS-H Bridge Management System and the HIPS-HUPMS network level Pavement Management System are based on the use of Markov transition probability matrices (Bakó et al., [10]). As a result, their identical structures allow the joint optimisation of both systems. This activity is especially important when the aim is the distribution of the funds available between the two infrastructure elements (road pavements and bridges).

The mathematical-engineering model of this BMS-PMS (PBMS) a common model has already been completed. Its implementation is planned for the near future.

As mentioned above, the deterioration sub-model of the Hungarian network level PMS (HUPMS) utilises Markov transition probability matrices. The bridge management model, the PONTIS, also uses them. However, a combined pavement-bridge management model cannot be developed using them because their module structures are different. That is why the mathematical model (PBMS-model) of the network-level pavement-bridge management has been developed which optimizes in a single model.

The structure of this model is presented in Figure 1. It has two columns. The first column (P1 and P2) contains the elements of the PMS model introduced earlier when discussing HUPMS (see conditions set out in Eqs. 3-5, 7).

In the right-hand column the relevant BMS conditions (Golabi et al [23]; Agárdy et al [1]) can be seen; the yearly cost boundary for the Bridge Management System is as follows:

$$\sum_{d=1}^{D}\sum_{e=1}^{E}\sum_{f=1}^{F}\sum_{g=1}^{G} r^{(t-1)} V_{defg} H_{defg} \leq r^{(t-1)}\, B \qquad t = 1,2,...T \tag{10}$$

where	V_{defgt}	intervention costs
	H_{defg}	user costs
	r	discount factor
	B	yearly cost boundary
	d	bridge span
	e	bridge element
	f	level of exposure
	g	intervention type

The object is to define a vector series which fulfils the conditions defined, and minimises the weighted sum of the intervention and user costs, that is:

$$\gamma$$

$$\sum_{d=1}^{D}\sum_{e=1}^{E}\sum_{f=1}^{F}\sum_{g=1}^{G}\sum_{t=1}^{T} r^{(t-1)} V_{defg} H_{defg} + \mu \sum_{d=1}^{D}\sum_{e=1}^{E}\sum_{f=1}^{F}\sum_{tg=1}^{T} r^{(t-1)} W_{deft} I_{def} \longrightarrow MIN \tag{11}$$

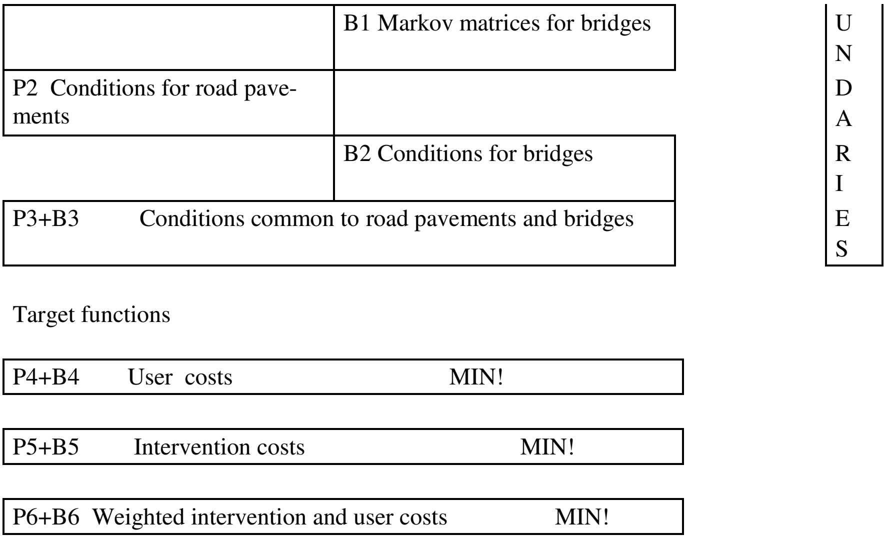
where the elements of I_{def} related to user costs are different from 0.

The PBMS model can also have common conditions, for example relating to the annual sum which is commonly available, that is, the sum of the conditions set out in Eqs. 6 and 9:

$$\sum_{ijk} r^{(t+1)} C_{ijkt}\, X_{ijkt} + \sum_{defg} r^{(t+1)} V_{defgt} H_{defgt} \leq r^{(t-1)} (M + B), \qquad t = 1,2,...T \tag{12}$$

As target function, the sum of the object functions of pavement and bridge models is taken. The object can be here the minimisation of the intervention costs (P4 + B4), the minimisation of user costs (P5+B5) or the weighted sum of these costs when none of the weighting factors is equal to 0 (P6+B6). By varying the parameters, any arbitrary combination of the target function can be produced. For example, the minimisation of the sum of road (pavement) user costs and bridge intervention costs.

P1 Markov matrices for road pavements	B O



B1 Markov matrices for bridges

P2 Conditions for road pavements

B2 Conditions for bridges

P3+B3 Conditions common to road pavements and bridges

U N D A R I E S

Target functions

P4+B4 User costs MIN!

P5+B5 Intervention costs MIN!

P6+B6 Weighted intervention and user costs MIN!

Figure 1
Combined model of PMS and BMS

6 Consideration of Climate Change

Typically, road asset management models usually do not consider environmental load (connected with climate change consequences) (Gáspár et al. [20]).

In case of long-term, multi-time period models, two approaches could be:

A) Environmental effects forecasted for the whole planning period, M+R actions are calculated accordingly,

B) Following forecasting in model A, environmental consequences are calculated after each time period resulting in an input of next time period (more accurate results)

$$\sum_{s=1}^{S}\sum_{f=1}^{F}\sum_{p=1}^{P}\left|\overline{\mathbf{X}}_{\mathbf{sfp}}\mathbf{C}_{\mathbf{sfp}}-\mathbf{X}_{\mathbf{sfpt}}\mathbf{C}_{\mathbf{sfp}}\right|\to \min,\quad t=1,\ldots,T \tag{13}$$

As a next step, the target function is linearized. The two artificial variables are denoted by $\mathbf{u}_{ijkt}$ and $\mathbf{v}_{ijkt}$. Besides the following equation has to be met:

$$\begin{gathered}\mathbf{u}_{\mathbf{sfpt}}-\mathbf{v}_{\mathbf{sfpt}}=\overline{\mathbf{X}}_{\mathbf{sfp}}\mathbf{C}_{\mathbf{sfp}}-\mathbf{X}_{\mathbf{sfpt}}\mathbf{C}_{\mathbf{sfp}},\ \text{ahol}\ \mathbf{u}_{\mathbf{sfpt}}\geq\mathbf{0}\ \text{és}\ \mathbf{v}_{\mathbf{sfpt}}\geq\mathbf{0},\\ s=1,\ldots,S,\quad f=1,\ldots,F,\quad p=1,\ldots,P,\quad t=1,\ldots,T.\end{gathered} \qquad (14),$$

and the new objective function in this case is

$$\sum_{s=1}^{S}\sum_{f=1}^{F}\sum_{p=1}^{P}(\mathbf{u}_{\mathbf{sfpt}}+\mathbf{v}_{\mathbf{sfpt}})\rightarrow \min,\quad t=1,\ldots,T \tag{15}$$

Just one of the coordinates $\mathbf{u}_{\mathbf{sfpt}}$ and $\mathbf{v}_{\mathbf{sfpt}}$, can be different to 0 any time. The remaining steps are identical to the ones mentioned before. The conditions (constraints) in Eq. 13 change in every planning time period t.

Besides, two new Bridge Management models were also developed. The above mentioned PONTIS and its Hungarian version seemed to be rather far from the real processes. The new models could handle the deterioration process of bridge elements more realistically.

7 Some Related Models

Some other related management models were also developed that are presented briefly.

7.1 Model for Funds Distribution

One of the outputs of the network level HUPMS is the "optimal" distribution of available highway funds for country-wide links of Hungarian public road network. The next necessary step is the continuation of funds distribution (allocation), among others, to the road network of various counties and the motorway network. To solve this problem, a computerized model was developed (Bakó [6]; Bakó [7]) with the following features.

First, the so-called expenditure groups (e.g. patching, grass mowing, bridge management, overhead of road management organisations, etc.) were identified. The task is to distribute "optimally" the available highway funds among these expenditure groups. The expenditure groups are denoted by "i", their number is "I". The running index of road management units (e.g. County Highway Directorate) should be "j", while their number amounts to "J". The task is to determine an unknown $X = (x_{ij})$ matrix an element of which is x_{ij}, the sum coming from the funds "i" and destined to the road management unit "j". Denote the sum available for the expenditure group "i" by "b_i". This sum can be determined either by the actual needs or by the so-called basis allocation in the previous year or, eventually, using another methodology.

The sums to be allocated to a road management unit is influenced by its special quantitative parameters, like total length of the road network managed, traffic amount, number of traffic signs, etc. These qualitative parameters are usually proportional with the works to be done and the related sum of money. The first qualitative parameter of the expenditure group "i" and road management unit "j" is designated by $z_{ij}^{(1)}$, the second one by $z_{ij}^{(2)}$ and the kth one by $z_{ij}^{(k)}$. For the sake of simplicity, a special methodology was used for the calculation of a characteristic rate of the road management unit in order to apply a single qualitative parameter for a unit.

In addition to the qualitative parameters, unit costs were also given for each expenditure group and road management unit. These unit costs can be the same for each road management unit (e.g. road pavement condition evaluation), but they can be different for various management units as a function of their location, natural features, etc. The unit cost of the expenditure group "i" and road management unit "j" should be denoted by e_{ij}. It is supposed that the benefit of the activity in question for the expenditure group "i" and road management unit "j" amounts to h_{ij} for 1 HUF expenditure. The task is to perform the optimal distribution of the funds available. There are several solution methodologies (optimization procedures resulting linear programming tasks, heuristic methods, simulation, expert system, etc.) depending on the targets set, the amount of inputs available and some other parameters.

The linear programming model distributes (allocates) optimally the funds available to the expenditure groups when also the benefits are known. One of the conditions for the use of the model is that, in case of a fixed expenditure group, the total funds allocated by this title for the road management units should reach b_i that is destined for the expenditure group:

$$\sum_{j=1}^{J} x_{ij} \geq b_i, \quad \mathrm{i} = 1, 2, \ldots, \mathrm{I} \tag{15}$$

Another constraint is the individual lower limit for each variable

$$x_{ij} \geq \frac{b_i \cdot z_{ij}}{\sum_{k=1}^{J} x_{ik}} \quad \begin{matrix} i = 1, 2, \ldots, I \\ j = 1, 2, \ldots, J \end{matrix} \tag{16}$$

The target function should be the maximization of the benefit coming from the maintenance-rehabilitation action. Since neither x_{ij}, nor k_{ij} are positive values, the task would be unlimited. That is why an additional limiting condition is defined for which the sum K is needed, the total financial means available for highway purposes. It is supposed that the following relationship between the limit given for an expenditure group and the sum K is valid:

$$\sum_{i=1}^{I} b_i \leq K \tag{17}$$

If this relationship is not valid, another model will be used for solving the task. This case, the target value is the maximization of benefit:

$$\sum_{i=1}^{I}\sum_{j=1}^{J} h_{ij}x_{ij} \to Max \tag{18}$$

As a summary, the model can be defined as follows: let us determine the unknown matrix X=(x_{ij}), which fulfils the following constraints:

$$\sum_{j=1} x_{ij} \geq b_i, \quad \mathrm{i}=1,2,\ldots,\mathrm{I}$$

$$\sum_{i=1}^{I}\sum_{j=1}^{J} x_{ij} \leq K \tag{19}$$

$$x_{ij} \geq \frac{b_i z_{ij}}{\sum_{k=1}^{J} x_{ik}} \qquad \begin{matrix} i=1,2,\ldots,I \\ j=1,2,\ldots,J \end{matrix}$$

and the value of target function would be maximal:

$$\sum_{i=1}^{I}\sum_{j=1}^{J} x_{ij}\cdot h_{ij} \to MAX\ \ ! \tag{20}$$

The above task is a linear programming model that consists of linear conditions and a target value. Some of the expenditure groups are not included in the optimization since they are of fixed costs, as, for example, the operation expenditures of the road management unit. This kind of cost item is known, so, it can be simply deducted from the whole sum destined to road management. Of course, a model can be developed also for the determination of the operation costs of these organizations, and the nearly objective allocation of these sums.

7.2 Model for the Allocation of the Operation Costs of Road Management Units

Denote the operation costs needed (or actually used in the previous year) for the first unit by L_1, for the second one by L_2,….and for the jth organization by L_j. (Bakó [7]). Supposing that these needed or previously actually used sums of money are not inaccurate, the following total sum has to be spent for the operation of road management units:

$$L=\sum_{j=1}^{J} L_j \tag{21}$$

The allocation (distribution) of this total sum among road operators can be determined more or less objectively. The task can be solved by using the quantitative parameters mentioned before; some of these parameters can be:

a) The total length of the road network managed by the organisation

b) The sum of the road sections managed weighted by their traffic volumes

c) Weighted operation tasks of the unit considering several qualitative parameters as total road length, traffic size, number, types and surfaces of bridges, number of traffic signs, etc.

d) Sizes proportional to other operational tasks to be done

The cases a.) and b.) will be presented briefly

For the case a.), the funds needed for a road management unit can be allocated based on the total road length managed by the unit in question. Denote the total road lengths managed by the 1st, 2nd,….jth road management units by *m₁, m₂,…..mⱼ*. Then the operational costs of the organization projected to 1 km road length are as follows: $fk = \frac{\sum_{j=1}^{J} L_j}{\sum_{j=1}^{J} m_j} = \frac{L}{M}$ (22)

The operational costs of the jth road management unit can be determined using m_j:

$$fk_j = m_j * fk \tag{23}$$

where fk_j denotes the sum destined to jth road management unit in the expenditure group in question (that is x_{lj}, where i denotes the row related to the operation of the organisation in the matrix X).

In the allocation variant b.), also the traffic volume A_s is known for each road section u_s. The traffic volume can be characterized by AADT, ESAL or a modified ESAL (Gáspár [16]). Then the road section lengths weighted by their traffic volumes have to be calculated for any jth road management unit.

$$m_j^{'} = \sum_{s=1}^{p} u_s A_s \tag{24}$$

where p is the number of homogeneous road sections managed by the jth road management unit

After having calculated the weighted $m_j^{'}$ values, the sum of money for a weighted 1km long section is to be determined using the following equation:

$$fk^{'} = \frac{\sum_{j=1}^{J} L_j}{\sum_{j=1}^{J} m_j^{'}} \tag{25}$$

Then the funds needed by road management units can be calculated without any problem. The value $fk^{'}$ for 1 km road section weighted by traffic size and the values $m_j^{'}$ (j = 1, 2, ….J) are used in the determination of funds need:

$$fk_j^{'} = m_j^{'} * fk^{'} \tag{26}.$$

The values fk_j (j=1, 2,...,J) can be fine tuned if other tasks of the road management unit are included in the weighting process.

7.3 Simultaneous Consideration of Several Quantitative Parameters

It is supposed that P various quantitative parameters are related to the ith road management unit. It means that this expenditure group is connected with tasks on various quantitative parameters. For the sake of simplicity, the index i will be omitted, that is $z_j^{(k)}$ is used instead of $z_{ij}^{(k)}$ (j = 1, 2,…..J, k = 1, 2,…….P). Accordingly, x_j is applied instead of x_{ij}, that is the index i is omitted. The sums related to quantitative parameters are nationally fixed. These values are given from the actual use of previous year or come from expert or professional political decisions.

The task is the determination of the sum of x_j (allocated to jth road management unit in this expenditure group) based on the known $z_j^{(k)}$ quantitative parameters and the $W^{(k)}$row sums related to the given qualitative parameters.

Since there are several qualitative parameters in the expenditure group in question, the x_j can be calculated as the sum of $x_j^{(k)}$ elements related to the qualitative parameters k = 1, 2. ….P. The elements $x_j^{(k)}$ are calculated in the ratio of the connected qualitative parameters.

The quantities x_j and $x_j^{(k)}$ have to satisfy the following relationships:

$$\sum_{j=1}^{J} x_j = w$$

$$\sum_{j=1}^{J} x_j^{(k)} = w^{(k)}, \qquad \text{k=1,2,...,P}$$

$$\sum_{k=1}^{K} w^{(k)} = w \tag{27}$$

$$\sum_{k=1}^{K} x_j^{(k)} = x_j, \qquad \text{j=1,2,...,J}$$

where the values w and $w^{(k)}$ are known.. The values $x_j^{(k)}$ are calculated, in the ratio of k^th^ qualitative parameter, as follows:

$$x_j^{(k)} = \frac{w^{(k)} z_j^{(k)}}{\sum_{l=1}^{J} z_l^{(k)}} \qquad \mathrm{k} = 1,2,\dots.,\mathrm{P} \tag{28}$$

The x_j sum of money related to j^th^ road management unit can be even directly calculated using the above equations, as follows:

$$x_j = \sum_{k=1}^{K} \frac{w^{(k)} z_j^{(k)}}{\sum_{k=1}^{J} z_l^{(k)}} \qquad \mathrm{j} = 1,2,\dots.,\mathrm{J} \tag{29}$$

Another option for the calculation of the funds allocated to the road management unit, to consider the ratio between the actual expenditures in the previous year. The only difference from the procedure presented before is the use of following equation:

$$x_j = \frac{W \cdot L_j}{\sum_{l=1}^{J} L_l} \qquad \text{j=1, 2,...,J} \tag{30}$$

It should be noted that the above algorithm can be fine-tuned by the inclusion of additional parameters.

7.4 Treating with Insufficient Funds

It is a usual situation that the sum of needed funds exceeds the available ones, consequently:

$$\sum_{i=1}^{I} b_i > K \tag{31}$$

where K equals to the whole funds available. This case, another optimization model could be used that will be briefly shown.

Since the sum of elements bi (actually the total demand) is above the financial resources available, the following constraint is set for the sum of the funds to be allocated:

$$\sum_{i=1}^{I} \sum_{j=1}^{J} x_{ij} = K \tag{32}$$

The sum of the funds to be distributed to each expenditure group cannot exceed the total need in the same expenditure group:

$$\sum_{j=1}^{J} x_{ij} \le b_i \qquad \mathrm{i} = 1, 2, \ldots, \mathrm{I} \tag{33}$$

The equality of the sums of rows is also required in the model:

$$\sum_{j=1}^{J} x_{ij} = \gamma_j b_j \tag{34}$$

This case, the target function is the minimization of the needed and the allocated sums of rows:

$$\prod_{j=1}^{J} \gamma_j \to MIN \tag{35}$$

This target function is nonlinear, that is why the task can be formulated in a non-linear model. Following this principle, a more general model was formulated.

In this model, it is supposed that the real value of the funds available in the previous years exceeded the funds that are presently available. Denote the funds used in previous (e.g. preceding) years the matrix F the element f_{ij} of which is the financial means used by the jth road management unit in the ith expenditure group. The sum of the row I of Matrix F, that is the funds used in ith expenditure group would be denoted by f_i. The symbol of $f^{(j)}$ means the column sum j, that is the total funds used by the jth road management unit. Furthermore the available funds K are also known. It is supposed to be less than the funds used in the previous years or needed in the present year.

The task is the calculation of the matrix X=(x_{ij}), actually the funds for ith expenditure group and jth road management unit. The sum of all elements of matrix X has to be equal to the value of funds K:

$$\sum_{i=1}^{I}\sum_{j=1}^{J} x_{ij} = K \tag{36}$$

If the sum of b_j j = 1. 2, ….,J is known, the following relationship has to be fulfilled:

$$\sum_{i=1}^{I} x_{ij} \geq b_j, \quad \mathrm{j}=1,2,\ldots,\mathrm{J} \tag{37}$$

The target is to determine a matrix that is similar to the other one as much as possible. For the measure of similarity, any of the known parameters can be applied, e.g. the Kulback measure (Klafszky, [26]). In the case of the vectors a= $(a_1, a_2, \ldots a_n)\rangle 0$ and b=$(b_1, b_2, \ldots., b_n)\rangle 0$ is given by the following equation:

$$\sum_{i=1}^{n} (a_i \log \frac{a_i}{b_i} - a_i + b_i) \tag{38}$$

Using the Kulback measure, the similarity can be determined, in the case of matrices F and X, in the following cases:

a) Similarity of the sums (that is sums of rows) allocated to expenditure groups

b) Similarity between the funds allocated to the road management units in the preceding year and this year

c) Similarity between the sums distributed in the preceding and the present year, actually the similarity of the matrices X and F

So, the target function is the minimization of the measures a.)-c.). In the case of expenditure group comparison – version a.) –, the target function is:

$$g_1(f_j, x_{ij}) = \sum_{j=1}^{J} (f_j \log \frac{f_j}{\sum_{i=1}^{I} x_{ij}} - f_j + \sum_{i=1}^{I} x_{ij}) \rightarrow MIN \tag{39}$$

In the case of similarity between the funds allocated to the road management units, – version b.) –following relationship has to be minimized:

$$g_2\left(f^{(j)}, x_{ij}\right) = \sum_{i=1}^{I} \left(f^{(i)} \log \frac{f^{(i)}}{\sum_{j=1}^{J} x_{ij}} - f^{(i)} + \sum_{j=1}^{J} x_{ij} \right) \rightarrow MIN \tag{40}$$

In the case of the similarity between the sums distributed in various years, – variant c.) – following target function has to be applied:

$$g_3(f_{ij}, x_{ij}) = \sum_{i=1}^{I}\sum_{j=1}^{J}\left(f_{ij}\log\frac{f_{ij}}{x_{ij}} - f_{ij} + x_{ij}\right) \to MIN \tag{41}$$

If all the three cases are considered, the target function is the weighted minimization of the cases a.)-c.):

$$\alpha_1 g_1(f_j, x_{ij}) + \alpha_2 g_2(f^{(i)}, x_{ij}) + \alpha_3 g_3(f_{ij}, x_{ij}) \to MIN \tag{42}$$

The above target function includes the preceding ones, as well, because in case of $\alpha_2 = \alpha_3 = 0$, the target function of variant a.), in the case of $\alpha_1 = \alpha_3 = 0$, the target function of variant b.), while in the case of $\alpha_1 = \alpha_2 = 0$, the target function of variant c.) are given. The determination of the parameters α_i can be the result of a professional-political decision, since the primary goal and the stimulus have to be always considered.

Summarizing the model, the following nonlinear task has to be solved. Determine the matrix X=(x_{ij}) for which the following conditions are met:

$$\sum_{i=1}^{I}\sum_{j=1}^{J} x_{ij} = K \tag{43}$$

$$\sum_{j=1}^{J} x_{ij} \geq b_i \qquad \text{i=1,2,...,I}$$

and the following target function is minimal:

$$\alpha_i \sum g_1(f_j, x_{ij}) + \alpha_2 g_2(f^{(i)}, x_{ij}) + \alpha_3(f_{ij}, x_{ij}) \to MIN \tag{44}$$

Conclusions

As mentioned earlier, several subsystems exist already in Hungary, in the field of Transport Asset Management. The systematic trial section monitoring has begun more than a decade ago. Asset value calculations, related to bridges and roads, is also performed regularly. We have urban, motorway and highway PMS systems, as well.

A combined PMS-BMS model has been also completed. The generalization of this model system is under development.

The first version of the model family consists of the following parts:

- The exact mathematical model (e.g. BMS + PMS)

- Normative model for some other elements
- Cost/benefit type models

The network-level multi-stage HUPMS model was developed further by applying climate-dependent and traffic-dependent parameters.

Furthermore, some other related management models (model for funds distribution; allocation of the operation costs of road management units; simultaneous consideration of several quantitative parameters; treating with insufficient funds) developed were also presented briefly.

Future plans are to develop further the above models for their inclusion into an effective Road Asset Management System.

References

[1] Agárdy, Gy., Bakó, A., Gáspár, L., Kolozsi, Gy., Lublóy, L., Molnár, I.: Adaptation of PONTIS BMS to Hungarian conditions. Proceedings of 4th Bridge Engineering Conference, AUSTROADS, Adelaide (South Australia) 2000, pp. 61-70

[2] Asset Management for the Road Sector, 2001, OECD, 83 p.

[3] Asset Management Premier, US Department of Transportation, FHWA, Office of Asset Management, 1999, 30 p.

[4] Bakó, A., Gyulai L., Erben, P: Structure of the Road Data Bank, Proceedings of the Pavement Management System, Budapest, 1989, pp. 43-47

[5] Bakó, A.: Mathematical Model for the first Hungarian network level PMS, Közlekedésépítési és Mélyépítéstudományi Szemle No. 2, 1991, pp. 68-72 (In Hungarian)

[6] Bakó A.: Programming system for funds distribution and its handling, Algorithm and Program Handling Guide. Ministry for Transport, Telecommunication and Water Management, Budapest (In Hungarian), 1992

[7] Bakó A.: Solving funds need and distribution using computer, Transport and Civil Engineering Scientific Review No. 1, 1994, pp. 39-44 (In Hungarian)

[8] Bakó, A.: Linear Multistage Optimization System, Periodica Politechnica No. 4, 1996, pp. 53-63

[9] Bakó A.: Combined network level road pavement-bridge management system, Közúti Közlekedés és Mélyépitéstudományi Szemle No. 3, 1997, pp. 96-100 (In Hungarian)

[10] Bakó A., Csicsely-Tarpay M., Gáspár L., Szakos P. The Development and Appli-cation of a Combined Highway Pavement Management System in Hungary, Proceedings of the 4th International Conference on Managing Pavements. Durban (South Africa), Volume 3, 1998, pp. 1091-1105

[11] Bakó, A., Gáspár, L.: PMS models in Hungary, CD Publishing, Proceedings of 1st European Pavement Management Systems Conference, Budapest, 2000, 8 p.

[12] Bakó, A., Földesi, P., Gáspár, L.: Using Traffic Forecasting Models in Asset Management, CD-ROM Proceedings of the 8th International Conference on Managing Pavement Assets, Santiago, Chile, 2011, 11 p.

[13] Csicselyné, T. M.: Funds Distribution is the Primary Task of the Road Management, Közlekedésépítés és Mélyépítéstudományi Szemle No. 9, 1993, pp. 291-295 (In Hungarian)

[14] Fábián, C. I., Prékopa, A., Ruf-Fiedler, O..“On a Dual Method for a Specially Structured Linear Programming Problem”, Optimization Methods and Software 17, 2002, pp. 445-492

[15] Feighan, K. J., Shanin, M. Y., Sinha, K. C.: A Dynamic Programming Approach to Opti-mization for Pavement Management Systems, Proceedings of 2nd North American Pavement Management Conference, 1988, pp. 2.2195-2.2206

[16] Gáspár L.: Economic asphalt pavement preservation, DSc (Doctor of Sciences Thesis), Hungarian National Academy of Sciences, Budapest, 1988, 263 p.

[17] Gáspár, L.: Development of the first Hungarian Network Level PMS, Közlekedéstudományi Szemle No. 4, 1991, pp. 132-141 (In Hungarian)

[18] Gáspár, L., Bakó, A.: Le systéme hongrois de gestion de l'entretien. Revue Générale des Routes et des Aérodromes No. 710, 1993, pp. 34-36

[19] Transportation Research Record 1455. Pavement Management Systems National Academy Press. Washington, D.C., pp. 22-30

[20] Gáspár, L., Bakó, A.: Further Development of Hungarian Road Asset Management due to Climate Change. CD-ROM Proceedings of 4th European Pavement and Asset Manage-ment Conference (Session 5), Malmö, 2012, Sweden, 10 p.

[21] Gáspár, L., Karoliny, M.: Investigation and design of durable pavement structure rehabilitation. LAP Lambert Academic Publishing, Saarbrücken, 2015, 101 p.

[22] Gáspár, L.: Actual efficiency of road pavement rehabilitation. CETRA2016 Proceedings of the 4th International Conference on Road and Rail Infrastructure, Sibenik, Croatia, 2016, pp. 181-186

[23] Golabi, K., Thompson, P. D., Hyman, W. A.: PONTIS Version 2.0 Technical Manual. A Network Optimization System for Bridge Improvements and Maintenance, FHWA-SA-94-031, 1993

[24] Hudson, W. R., Hudson, S. W.: Pavement Management System Proceedings of the 3th International Conference on Managing Pavement, San Antonio (USA), 1994 , pp. 99-111

[25] Hudson W.R., Haas R., Uddin W.: Infrastructure Management, McGraw-Hill, New York, 1997, 393 p.

[26] Klafszky E.: On the forecast of Input-Output tables, Bulletin of the Hungarian Academy of Sciences SZTAKI, Budapest, Vol. 10, 1973, pp. 1-13 (In Hungarian)

[27] Männistö V.: Network Level PMS Research Report, Budapest, 1995, 35 p.

[28] What is Asset Management Federal Highway Administration, Office of Asset Management, Washington, D.C., 2001

Stable vector operation implementations, using Intels SIMD architecture

József Smidla, István Maros

University of Pannonia, 10. Egyetem Str., Veszprém, Hungary H-8200, smidla@dcs.uni-pannon.hu, maros@dcs.uni-pannon.hu

Most floating point computations, in particular the additive operations, can have an annoying side-effect: The results can be inaccurate (the relative error can be any large) which may lead to qualitatively wrong answers to a computational problem. For the most prevalent applications there are advanced methods that avoid unreliable computational results in most of the cases. However, these methods usually decrease the performance of computations. In this paper, the most frequently occurring vector-vector addition and dot-product operations are investigated. We propose a faster stable add method which has been made possible by the appropriate utilization of the advanced SIMD architecture. In this work we focus only on the widely used Intel systems that are available for most of users. Based on it, stable vector addition and dot-product implementations are also introduced. Finally, we show the performance of these techniques.

Keywords: Scientific computations, Numerical accuracy, Accelerating stable methods, Heuristics, Intel's SIMD architecture, Cache techniques

1 Introduction

Several pieces of software use vector operations like vector addition and dot product. Scientific applications require double precision floating point number representation because of the required accuracy of the computations. However, there are cases when even this representation is not sufficient. One example is the simplex method, where wrongly generated non-zeros can slow down the solution algorithm and can lead to false results. One possible way to reduce the chances of the occurrence of such events is implementations using absolute and relative tolerances in order to mitigate numerical errors. There are several open-source linear algebraic libraries like BLAZE, but they do not handle numerical errors. There are techniques that can greatly improve the accuracy of floating point additive arithmetic operations [1, 2, 3] but they are very slow. For example, sorting of the addends can increase the result's accuracy [4], but the drastic slow-down is unacceptable in many applications. Our aim is to develop an efficient linear algebraic library that supports increased accuracy by heuristics while the incurred slowdown factor is minimal. We can achieve this by utilizing some future proof advanced features of Intel's SIMD processors. Of course, the real benefit in speed of these methods appears in computationally demanding applications like in Gaussian elimination, matrix multiplications. Particularly, since matrix by matrix multiplication needs dot-products, this operation can generate a significant amount of numerical error.

The paper is organized as follows. The behavior of numerical errors is shown in Section 2. Intel's SIMD architecture and the cache system are introduced in Section 3. We propose our simplified stable addition operation in Section 4, its SIMD implementation is also presented. The next section introduces our conditional branching free stable dot-product technique for the C programming language, and the SIMD based dot-product operation. In Section 6 the computational performance of the introduced implementations is compared and explained. Finally, in Section 7 we present our conclusions.

The introduced methods in this paper often use bit manipulation operations. One bit can be 1 or 0. The value of a bit can be changed by hardware and/or software. If a bit is changed to 1, we say that we *set* the bit. On the other hand, if we want to ensure that the bit is 0, we say the bit is *cleared*.

2 Numerical errors in scientific computations

The numbers used in scientific and engineering computations can be represented in several ways. A very reliable and accurate method is the symbolic number manipulation provided by Maple [5], MATLAB's Symbolic Math Toolbox [6], or Mathematica [7, 8]. However, there are applications, where these tools are not available. We are focusing on the floating point numbers [9, 10], which are easier to use, but they can have accuracy problems [11]. Floating point numbers (whether single or double precision) have three fields: Sign bit, significand and the exponent. The number of significand bits is fixed. The painful consequence of this principle is that the numbers with large magnitude have lower precision than smaller numbers. This can lead to the so called *rounding error*. Let a and b be nonnegative numbers with $a \gg b$. If we compute the value of $a+b$, it can happen that the result will be just a.

The other source of errors is *cancelation*, also known as *loss of significant digits*. If in theory $a = -b$, and $a, b \neq 0$, we expect that $a + b = 0$. However, if a and b may carry numerical inaccuracies the computed sum can be a small number ε which is usually computational garbage. If the execution of the program depends on the zero test of the result it can go in the wrong direction. Our primary interest is the numerical behavior of optimization software. In this area considerable efforts have been made to properly handle the emerging cancelation errors. Our stable vector operations also have been designed to handle this type of error in an efficient way.

3 Intel's SIMD architecture

The SIMD (Single Instruction, Multiple Data) architecture provides a tool to perform the same low level operations on multiple data in parallel [12]. It was successfully used in the simplex method [13], and in other numerical algorithms [14]. The old Intel CPUs used a stack for storing 32, 64 or 80 bit floating point numbers. This architecture can perform the current operation only on a single data. In 1999 Intel introduced the SSE (Streaming SIMD Extensions) instruction set in the Pentium III processor. It contains 70 new instructions, which can operate on single precision numbers. The processor has 8 brand new, 128 bit wide registers, named XMM0, XMM1, ..., XMM7. One XMM register can store 4 single precision numbers. The arguments of the operations are the registers, and the result will be stored in such a register. For example, if we add the content of register XMM1 to XMM0, the CPU adds the first number in XMM0 to the first number of XMM1, and so on, as it is shown in Figure 1.

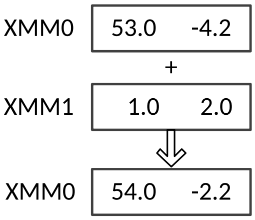


Figure 1
Addition using XMM registers

However, SSE does not support 64 bit floating point operations. Pentium 4 introduced SSE2, which supports 64 bit numbers as well. The XMM registers are still 128 bit wide, so one register can contain 2 double precision floating point numbers. SSE3 and SSE4 processors have added some expansion to the instruction set, like dot product and integer number support.

In 2011, Intel added the AVX (Advanced Vector Extensions) instruction set to the CPUs. This extension doubles the size of the XMM registers to 256 bit wide. It is called YMM. Now one register can store four 64 bit floating point numbers. Moreover, AVX's have 16 YMM registers. The AVX2 instruction set provides some integer operations with the new registers.

The modern Intel CPUs have one more useful feature. Namely, they have two memory ports. It means that, while the CPU calculates, they can load some other data from the memory in parallel.

Cache

In this section a brief summary of the CPU caching is given because it has some non-intuitive properties: The wrong utilization of the cache cannot achieve the highest performance. The communication between the CPU and the main memory is much slower than the speed of the CPU, i.e., while the CPU is waiting for the memory, it can execute a lot of instructions. To keep the CPU working engineers have designed cache memory between the CPU and the main memory. The cache uses faster circuit elements, but it is more expensive, so the cache size is limited relative to the main memory. Typical cache sizes are 3-10 Mbytes today, while the main memory can be

32 Gbytes. Moreover, modern CPUs have more cache levels, where the lower levels are smaller, but they are faster.

The memory is divided into so called cache lines, which are certain lengths of memory partitions. Typical lengths are 32 or 64 bytes. If an instruction reads some bytes from a given memory address, the total cache line that contains the required data moves to the cache. If later instructions load an adjacent memory address its content will already be in the faster cache, thus the reading time is reduced. However, as the cache size is limited, the CPU has to make room for a new cache line if the required data item is not in the cache. In this case a formerly used cache line is dropped out and its content is written back to the main memory if it is necessary. When an instruction writes to a given address, the corresponding cache line is loaded into the cache, and the instruction writes there. In this case, the content of that memory address has a copy in the cache, which is different. This cache line is called dirty, but if we write this content back to memory, this flag is cleared.

There is a little intelligence in the cache controller. If the CPU senses that the software accesses adjacent memory addresses it loads some next cache lines. So, if we read or write a memory region from its beginning to its end, the currently needed data will already be in the cache.

The SSE2 and AVX support bypassing the cache for memory writing. In this case the cache line of the current memory address is not loaded and the CPU writes into the main memory directly. We call this non-temporal writing. Obviously, this mode is much slower in itself. However, we can keep the more important data in the cache. What happens if we add two large vectors (larger than the cache), and the result is stored in a third vector? Without bypassing, the CPU reads the next two terms of the sum and it has to write the result to the memory. At first, the result is placed into the cache. However, since the vectors are too large, and their contents fill the cache, the CPU has to drop out an older cache line to the memory. If the cache line of the result is not prepared for the cache the CPU has to load that cache line and, obviously, drops out an older line too. The non-temporal writing prevents the CPU from loading the cache-line of the destination, so it drops out older cache lines if and only if there is no more room for the input data. Finally, the performance of this algorithm is improved.

4 Vector addition

In computational linear algebra (on which many optimization algorithms rely) vector addition is one of the most frequently used operations.

Let $\mathbf{a}$ and $\mathbf{b}$ be two n dimensional vectors, $\mathbf{a},\mathbf{b} \in \mathbb{R}^n$. We propose our implementations of the following vector addition operation:

$$\mathbf{a} := \mathbf{a} + \lambda \mathbf{b},$$

where $\lambda \in \mathbb{R}$. In detailed form:

```
Input: $\mathbf{a}, \mathbf{b}, \lambda$
Output: $\mathbf{a}$
for $i := 1$ to $n$
    $a_i := a_i + \lambda b_i$
end
```

Figure 2
Naive vector addition

If cancelation errors occur, the implementation shown in Figure 2 can generate fake non zeros. This error can be handled with an absolute tolerance ε_a. If the absolute value of the sum is smaller than the ε_a, we set the result to zero as in Figure 3.

```
Input: $\mathbf{a}, \mathbf{b}, \lambda$
Output: $\mathbf{a}$
for $i := 1$ to $n$
    $a_i := a_i + \lambda b_i$
    if $|a_i| < \varepsilon_a$ then
        $a_i := 0$
    end
end
```

Figure 3
Vector addition using absolute tolerance

The absolute tolerance cannot adapt to the magnitudes of the input values. The solution can be the use of a relative tolerance ε_r. In 1968 William Orchard-Hays [15] suggested the following method using this tolerance: If the sum is much smaller relative to the largest absolute value of the input numbers the result is set to zero, see Figure 4.

```
Input: $\mathbf{a}, \mathbf{b}, \lambda$
Output: $\mathbf{a}$
for $i := 1$ to $n$
    $c := a_i + \lambda b_i$
    if $\max\{|a_i|, |\lambda b_i|\}\varepsilon_r \geq |c|$ then
        $c := 0$
    end
    $a_i := c$
end
```

Figure 4
Vector addition using relative tolerance, Orchard-Hays's method

Determining the maximum of two numbers uses conditional branching. We propose a simplified method which uses fewer operations. It is sufficient to multiply the absolute value of one of the input numbers by the relative tolerance. In this way we can save an absolute value and a conditional branching step. The result can be close to zero if the input values have the same order of magnitude and their signs are different. The useful value of the relative tolerance can be different from that of the Orchard-Hays method.

```
Input: $\mathbf{a}, \mathbf{b}, \lambda$
Output: $\mathbf{a}$
for $i := 1$ to $n$
    $c := a_i + \lambda b_i$
    if $|a_i|\varepsilon_r \geq |c|$ then
        $c := 0$
    end
    $a_i := c$
end
```

Figure 5
Vector addition using simplified relative tolerance

The implementation shown in Figure 3 requires one compare and a conditional jump instruction. Our simplified implementation with relative tolerance uses one addition, two multiplications, two assignments, two absolute values, one compare and one conditional jump. Orchard-Hays's implementation needs one more absolute value and a conditional branching. The additional operations cause overhead in time, so these implementations are slower than the naive one.

4.1 SIMD vector addition

Conditional jumping slows down the execution of the program because it breaks the pipeline mechanism of the CPU. So it is worthy to try to implement the algorithms in a way that avoids conditional jumps. Intel's SIMD architecture contains several instructions which help us design such an implementation. We will use the following instructions:

- *Move:* Moves the content of a register to another register.
- *Multiply:* Multiplies the number pairs of two registers.
- *Add:* Adds the number pairs of two registers.
- *And:* Performs a bitwise AND between two registers.
- *Compare:* Compares the number pairs of two registers. If they are identical the destination register will contain a bit pattern filled by 1's, otherwise 0.
- *Max:* Chooses the larger of two numbers stored in two registers. It is used for the implementation of Orchard-Hays's addition method.

The detailed description of these instructions can be found in [16]. The key point of the conditional jump aware implementations (called accelerated stable addition in this paper) is the compare instruction. It compares the number pairs and stores the results in a register. If the register contains two double pairs then the comparator puts two bit patterns in the destination area. One pattern can be filled by 1 if the result of the comparison for the related number pair is true, otherwise 0 as it is shown in Figure 6. These bit patterns can be used for bit masking.

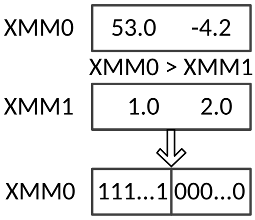


Figure 6
The compare instruction of the SSE2 instruction set

Figures 7 and 8 show the flowchart of the SSE2 implementations of our stable add operations with relative and absolute tolerances. The algorithms add two number pairs loaded to registers XMM0 and XMM1. The final result is placed in XMM2.

The implementations have two main phases: initialization, and process. We prepare some registers to store the value of λ (XMM7), ε_r (XMM4), ε_a (XMM6) and the absolute value mask (XMM5). In the process phase we perform the stable add operations for the successive number pairs, without modifying registers XMM4-XMM7. Figures 7 and 8 show only one iteration in the processing phase. One iteration of the absolute tolerance version stable adder performs 6 steps:

1. Multiply XMM1 and XMM7, store the result in XMM1, XMM1 will store λb_i.
2. Add XMM1 to XMM0, so XMM0 stores $c = a_i + \lambda b_i$.
3. Move XMM0 to XMM2. We have to store the original value of c, in order to use its absolute value in later steps.
4. Bitwise AND between XMM2 and XMM5, store the result in XMM2. Therefore XMM2 stores $|c|$.
5. Now we have to compare $|c|$ and ε_a. If $|c| < \varepsilon_a$, then the CPU sets the bits of the corresponding floating point number in XMM2, otherwise clears them.
6. Bitwise AND between XMM2 and XMM0. After this step, if $|c| < \varepsilon_a$ then XMM2 stores zero, because of the cleared bit mask in XMM0, otherwise XMM2 stores c.

The stable add operation that uses relative tolerance performs 9 steps in one iteration:

1. Multiply XMM1 and XMM7, store the result in XMM1, XMM1 will store λb_i.
2. Move XMM0 to XMM2. We have to store the original value of a_i and λb_i, in order to use their absolute value in the later steps.
3. Add XMM1 to XMM2, so XMM1 stores $c = a_i + \lambda b_i$.

4. Move XMM2 to XMM3, because we will use the absolute value of c in the next steps, but we will need the original value of c as well.
5. Bitwise AND between XMM3 and XMM5, store the result in XMM3. Therefore XMM3 stores $|c|$.
6. Bitwise AND between XMM0 and XMM5, XMM0 stores $|a_i|$.
7. Multiply XMM0 and XMM4, and store the result in XMM0, so XMM0 stores $|a_i|\varepsilon_r$.
8. Now we have to compare $|a_i|\varepsilon_r$ and $|c|$. If $|a_i|\varepsilon_r < |c|$, then the CPU sets the bits of the corresponding floating point number in XMM0, otherwise clears them.
9. Bitwise AND between XMM2 and XMM0. After this step, if $|a_i|\varepsilon_r \geq |c|$ then XMM2 stores zero, because of the cleared bit mask in XMM0.

Each operation above belongs to exactly one SSE2 or AVX instruction, so the reader can easily reproduce our results. These implementations use several additional operations on top of the one addition and multiplication, so they have an overhead compared to the naive implementation. They use some additional bit masking steps, because the Intel's SIMD instruction sets have no absolute value operations. However, we can obtain the absolute value of a floating point number by clearing the sign bit. Therefore, we have to apply a bit masking technique to get the absolute values, as in the steps 5-7, in relative tolerance adder, and step 4 in absolute tolerance adder.

However, SSE2 performs every instruction between two number pairs in parallel, so this overhead is not significant. Moreover, AVX can execute the instructions between 4 number pairs, consequently, the overhead will be even lower. In order to improve the speed of the algorithms, our implementations utilize the two memory ports mentioned in Section 3: While one number pair is being processed, the next pair is loaded to other unused registers, so the delay of memory operations is decreased. This technique is used in our dot-product implementations. In the future, AVX-512 processors will further increase the performance.

We modified the above relative tolerance adder procedure to implement Orchard-Hays's method. After step 6, two additional steps are inserted:

1. Bitwise AND between XMM1 and XMM5, XMM1 stores $|\lambda b_i|$.
2. Use MAX operation between XMM0 and XMM1, XMM0 stores $\max\{|a_i|, |\lambda b_i|\}$.

5 Vector dot-product

The dot-product between two n dimensional vectors **a** and **b** is defined as:

$$\mathbf{a}^T\mathbf{b} = \sum_{i=1}^{n} a_i b_i.$$

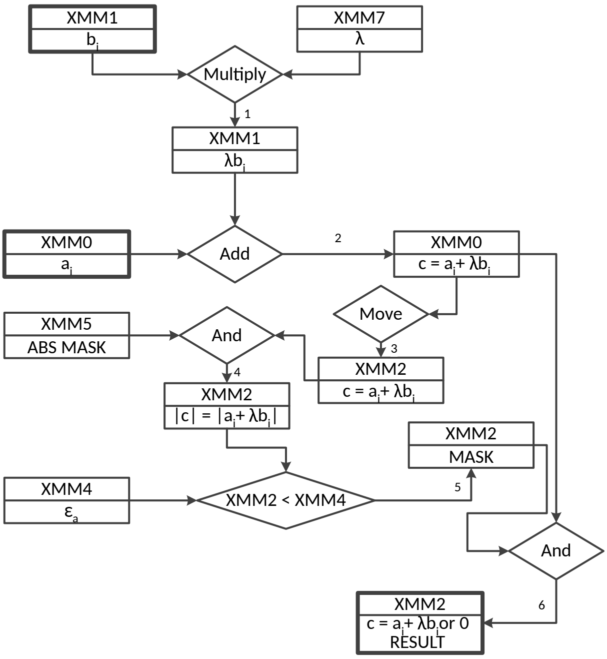


Figure 7
Flow chart of the stable add implementation, using absolute tolerance. Arrow numbers show the order of the operations.

Figure 9 shows the pseudo code of its naive implementation. The problem is that the add operation in line 3 can cause a cancelation error.

This error can be greatly reduced by using a *pos* and a *neg* auxiliary variables as introduced by Maros and Mészáros in 1995 [17]. Positive (negative) products accumulate in variable *pos* (*neg*). Finally, the result is the sum of *pos* and *neg* as shown in Figure 10. This final add is a stable add operation introduced in Section 4.

The conditional jump in line 5 breaks the pipeline mechanism and the execution slows down accordingly. We have developed a solution for C/C++ programs, where the conditional jump can be avoided and substituted by pointer arithmetic. This method can be used if the later introduced SIMD based methods are not available, for example the AVX is disabled by the operating system. The elements of an array are stored in adjacent memory addresses. If a pointer is increased by 1 in C/C++, it will refer to the next object. The most significant bit in the bit pattern of a double type variable stores the sign bit. If this bit is 1, the number is negative, otherwise it is positive. The conditional jump free implementation uses a double type array, where the first element stores the positive, the second one stores the negative sums. The current product is added to one of these elements. The address of the current sum variable is obtained by a C/C++ expression: The address of the array is shifted by the product's sign bit, as Figure 11 shows.

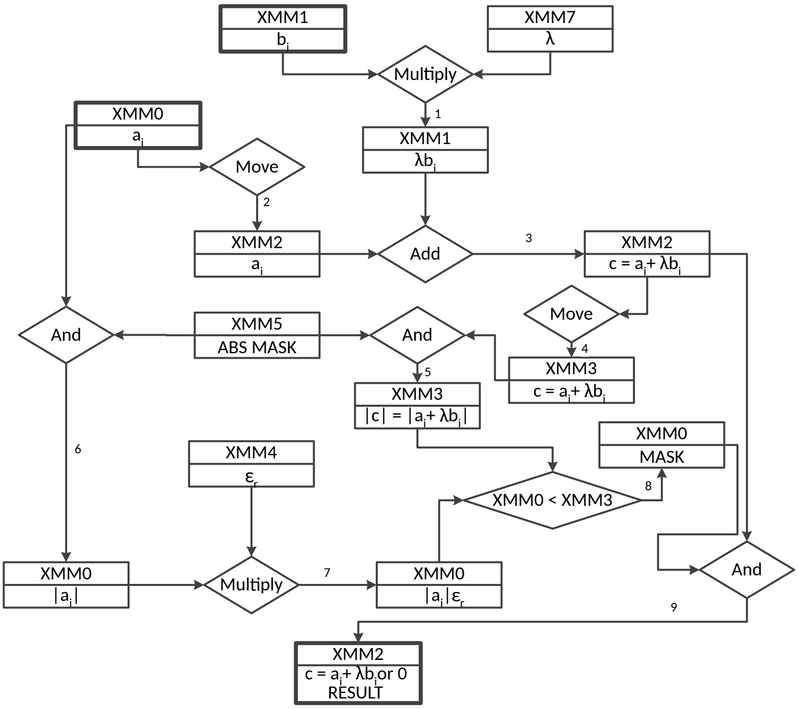


Figure 8
Flow chart of the stable add implementation, using relative tolerance. Arrow numbers show the order of the operations.

```
Input: $\mathbf{a}, \mathbf{b}$
Output: $dp$
dp := 0
for $i := 1$ to $n$
        $dp := dp + a_i b_i$
end
```

Figure 9
Naive dot-product

SIMD dot-product

The SIMD version of the dot product uses similar techniques introduced in Section 4.1. This implementation also has two phases, initialization and processing. We use XMM1 to store the negative products, XMM2 stores the positive products, and XMM4 contains zero for the comparison.

In the first step the product is loaded into XMM0. Of course, the multiplication can be supported by SSE2. The separation of positive and negative products can be implemented in 7 steps:

```
Input: $\mathbf{a}, \mathbf{b}$
Output: $dp$
$dp := 0$
$pos := 0$
$neg := 0$
for $i := 1$ to $n$
    if $a_i b_i < 0$ then
        $neg := neg + a_i b_i$
    else
        $pos := pos + a_i b_i$
    end
end
$dp := StableAdd(pos, neg)$
```

Figure 10
Stable dot-product, where *StableAdd* is an implementation of the addition, which can use tolerances

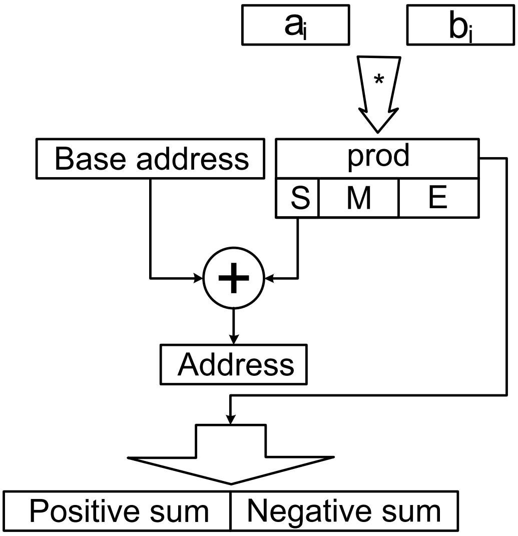
1. In order to keep the value of the product $a_i b_i$, we save the content of XMM0 to XMM5.
2. Move the content of XMM0 to XMM3, in order to perform the comparison between zero and the product.
3. Compare XMM3 with XMM4, if $a_i b_i < 0$, then the CPU sets the bits of the corresponding floating point number in XMM3, otherwise clears them.
4. Bitwise AND between XMM5 and XMM3. If $a_i b_i < 0$, then XMM5 stores $a_i b_i$, otherwise zero.
5. Add XMM5 to XMM1, i.e if $a_i b_i < 0$, then we add this negative value to XMM1, otherwise we add zero.
6. Bitwise AND between the inverse of XMM3 and XMM0. If $a_i b_i \geq 0$, then XMM3 stores $a_i b_i$, otherwise zero.
7. Add the content of XMM3 to XMM2, that is we update the positive sum.

Similarly to the SIMD accelerated vector addition, this dot product algorithm can be improved using AVX. The stable dot product uses fewer instructions than the stable add, so the performance of this implementation is better, as we will see in Section 6.

6 Computational experiments

In this section some benchmarking results are presented. The tests were performed on a computer with the following parameters:

- CPU: Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz
- Level 1 cache: 32 Kbyte
- Level 2 cache: 256 Kbyte



```
union Number {
 double num;
 unsigned long long int bits;
} number;
    double negpos[2] = {0.0, 0.0};
    [...]
    const double prod = a * b;
    number.num = prod;
    *(negpos + (number.bits >> 63)) += prod;
```

Figure 11
Handling positive and negative sums with pointer arithmetic without branching, where *S* is the sign bit, *M* is the significand and *E* is the exponent

- Level 3 cache: 3072 Kbyte
- Memory: 8 Gbyte
- Operating system: Debian 8, 64 bit
- Window manager: IceWM

The i5-3210M CPU has three cache levels. Intel processors have an inclusive cache architecture. It means that the higher level caches include the lower levels, so the test CPU has 3 Mbyte cache in total. Moreover, this CPU has two cores, where the L1 and L2 caches are unique in each core. However, the cores share the L3 cache, so it can happen that more than one process uses the L3 cache [18].

Our SSE2 and AVX implementations are written in assembly and compiled by NASM, version 2.11.08. In our C language implementations we used C++11 for

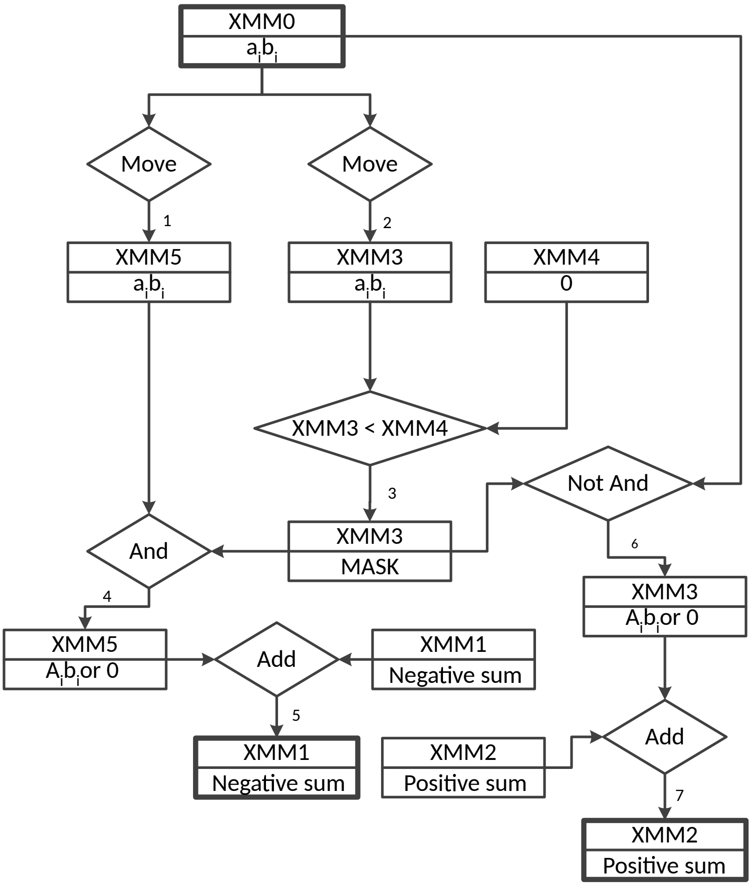


Figure 12
Flow chart of the stable dot product implementation. Arrow numbers show the order of the operations.

the benchmarking software, and the non-SIMD algorithm versions. The C++ compiler was gcc 4.9.2.

We have performed 80 measurements with different sizes of test vectors. In the sequel, s_i $(0 \leq i < 80)$ denotes the size of one vector in the i^{th} test. In the first test, the size of a vector is 1000 elements ($s_0 = 1000$). The vector sizes grew exponentially: $s_i = \lfloor 1000 * 1.1^i \rfloor$, thus the largest vector size is 1862182 elements. Since one element is 8 bytes long, the smallest vector needs 8000 bytes, while the largest is 14.2 Mbytes long.

6.1 Vector addition

Each test was repeated 5000 times, and the execution time was measured. Based on the vector lengths and the execution time, the performance was calculated in number of FLOPS (Floating-point Operations Per Second). We have counted only the effective floating point operations, i.e. the multiplication by λ and the addition. The number of effective floating point operations expresses how long input vectors can be processed by the current implementation. If an implementation uses additional auxiliary floating point operations (like multiplying by a ratio), that operations do not count.

The input vectors were randomly generated. If we add two numbers, then we have two cases: (1) The result is stable, so we keep it, (2) or the result violates a tolerance, so it is set to zero. Hence we have generated the input vectors in such a way that the likelihood for setting the result to zero is 1/2. This method moderately supports the efficiency of the CPU's branching prediction mechanism. Moreover, if it is required to set zero half of the results, it ensures that the non-vectorized implementations have to execute all of their branches.

We have to distinguish two cases of the vector addition operation:

1. $\mathbf{c} = \mathbf{a} + \lambda\mathbf{b}$, three vectors case
2. $\mathbf{a} := \mathbf{a} + \lambda\mathbf{b}$, two vectors case

where the memory areas of the vectors **a**, **b** and **c** are different. Since these cases use the memory in different ways we have tested them for every implementation.

6.1.1 Results for three vectors

If three different memory areas are used with cache, the cache is divided into 3 partitions, so the performance is decreased. However, if non-temporal memory writing is used, then larger vectors can be placed in the cache. Moreover, if the larger cache is still tight the non-temporal writing saves unnecessary memory operations. Therefore, this writing mode is recommended for large vectors. Figure 13 shows the results for the unstable implementations. It can be seen that the AVX is the best alternative, because it can perform four floating point operations per CPU cycle. The performance decreases if the vectors grow out of the available cache sizes. Since the L3 cache is shared among the cores, our process cannot use the whole cache, so the efficiency decreases sooner as the total vector sizes exceed the size of larger caches.

If the vectors are too large, the non-temporal SSE2 and AVX implementations have the same performance because they execute quick calculating operations, but the speed of memory operations is much slower than a floating-point operation. This holds for the cache writing implementations too, but their performance is the half of that of the non-temporal versions, because they use slower memory operations.

Figure 14 shows the results for the stable add implementations, where relative tolerance is used. Since more operations are used for one stable add step, the performance is lower than in the unstable case. If the vectors are larger then the non-temporal writing version with AVX is a little faster than the non-temporal SSE2 because the AVX instructions have to wait fewer times to read data from memory. While the 9 steps of the stable add are executed the CPU can read the next data into the cache.

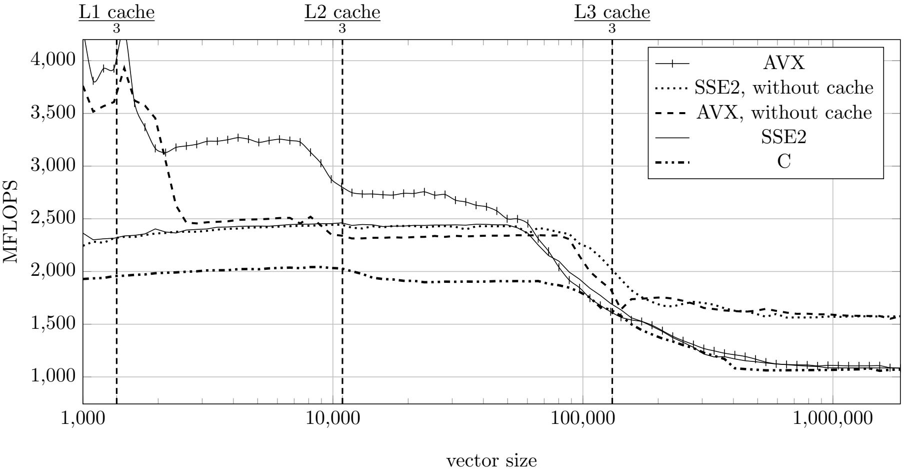


Figure 13
Performances of the unstable add vector implementations for three vectors

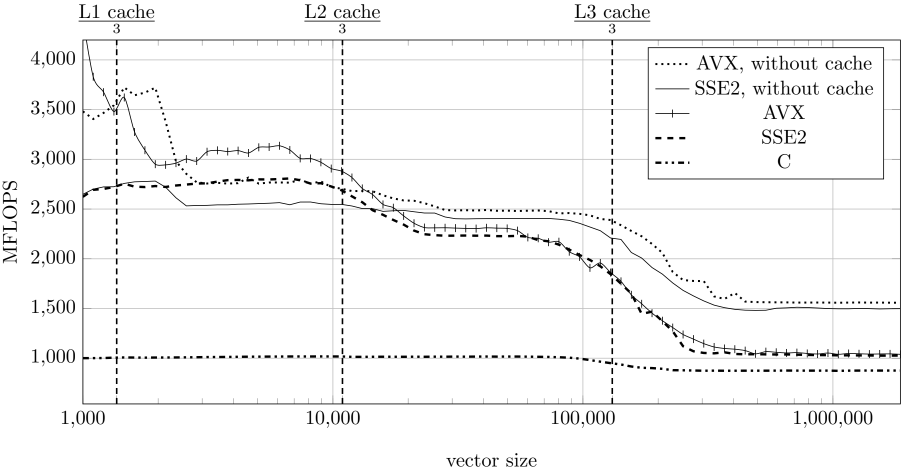


Figure 14
Performances of the stable add vector implementations, using relative tolerance, for three vectors

As Figure 15 shows, the performance of the absolute tolerance versions has a similar behavior to the unstable implementations but, of course, in this case the performance is lower.

6.1.2 Results for two vectors

If two vectors are used and one of them is the result the cache line of the current result memory area is in the cache. This involves that there is no additional communication between the cache and the memory, so the performance increases. Obviously, bypassing the cache is not unprofitable in this case, as Figures 16, 17,

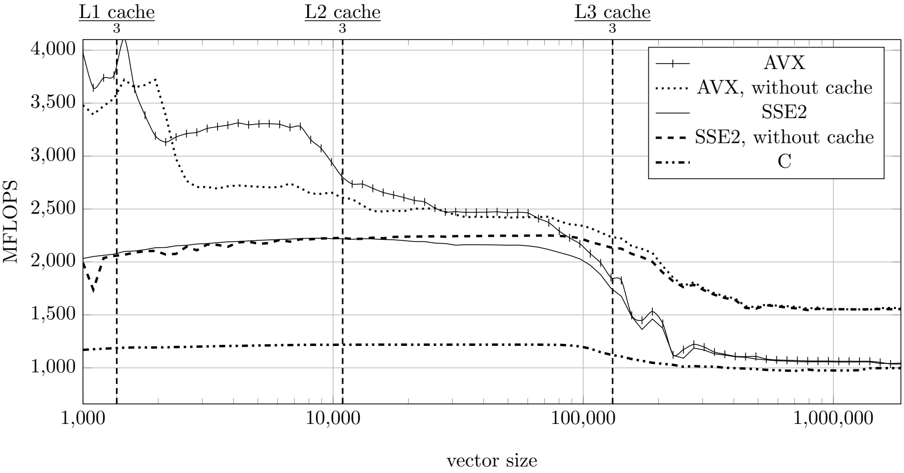


Figure 15
Performances of the stable add vector implementations, using absolute tolerance, for three vectors

and 18 show. If the cache is not bypassed, the overall performance is better than in the three vectors case.

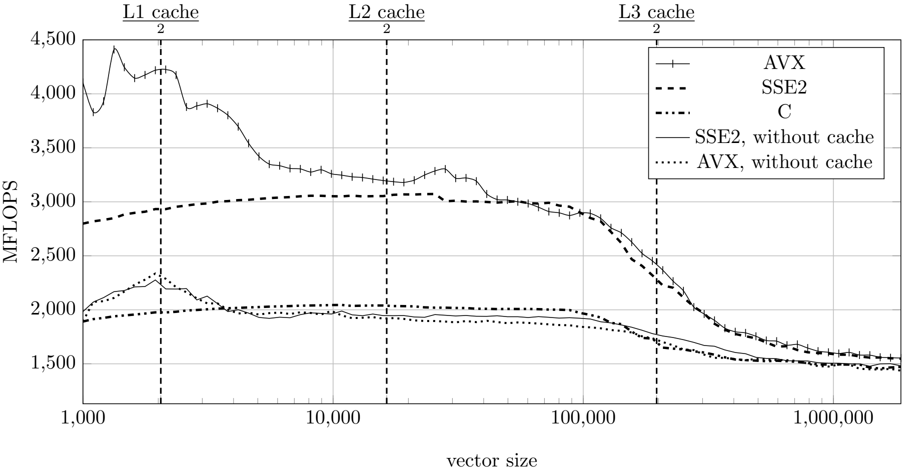


Figure 16
Performances of the unstable add vector implementations for two vectors

6.1.3 Orchard-Hays's relative tolerance method

Since SSE2 and AVX have a MAX operation which selects the maximum of two numbers, Orchard-Hays's relative tolerance test can be implemented on Intel's SIMD architecture. As mentioned in subsection 4.1 two additional operations are inserted into the assembly code; the max selector and an absolute value operation. The

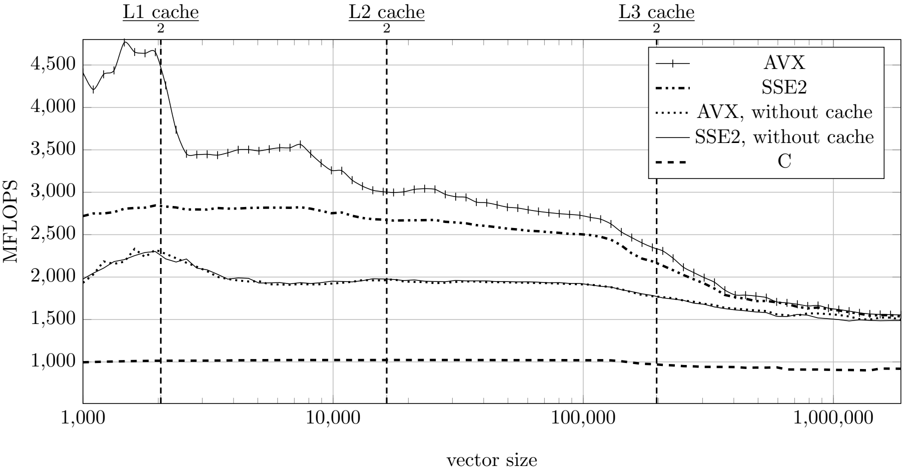


Figure 17
Performances of the stable add vector implementations, using relative tolerances for two vectors

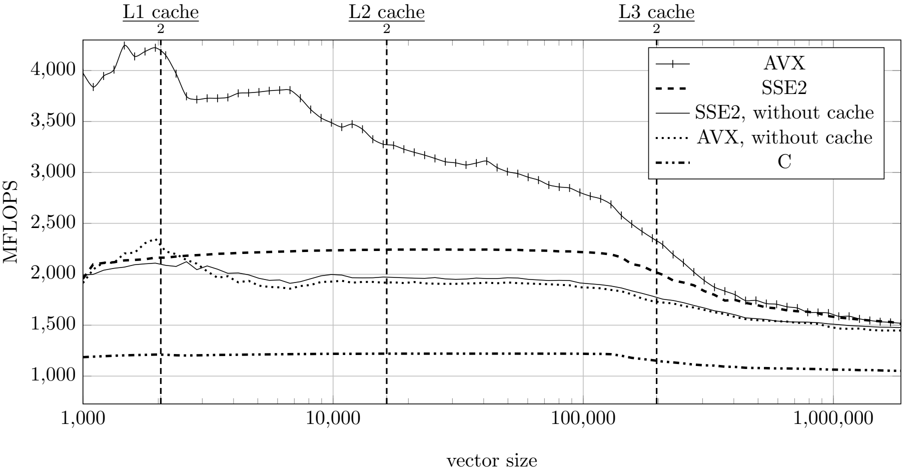


Figure 18
Performances of the stable add vector implementations, using absolute tolerance, for two vectors

modified implementation uses 11 instructions, where the max operation requires significant amount of execution time, as Figures 19-22 show. There were 800 measurement points that compare Orchard-Hays's and our method. In 607 cases, our algorithm is the fastest, the highest speedup ratio was 1.245 in the 3 vector SSE2 test, using cache. Our method behaved worse in the remaining 193 test points, the worst ratio was 0.905 in the 3 vector, AVX, and cache-free case. We mention that this is a very extreme case, in most of the cases, if our approach is worse, the ratio moves around 0.98. However, as we saw, a simple policy can be constructed: Depending on the vector's size, and numbers (2 or 3 vectors), we can choose between the cache and cache-free implementations. We can avoid most of the situations, when our method's performance is lower than the original.

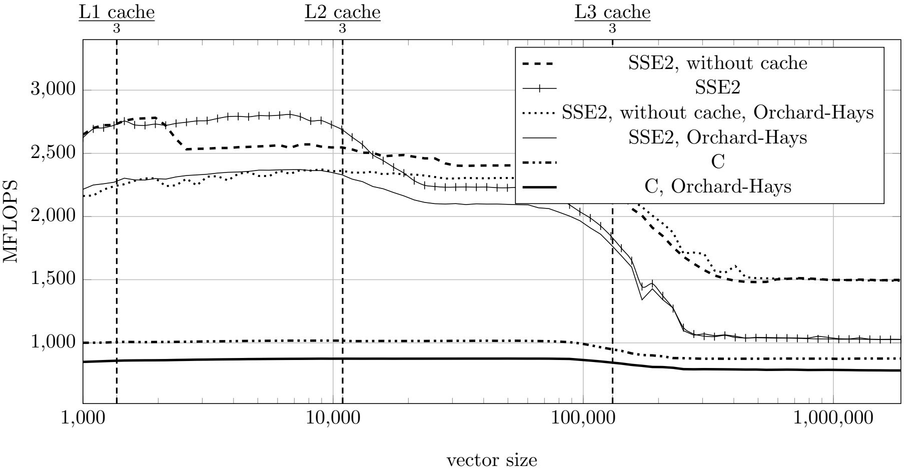


Figure 19
Performance comparison of our stable add implementations and the method of Orchard-Hays, with SSE2, using relative tolerance, for three vectors

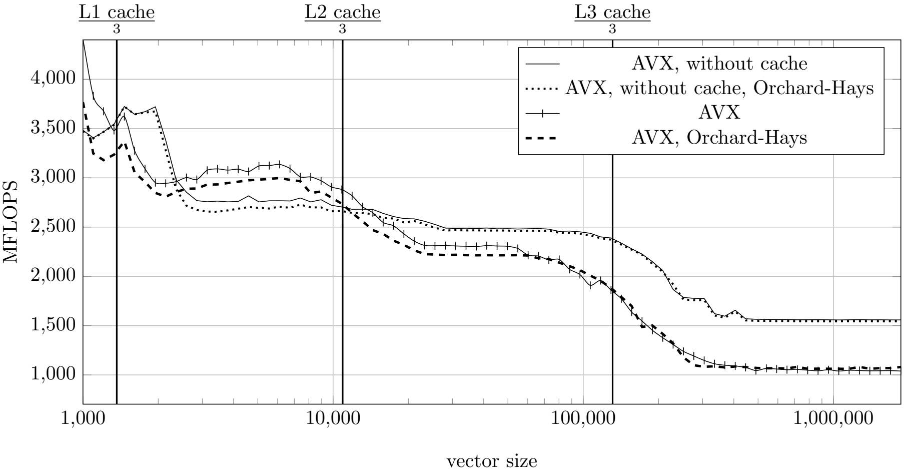


Figure 20
Performance comparison of our stable add implementations and the method of Orchard-Hays, with AVX, using relative tolerance, for three vectors

6.2 Dot-product

The dot product requires only two vectors and the result is a scalar value. Since, in general, the input vectors have much more than one element, writing time of the result to the memory is irrelevant. The stable AVX implementation uses only 7 instructions in addition to the loading, multiplying, and add operations, so its

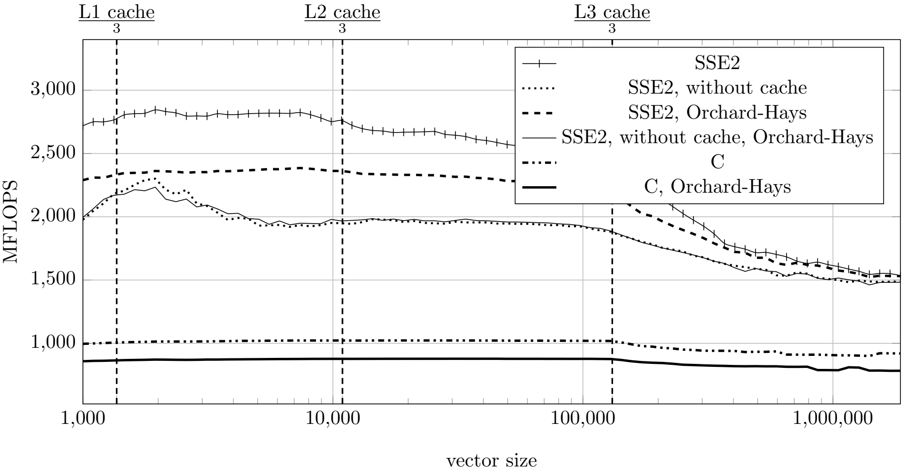


Figure 21
Performance comparison of our stable add implementations and the method of Orchard-Hays, with SSE2, using relative tolerance, for two vectors

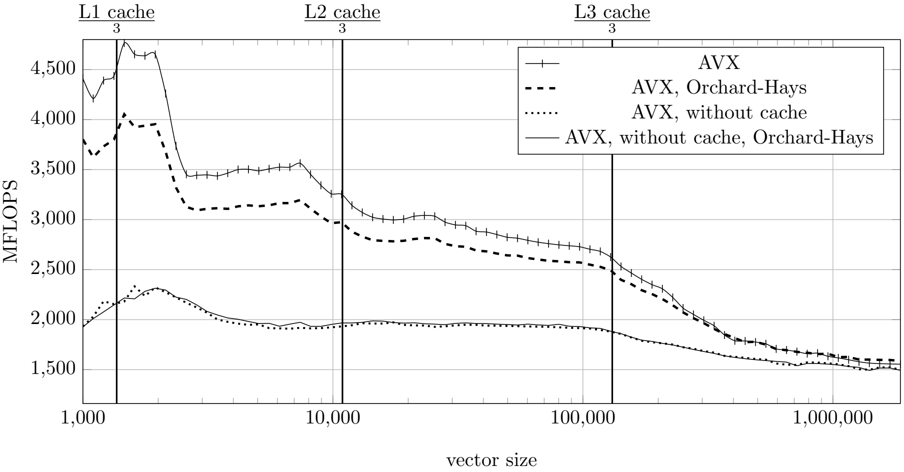


Figure 22
Performance comparison of our stable add implementations and the method of Orchard-Hays, with AVX, using relative tolerance, for two vectors

performance is better than the stable add. As Figure 23 shows, the performance of stable AVX dot product is close to the unstable AVX version. The stable SSE2 requires more cycles, so the performance is considerably lower than the unstable SSE2 version. The figure shows that if there is no SIMD support, the branching-free techniques can be very useful if the input vectors are sufficiently large.

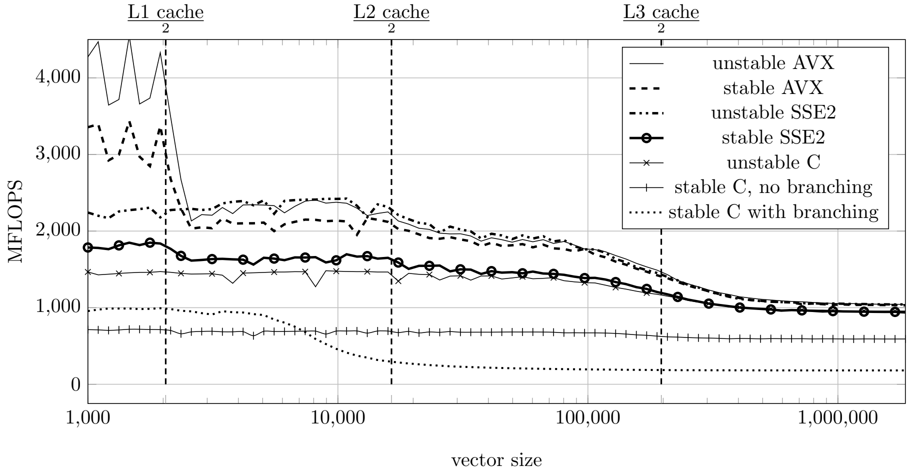


Figure 23
Performances of the dot product implementations

7 Conclusions

As the performance tests prove, our simplified stable add method is faster than Orchard-Hays's method. The applicability of our method is also tested by our simplex method implementation; the test problems of NETLIB were successfully solved. It is clear that our pointer arithmetic based stable dot-product implementation is much more efficient than the conditional branching version if the input vectors are sufficiently large. Moreover, the tests show that using Intel's SIMD instruction sets provides strong tools in order to implement the stable algorithms in an efficient way.

Modern Intel CPUs have at least two memory ports. So, while the next data set is loading from the memory, the CPU can execute complex computations on the previous set. This is why the AVX is so efficient in high performance stable computations.

References

[1] Ogita, T., Rump, S. M., and Oishi, S. (2005) Accurate sum and dot product. *SIAM J. Sci. Comput.*, **26**, 1955–1988.

[2] Langou, J., Langou, J., Luszczek, P., Kurzak, J., Buttari, A., and Dongarra, J. (2006) Exploiting the performance of 32 bit floating point arithmetic in obtaining 64 bit accuracy (revisiting iterative refinement for linear systems). *Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*, New York, NY, USA SC '06. ACM.

[3] Fousse, L., Hanrot, G., Lefèvre, V., Pélissier, P., and Zimmermann, P. (2007) Mpfr: A multiple-precision binary floating-point library with correct rounding. *ACM Trans. Math. Softw.*, **33**.

[4] Dekker, T. J. (1971) A floating-point technique for extending the available precision. *Numer. Math.*, **18**, 224–242.

[5] Waterloo Maple Inc. Maple.

[6] MathWorks, I. (2005) *Symbolic Math Toolbox for Use with MATLAB: User's Guide*. MathWorks, Incorporated.

[7] Wolfram Research Inc. Mathematica.

[8] Wolfram, S. (1999) *The Mathematica Book (4th Edition)*. Cambridge University Press, New York, NY, USA.

[9] Knuth, D. E. (1997) *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[10] Muller, J.-M., Brisebarre, N., de Dinechin, F., Jeannerod, C.-P., Lefèvre, V., Melquiond, G., Revol, N., Stehlé, D., and Torres, S. (2009) *Handbook of Floating-Point Arithmetic*, 1st edition. Birkh&#228;user Basel.

[11] Higham, N. J. (2002) *Accuracy and Stability of Numerical Algorithms*, second edition. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

[12] Hassaballah, M., Omran, S., and Mahdy, Y. B. (2008) A Review of SIMD Multimedia Extensions and their Usage in Scientific and Engineering Applications. *Comput. J.*, **51**, 630–649.

[13] Thomadakis, M. E. and Liu, J. (1997) An Efficient Steepest-Edge Simplex Algorithm for SIMD Computers. Technical report., College Station, TX, USA.

[14] Takahashi, A., Soliman, M. I., and Sedukhin, S. (2003) Parallel LU-decomposition on Pentium Streaming SIMD Extensions. In Veidenbaum, A. V., Joe, K., Amano, H., and Aiso, H. (eds.), *ISHPC*, October, Lecture Notes in Computer Science, **2858**, pp. 423–430. Springer.

[15] Orchard-Hays, W. (1968) *Advanced linear-programming computing techniques*. McGraw-Hill, New York.

[16] Intel (2013) *Intel 64 and IA-32 Architectures Software Developers Manual - Volume 2 (2A, 2B & 2C): Instruction Set Reference, A-Z.*

[17] Maros, I. and Mészáros, C. (1995) A numerically exact implementation of the simplex method. *Annals of Operations Research*, **58**, 1–17.

[18] Intel (2013) *Intel 64 and IA-32 Architectures Software Developers Manual - Volume 3A: System Programming Guide, Part 1.*

Comparison of Vector Operations of Open-Source Linear Optimization Kernels

Péter Böröcz, Péter Tar, István Maros

University of Pannonia
10. Egyetem Str., Veszprém, Hungary H-8200
Tel.: +36-88-624020
{borocz, tar, maros}@dcs.uni-pannon.hu

Optimization is a widely used field of science in many applications. Optimization problems are becoming more and more complex and difficult to solve as the new models tend to be very large. To keep up with the growing requirements the solvers need to operate faster and more accurately. An important field of optimization is linear optimization which is very widely used. It is also often the hidden computational engine behind algorithms of other fields of optimization. Since linear optimization solvers use a high amount of special linear algebraic vector operations their performance is greatly influenced by their linear algebraic kernels. These kernels shall exploit the general characteristics of large-scale linear optimization problem models as efficiently as possible. To construct more efficient linear algebraic kernels the critical implementational factors influencing operation performance were identified via performance analysis and are presented in this paper. With the results of this analysis a new kernel has been developed for the open-source linear optimization solver called Pannon Optimizer developed at the Operations Research Laboratory at the University of Pannonia. A novel application of indexed dense vectors is also introduced which is designed specifically for linear optimization solvers. Finally a computational study is performed comparing the performance of vector operations of different linear optimization kernels to validate the high efficiency of our kernel. It shows that in case of large scale operations the indexed dense vector outperforms the state-of-the-art open-source linear optimization kernels.

Keywords: Linear optimization; Simplex method; Optimization software; Computational linear algebra; Sparse data structures

1 Introduction

Nowadays, there are numerous performance-critical algorithms based on linear algebraic operations. As the efficiency of such algorithms is strongly influenced by the characteristics of the used linear algebraic kernels, the usage of appropriate data structures and their implementations are very important. In many cases it is possible to achieve better overall performance with a kernel exploiting the characteristics of the nature of the problem. A common characteristic is sparsity, which heavily appears in the field of linear optimization (LO). The implementation of data structures used by LO solvers is not a trivial matter. Several critical factors highly influence the performance of operations.

In this paper the key implementational factors of sparse linear algebraic data structures are gathered and analyzed. Based on these a linear algebraic kernel has been created for the open-source simplex-based LO solver, Pannon Optimizer [1], developed by the Operations Research Laboratory at the University of Pannonia. A specialized indexed dense vector has also been designed and implemented to maximize performance in such applications [2]. Usage and efficiency of such vectors have not been published in the literature of linear optimization. The aim of this paper is to show that using a specialized version of indexed dense storage can lead to major performance improvements in linear optimization.

Besides the used data structures LO has many interesting computational aspects. Numerical stability can be crucial depending on the nature of the problem since the side effects of floating point numerical computations can heavily affect the performance of the solution algorithm [3]. It can also happen that numerical errors lead to cycling or stalling [4] of the algorithm which can prevent it from finding the optimal solution in reasonable time. Although these aspects play important role in the performance of LO solvers they are out of the scope of this paper since these are typically handled with high level logics.

The paper is structured as follows. The introduction briefly describes sparse computing techniques and the commonly used linear algebraic data structures of an LO solver. It also includes the standard form of the LO problem but it does not explain LO in a detailed manner as the focus of the paper is about the implementation aspects of low-level data representation and usage. The second section discusses how linear optimization solvers benefit from the sparse data structures during vector transformations and highlight the importance of these operations. The third section introduces appropriate tools for benchmarking these data structures, while section four gives an overview of the most widely used open-source LO kernels and the Pannon Optimizer. Finally, the performance of the new linear algebraic kernel and two open-source kernels are compared using the CLP [5] and GLPK [6] solvers to support our findings.

1.1 Sparse computing techniques

If z denotes the number of nonzero valued elements in an m dimensional vector v the density of v is defined as: $\rho(v) = \frac{z}{m}$. Sparsity is nothing but low density. Vectors and matrices of real-life problems submitted to LO solvers are usually very sparse. To provide high-performance data structures for LO solvers it is necessary to exploit sparsity, which is a key issue for a meaningful implementation of the revised simplex method. Stored sparse vectors generally do not explicitly keep information about every element. They only store the index-value pairs of nonzero entries. The storage of sparse vectors is much more efficient in this way but this representation lacks the direct accessibility of elements. There are many factors influencing the performance of such data structures. As an example, element access can be speeded up by storing the pairs in a sorted order by indices. However, the initialization and insertion of a new element are slower due to the necessity of sorting (Table 1) [7].

Table 1
Operation complexity for different vector types

Operation	Dense	Sparse sorted	Sparse unsorted
Access element	$O(1)$	$O(log(z))$	$O(z)$
Initialization	$O(m)$	$O(z \cdot log(z))$	$O(z)$
Insert new element	$O(m)$	$O(log(z))$	$O(z)$

1.2 Operations of sparse linear algebraic kernels

Sparse linear algebraic kernels usually implement both dense and sparse vectors since using sparse storage techniques on vectors with high density is very inefficient. Because of this and the basic differences between dense and sparse vectors dense—dense, sparse—dense and sparse—sparse operations need to be implemented separately. The efficiency of these operations is heavily influenced by the characteristics of the vector implementations and the algorithm that use these structures. Since sparse—dense operations are faster most sparse—sparse operations are more efficient if one of the vectors is converted to dense representation and after the operation is executed the result is converted back to sparse.

Converting a sparse vector into dense representation is called scattering and converting a dense vector into sparse representation is called gathering. Scatter and gather are elementary operations of sparse linear algebraic kernels.

When a high amount of sparse—sparse operations is to be executed and the maximal dimension of the vectors is known it is more efficient to maintain a static array as working vector and use it for scattering sparse vectors. A static array means that it is a fixed dimension array allocated at the beginning of the algorithm and is kept throughout the whole solution process. This working vector needs to be cleared after every operation which can be done very efficiently if the nonzero pattern of the vector is known.

1.3 Linear optimization problems

The history of linear optimization was originated in the 1950's when Dantzig formulated the LO problem [8]. One of the formulations of the LO problem is the standard form:

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^T\mathbf{x}, \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$; $\mathbf{c}, \mathbf{x} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^m$.

The two main solution algorithms are the simplex method [9, 10, 11] and the interior point methods [12, 13] which give the global optimum of a linear optimization problem. In this paper, we show our results using the revised simplex method from which the main operations are highlighted in section 2. The simplex method is an iterative algorithm. The whole process is started from an initial basis and results in an optimal one if it exists, while the method iterates through a series of neighboring bases of the linear equation system. During these a large amount of linear algebraic operations must be performed. This paper is not intended to compare or evaluate different theoretical approaches of the simplex method but focuses on implementational details, which can be commonly applied for each method.

1.4 Linear algebraic data structures of LO solvers

In the past five decades the performance of LO solvers increased by 5 orders of magnitude. From this 3 orders of magnitude are due to the hardware development of computers and 2 are due to algorithmic improvements. The need for further performance improvements is still an issue, significant breakthroughs were not published in the past decade. Since the solution time of LO problems is crucial, efficient data structures can be exploited to achieve good optimization performance. Today's LO solvers generally use three different vector representations for storing operational data as they are shown in figure 1. They have different characteristics in term of operational complexity and memory requirements [7].

Dense storage

Value	2	0	0	5	0	-3

Sparse storage

Value	2	5	-3
Nonzero index	3	0	5
Dimension	6		

Indexed storage

Value	2	0	0	5	0	-3
Nonzero index	3	0	5			
Nonzero count	3					

Figure 1
The storage types used in large-scale LO solvers. Note, components of a vector are indexed $0, \ldots, m-1$.

Dense vectors are stored as simple arrays. This allows direct access to the elements and an efficient way of storage if the vector is static (e.g. an auxiliary working vector that speeds up some computations) and not very sparse. This storage is wasteful in case of sparse vectors since zeros are stored explicitly. Furthermore dense storage can cause serious memory issues if large-scale LO problems are considered.

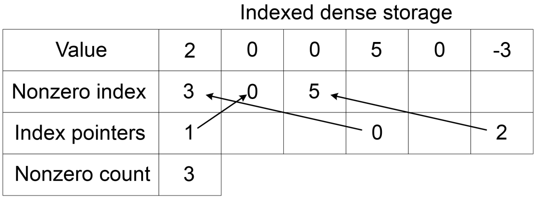
Sparse vectors are stored as index-value pairs. In this way, elements are only accessible via searching, but the storage need is low if the vector is highly sparse. The mathematical models of LO problems are generally stored in sparse format. Since the model is nothing but a sparse matrix it can be stored as a set of sparse row or column vectors. The state-of-the-art solvers usually sacrifice some memory and store the coefficient matrix in both ways in order to enhance computational efficiency of the simplex method [3].

Indexed storage of vectors maintains dense and sparse representations of a vector in parallel [14]. This enables direct element access and exploits sparsity at the same time. However, it uses considerably more memory. Operations executed with indexed vectors are even more efficient than with sparse vectors, but changing element values is costly due to the necessity of maintaining both representations. Indexed vectors are generally used in sparse operations that involve scattering a sparse vector into a dense array.

A fourth storage type is based the indexed dense vector introduced in [2], the modified version of this vector used for linear optimization is introduced by the kernel of Pannon Optimizer which is shown in figure 2 [15]. Indexed dense vectors are similar to indexed vectors with the addition of an index pointer vector. This index pointer vector is a full-length array connecting the nonzero values of the dense representation to their copies in the sparse representation. If the element at index i is nonzero the index pointer vector has a pointer at index i pointing to the sparse representation value i. With this the complexity of changing values is reduced to constant. Indexed dense vectors generally offer better performance than traditional indexed vectors as it will be shown in section 5. There are several differences with our vector implementation and the previously published method. The differences of our indexed dense implementation is as follows:

- In our case the vector uses permanent storage capacity for each indexed dense vector while traditional methods collect the indices of nonzero pointers for temporary usage only. In our case the pointers are stored and maintained while temporary usage of these pointers have to be initialized at the beginning of each vector operation.
- To utilize index pointers the temporary storage only exploits the pointers for one operand of the vector operation.
- Traditional methods do not use index pointers to handle canceled nonzero elements. Numerically sensitive situations often generate many zero elements which shall be noticed to maintain the sparsity of the representation and the efficiency of further calculations.

An LO implementation usually uses multiple vector types to store data. Matrix $\mathbf{A}$ is represented using sparse storage for real-life large-scale problems because with dense methods it can take a huge amount of memory. For example storing a matrix with $n = m = 50000$ takes 20 GB of RAM using double precision floating point numbers which makes it impossible to handle with commonly available computing hardware. Conversely, sparse-sparse operations are not efficient on their own, because the nonzeros are usually not ordered (ordering takes computational time) and



Indexed dense storage

Value	2	0	0	5	0	-3
Nonzero index	3	0	5			
Index pointers	1			0		2
Nonzero count	3					

Figure 2
The indexed dense storage type. Note, components of a vector are indexed $0,\dots,m-1$.

searching must be used to access elements (Table 1). To avoid searching working vectors containing dense arrays (dense, indexed or indexed dense) should be used to scatter the elements thus making them easily available for computation. When a series of computations should be done using the same vectors this working vector is extremely valuable and the result can be quickly gathered to a sparse format. In section 5 the impact of using different vector types as a working vector is investigated in detail.

2 Computational elements of the simplex method

In this section, the most commonly used elementary linear algebraic operations of the revised simplex method are presented. The following data structures are used in the state-of-the-art simplex based LO solvers to achieve high performance [3]. The tools supporting the identification of critical implementational factors are performance analysis tools presented in section 3.

When we deal with sparse problems the application of the revised simplex method is inevitable which uses some special representation of the basis. The two main representations are the Lower-Upper (LU) factorization [16] and the Product Form of the Inverse (PFI) [17]. Most of the solvers use the LU factorization, but it has been shown that none of them is superior because the PFI with a proper implementation can be as good as the LU [18] in several cases. The two most time consuming linear algebraic operations of the simplex algorithm are the FTRAN and BTRAN operations [3]. They involve the computation of $\mathbf{B}^{-1}\mathbf{a}$ and $\mathbf{a}^T\mathbf{B}^{-1}$, where $\mathbf{a} \in \mathbb{R}^m$, and $\mathbf{B}$ is the actual basis. In the formulas below we use the PFI form to represent the computational aspects of vector transformations but it can be adopted to the LU form as well.

FTRAN is a sequence of simple elementary transformations of the form [3]:

$$\begin{bmatrix} 1 & & \eta 1 & & \\ & \ddots & \vdots & & \\ & & \eta^p & & \\ & & \vdots & \ddots & \\ & & \eta^m & & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_p \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} a_1 + a_p\eta^1 \\ \vdots \\ a_p\eta^p \\ \vdots \\ a_m + a_p\eta^m \end{bmatrix}, \tag{1}$$

which can be written in vector form as:

$$\mathbf{c} = \mathbf{a} + \lambda \mathbf{b} \tag{2}$$

This is called a daxpy product of two vectors if double precision values are used. Since we are dealing with sparse basis representations the daxpy product is widely used on sparse vectors throughout the solution process.

BTRAN can be decomposed similarly. It is also a sequence of elementary steps, which are of the form:

$$[a_1,\ldots,a_p,\ldots,a_m]\begin{bmatrix} 1 & & \eta 1 & & \\ & \ddots & \vdots & & \\ & & \eta^p & & \\ & & \vdots & \ddots & \\ & & \eta^m & & 1 \end{bmatrix} = \left[a_1,\ldots,\sum_{i=1}^{m} a_i \eta^i,\ldots,a_m\right]. \tag{3}$$

It can be written in vector form as a dot product:

$$a'_p = \mathbf{a}^{\mathbf{T}} \eta \tag{4}$$

Since these computations usually take $> 50\%$ of the total solution time, the performance of them is critical and the data structures must be highly efficient.

3 Tools for performance analysis

Performance of different implementations of a given sparse linear algebraic operation can be compared by measuring the execution times on a fixed architecture. Some open-source tools exist to make such measurements. However, creating a comprehensive performance analysis is cumbersome due to the lack of tools to process and display the obtained data. As a measuring engine the Blazemark benchmark suite of the open-source Blaze library was used [19].

The Blazemark suite has several features making it a good choice to measure performance of sparse linear algebraic operations. It gives measurement results in million floating point operations per second (MFLOPS) rather than execution time making comparison between different vector sizes and the theoretical peak of the processor possible. In order to provide credible results for sparse computations as well, only the necessary floating point operations are considered. It means that the minimal number of additions that must be computed in order to get the proper result. In case of sparse-sparse addition the number of effective operations is determined by the number of nonzero elements rather than vector dimensions. Blazemark also makes measurements iteratively to filter out false results. It can be parameterized to measure operations with data structures of the desired size and sparsity.

In order to extend the functionality of the Blazemark suite and support advanced measurements and performance analysis we have created a tool called Blazemark-Analyser [15]. It is able to configure and run the Blazemark suite, parse the test results and store them in a database. It supports library-specific parameterization such as tolerances or other simplex-related numerical settings. It is also able to execute parameter sweeps and draw a plot of performance as a function of a given parameter (e.g. dimension or sparsity). The software offers a graphical user interface and can be used conveniently to interpret and compare the results. This software makes it possible to identify how different implementational factors influence operational performance. With the help of performance analysis a new linear algebraic kernel was created for Pannon Optimizer.

4 Kernels of open source LO solvers

There are several open-source LO solvers from which only a few is capable of solving large-scale LO problems. The linear algebraic kernels used in them need to fulfil the requirements of the LO solver and offer the best possible performance. This section describes the main characteristics of the kernels of two of the most widely used open-source LO solvers GLPK [20], CLP [21] and the Pannon Optimizer.

4.1 GNU Linear Programming Kit

The GNU Linear Programming Kit (GLPK) is a collection of ANSI C implementations of mathematical optimization solvers including linear optimization [6]. The GLPK kernel uses computer memory very efficiently. It allocates blocks and stores vectors in a way to minimize the caching operations of the processor. Just as other solver algorithms it implements the revised simplex method. GLPK has a lightweight implementation of dense, sparse and indexed vector representations with minimal overhead. All vector types consist of only the necessary arrays and the vector operations are also implemented without any overhead of sophisticated implementations. This implies that it does not pay particular attention to numerical stability at operational level.

4.2 COIN-OR Linear Program solver

The COIN-OR Linear Program solver (CLP) is an open-source large-scale optimization problem solver written in C++ [5]. It includes an object-oriented implementation of the revised simplex method. The linear algebraic kernel of CLP offers three vector types for the solver: dense, sparse and indexed representations. Dense vectors are implemented traditionally using arrays. The sparse vector representation is sorted by index and only used to store the mathematical model. Indexed vector representation is used during all sparse operations. The CLP kernel is capable of mitigating the negative effects of numerical problems by setting elementary operation results below a given threshold to a pre-determined tiny value. It should be noted that throughout the solution process of CLP it overrides the default behavior of its own kernel with more efficient array operations (similar to GLPK).

4.3 Pannon Optimizer

Pannon Optimizer [1] is a large-scale LO solver using a high-level C++11 implementation of the revised simplex method [22]. It is being developed specifically for research purposes, making the performance impact of subalgorithms measurable. The linear algebraic kernel of Pannon Optimizer was developed considering the results of performance analysis with BlazemarkAnalyzer. It implements dense and sparse vector representations as well as the indexed dense vector which is a uniquely extended implementation of the indexed vector.

5 Computational study

This section presents the summarized results of a computational study on the linear algebraic kernels of the solvers mentioned above. The testing environment was a laptop computer with an Intel Core i5-3230M CPU with fixed clock speed at 2.60GHz, 3MB L3 cache and AVX (Advanced Vector Extensions) support, with 4 Gb DDR3 RAM. The operating system was an Ubuntu 14.04 64-bit system.

The theoretical peak performance of this system is 10400 MFLOPS with 4 double precision floating point operations per clock cycle (2 additions and 2 multiplications). BlazemarkAnalyzer was used for the measurements and it also provided the diagrams that we present in this section. All the figures show the results normalized according to the maximal measured MFLOPS value on the Y axis of the diagrams. The X axis showing the dimensions of the vectors uses a logarithmic scale.

The test vectors used in our performance analysis were composed in a way to mimic the structure of real LP problems. In order to achieve this we have used different vector patterns throughout the measurements. In case of vector additions 70% of the operations were standard additions where the operands and result are nonzero. 10% of the operations were cancellations, where the operands are nonzero but the result is numerically zero. The last 20% were non-overlapping values, thus one of the operands of these operations was zero. In case of dot product operations the vectors did not have the same number of nonzero elements. During the measurements of dot product operations $\mathbf{a}^\mathbf{T}\mathbf{b}$ was always performed together with $\mathbf{b}^\mathbf{T}\mathbf{a}$. This measures the effect of traversing different nonzero patterns.

In the case of dense operations such as the dense—dense vector addition (Figure 3) or the dense—dense vector daxpy product (Figure 4) low-level memory management can result in significantly better performance. This means that simple array implementations are the best for dense—dense operations.

In the case of sparse—sparse vector dot products kernels with sophisticated memory management perform significantly better (Figure 5). Since this operation does not change values in either vectors and the dot product implementation is based on the nonzero indices, the difference between using a sparse, an indexed vector or an indexed dense vector is negligible. This is one of the most performance-critical operations of LO solvers since it forms the foundation of an elementary step of the

BTRAN operation. The results prove that with the application of performance analysis very high performance of linear algebraic operations can be achieved. When it comes to large-scale simplex specific operations the performance of Pannon Optimizer kernel exceeds the other two kernels which it has been compared to.

For sparse operations that result in the change of values a working vector having a dense representation is advisable since during the FTRAN operation multiple daxpy products are to be done using the same vector. The performance of such operations can be significantly increased with the use of indexed dense vector instead of regular indexed vector (Figure 6). The extent of this improvement depends on the number of newly created zero elements. Changing a nonzero element to zero in an indexed vector has logarithmic or linear complexity depending on whether the vector is sorted or not, while in an indexed dense vector it has constant complexity. In the case of operations with very small vectors in dimension lightweight kernels with minimized overheads (additional computing only needed for memory management) perform better.

It is not trivial whether the traditional indexed vector or the new indexed dense vector would perform better as a static operation vector for LO solvers. To examine this issue, numerous measurements were made to compare the indexed dense vector of Pannon Optimizer with regular indexed vector implementations used in CLP and GLPK. When adding a sparse vector to an indexed or indexed dense vector with values of opposite sign where the result is expected to be a null-vector, the operation performance of regular indexed vectors reduces logarithmically with the growth of the dimension of the vectors. In the case of indexed dense vectors operation performance does not reduce (Figure 7). This further emphasizes the efficiency of the indexed dense vector.

When comparing the performance of traditional indexed vectors with indexed dense vectors on the dot product operation with dense vectors the results show that when working with small vectors regular indexed vectors are advisable to be used but in the case of large vectors the indexed dense vector performs significantly better (Figure 8). These characteristics prove that the indexed dense vector is a very efficient working vector of large-scale LO solvers.

Conclusions

The linear algebraic kernel of the Pannon Optimizer was developed, based on results of the performance analysis of sparse data structures and with consideration of computationally heavy simplex-specific operations such as FTRAN and BTRAN. This investigation led us to develop and release a high performance sparse linear algebraic kernel that performs better than its predecessors for solving linear optimization problems.

We have introduced a new kind of indexed vector based on the experiences that we gathered from its alternatives. It appears that our vector type performs very well as a static working vector of large-scale LO solvers, surpassing the performance of traditional indexed vector implementations that most LO solvers use. When working with large vectors, both sparse and dense operations are faster with the new vector

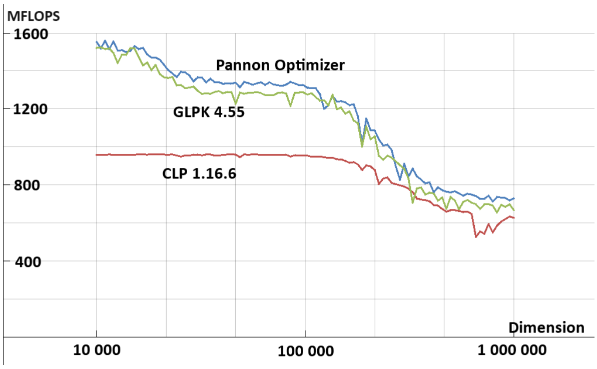


Figure 3
Performance of different implementations of the addition of two dense vectors.

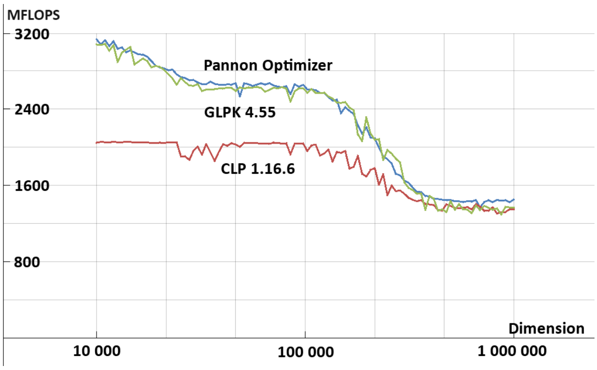


Figure 4
Performance of different implementations of the daxpy product of two dense vectors.

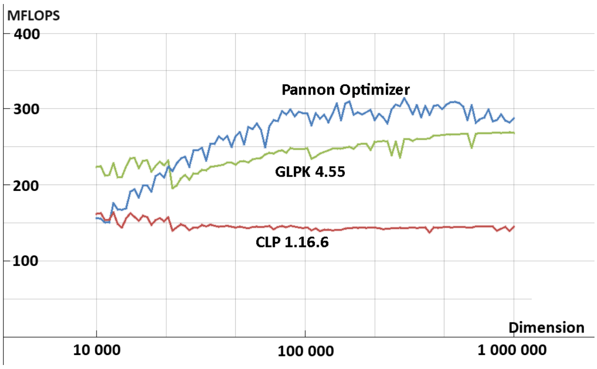


Figure 5
Performance of different implementations of the dot product of two sparse vectors (0.1% density).

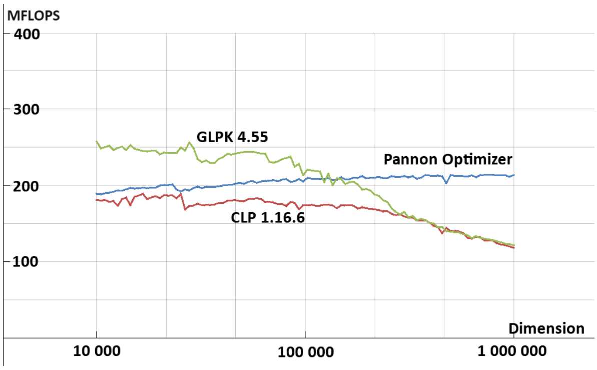


Figure 6
Performance of different implementations of the daxpy product of two sparse vectors (0.1% density) using an indexed (CLP, GLPK) or an indexed dense (Pannon Optimzier) static working vector.

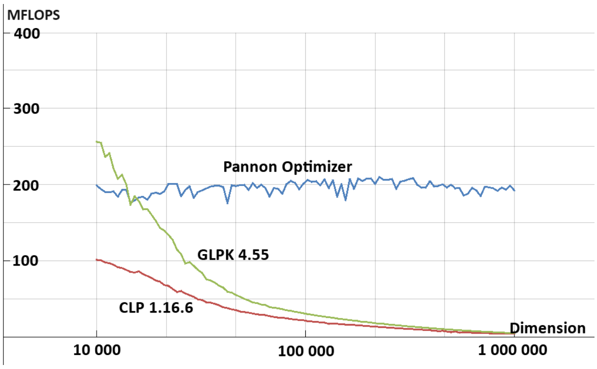


Figure 7
Performance of the addition of a traditional indexed vector (CLP, GLPK) or an indexed dense vector (Pannon Optimizer) (0.1% density) and a sparse vector (0.1% density) using a nonzero pattern where the result of the addition is algebraically a null-vector.

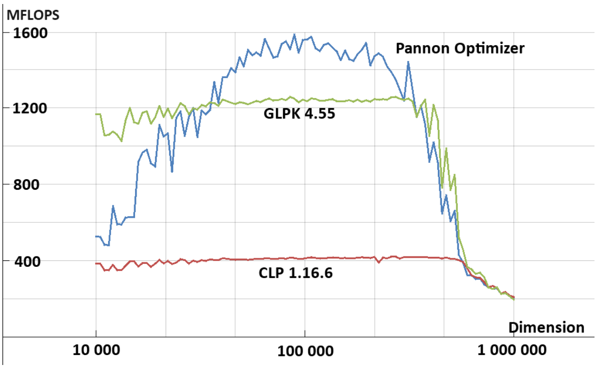


Figure 8
The performance of the dot product operation of a traditional indexed vector (CLP, GLPK) or an indexed dense vector (Pannon Optimizer) (0.1% density) and a dense vector.

type that has been validated by performance analysis. As a conclusion of the performance analysis, we highly recommend the usage of the indexed dense vector if the dimension of the vectors is greater than 10^4. It can also be noted that the usage of the indexed dense vector instead of other vector types does not affect the performance negatively if used as a static working vector. Altogether, the overall performance of the linear algebraic kernel of the Pannon Optimizer seems to be better than kernels of other open-source, large-scale LO solvers.

Acknowledgement

This publication/research has been supported by the European Union and Hungary and co-financed by the European Social Fund through the project TÁMOP-4.2.2.C-11/1/KONV-2012-0004 - National Research Center for Development and Market Introduction of Advanced Information and Communication Technologies.

References

[1] University of Pannonia: Pannon Optimizer, http://sourceforge.net/projects/pannonoptimizer/, Online, 2017.09.

[2] S. Pissanetzky: Sparse Matrix Technology, Academic Press, 1986.

[3] I. Maros: Computational Techniques of the Simplex Method, Kluwer Academic Publishers, Norwell, Massachusetts, 2003.

[4] M. Padberg: Linear Optimization and Extensions, Springer, Berlin, 1999.

[5] COIN-OR Project: COIN-OR Linear Optimization Solver, https://projects.coin-or.org/Clp, Online, 2017.09.

[6] GNU Project: GNU Linear Programming Kit, https://www.gnu.org/software/glpk/, Online, 2017.09.

[7] I. Maros: Essentials of Computational Linear Algebra, Department of Computer Science, Veszprém, 2009.

[8] G. B. Dantzig: Maximization of a Linear Function of Variables Subject to Linear Inequalities In T.C. Koopmans, editor, *Activity analysis of production and allocation*, pages 339–347. Wiley, 1951.

[9] G. B. Dantzig: Linear Programming and Extensions, Princeton University Press, 1963.

[10] K. G. Murty: Linear and Combinatorial Programming, John Wiley & Sons, 1976.

[11] A. Prékopa: A Very Short Introduction to Linear Programming, RUTCOR Lecture notes, pages 2–92, 1992.

[12] C. Roos, T. Terlaky, and J-Ph Vial: Theory and Algorithms for Linear Optimization: An Interior Point Approach, John Wiley & Sons, 1997.

[13] T. Illés and T. Terlaky: Pivot Versus Interior Point Methods: Pros and Cons, European Journal of Operations Research, 140:6–26, 2002.

[14] A. Koberstein: Progress in the dual simplex algorithm for solving large scale LP problems: techniques for a fast and stable implementation, Computational Optimization and Applications, 41(2):185–204, 2008.

[15] P. Böröcz, P. Tar, and I. Maros: Performance Analysis of Sparse Data Structure Implementations, MACRo 2015, 1(1):283–292, 2015.

[16] H. M. Markowitz: The Elimination Form of the Inverse and Its Application to Linear Programming, Management Science, 3(3):255–269, 1957.

[17] G.B. Dantzig and Wm. Orchard-Hays: The Product Form for the Inverse in the Simplex Method, Mathematical Tables and Other Aids to Computation, 8(46):64–67, 1954.

[18] P. Tar and I. Maros: Product Form of the Inverse Revisited, OpenAccess Series in Informatics (OASIcs), 22:64–74, 2012.

[19] K. Iglberger, G. Hager, J. Treibig, and U. Rüde: Expression Templates Revisited: A Performance Analysis of the Current ET Methodology, SIAM Journal on Scientific Computing, 34(2):42–69, 2012.

[20] J. L. Gearhart, K. L. Adair, R. J. Detry, J. D. Durfee, K. A. Jones, and N. Martin: Comparison of Open-Source Linear Programming Solvers, Technical report, 2013.

[21] B. Meindl and M. Templ: Analysis of Commercial and Free and Open Source Solvers for the Cell Suppression Problem, Transactions on Data Privacy, 6:147–159, 2013.

[22] B. Stágel, P. Tar, and I. Maros: The Pannon Optimizer - A Linear Programming Solver for Research Purposes, Proceedings of the 5th International Conference on Recent Achievements in Mechatronics, Automation, Computer Science and Robotics, 1(1):293–301, 2015.

Secondary Stochastic Processes and Storage Reservoir Optimization

András Prékopa, **Tamás Szántai**, **István Zsuffa**

Department of Differential Equations, Budapest University of Technology and Economics, Műegyetem rkp. 3-9, 1111 Budapest, Hungary
e-mail: szantai@math.bme.hu

Abstract: In the first two sections of the paper, stream flow is investigated on a probability theoretical basis. We will show that under some realistic conditions its probability distribution is of gamma type. In the model of the third section the optimal capacity of a storage reservoir is determined. In the model of the fourth section optimal water release policy is sought, given that water demands should be met by a prescribed large probability. Finally, in the last fifth section, in addition to the before mentioned reliability type constraint an upper bound is imposed on the number of days when demands may not be met and the cost of the intake facility is to be minimized[1].

Keywords: reservoir capacity; release policy; stochastic programming

1 Secondary Stochastic Processes Derived by a Poisson Process

The use of Poisson type stochastic processes is frequent in hydrology. Presently, we assume that the sequence of rainfall events follows a Poisson process. That is, if $\xi(I)$ denotes the (random) number of rainfalls in a time interval *I*, then

a) for all $I_1,\ldots,I_n$ interval systems, where any two intervals have no common inner points, the random variables $\xi(I_1),\ldots,\xi(I_n)$ are independent,

b) $\xi(I)$ has Poisson distribution with parameter $\lambda(I) \geq 0$.

[1] The problem of finding storage reservoir capacity was formulated by István Zsuffa many years ago. The detailed elaboration of the problem is more recent and is due to the first two authors who offered the Hungarian version of this paper appeared in Alkalmazott Matematikai Lapok **27** (2010) 175-188 to the memory of their friend and co-worker, István Zsuffa. The first author many years ago planned to publish the paper in English, too. After András Prékopa passed away last year, this task remained to the second author, who offers this paper to the memorial volume of Acta Polytechnica Hungarica.

A secondary process derived by a Poisson process means that to the random events of the Poisson process, in our case to the time points of rainfalls, a random secondary phenomenon is ordered, which is now a random flood wave. Let denote the random field of the secondary events *Y*. On this random field more probability measures are defined. For discussing secondary processes an appropriate tool is the so called product space method, see [3]. This consists of regarding the secondary process in the set of the element pairs (*t*, *y*), with other words in the product space $(T \times Y)$, where *T* is a subset of the time axis and *t* is one of its elements. A special run of the secondary process, that is its realization means a random point system in the space $T \times Y$. Indeed, if $K, t_{-1}, t_0, t_1, t_2, K$ is the Poisson-type point process and $K, y_{-1}, y_0, y_1, y_2, K$ is the series of the appropriate secondary phenomena then the realization of the secondary process can be characterized by the

$$K, (t_{-1}, y_{-1}), (t_0, y_0), (t_1, y_1), (t_2, y_2), K$$

random point system in the space $T \times Y$.

The main theorem of the product space theory on secondary processes [3] claims the following.

If the selection from the space *Y* of the secondary phenomena belonging to different points of the Poisson process is serially independent and identically distributed with the same probability measure μ, then the random point system in the space $T \times Y$ is also of Poisson type with parameter measure $\lambda \times \mu$.

It may occur that the secondary phenomena belonging to the points of the Poisson process are serially independent but their probability distribution depends on *t*. This means that the recession of a flood wave depends on the time when the flood wave was initiated. In this case, one has to use measures μ_t instead of the single measure μ. Then the parameter measure belonging to a set *D* of the random Poisson type point system in the product space, is determined by the following integral

$$\int_C \mu_t(D_t)\lambda(\mathrm{d}t), \tag{1}$$

where *C* is the projection of *D* on the set *T* and D_t is the intersection of the set *D* with that subset of $T \times Y$ on which *t* is constant i.e. $D_t = \{y \mid (t, y) \in D\}$.

The number of random points belonging to the set *D* of the product space $T \times Y$ can be denoted as $\eta(D)$. So integral (1) equals to $E(\eta(D))$.

For simplicity we suppose in the following that μ_t is independent of *t*.

A different treatment of the theory of secondary processes can be found in [10].

2 Streamflow Probability Model Based on the Theory of Secondary Processes

Let the flow response to rainfall depth κ be characterized by function $f(t,\kappa)$, where κ is a random variable. One possible empirical version of this function is

$$f(t,\kappa)=\kappa t^{\alpha-1}\mathrm{e}^{-\beta t}, t\geq 0\,, \tag{2}$$

where α and β are positive parameters depending on watershed characteristics.

Let the relationship between rainfall and runoff at time point t_i be described by the function

$$f(t-t_i,\kappa_i),\ t\geq t_i\ , \tag{3}$$

where the random variables κ_i are serially independent. Streamflow η_t is described by the superposition of the functions (3), i.e. the function:

$$\sum_{t_i\leq t} f(t-t_i,\kappa_i)=\xi_t\,.$$

We determine the probability distribution of the random variable ξ_t for the case of function (2).

From our main theorem it follows that the number of runoff events between limits (*a*,*b*) follows Poisson distribution with a parameter given by the integral

$$\int_{-\infty}^{t} P\left(a\leq\kappa(t-x)^{\alpha-1}\mathrm{e}^{-\beta(t-x)}\leq b\right)\lambda(\mathrm{d}x)\,. \tag{4}$$

In the case of *a*=*y*, *b*=*y*+d*y* and supposing that $\lambda(\mathrm{d}x)=\lambda\mathrm{d}x$, where $\lambda\geq 0$ constant, we get for this:

$$\int_{-\infty}^{t} P\left(y\leq\kappa(t-x)^{\alpha-1}\mathrm{e}^{-\beta(t-x)}\leq y+\mathrm{d}y\right)\lambda\mathrm{d}x$$

$$=\int_{0}^{\infty} P\left(y\leq\kappa v^{\alpha-1}\mathrm{e}^{-\beta v}\leq y+\mathrm{d}y\right)\lambda\mathrm{d}v$$

$$=\int_{0}^{\infty}\frac{\mathrm{d}}{\mathrm{d}y}\left[1-\mathrm{e}^{-y\delta\mathrm{e}^{\beta v}v^{1-\alpha}}\right]\mathrm{d}y\lambda\mathrm{d}v$$

$$=\lambda\int_{0}^{\infty}\delta\mathrm{e}^{\beta v}v^{1-\alpha}\mathrm{e}^{-y\delta\mathrm{e}^{\beta v}v^{1-\alpha}}\mathrm{d}v\mathrm{d}y$$

where κ is exponentially distributed with expected value $1/\delta$. It is not essential to suppose the exponential distribution; we may use any other probability distribution, too.

The probability distribution function of streamflow at time point t can be determined in the following way. Let denote $\eta(I)$ the number of individual runoff events in interval *I*. Then according to the earlier results $\eta(I)$ is Poisson distributed with parameter (4) in the case of $I=(a,b)$. Accordingly, the characteristic function of the probability distribution we are looking for is:

$$\mathrm{e}^{\int_0^\infty \left(\mathrm{e}^{iuy}-1\right)E(\eta(\mathrm{d}y))} = \mathrm{e}^{\lambda\int_0^\infty \left(\mathrm{e}^{iuy}-1\right)\left[\int_0^\infty \delta\mathrm{e}^{\beta v} v^{1-\alpha}\mathrm{e}^{-y\delta\mathrm{e}^{\beta v} v^{1-\alpha}}\mathrm{d}v\right]\mathrm{d}y} \tag{5}$$

In the case of $\alpha=1$ we get as result:

$$\mathrm{e}^{\lambda\int_0^\infty \left(\mathrm{e}^{iuy}-1\right)\frac{\mathrm{e}^{-\delta y}}{\beta y}\mathrm{d}y}$$

which is the characteristic function of a gamma distribution. Namely, if $\alpha=1$, then the equation (5) can be continued as

$$\mathrm{e}^{\lambda\int_0^\infty \left(\mathrm{e}^{iuy}-1\right)\left[\int_0^\infty \delta\mathrm{e}^{\beta v} v^{1-\alpha}\mathrm{e}^{-y\delta\mathrm{e}^{\beta v} v^{1-\alpha}}\mathrm{d}v\right]\mathrm{d}y} = \mathrm{e}^{\lambda\int_0^\infty \left(\mathrm{e}^{iuy}-1\right)\frac{1}{\beta y}\mathrm{e}^{-\delta y}\mathrm{d}y} \tag{6}$$

The form (6) of the characteristic function of gamma distribution can be found on page 92 of book [2].

Considerations applied in this section can be transferred to different, possibly more complicated $f(t,\kappa)$ functions that include rainfall-runoff relationships too. The result not necessarily can be expressed by a formula; however, it always can be calculated numerically. As a result we can always provide the probability distribution of ξ_t.

3 A Stochastic Programming Model for Determining the Optimal Capacity of Irrigation Reservoirs

Let us regard consecutive time sections (periods) and introduce the following notations:

η_k water demand in period *k*: $\eta_k = h_k - \gamma_k$, where h_k is constant meaning the total amount of demand, γ_k is the amount of rainfall in period *k*

ξ_k	streamflow in period *k*
m	storage capacity, the decision variable
M	reasonable upper bound for the capacity *m*
$c(m)$	cost of the reservoir as a function of its capacity
$\varsigma_k = \min(m, \xi_k)$	amount of water released in period *k*
c_k	benefit per water unit in period *k*
K	number of periods
N	number of years
$p > 0$	inflation rate supposed to be constant up to year *N*

Let us suppose that the damage in period *k* is proportional to the amount of water shortage.

The model to be discussed can be formulated for the case of nonlinearly increasing penalty, too.

The random amount of damage generated in period *k* is described by the random variable

$$\chi_k = c_k\left[(\eta_k - \varsigma_k)\right]_+ = \begin{cases} c_k(\eta_k - \varsigma_k), & \text{if } \eta_k > \varsigma_k \\ 0, & \text{otherwise.} \end{cases}$$

Regarding the number of *K* consecutive periods, the expected value of the total amount of generated damages will be $\sum_{k=1}^{K} E(\chi_k)$. If we want to minimize the expected value of the total amount of generated damages summarized over the current and the next consecutive *N* years, then regarding the expected present value of the damages, we have to solve the following optimization problem:

$$\min\left[c(m) + \sum_{i=0}^{N}\left(\sum_{k=1}^{K} E(\chi_k)\right)\frac{1}{(1+p)^i}\right], \quad \text{supposing} \quad 0 \le m \le M \tag{7}$$

Problem (7) is a single variable optimization problem, and the minimum of the objective function is sought on the interval $[0, M]$. We show that the sum in the objective function is a convex function of *m*. It is enough to show the convexity for one term of the sum. Let G_k and F_k denote the probability distribution function of the random variables η_k resp. ξ_k, and f_k the probability density function according to F_k. Then by definition of the random variable ς_k we get:

$$\frac{1}{c_k}E(\chi_k)=E([\eta_k-\varsigma_k]_+)$$

$$=\int_0^m E([\eta_k-z]_+)f_k(z)\mathrm{d}z+\int_m^\infty E([\eta_k-m]_+)f_k(z)\mathrm{d}z \tag{8}$$

$$=\int_0^m\left(\int_z^\infty(1-G_k(x))\mathrm{d}x\right)f_k(z)\mathrm{d}z+\int_m^\infty(1-G_k(x))\mathrm{d}x(1-F_k(m))$$

$$=\int_0^m\int_0^\infty(1-G_k(y+z))f_k(z)\mathrm{d}y\mathrm{d}z+\int_m^\infty(1-G_k(x))\mathrm{d}x(1-F_k(m))$$

Here we used the fact that if a random variable ξ has probability density function $f(x)$, probability distribution function $F(x)$ and its expected value exists, then it is easy to check by partial integration that for any real number z we have

$$E([\xi-z]_+)=\int_z^\infty(x-z)f(x)\mathrm{d}x=\int_z^\infty(1-F(x))\mathrm{d}x$$

One can check the convexity of the function $(1/c_k)E(\chi_k)$ by differentiating twice the formula (8). As $c_k>0,\ k=1,\mathrm{K},N$, it follows that $E(\chi_k)$ and the sum of these is also convex. As $p>0$ it is clear that the expected damage summed for *N* years and transformed to present value is also a convex function of the variable *m*. If the function $c(m)$ is also convex then the whole objective function is convex. However, if $c(m)$ is not convex, then the convexity of the objective function cannot be proved, but in some special cases it may be convex as it can be seen also in our example. The optimization can be done relatively simply. The distribution of streamflow can be selected to be gamma and the distribution of water demands can be supposed to be normal or gamma, too.

The model (7) can be extended by prescribing reliability type constraints for the random water demand to be met with a high probability.

We will illustrate the model (7) with an example provided in [5] including the stochastic programming model applied to a serially linked water reservoir system. Now we regard only the first reservoir out of the two serially linked reservoirs for three consecutive periods (June, July and August). We suppose that the random variables η_1,η_2,η_3, describing the random water demands, are independent of each other and the random streamflow is gamma distributed with the following parameters:

Table 1

Parameters of the gamma distributed random water demands

	expected value (m^3)	standard deviation(m^3)	ϑ	λ
η_1	215 760	327 120	0.000 002 016	0.435 038 479
η_2	433 608	243 600	0.000 007 307	3.168 400 000
η_3	484 416	214 368	0.000 010 541	5.106 426 041

Similarly we suppose that the random variables ξ_1, ξ_2, ξ_3, describing random streamflow values are independent of each other and of the random water demands and have gamma distribution with the following parameters:

Table 2

Parameters of the gamma distributed random streamflow values

	expected value (m^3)	standard deviation(m^3)	ϑ	λ
ξ_1	464822	186984	0.000013295	6.179658245
ξ_2	320576	266040	0.000004529	1.452005071
ξ_3	266040	234040	0.000004857	1.292152284

The cost in HUF of a reservoir with capacity *m* let be the following piecewise linear function

$$c(m) = \begin{cases} 100m, & \text{if } m \le 500000 \\ 50000000 + 150(m - 500000), & \text{if m} > 500000 \end{cases}$$

and let us suppose that we cannot build up any reservoir with capacity greater than $25\,000\,000\, m^3$.

The benefit of water/ m^3 in the consecutive periods let be $c_1 = 200$ HUF, $c_2 = 300$ HUF, $c_3 = 250$ HUF. Let $N = 10$ and the constant inflation rate $p = 0.05$. Then the single variable optimization problem (7) can be solved by some standard Matlab routines (gamma, gammainc, quad, dblquad, fminbnd).

The optimal solution of the above described test problem is $m = 580\,391\, m^3$ and the optimum value according to this solution equals to $523\,146\,000$ HUF. Fig. 1 shows the objective function values of the optimization problem (7) for its whole domain.

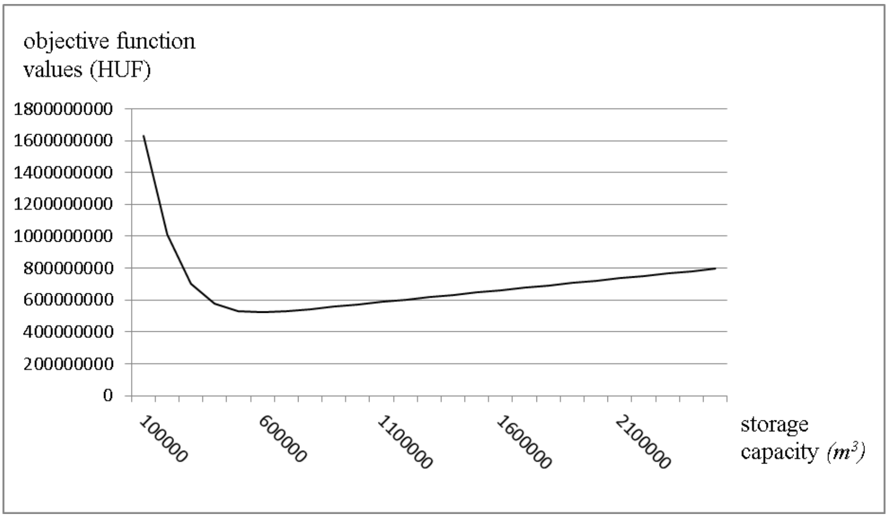


Figure 1
Diagram of the objective function values of optimization problem (7)

4 Optimization of Reservoir Release Policy

Let us regard consecutive periods and introduce the following notations:

ξ_0 amount of water in the reservoir at beginning the first period

ξ_k amount of streamflow in period *k*

$a_k (b_k)$ smallest (largest) allowed amount of water in the reservoir in period *k*

z_k amount of release in period *k*, the decision variable

N number of periods

$f(z_1,\mathrm{K}\ ,z_N)$ present value of the benefit of released water $z_1,\mathrm{K}\ ,z_N$ in consecutive periods

m reservoir capacity

$c(m)$ cost of the reservoir as a function of its capacity

K upper bound - the cost of building the reservoir with capacity *m*

p reliability level prescribed, close to one

The optimization problem is formulated as

$$\max\left[f(z_1,\ldots,z_N)-c(m)\right] \quad \text{supposing that}$$

$$P\left\{a_k \le \xi_0+\sum_{j=1}^{k}\xi_j-\sum_{j=1}^{k}z_j \le b_k,\ k=1,\ldots,N\right\}\ge p \tag{9}$$

$$0\le z_k \le m, \quad k=1,\ldots,N\,.$$

If *m* is given then we don't regard it as a variable, otherwise the problem remains unchanged. If we want to build into the model the random water demands η_k, it may be done without any further as in Section 3 was discussed. The numerical solution of problem (9) is possible if we put some special assumptions on the random variables $\xi_1,\ldots,\xi_N$, see the papers [7], [8], [9]. The model (9) can be successfully applied to scaling the capacity value *m*.

A further variant of model (9) is when the decision-maker may give an upper bound *K* on the cost of building the reservoir with capacity *m, c(m)*. In this case it is not necessary to subtract the value $c(m)$ from the objective function and the problem of the modified model can be formulated as

$$\max\left[f(z_1,\ldots,z_N)-c(m)\right] \quad \text{supposing, that}$$

$$P\left\{a_k \le \xi_0+\sum_{j=1}^{k}\xi_j-\sum_{j=1}^{k}z_j \le b_k,\ k=1,\ldots,N\right\}\ge p \tag{10}$$

$$c(m)\le K, \quad 0\le z_k \le m, \quad k=1,\ldots,N$$

It's worth mentioning that if the probability distribution of the random variables $\xi_1,\ldots,\xi_N$ is continuous and their density function is logarithmically concave, then the $m,z_1,\ldots,z_N$ feasible domain of problems (9) and (10) is convex (see for example Prékopa [6]). So if $f(z_1,\ldots,z_N)$ and $c(m)$ are convex functions, then the problems (9) and (10) are convex.

Let us regard a reservoir for four consecutive months, say April, May, June and July, as an example of Problem (10). Let streamflow data follow joint normal probability distribution with the following parameters:

Table 3

Parameters of joint normal distribution of the random streamflow values

	expected value ($10^6 m^3$)	standard deviation ($10^6 m^3$)	correlation coefficients			
ξ_1	79.74	83.51	1.000	0.284	-0.017	0.047
ξ_2	29.78	63.11	0.284	1.000	0.333	0.198

ξ_3	-4.52	73.98	-0.017	0.333	1.000	0.579
ξ_4	-43.44	73.96	0.047	0.198	0.579	1.000

Create the aggregated random variables

$$\varsigma_1 = \xi_1$$

$$\varsigma_2 = \xi_1 + \xi_2$$

$$\varsigma_3 = \xi_1 + \xi_2 + \xi_3$$

$$\varsigma_4 = \xi_1 + \xi_2 + \xi_3 + \xi_4 .$$

These random variables as linear transforms of $\xi_1, \xi_2, \xi_3, \xi_4$ have also normal distribution with the transformed expected values, standard deviations and correlation coefficients:

Table 4

Parameters of joint normal distribution of the random stream flow values

	expected value $(10^6 m^3)$	standard deviation $(10^6 m^3)$	correlation coefficients			
ς_1	79.740	83.510	1.000	0.859	0.670	0.542
ς_2	109.520	118.112	0.859	1.000	0.873	0.736
ς_3	105.000	149.408	0.670	0.873	1.000	0.935
ς_4	61.560	191.201	0.542	0.736	0.935	1.000

Let us suppose that in the optimization problem (10) $f(z_1, z_2, z_3, z_4) = 40z_1 + 70z_2 + 80z_3 + 50z_4$, i.e. the total benefit of released water is a linear function. Let the cost of the reservoir of capacity m also be linear function: $c(m) = 50m$. For the smallest water level of the reservoir let be prescribed $a_k = 100$ in all periods $k = 1,2,3,4$; for the largest water level of the reservoir let be prescribed $b_k = 1000$ in all periods $k = 1,2,3,4$; and let us suppose that at the beginning of the first period the season starts with full reservoir. If we solve the arising optimization problem with different bounds on the building cost then the decision-maker can select the economically reasonable capacity. Introducing new variables for simplifying the terms inside the probability expressing the reliability-type constraint, we solved the following optimization problem for different building up cost bounds *K*:

$$\max(40z_1 + 70z_2 + 80z_3 + 50z_4) \quad \text{supposing that}$$

$$l_1 = 100 + z_1 - 1000$$

$$l_2 = 100 + z_1 + z_2 - 1000$$

$$l_3 = 100 + z_1 + z_2 + z_3 - 1000$$

$$l_4 = 100 + z_1 + z_2 + z_3 + z_4 - 1000$$

$$u_1 = 1000 + z_1 - 1000$$

$$u_2 = 1000 + z_1 + z_2 - 1000$$

$$u_3 = 1000 + z_1 + z_2 + z_3 - 1000$$

$$u_4 = 1000 + z_1 + z_2 + z_3 + z_4 - 1000$$

$$100P\begin{pmatrix} l_1 \le \varsigma_1 \le u_1 \\ l_2 \le \varsigma_2 \le u_2 \\ l_3 \le \varsigma_3 \le u_3 \\ l_4 \le \varsigma_4 \le u_4 \end{pmatrix} \ge 90.00$$

$$50m \le K,\ z_1 \le m,\ z_2 \le m,\ z_3 \le m,\ z_4 \le m\,.$$

Notice that the probabilistic constraint has been multiplied by 100. As a result, the problem can be solved numerically in a more stable way. Then the only difficulty is the calculation of the probability values and its partial derivatives. For this we can write the probability value in the following form:

$$\begin{aligned} P\begin{pmatrix} l_1 \le \varsigma_1 \le u_1 \\ l_2 \le \varsigma_2 \le u_2 \\ l_3 \le \varsigma_3 \le u_3 \\ l_4 \le \varsigma_4 \le u_4 \end{pmatrix} &= F(u_1,u_2,u_3,u_4) - F(l_1,u_2,u_3,u_4) \\ &- F(u_1,l_2,u_3,u_4) - F(u_1,u_2,l_3,u_4) \\ &- F(u_1,u_2,u_3,l_4) + F(l_1,l_2,u_3,u_4) \\ &+ F(l_1,u_2,l_3,u_4) + F(l_1,u_2,u_3,l_4) \\ &+ F(u_1,l_2,l_3,u_4) + F(u_1,l_2,u_3,l_4) \\ &+ F(u_1,u_2,l_3,l_4) - F(l_1,l_2,l_3,u_4) \\ &- F(l_1,l_2,u_3,l_4) - F(l_1,u_2,l_3,u_4) \\ &- F(u_1,l_2,l_3,l_4) + F(l_1,l_2,l_3,l_4) \end{aligned}$$

where $F(x_1, x_2, x_3, x_4)$ denotes the joint normal probability distribution function of the random variables $\varsigma_1,\varsigma_2,\varsigma_3,\varsigma_4$ with parameters given in Table 4. This means that for the calculation of one probability value we have to calculate $2^4 = 16$ four dimensional normal probability distribution function values.

We prepared the AMPL model of the above defined nonlinear programming problem. To calculate values of the multivariate normal probability distribution functions the numerical integration code developed by A Genz ([1]) has been added to the AMPLE model. Then the problem was solved by the solver LOQO for different values of cost bound *K*. The results are summarized in Table 5.

Table 5

Total benefit values of the test problem for different values of *K*

Here *m* is the optimal capacity of the reservoir and z_k is the optimal amount of water release in period *k. All of them are given in* $10^6 m^3$. *K* and the total benefit values are given in millions of HUF.

Number	*K*	total benefit	*m*	z_1	z_2	z_3	z_4
1	10000	36634.493	200.001	200.001	180.665	199.848	0.000
2	10500	39682.114	210.011	210.011	206.903	209.975	0.010
3	11000	41250.695	220.003	220.003	212.155	219.996	0.001
4	11500	42270.302	230.004	230.004	209.583	229.990	0.003
5	12000	42948.048	240.010	240.010	202.120	239.990	0.003
6	12500	43378.927	250.012	250.012	191.146	249.973	0.009
7	13000	43615.155	259.999	259.999	177.390	259.973	0.003
8	13500	43741.739	269.943	268.574	162.960	269.943	0.002
9	14000	43792.337	279.971	268.973	151.969	279.971	0.001
10	14500	43836.424	289.895	268.947	141.354	289.895	0.000
11	15000	43861.877	299.046	268.963	132.222	299.046	0.002

Figure 2 represents the possible benefit values according to different cost bounds *K*. This graph may be useful for decision-makers when deciding how much should be spent for providing a given reservoir capacity. The graph shows the benefit increase of water releases if the amount of money spent for building the reservoir is increased. The decision should take into account, of course, that the cost of reservoir occurs only once and the benefit of released water can be realized for many years.

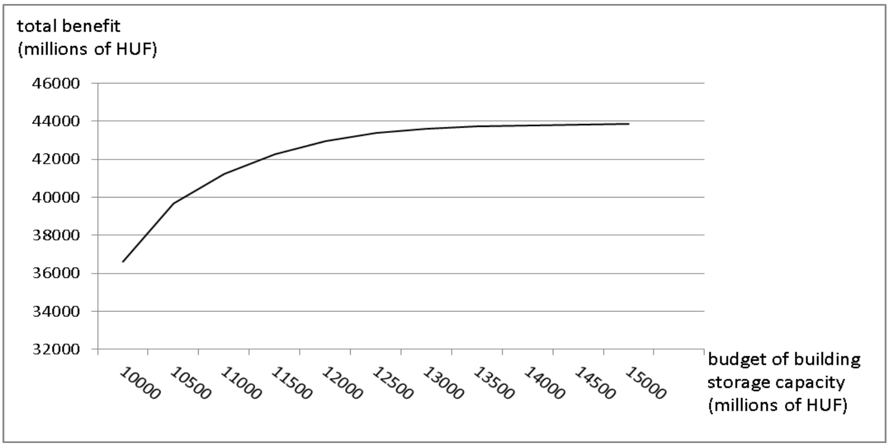


Figure 2
The benefit of irrigation water in function of the money spent for building a reservoir

5 Probability Constrained Stochastic Programming Model for an Intake Facility

The capacity of an intake facility, say a pumping station is considered to satisfy random water demands (e.g. irrigation) utilizing the available streamflow. Let us regard a given time period which can be a month, say August of the year. We will prescribe that the number of days with unsatisfied water demands should not exceed a given value with a high probability. The model will be described for a time interval of *n* days. We introduce the notations:

$\xi_1,\mathrm{K}\,,\xi_n$	daily available stream flows
$\gamma_1,\mathrm{K}\,,\gamma_n$	daily rainfalls
$\eta_1,\mathrm{K}\,,\eta_n$	daily water demands
m	daily capacity of the intake facility, the decision var
M	upper bound for capacity *m*
$c(m)$	cost of the facility
b	maximum number of days with unsatisfied water given time period
p	reliability level prescribed, close to one

There is enough water on the k th day if and only if the following relation holds

$$\min\left(\xi_k, m\right) + \gamma_k \geq \eta_k . \tag{11}$$

Let $x_1, \ldots, x_n$ be deterministic variables which take on values 0 and 1, only. The following relation doesn't mean any constraint if $x_k = 0$, but if $x_k = 1$ it is equivalent to the constraint (11):

$$\min\left(\xi_k, m\right) + \gamma_k \geq x_k \eta_k . \tag{12}$$

Beside (12) for all $k = 1, \ldots, n$ prescribing the constraint

$$x_1 + \cdots + x_n \geq n - b$$

we require that at least $n - b$ out of the constraints (11) be met, i.e. at least $x_1, \ldots, x_n$ times the opposite of the constraint (11) be met. Then our model can be formulated as

$$\min c(m) \quad \text{supposing, that}$$

$$P\left\{\min\left(\xi_k, m\right) + \gamma_k \geq x_k \eta_k,\ k = 1, \ldots, n - b\right\} \geq p$$

$$x_1 + \cdots + x_n \geq n - b \tag{13}$$

$$x_k = 0 \text{ or } 1,\ k = 1, \ldots, n,\ 0 \leq m \leq M .$$

Like the earlier models, this model also has more variants. Among others, one can build into the objective function, a cost factor, that depends on the number of days, b. If the random variables $\xi_k, \eta_k, \gamma_k, k = 1, \ldots, N$ have continuous joint probability distribution and their joint density function is logarithmically concave then the constraints of problem (13) except the constraints $x_k = 0 \text{ or } 1$, define a convex feasible set.

References

[1] A. Genz, Numerical Computation of the Multivariate Normal Probabilities, *Journal of Computational and Graphical Statistics*, **1,** 141-150, 1992

[2] B. V. Gnedenko and A. N. Kolmogorov, *Limit distributions for sums of* ***independent*** *random variables*, Translated and annotated by K. L. Chung, Addison-Wesley, Cambridge, 1954

[3] A. Prékopa, On secondary processes generated by random point distributions of Poisson type, *Annales Univ. Sci. R. Eötvös, Sectio Math.*, **2**, 139-146, 1959

[4] A. Prékopa, Contributions to the theory of stochastic programming, Mathematical Programming, 4, 202-221, 1973

[5] A. Prékopa, T. Rapcsák and I. Zsuffa, A new method for serially linked reservoir system design using stochastic programming (in Hungarian), *Alkalmazott Matematikai Lapok*, **2**, 189-2001, 1976

[6] A. Prékopa, *Stochastic Programming*, Kluwer Scientific Publishers, Dordrecht, Boston, 1995

[7] A. Prékopa and T. Szántai, On Optimal Regulation of a Storage Level with Application to the Water Level Regulation of a Lake, *European Journal of Operational Research*, **3**, 175-189, 1979

[8] T. Szántai, Evaluation of a Special Multivariate Gamma Distribution, *Mathematical Programming Study*, **27**, 1-16, 1986

[9] T. Szántai, A Computer Code for Solution of Probabilistic Constrained Stochastic Programming Problems, In: *Numerical Techniques for Stochastic Optimization* (Yu. Ermoliev and R. J.-B. Wets, eds.), Springer, New York, 229-235, 1988

[10] L. Takács, Secondary processes generated by Poisson process and their applications in physics (in Hungarian), *MTA Matematikai és Fizikai Tudományok Osztályának Közleményei,* **4**, 473-504, 1954

Inventory Control in Sales Periods

Tamás Szántai, Edith Kovács, Attila Egri

Department of Differential Equations, Budapest University of Technology and Economics, Műegyetem rkp. 3-9, 1111 Budapest, Hungary
e-mail: szantai@math.bme.hu, kovacsea@math.bme.hu, egri@math.bme.hu

Abstract: Sales promotion aims to capture the market and increase sales volume. Therefore, an important task is the forecasting of the demand during the sales period. We present two dynamic methodologies for calculating the quantity which has to be placed on the shelves at the beginning of each day such that we keep some constraints expressing lower and upper bounds on the quantities. Both methodologies are new to this field and are useful because of some specific properties of the problem. Our new methods use historical data of the demands in previous promotions and the consumptions registered in the previous days. Since the promotion period is relatively short, other methods such as time series analysis can hardly be used.

Keywords: inventory control; dynamic forecasting; information driven forecasting

1 Introduction

Many businesses use sales promotions to increase the demand of a product or service. Promotions and sales are important strategies of a successful business. Their effects include growth within the market segment involved, the discovery of new products. Promotions attract new and old customers and can keep the company relevant when competitors appear. Price reductions can substantially boost the sales of the given product, but also cause brand switching.

Effective sales promotions lead to inventory reductions, because customers buy more products. Therefore companies use these actions at the end of a buying season. For example, when Christmas Eve is past, very often, retailers offer discounts to make room on the shelves for other products.

Paper [12] highlights how promotions affect the buying habits of costumers as a consequence of a changed price conditions.

Some interesting statistics on demand in sales period can be found in [1]: "Demand during many promotions is often dramatically greater than median daily demand: demand in 54% of promotions is > 15 standard deviations greater than daily median demand, and demand in 3% of promotions is > 100 standard deviations greater than normal daily demand. However, promotion demand represents a relatively small percentage of total yearly demand for most products. For 90% of products, promotion demand is <15% of total yearly demand and promotion demand is <20% of total demand among 90% of the products with one or more promotions."

The time series method forecasts the new demand values, on the basis of historical demand data. In [11] time series forecasting models with extending an exponential smoothing approach were proposed. However, exponential smoothing methods have been criticized for their inability to capture the effects of special events such as promotions, announcements. When demand for an item is being driven by such factors as trends and seasonal patterns, time series methods tend to work quite well [6]. However, business data often contain responses to actions, such as promotions, that cannot be captured as part of the level, trend and seasonal components. When a significant amount of demand is being driven by these types of events, time series methods will not work very well.

Fildes and Goodwin ([4]) indicated promotional and advertising activity as one of the main drivers behind adjustments of the statistical forecasts by managerial judgments.

An alternative approach to the problem of forecasting promotional sales is to use regression models, which use past promotional information to formulate causal models, Fildes et al. [5].

Although the information of human judgment cannot be captured by simple promotional models, yet Trapero et al. in [12] showed that a simple model could beat judgmental forecasting. Therefore, there is a need for developing more sophisticated promotional models.

In a recent paper [3], different models of forecasting the demand during a promotion are developed and tested, including a moving average forecast and several regression models. In the paper it is investigated how different factors such as price variation, advertising influence the demand.

Another recent paper on the topic of forecasting demand in sales period is [7]. The presented method consists of the identification of potentially influential categories, and then of the selection of the explanatory variables by using multistage LASSO regression and of the use of a rolling scheme to generate forecasts. The success of the method is also based on dealing with high dimensionality which brings improvements in forecasting accuracy compared to other methods which used also a reduced variable space.

In [13] Trapero et al. proposed a Principal Component Analysis based promotional model that overcomes the limitations caused by multicollinearity and high dimensionality.

In this paper we address the problem of stochastic inventory control during a retail or promotion time. The pricing of the products in sales period is also an important optimization problem, but in this paper we suppose the promotion price was already fixed. This paper is dealing with the problem of daily updating the quantity of a given product on the shelves during a promotion sales period. A specific characteristic of this problem is that, products are sold at a lower price during a relatively short period only. We present two methodologies for making decisions on the quantity of product which has to be placed on the shelves. These are based on historical data of similar promotions that have occurred in the past. To our best knowledge we are the first who introduce the following models to the problem of inventory management during promotional sales.

The dynamic of our models is as follows. At the beginning of each day, a quantity of a given product is placed on the shelves. The demand on each day is observed and based on this cumulative set of information one has to decide the quantity to be placed on the shelves at the beginning of the next day. This way the decision is made day by day and uses beyond the information accumulated from the previous days also historical data collected from previous sales periods. In addition the experts may put some constraints on the quantity of products being on the shelves.

2 Dynamic Inventory Control in Sales Periods by Adapting the Lake Balaton Water Level Regulation Model

2.1 Preliminaries

In paper [8] the following dynamic control model was developed for regulation of the water level of Lake Balaton.

Let us introduce the following notations:

V_0 initial water content of the lake,

ξ_k random water input in month k,

z_k water quantity to be released through the channel Sio in month k,

$\varsigma_k = V_0 + \sum_{i=1}^{k} \xi_i$ initial water content plus the cumulated monthly random water inputs at the end of month k,

a_k lower bound for the water quantity being in the lake at the end of month k,

b_k upper bound for the water quantity being in the lake at the end of month k.

For determining the optimal decisions $z_1^*,\ldots,z_N^*$ according to the first N months one should solve the following stochastic programming problem:

$$\max P\left\{a_k \le \varsigma_k - \sum_{i=1}^{k} z_i \le b_k,\ k = 1,\ldots,N\right\}$$

supposing that (1)

$$0 \le z_k \le K, \quad k = 1,\ldots,N,$$

where K is the monthly capacity of the channel Sio.

The authors of paper [8] proposed to accept the optimal value $z_1 = z_1^*$ of the first decision variable only, apply it as water release in the actual month and then formulate the next stochastic programming problem of type (1) and so on.

If one observed the realized values $x_1,\ldots,x_n$ of the random water inputs $\xi_1,\ldots,\xi_n$ and the realized water releases were $z_1^*,\ldots,z_n^*$ in the first n months, then the knowledge of these values can also be utilized in the following way. Let us modify the initial water content of the lake for the water content at the end of the n^{th} month, i.e. let be $V_n = V_0 + \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} z_i^*$ and $\varsigma_k = V_n + \sum_{i=n+1}^{k} \xi_i$. Then instead of stochastic programming problem (1) one can regard the problem

$$\max P\left\{a_k \le \varsigma_k - \sum_{i=n+1}^{k} z_i \le b_k,\ k = n+1,\ldots,n+N \middle| \xi_1 = x_1,\ldots,\xi_n = x_n\right\}$$

supposing that (2)

$$0 \le z_k \le K, \quad k = n+1,\ldots,n+N.$$

Now one may accept the optimal value of the first decision variable $z_{n+1} = z_{n+1}^*$ only, apply it as water release in the actual month and then formulate the next stochastic programming problem of type (2) and so on.

If the random process $\xi_1, \xi_2,\ldots$ is Gaussian, these stochastic programming problems can be solved as it was shown in [8]. In this paper the special case of $n = N = 2$ was taken and the authors successfully applied this method for the monthly dynamic control of the water level of the Lake Balaton for a fifty years long time horizon.

2.2 Adaptation of the Lake Balaton Water Level Regulation Model

Let us now regard a line of goods in a department store which is on sale for a fourteen days' time period. The main difference is that while the water level of the Lake Balaton can be controlled only by decreasing its value, in this case the amount of the line on the shelves can be controlled only by increasing its value. In the same time while the water level of the Lake Balaton increases (changes) randomly and it can be decreased deterministically, in this case the amount of the line on the shelves decreases randomly and it can be increased deterministically.

For describing the stochastic programming models let us now introduce the following notations:

V_0	the starting amount of the line on the shelves at the beginning of the sale,
ξ_k	the random consumptions of the line on the k th day of the sale,
z_k	decision variable belonging to the k th day of the sale, this is the quantity of the line to be placed on the shelves when opening the k th day of the sale,
$\varsigma_k = \sum_{i=1}^{k} \xi_i$	the cumulated daily random consumption at the end of the k th day of the sale,
K	the capacity of the shelves over the sale,
a_k	lower bound for the line amounts to be placed on the shelves at the end of the k th day of the sale,
b_k	upper bound for the line amounts to be placed on the shelves at the end of the k th day of the sale.

The notations above are introduced for all days $k = 1,2,\ldots,14$ of the sale.

If we suppose that the line of goods is put on the shelves each day morning then the following inequalities must be fulfilled:

$V_0 + z_1$	$\le b_1$	quantity at first day opening time,
$V_0 + z_1 + z_2 - \xi_1$	$\le b_2$	quantity at second day opening time,
$\vdots$	$\vdots$	$\vdots$
$V_0 + z_1 + \cdots + z_N - \xi_1 - \cdots - \xi_{N-1}$	$\le b_N$	quantity at N th day opening time.

Taking into account the daily random consumption values at the end of the day the following inequalities must be fulfilled:

$$\begin{array}{lll}
V_0 + z_1 - \xi_1 & \geq a_1 & \text{quantity at first day closing time,} \\
V_0 + z_1 + z_2 - \xi_1 - \xi_2 & \geq a_2 & \text{quantity at second day closing time,} \\
\vdots & \vdots & \vdots \\
V_0 + z_1 + \ldots + z_N - \xi_1 - \ldots - \xi_N & \geq a_N & \text{quantity at } N^{th} \text{ day closing time.}
\end{array}$$

For determining the optimal decisions $z_1^*, \ldots, z_N^*$ according to the first $N(\leq 14)$ days one should solve the following stochastic programming problem:

The problem which accords with/corresponds to problem (1) is formally the following:

$$\max P\left\{\begin{array}{c} V_0 + \sum_{i=2}^{k+1} z_i - b_{k+1} \leq \varsigma_k \leq V_0 + \sum_{i=1}^{k} z_i - a_k,\ k = 1, \ldots, N-1 \\ \varsigma_N \leq V_0 + \sum_{i=1}^{N} z_i - a_N \end{array}\right\}$$

supposing that (3)

$$z_1 \leq b_1 - V_0,\ z_1 \geq 0, z_2 \geq 0, \ldots, z_N \geq 0.$$

If one observed the realized values $x_1, \ldots, x_n$ of the random consumptions $\xi_1, \ldots, \xi_n$ and the quantities $z_1^*, \ldots, z_n^*$ of the line placed on the shelves on the first n days of the sale, then the knowledge of these values can also be utilized in the following way. Let us modify the starting amount of the line on the shelves in the morning of the $n+1$ th day, i.e. let be $V_n = V_0 + \sum_{i=1}^{n} z_i^* - \sum_{i=1}^{n} x_i$. Then instead of stochastic programming problem (3) one can regard the problem

$$\max P\left\{\begin{array}{c|c} V_n + \sum_{i=n+2}^{n+k+1} z_i - b_{n+k+1} \leq \varsigma_{n+k} \leq V_n + \sum_{i=n+1}^{n+k} z_i - a_{n+k},\ k = 1, \ldots, N-1 & \xi_1 = x_1 \\ & \vdots \\ \varsigma_{n+N} \leq V_n + \sum_{i=n+1}^{n+N} z_i - a_{n+N} & \xi_n = x_n \end{array}\right\}$$

supposing that (4)

$$z_{n+1} \leq b_{n+1} - V_n,\ z_{n+1} \geq 0, z_{n+2} \geq 0, \ldots, z_{n+N} \geq 0.$$

Now one may accept the optimal value of the first decision variable $z_{n+1} = z_{n+1}^*$ only, apply it as quantity of the line to be placed on the shelves in the actual day and then formulate the next stochastic programming problem of type (4) and so on.

If the random process ξ_1, ξ_2, K is Gaussian, these stochastic programming problems can be solved. For detailed calculation procedure, see paper [8]. Relatively small values of n and N (say, $n = N = 2$) may be enough for achieving good control in this case, too.

2.3 Application of the Algorithm

As it can be seen in Table 1, the random consumptions had relatively large standard deviations according to their mean values, so the modified Lake Balaton inventory control model was not applicable for these data. This model could be applied when the standard deviation of the random consumptions is not larger than one third of the mean value, otherwise one should be able to interpret negative valued consumptions.

3 Dynamic Inventory Control in Sales Periods by using Information-driven Forecasting

3.1 Preliminaries

Sales periods are relatively short, one or two weeks typically, therefore, the popular time series forecasting methods cannot be applied for the goods in promotion sales.

We distinguish the following two kinds of promotional sales. The first one is the case of a product which already exists on the market, the second one is the sale promotion applied to a new product which has to be introduced into the market.

For the second case we have no proper historical data. To overcome this drawback, we can search for products which are similar to the new one, and use their historical data. Having these we may apply our methodology.

We consider now the case of forecasting the demand of an existing product on the market, for which we have earlier data registered, during the sale periods of the same length. We associate a random variable X_i to the daily consumption registered at the end of each day. We can define a random vector $\mathbf{X} = \left(X_1, \mathrm{K}\ , X_d\right)$, where d is the length of the sales period expressed in days.

At the end of the i th day, $i = 1, \mathrm{K}\ , d-1$ we have to decide on the quantity of product to place on the shelves. For this we have to forecast the consumption of day $i+1$ based on the consumption of the first i days. Based on the forecasted consumption we make sequential decisions on the quantity which have to be displayed.

We regard now the problem of forecasting the consumption of the $i+1$ th day.

We consider the random vector: $\mathbf{X}=\left(X_1,\mathrm{K}\ ,X_{i+1}\right)$ which is a margin of the random vector $\mathbf{X}=\left(X_1,\mathrm{K}\ ,X_d\right)$.

The main idea behind our method is that we use 1, 2, 3 (rarely 4) out of the previous days to forecast the consumption of day $i+1$. We emphasize here that we do not use necessarily the days *i*, *i*-1, *i*-2. Instead we will choose those days from all previous days which minimize the uncertainty of the day *i*+1.

For this task, we use the following informational theoretical concepts.

The uncertainty amount of a random vector can be quantified by its entropy. The entropy does not depend on the values of the random vector; it depends only on the probabilities with which the different values are taken on.

The concept of entropy has its roots back in 1854 in a memoir of Rudolph Clausius. However, in this paper we will use the expression given by Claude Shannon published in his famous paper [10]. More general definitions for entropy were also given by Rényi [9].

We introduce the reader into some information theoretical concepts, which have to be reminded for the understanding of our method. The interested reader can find more details about these concepts in [2].

In the present work, we use the following formula for entropy, which is related to a random vector with *m* realizations. If χ_i represents the range of the variable X_i then the range of $\mathbf{X}$ is a subset of $\bigotimes_{i=1}^{d}\chi_i$

$$H(\mathbf{X})=-\sum_{k=1}^{m}p_k\ln p_k$$

where m indicates the number of all distinct realizations of the random vector **X** and p_k denotes the probability of the *k* th realization of the random vector **X** (the ordering of the realization has no importance, but is fixed).

In order to quantify how much the uncertainty of a given random variable is reduced by knowing the values taken on by the other random variables we use the concept of conditional entropy denoted by $H\left(X_i\,|\,\mathbf{X}_{V-i}\right)$, where we use the notation $\mathbf{X}_{V-i}$ for the random vector of all random variables with indices in *V* except X_i.

For a better understanding of the concept of conditional entropy we first define the following random variables. Let us fix $i \in V$ and an arbitrary realization $\mathbf{x}_{V-i}^{k}$ of $\mathbf{X}_{V-i}$. The conditional random variable denoted by $X_i \,|\, \mathbf{x}_{V-i}^{k}$ takes on the values x_{i_j} by probabilities $p_{x_{i_j}|\mathbf{x}_{V-i}^{k}} = P\left(X_i = x_{i_j} \middle| \mathbf{X}_{V-i} = \mathbf{x}_{V-i}^{k}\right), \quad j = 1,\ldots,i_s$:

$$X_i \,|\, \mathbf{x}_{V-i}^{k}: \quad \begin{pmatrix} x_{i_1} & \ldots & x_{i_j} & \ldots & x_{i_s} \\ p_{x_{i_1}|\mathbf{x}_{V-i}^{k}} & \ldots & p_{x_{i_j}|\mathbf{x}_{V-i}^{k}} & \cdots & p_{x_{i_s}|\mathbf{x}_{V-i}^{k}} \end{pmatrix}.$$

Here $\mathbf{x}_{V-i}^{k}$ stands for the k^{th} realization of the random vector $\mathbf{X}_{V-i}$. This way a conditional random variable $X_i \,|\, \mathbf{x}_{V-i}^{k}$ is assigned to each realization $\mathbf{x}_{V-i}^{k}$, $k = 1,\ldots,m_{V-i}$. We take now their entropies denoted by $h\left(X_i \,|\, \mathbf{x}_{V-i}^{k}\right)$, $k = 1,\ldots,m_{V-i}$ and define a new random variable as follows.

$$\tilde{h}\left(X_i \,|\, \mathbf{X}_{V-i}\right): \quad \begin{pmatrix} h\left(X_i \,|\, \mathbf{x}_{V-i}^{1}\right) & \ldots & h\left(X_i \,|\, \mathbf{x}_{V-i}^{k}\right) & \ldots & h\left(X_i \,|\, \mathbf{x}_{V-i}^{m_{V-1}}\right) \\ p_{\mathbf{x}_{V-i}^{1}} & \cdots & p_{\mathbf{x}_{V-i}^{k}} & \cdots & p_{\mathbf{x}_{V-i}^{m_{V-1}}} \end{pmatrix},$$

where $p_{\mathbf{x}_{V-i}^{k}} = P\left(\mathbf{X}_{V-i} = \mathbf{x}_{V-i}^{k}\right), \quad k = 1,\ldots,m_{V-i}$.

Finally, we arrive to the definition of conditional entropy $H\left(X_i \,|\, \mathbf{X}_{V-i}\right)$ that is defined as the expected value of $\tilde{h}\left(X_i \,|\, \mathbf{X}_{V-i}\right)$.

From these it can be seen, that the conditional entropy $H\left(X_i \,|\, \mathbf{X}_{V-i}\right)$ quantifies the amount of the uncertainty of X_i when there are given the realizations of X_{V-i}. As the smaller the conditional entropy is the better we can reduce the uncertainty of X_i by knowing realizations of $\mathbf{X}_{V-i}$.

This leads to the introduction of the concept of mutual information $I\left(X_i, \mathbf{X}_{V-i}\right)$, which is defined as the following difference:

$$I\left(X_i, \mathbf{X}_{V-i}\right) = H\left(X_i\right) - H\left(X_i \,|\, \mathbf{X}_{V-i}\right).$$

3.2 The Central Idea of our Method

The main idea of our method is the way we decide on the quantity which has to be placed on shelves next day. The decision is based on the consumptions registered in few previous days. These days are chosen in such a way that these minimize the conditional entropy of the next day’s consumption. This is equivalent with maximizing the information gain.

Since the sales period is relatively short, one or two weeks long, we use only one, two or three previous days in forecasting.

From a theoretical point of view taking more than three days leads to over fitting and poor generalization of the model.

3.3 The Consumption Forecasting Algorithm

Based on the historical data we have the joint empirical probability distribution of the random vector $\mathbf{X} = (X_1, \ldots, X_d)$, where the random variables $X_i, i = 1, \ldots, d$ represent the daily consumptions.

We introduce the following notations.

The realized consumption of the i th day in the actual sales period is denoted by C_i. Forecasting the consumption of the i+1 th day means choosing one of the possible realizations of the random variable X_{i+1} in a certain way. Let Q_{i+1}^k $k = 1, \ldots, s_{i+1}$ denote the possible values of X_{i+1}. The forecasted consumption of the i+1 th day is denoted by Q_{i+1}.

We introduce the following notations:

$\Lambda_k = \{X_1, \ldots, X_k\}$,

$\Lambda_k^2 = \{(X_l, X_m) \mid X_l, X_m \in \Lambda_k, l < m\}$

$\Lambda_k^3 = \{(X_l, X_m, X_n) \mid X_l, X_m, X_n \in \Lambda_k, l < m < n\}$

- In the first day we usually consider the mean value of X_1 as forecasted consumption. This will be the quantity Q_1.
- In the second day we can use the registered consumption C_1 of the first day and forecast the second day consumption by the following maximization:

  $$Q_2 = \arg\max_{k=1,\ldots,s_2} P\left(X_2 = Q_2^k \mid X_1 = C_1\right)$$

- From i=3 to n

  Step 1: Choose a number f^* from 1 to 3, this is the number of the previous days used in the forecast – this decision can be made by interacting with the user.

  Step2. The selection of the informative variables:

$$\text{if } f^* = 1, \; X_l^* = \arg\min_{X_l \in \Lambda_i} H(X_{i+1} \mid X_l)$$

$$\text{if } f^* = 2, \; (X_l^*, X_m^*) = \arg\min_{(X_l, X_m) \in \Lambda_i^2} H(X_{i+1} \mid X_l, X_m)$$

$$\text{if } f^* = 3, \; (X_l^*, X_m^*, X_n^*) = \arg\min_{(X_l, X_m, X_n) \in \Lambda_i^3} H(X_{i+1} \mid X_l, X_m, X_n)$$

Step 3. The forecast of the consumption of day $i+1$:

$$Q_{i+1} = \arg\max_{k=1,\mathrm{K},s_{i+1}} P\left(X_{i+1} = Q_{i+1}^k \mid \mathbf{X}^* = \mathbf{C}^*\right).$$

Here the notation $\mathbf{X}^* = \mathbf{C}^*$ stands for

$X_l^* = C_l$, if $f^* = 1$,

$X_l^* = C_l, X_m^* = C_m$, if $f^* = 2$,

$X_l^* = C_l, X_m^* = C_m, X_n^* = C_n$, if $f^* = 2$.

The algorithm was implemented in such a way that the user can make interactive decisions when the code is running. In Step 1 the user can specify, based on the value of conditional entropies, how many previous days should be taken into account in the forecast.

In Step 3 two, three or four dimensional marginal probability distributions are used, depending on conditioning one, two or three earlier days consumptions. We may face to the following problems:

a) X_{i+1} takes on more values with the same probability. In this case, we take their mean value as forecast.

b) In the marginal probability distribution of the historical data never occurs the realization $\mathbf{X}^* = \mathbf{C}^*$. We overcome this problem by using lower marginals.

For example, it may happen that the probability

$$P\left(X_{i+1} = Q_{i+1}^k \mid X_l^* = C_l, X_m^* = C_m, X_n^* = C_n\right)$$

cannot be calculated since the conditioning realization did not occur in the historical data. For these cases we have to apply a forecasting scheme which is based on lower marginal probability distributions.

Let us denote by *r* the dimension of the largest marginal probability distribution of $\mathbf{X}^*$ with the property that the corresponding conditioning set occurs with positive probability.

If the conditioning set contains 3 variables, *r* can be 2 or 1.

For illustration let us suppose that *r*=2. In this case at least one of the following cases occurs. There exists at least one *k* such that

$$P\left(X_{i+1} = Q_{i+1}^k \mid X_l^* = C_l, X_m^* = C_m\right) > 0\,, \tag{5}$$

or

$$P\left(X_{i+1} = Q_{i+1}^k \mid X_l^* = C_l, X_n^* = C_n\right) > 0\,, \tag{6}$$

or

$$P\left(X_{i+1} = Q_{i+1}^k \mid X_m^* = C_m, X_n^* = C_n\right) > 0\,. \tag{7}$$

Let us denote by $P_{X_{i+1}}^r$ the sum of the above nonzero probabilities for all $k = 1,\ldots,s_{i+1}$.

For each *k* we calculate a probability $p_{Q_{i+1}^k}$ as the sum of nonzero probabilities of (5)-(7) for which $X_{i+1} = Q_{i+1}^k$ is taken on, divided by $P_{X_{i+1}}^r$. Using these we define the following random variable.

$$\gamma_r : \begin{pmatrix} Q_{i+1}^1 & \mathrm{K} & Q_{i+1}^k & \mathrm{K} & Q_{i+1}^{s_{i+1}} \\ p_{Q_{i+1}^1|} & \mathrm{K} & p_{Q_{i+1}^k} & \Lambda & p_{Q_{i+1}^{s_{i+1}}} \end{pmatrix}$$

The forecasted consumption of the *i*+1 *th* day is $Q_{i+1} = \arg\max_{k=1,\mathrm{K},s_{i+1}} p_{Q_{i+1}^k}$.

3.4 Decision Making Procedure

Let us suppose that we decided on the amounts of goods to be placed on shelves in the first i days. We have to decide on the amount of goods to be placed on the shelves in the morning of the $i+1$ *th* day, based on the forecasted consumption. We want the end-of-day amount on display to be equal to the arithmetic mean of the prescribed lower and upper levels. If z_{i+1} denotes the quantity of the line to be placed on the shelves at the beginning of the $i+1$ *th* day its value have to fulfill the following equality:

$$V_0 + z_1^* + \Lambda + z_i^* - C_1 - \Lambda - C_i + z_{i+1} - Q_{i+1} = \frac{a_{i+1} + b_{i+1}}{2}\,, \tag{8}$$

where V_0 is the starting amount of line on the shelves at the beginning of the sale; $z_1^*,\mathrm{K}\,,z_i^*$ are the decisions applied in the first i days; $C_1,\mathrm{K}\,,C_i$ are the realized random consumptions in the first i days; Q_{i+1} is the forecasted consumption in the $i+1$ *th* day, and a_{i+1}, b_{i+1} are the prescribed lower resp. upper bounds on the amount of goods placed on display at the end of the $i+1$ *th* day.

Solving the equation (8) we get for the optimal decision:

$$z_{i+1}^{*} = \frac{a_{i+1} + b_{i+1}}{2} - \left(V_0 + z_1^{*} + \Lambda + z_i^{*} - C_1 - \Lambda - C_i\right) + Q_{i+1} \,.$$

3.5 Data Preprocessing

In the practical application of our model the following problem may appear which have to be solved before running the algorithm.

The problem is caused by the relatively small learning data set, and the relatively large range of values which are taken on by each random variable. Therefore we decided to group the values into intervals. Based on the historical data the range of the consumption for each day was divided into 4 intervals as follows.

For each day there was calculated the minimum consumption, maximum consumption, mean value and standard deviation. These divide the range into four intervals. The intervals were delimited by the minimum consumption, the mean value minus the standard deviation, the mean value, the mean value plus the standard deviation and the maximum consumption. Each interval was characterized by the mean value calculated from the historical data.

First we forecasted an interval then on the basis of this we accepted the mean value assigned to this interval as forecast for the consumption.

3.6 Application of the Algorithm

Our dynamic decision making algorithm has been applied to the real-life data set of a 14 day sales period. We got observed data for 46 sales periods. Data of randomly selected 40 sales periods was used as learning data set and data of the remaining 6 sales periods was used as testing data set. In Table 1 there are given the mean values and the standard deviations of the daily consumptions and the prescribed lower resp. upper bounds for the amount of goods to be placed on the shelves. The runs of our dynamic control for the testing data sets can be seen on Figures 1-6.

Table 1

The mean values and standard deviations of the daily consumptions and the prescribed lower and upper bounds for the amount of goods to be placed on the shelves

		Exp. val.	Std. dev.	Lower bound	Upper bound
1	Friday	6.9500	3.5515	6	20
2	Saturday	5.9000	3.7403	6	20
3	Sunday	3.6000	3.3344	6	20
4	Monday	3.2250	2.8328	6	18

5	Tuesday	3.9250	3.1816	5	18
6	Wednesday	3.8250	3.2415	5	18
7	Thursday	4.4500	3.9481	5	16
8	Friday	3.5500	3.2734	5	16
9	Saturday	3.1750	2.5709	5	16
10	Sunday	2.3750	1.8904	5	14
11	Monday	2.1000	1.7802	4	14
12	Tuesday	3.2500	2.4469	4	14
13	Wednesday	2.3250	1.9133	4	12
14	Thursday	2.3250	1.9792	4	12

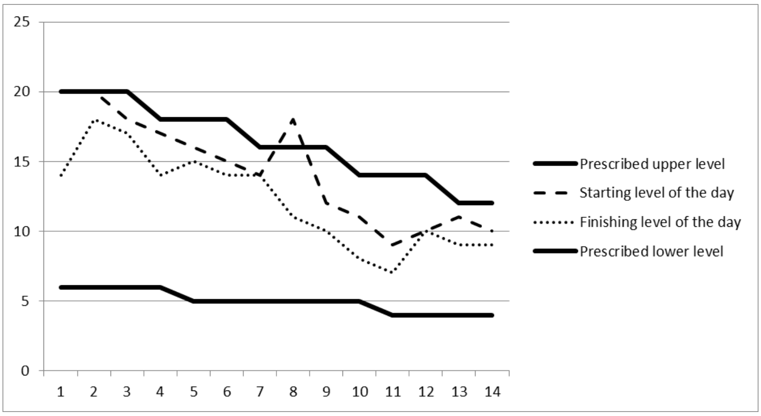


Figure 1
Run of the dynamic control for the first testing data set

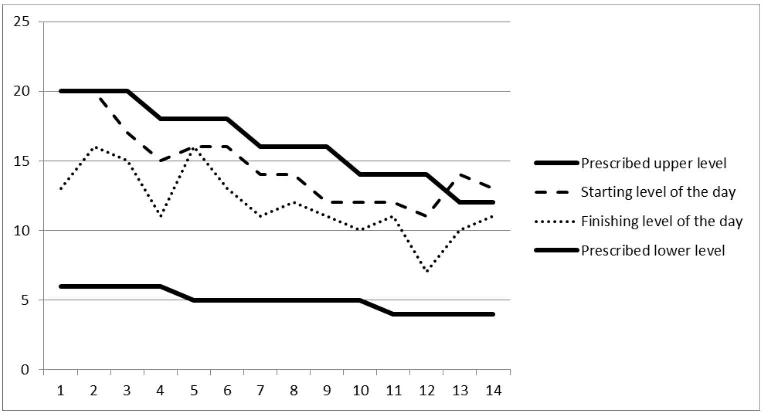


Figure 2
Run of the dynamic control for the second testing data set

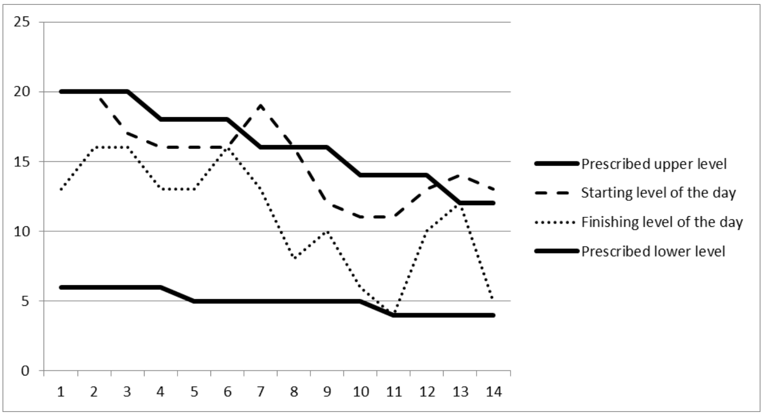


Figure 3
Run of the dynamic control for the third testing data set

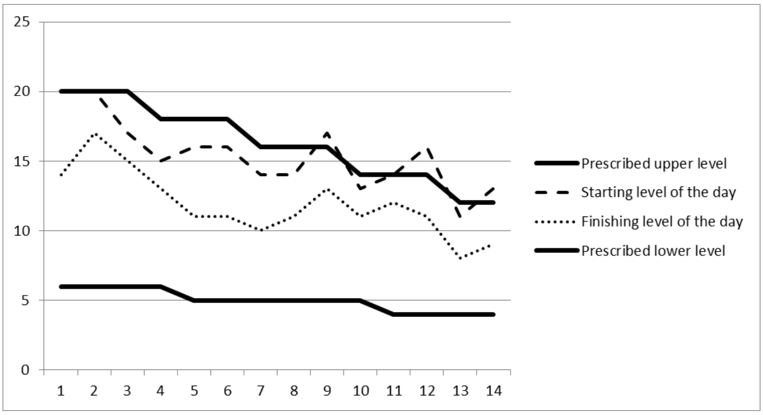


Figure 4
Run of the dynamic control for the fourth testing data set

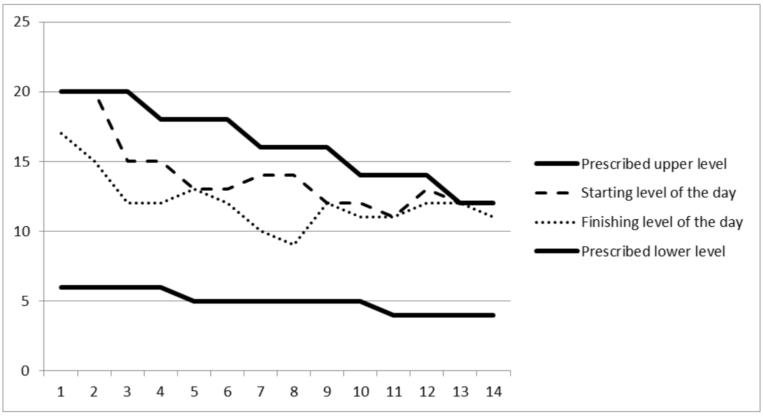


Figure 5
Run of the dynamic control for the fifth testing data set

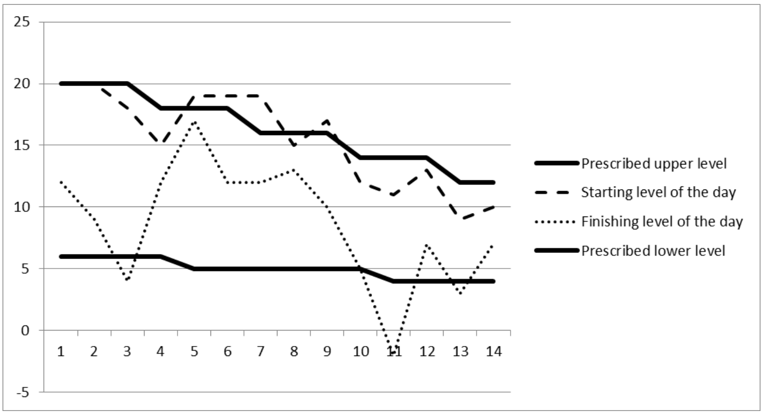


Figure 6
Run of the dynamic control for the sixth testing data set

4 Some Methods to Improve the Quality of the Registered Data and Ideas for Future Work

One of the problems, which occurred, is that the registered consumption was messy, due the fact that there were days when at the end of the day the shelves were empty. In such cases the registered consumptions were considered as the quantities displayed at the beginning of these days. We recommend that in such cases the registered quantity should somehow indicate this fact, for example by

C_i^+, and this fact should be taken into account in the procedure of forecasting the future consumptions.

Very often, the promotional sale for a given line, has effects on the demand of other goods. It would be important to investigate these effects and include them in the calculation.

Conclusions

We presented two new methods of dynamic forecasting for the consumptions within sales periods and a decision procedure based on the forecasted consumption and the prescribed levels. The first method can be applied in cases when the probability distribution of the random consumptions can be supposed to be normal, i.e. the standard deviation of the data is relatively small according to the mean value.

The advantage of the second method is that it needs no hypothesis on the theoretical probability distribution, but for accurate forecasting, it needs a larger historical dataset. The dataset could be enlarged by other observed sales periods for similar items.

Both methods presume that the sales periods were observed under equal market and advertising conditions.

References

[1] G. Cachon and M. Fisher, Campbell Soup's continuous replenishment program: Evaluation and enhanced inventory decision rules, *Production and Operations Management*, **6**, 266-276, 1997

[2] T. M. Cover and J. A. Thomas, *Elements of Information Theory,* John Wiley & Sons, New York, 2006

[3] K. H. Van Donselaar, J. Peters, A. De Jong and R. A. C. M. Broekmeulen, Analysis and forecasting of demand during promotions for perishable items, *International Journal of Production Economics*, **172**, 65-75, 2016

[4] R. Fildes and P. Goodwin, Against your better judgment? How organizations can improve their use of management judgment in forecasting, *Interfaces*, **37**, 570-576, 2007

[5] R. Fildes, K. Nikolopoulos, S. F. Crone and A. A. Syntetos, Forecasting and operational research: a review, *Journal of the Operational Research Society*, **59,** 1150-1172, 2008

[6] M. Leonard, Promotional analysis and forecasting for demand planning: a practical time series approach, https://support.sas.com/rnd/app/ets/papers/PromotionalAnalysis.pdf, 2001

[7] S. Ma, R. Fildes and T. Huang, Demand forecasting with high dimensional data: The case of SKU retail sales forecasting with intra-and inter-category

promotional information. *European Journal of Operational Research*, **249**, 245-257, 2016

[8] A. Prékopa and T. Szántai, On Optimal Regulation of a Storage Level with Application to the Water Level Regulation of a Lake, *European Journal of Operational Research*, **3**, 175-189, 1979

[9] A. Rényi, On measures of information and entropy, in: Proceedings of the fourth Berkeley Symposium on Mathematics, Statistics and Probability, 547-561, 1961

[10] C. E. Shannon, A mathematical theory of communication, *The Bell System Technical Journal*, **27**, 379-423, July 1948

[11] J. W. Taylor, Multi-item sales forecasting with total and split exponential smoothing, *Journal of the Operational Research Society*, 62, 555-563, 2011

[12] J. R. Trapero, D. J. Pedregal, R. Fildes and N. Kourentzes, Analysis of judgmental adjustments in the presence of promotions, *International Journal of Forecasting*, **29** 234-243, 2013

[13] J. R. Trapero, N. Kourentzes and R. Fildes, On the identification of sales forecasting models in the presence of promotions, *Journal of the Operational Research Society*, **66**, 299-307, 2015

Probability maximization by inner approximation

Csaba I. Fábián[1], Edit Csizmás[1], Rajmund Drenyovszki[1], Wim van Ackooij[2], Tibor Vajnai[1], Lóránt Kovács[1], Tamás Szántai[3]

[1] Department of Informatics, GAMF: Faculty of Engineering and Computer Science, John von Neumann University. Izsáki út 10, 6000 Kecskemét, Hungary.
[2] EDF Research and Development, Department OSIRIS. 1, avenue du Général de Gaulle, F-92141 Clamart Cedex France.
[3] Department of Differential Equations, Institute of Mathematics, Budapest University of Technology and Economics. Műegyetem rakpart 3-9, 1111 Budapest, Hungary.

E-mails: fabian.csaba@gamf.uni-neumann.hu, csizmas.edit@gamf.uni-neumann.hu, drenyovszki.rajmund@gamf.uni-neumann.hu, wim.van-ackooij@edf.fr, vajnai.tibor@gamf.uni-neumann.hu, kovacs.lorant@gamf.uni-neumann.hu, szantai@math.bme.hu.

Abstract: We solve probability maximization problems using an approximation scheme that is analogous to the classic approach of p-efficient points, proposed by Prékopa to handle chance constraints. But while p-efficient points yield an approximation of a level set of the probabilistic function, we approximate the epigraph. The present scheme is easy to implement and is immune to noise in gradient computation.

Keywords: stochastic programming; probabilistic constraints; applications.

1 Introduction

A probabilistic constraint is of the following type:

$$\mathrm{P}(g(x,\xi) \leq 0) \geq p, \tag{1}$$

where $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^k$ is a mapping, $\xi \in \mathbb{R}^m$ a multivariate random vector with associated probability measure P and $p \in [0,1]$ a user defined safety level. When $k \geq 1$, the terminology *joint probabilistic constraint* is also frequently employed, since we would like the random inequality system $g(x,\xi) \leq 0$ to hold with high-enough probability.

We are interested in two general optimization problems associated with (1), namely that of maximizing the probability function and a classic problem of optimizing under constraint (1). These appear under the following form:

$$\max \mathrm{P}(g(x,\xi) \le 0) \quad \text{subject to} \quad x \in X, \quad \text{and} \tag{2}$$

$$\min c^T x \quad \text{subject to} \quad \mathrm{P}(g(x,\xi) \le 0) \ge p,\ x \in X, \tag{3}$$

where X is a convex compact set. In many applications X is a polyhedral set $X = \{x \in \mathbb{R}^n\ : Ax \le b\}$. We will make the assumption that g is jointly-quasi concave and that ξ admits a density (with respect to the Lebesgue-measure) disposing of generalized concavity properties as well. Under these assumptions the mapping $x \mapsto \phi(x) := \mathrm{P}(g(x,\xi) \le 0)$ also disposes of generalized concavity properties. In particular problems (2) and (3) are convex optimization problems under these assumptions.

In the present paper we will deal with the special case when $g(x,\xi) = \xi - Tx$. Then the problems (2) and (3) become the following:

$$\max \mathrm{P}(Tx \ge \xi) \quad \text{subject to} \quad Ax \le b, \tag{4}$$

and the probabilistic constrained problem

$$\min c^T x \quad \text{subject to} \quad \mathrm{P}(Tx \ge \xi) \ge p,\ Ax \le b, \tag{5}$$

where the decision vector is x. Given are the matrices A,T and the vectors b,c, of corresponding sizes. The probability $1 > p > 0$ is set, and the distribution of the random vector ξ is known. We assume that the feasible domains are not empty and are bounded. We assume that ξ has a continuous, logconcave distribution. It follows that the cumulative distribution function $F(z) = \mathrm{P}(z \ge \xi)$ is logconcave.

Probabilistic constraints arise in many applications such as water management, telecommunications, electricity network expansion, mineral blending, chemical engineering etc. (e.g., [21, 41, 52, 53, 55, 69, 76, 78]). With the advance of infocommunication technologies, new areas of application are emerging, e.g., smart grids and transportation systems.

For an overview of recent theory and algorithmic treatment of probabilistic constraints we refer to [9,49,50]. Other monographs dealing (partially) with probabilistic constraints are [8,26,38] and [37], where the latter focussed more on algorithms.

A brief history of methods for solving probabilistically constrained problems

Programming under probabilistic constraints as a decision model under uncertainty, has been introduced by [7]. In this paper the authors use the term *chance constrained programming* for this model and its variants as well as extensions presented, among others, in the paper [6]. However these early chance constrained models were based on *individual chance constraints*, i.e., instead of a constraint of the type in problem (3), the following type constraints were used: $\mathrm{P}(g_i(x,\xi) \le 0) \ge p_i,\ \ i = 1,\ldots,k$. Programming under probabilistic constraint with a random right hand side vector ξ (as it stands in problem (5)), having stochastically independent components , was first considered by [39]. The more general problem (3), where ξ is allowed to have stochastically dependent components, was introduced by Prékopa [44,46] and further investigated by him and his followers. A significant step for the numerical treatment of probabilistic constraints was laid out when convexity statements based on the theory of logconcave measures were developed by Prékopa [45,47] and later generalized by [3,4,63]. Recent advances in convexity statements for probabilistic constraints are based on eventual convexity and can be found in [23,24,70]

In [52], Prékopa and co-authors developed a model (STABIL) for a planning problem in the Hungarian electrical energy sector, which is of the form (5). The resulting stochastic programming problem is solved by a feasible direction method of Zoutendijk [81]. It should be noted however that Zoutendijk's method lacks the global convergence property as shown in [64]. We refer to the discussion in [40] for further information.

Cutting-plane methods were also developed for the probabilistic constrained problem, approximating the level set $M(p) := \{x \in \mathbb{R}^n \ : \mathrm{P}(g(x,\xi) \leq 0) \geq p\}$. The method of Prékopa and Szántai [53] applies a Slater point to determine where to construct the next cut. (Namely, the intersection of the boundary of $M(p)$ on the one hand, and the interval connecting the Slater point with the current iterate on the other hand.) The method is related to that of Veinott [79]. In his solver built for the STABIL problem, Szántai [61] developed a careful interval bisection algorithm for safely computing the intersection point on the boundary of $M(p)$ when the probability values defining the probability constraints cannot be calculated with arbitrary high precision. He also applied Veinott's technique of moving the Slater point in course of the solution process, which results in faster convergence and makes the supporting hyperplane method equivalent to a method of Zoutendijk [81]. Mayer [37] proposed a central cutting plane method, an adaptation of Elzinga and Moore [13]. Cutting-plane methods converge in less iterations than feasible direction methods do, since former gradient information is retained. These methods obviously require that one is able to compute the gradient of $\phi(x) := \mathrm{P}(g(x,\xi) \leq 0)$ efficiently. Identifying conditions under which ϕ is differentiable has lead to the development of two main research directions. The first direction exploits no specific knowledge of ξ or its underlying distribution, but only differentiability properties of its density and differentiability of g. Then under several additional assumptions, including the assumption that $B(x) := \{z \in \mathbb{R}^m \ : g(x,z) \leq 0\}$ is bounded in a neighbourhood of x, one can represent the gradient of ϕ as an integral over $B(x)$ and/or its boundary $\partial B(x)$. We refer to [35,36,65–67] and the references contained therein for more on this research direction. We note here, that the condition that $B(x)$ remains bounded around a point x rules out the study of distribution functions. The second research direction exploits specific knowledge of the underlying distribution of ξ and tries to build a link between any component of the gradient of ϕ and the evaluation of a quantity akin to ϕ. This direction was explored in [22,44,54,60,73–75]. When combined with sophisticated software such as for instance Genz' code [17,19] for multivariate normal distributions, high dimensional problems can be solved with significant efficiency (e.g., a case with $k = 168$ is examined in [72]).

In the supporting hyperplane method, the inaccuracy of evaluating ϕ needs to be taken into account when computing the intersection point on the boundary of $M(p)$. We refer to [1] for such an approach. Still inaccuracy of $\nabla\phi$ may result in a cut cutting into the level set $M(p)$. This leads to the development of the notion of *upper-oracle* in [77] and specialized proximal ([77]) and level ([72]) bundle methods for probabilistically constrained problems with underlying convexity structure.

A non-standard dual formulation for problems of type (5) was proposed by Komáromi [27, 28]. This is a max-min formulation, the inner problem being minimization of a linear function over the level set $M(p)$. For the solution of the dual problem, a special feasible direction method is developed in [27].

We are going to focus on p-efficient point approaches. Other recent algorithmic approaches for probabilistically constrained programming are the penalty approach [14], scenario approximation [5], convex approximation [42], sample average approximation and integer programming [31–33,43], binarization approaches [29,30].

On p-efficient point approaches

When the mapping g is of the form $g(x,z) := z - h(x)$, the probabilistic constraint is said to be separable and properties of $\phi(x) = \mathrm{P}(g(x,\xi) \leq 0) := F_\xi(h(x))$ relate directly to that of the multivariate distribution function F_ξ. In this setting, Prékopa [48] initiated a new solution approach by introducing the concept of p-efficient points. A point z is p-efficient if and only if $F_\xi(z) \geq p$ and there exists no z' such that $z' \leq z$, $z' \neq z$, $F_\xi(z') \geq p$. Prékopa, Vizvári, and Badics [56] employ this concept in the solution of problems of the type (5), where the random parameters have a discrete finite distribution. They first enumerate all the p-efficient points, and based on these, propose a convex relaxation of the problem. The relaxed probabilistic constraint prescribes the existence of a point z in the convex hull of the p-efficient points such that $h(x) \geq z$ holds. The relaxed problem is then solved by a cutting-plane method. In essence, the cuts generated correspond to facets of the convex hull of the p-efficient points.

Prékopa [51] considers a problem equivalent to (5), where the random vector has a continuous logconcave distribution. He combines the cutting-plane method of [56] with the supporting hyperplane method of Szántai [61]. The resulting hybrid method simultaneously constructs inner and outer approximations of the level set $M(p)$. The supporting hyperplane method is used to generate p-efficient points in the course of the solution process. (More general stochastic programming models are also proposed in [51], but in the present paper we restrict ourselves to simpler formulations.)

Dentcheva, Prékopa, and Ruszczyński [12] consider problems of type (5), where the random parameters are integer valued. They prove that the probabilistic constraint is essentially convex, in case the random parameters have an r-concave distribution. The probabilistic constraint is formulated in a split form: $h(x) \geq z$, where z belongs to (a discrete version of) the level set $M(p)$. These authors construct a Lagrangian dual by relaxing the constraint $h(x) \geq z$, and observe that the dual functional splits into the sum of two functionals. The addend functionals are the respective optimal objective value functions of two simpler problems. The first auxiliary problem is a linear programming problem, and the second auxiliary problem is about minimizing a linear function over (a discrete version of) the level set $M(p)$. Once the dual problem is solved, a primal optimal solution can be constructed, though technical problems may occur and need to be overcome. These authors also develop a new specialized method which separates the generation of p-efficient points and the solution of the approximate problem based on known p-efficient points. The new method, called cone generation, employs the time-honoured concept of column generation. The inherent link with integer programming is given in [80].

Dentcheva, Lai, and Ruszczyński [10] extend these results to general convex problems, and general (r-concave) distributions. The probabilistic constraint is formulated in a split form, and the Lagrangian dual is constructed by relaxing the constraint $h(x) \geq z$. The dual functional splits into the sum of two functionals, like in the special case discussed in [12]. The first auxiliary problem, however, is a well-structured convex programming problem, instead of the linear programming problem of [12]. The difficult part is still the second auxiliary problem, minimizing a linear function over $M(p)$. These authors develop a dual method, and propose a way of recovering a primal solution. Moreover, they extend the cone generation method to a general primal-dual method.

Dentcheva and Martinez [11] developed a regularized version of the dual method of [10]. Moreover they developed a progressive augmented Lagrangian method that is a primal-dual-type method. The latter method turns out to be more efficient as it requires the solution of fewer minimization problems over the level set $M(p)$.

A solution framework that includes and extends various existing formulations was developed by Van Ackooij, Berge, de Oliveira and Sagastizábal [71].

Contribution

In the present paper, we construct polyhedral approximations of the epigraphs of the probabilistic functions in problems (4) and (5). This is analogous to the use of p-efficient points. But while p-efficient points yield an approximation of a level set, we approximate the epigraph. We formulate dual problems that are analogous to those of [12,27], and [10]. The present scheme yields very convenient duals, simple formulations using conjugate functions.

The solution approaches proposed in [12] and [10] can be adapted to the present approximation scheme and dual formulations. Finding a new approximation point in the present scheme is easier than finding a p-efficient point in the schemes of [12] or [10]. – In the latter schemes, finding a p-efficient point amounts to minimization over the level set $M(p)$. In the present scheme, an approximation point is found by unconstrained minimization.

The present simple models and methods expose an important contrast between column generation methods and direct cutting-plane methods. Direct cutting-plane methods for probabilistic functions are difficult to implement due to noisy gradient computation. A practicable implementation requires sophisticated tolerance handling. In contrast, the column generation approach is immune to noise in gradient computation.

2 Problem and model formulation

Using the distribution function $F(z)$, let $\phi(z) = -\log F(z)$. Of course it is a convex function, due to the logconcavity of $F(z)$. Taking into account the monotonicity of the distribution function, Problem (4) can be written as

$$\min\ \phi(z) \quad \text{subject to} \quad Ax - b \leq 0,\ z - Tx \leq 0. \tag{6}$$

This problem has an optimal solution, due to our assumption that the feasible domain of (4) is not empty and is bounded. Introducing non-positive multiplier vectors y, u to the respective constraints, we formulate the Lagrangian relaxation of (6):

$$\inf_{x,z} \{\ \phi(z) - y^T(Ax-b) - u^T(z-Tx)\ \}$$

$$= \quad \inf_z \{\phi(z) - u^T z\} \quad + \inf_x (-y^T A + u^T T)x \quad + y^T b.$$

The first addend is by definition $-\phi^\star(u)$, where $\phi^\star$ is the convex conjugate of ϕ. The second addend is finite iff $-y^T A + u^T T = 0^T$. Hence the Lagrangian dual of (6) can be written as

$$\max_{y,u\leq 0} \{y^T b - \phi^\star(u)\} \quad \text{subject to} \quad -y^T A + u^T T = 0^T. \tag{7}$$

According to the theory of convex duality, this problem has an optimal solution, since the primal problem (6) has an optimal solution.

Concerning the probabilistic constraint, let $\pi = -\log p$. We formulate (5) as

$$\min\ c^T x \quad \text{subject to} \quad Ax - b \leq 0,\ z - Tx \leq 0,\ \phi(z) - \pi \leq 0. \tag{8}$$

This problem has an optimal solution, due to our assumption that the feasible domain of (5) is not empty and is bounded. Introducing the multiplier vectors $-y \geq 0$, $-u \geq 0$, $\nu \geq 0$ to the respective constraints, we formulate the Lagrangian relaxation of (8):

$$\inf_{x,z} \{\ c^T x - y^T(Ax-b) - u^T(z-Tx) + \nu\,(\phi(z) - \pi)\ \}$$

$$= \quad \inf_z \{\nu\phi(z) - u^T z\} \quad + \inf_x (c^T - y^T A + u^T T)x \quad + y^T b - \nu\pi.$$

The first addend is by definition $-(\nu\phi)^\star(u)$. The second addend is finite iff $c^T = y^T A - u^T T$. Hence the Lagrangian dual of (8) can be written as

$$\max \left\{\ y^T b - \nu\pi - (\nu\phi)^\star(u)\ \right\} \quad \text{subject to}\ y, u \leq 0,\ \nu \geq 0,\ c^T = y^T A - u^T T. \tag{9}$$

Remark. The function $(\nu, u) \mapsto (\nu\phi)^\star(u) = \sup_z \{u^T z - \nu\phi(z)\}$ is convex by definition, and given $(\hat{\nu}, \hat{u})$ in the effective domain, a gradient can be computed by finding the optimal z.

In this paper we focus on unconstrained problems. The proposed algorithms can be extended to the constrained case.

Polyhedral models

Suppose we evaluated the function $\phi(z)$ in the points z_i $(i = 0,1,\ldots,k)$. These result the function $\phi_k(z)$, an inner approximation (polyhedral convex upper approximation) of $\phi(z)$, in the usual way: given z, let

$$\phi_k(z) = \min \sum_{i=0}^{k} \lambda_i \phi(z_i) \quad \text{such that} \;\; \lambda_i \geq 0, \; \sum_{i=0}^{k} \lambda_i = 1, \quad \sum_{i=0}^{k} \lambda_i z_i = z. \tag{10}$$

If $z \notin \mathrm{Conv}(z_0,\ldots,z_k)$, then we have $\phi_k(z) = +\infty$ by definition.

The following problem is the current polyhedral model of (6):

$$\min \; \phi_k(z) \quad \text{subject to} \quad Ax - b \leq 0, \; z - Tx \leq 0. \tag{11}$$

We assume that (11) is feasible, i.e., its optimum is $< +\infty$. This can be ensured by the selction of the vectors $z_0,\ldots,z_k$. The convex conjugate of ϕ_k can be computed by taking into account a finite set only, hence

$$\phi_k^\star(u) = \max_{0 \leq i \leq k} \{u^T z_i - \phi(z_i)\}. \tag{12}$$

– The above observation is in accordance with Chapter X Section 3.4 of Hiriart-Urruty and Lemaréchal [25]. – Of course $-\phi_k^\star$ is a cutting-plane approximation (polyhedral concave upper approximation) of $-\phi^\star$. Hence the following problem is a cutting-plane model of (7):

$$\max_{y,u \leq 0} \{y^T b - \phi_k^\star(u)\} \quad \text{subject to} \quad -y^T A + u^T T = 0^T. \tag{13}$$

It is easy to check that (11) and (13), considered as linear programming problems, form a primal-dual pair. We are going to examine the primal problem.

Linear programming formulation

Introducing the notation $\phi_i = \phi(z_i)$ $(i = 0,\ldots,k)$, the primal model problem (11) can be formulated as follows. – Dual variables corresponding to the different constraints are indicated in the right-hand column.

$$\begin{array}{llllll}
\min & \sum_{i=0}^{k} \phi_i \lambda_i & & & & \\
\text{such that} & \lambda_i \geq 0 & & (i = 0,\ldots,k), & & \\
& \sum_{i=0}^{k} \lambda_i & & = 1, & \perp & \vartheta \in \mathbb{R} \\
& \sum_{i=0}^{k} \lambda_i z_i & -Tx & \leq 0, & \perp & u \leq 0 \\
& & Ax & \leq b. & \perp & y \leq 0
\end{array} \tag{14}$$

Let us assume that the primal model problem (14) has a feasible solution. Let $(\overline{\lambda}_0,\ldots,\overline{\lambda}_k,\overline{x})$ and $(\overline{\vartheta},\overline{u},\overline{y})$ denote an optimal solution and an optimal dual solution, respectively – both existing due to our assumption. Let moreover

$$\overline{z} = \sum_{i=0}^{k} \overline{\lambda}_i z_i. \tag{15}$$

Observation 1. *We have* $\phi_k(\overline{z}) \;=\; \sum_{i=0}^{k} \phi_i \overline{\lambda}_i \;=\; \overline{\vartheta} + \overline{u}^T \overline{z}$.

Proof. The first equality follows from the equivalence of (14) on the one hand, and (10)-(11) on the other hand.

The second inequality is a consequence of complementarity. $\overline{\lambda}_i > 0$ implies that the reduced cost of the ith column is 0 in (14), hence $\overline{\vartheta} + \overline{u}^T z_i = \phi_i$. It follows that

$$\sum_{i=0}^{k} \phi_i \overline{\lambda}_i \;=\; \sum_{i=0}^{k} \left(\overline{\vartheta} + \overline{u}^T z_i\right) \overline{\lambda}_i \;=\; \overline{\vartheta} \sum_{i=0}^{k} \overline{\lambda}_i \;+\; \overline{u}^T \sum_{i=0}^{k} \overline{\lambda}_i z_i.$$

3 Column generation

We solve (6) by iteratively adding improving columns to the primal model (14). An optimal dual solution (i.e., shadow price vector) of the current model problem is $(\overline{\vartheta},\overline{u},\overline{y})$.

Given a vector z, we can add the corresponding column $(1,z,0)$ with objective component $\phi(z)$. This is an impoving column if its reduced cost is positive; formally, if $\overline{\rho}(z) > 0$ holds for

$$\overline{\rho}(z) \;:=\; \left(\overline{\vartheta},\overline{u}\right)^T (1,z) - \phi(z) \;=\; \overline{\vartheta} \;+\; \overline{u}^T z - \phi(z). \tag{16}$$

The vector yielding the best reduced cost can be found by maximizing $\overline{\rho}(z)$. Let $\overline{\mathcal{R}}$ denote the optimal objective value.

If $\overline{\mathcal{R}}$ is small, then $(\overline{x},\overline{z})$ is a near-optimal solution to (6). Otherwise an improving column can be constructed to (14).

A practical way of finding an improving column

In order to maximize the reduced cost, we can apply a steepest descent method to $-\overline{\rho}(z)$, a natural starting point being $\overline{z}$. However, we found the computational effort prohibitive. Hence we propose to perform just a single line search. As theoretical motivation, we put forward the following well-known theorem. (It can be found in [34] or [57].)

Theorem 1. *Let the convex function $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable. Assume that*

$$\alpha I \preceq \nabla^2 f(z) \preceq \omega I \qquad (z \in \mathbb{R}^n), \tag{17}$$

where $0 < \alpha \leq \omega$*, I is the identity matrix, and the relation* $U \preceq V$ *between matrices means that* $V - U$ *is positive semidefinite. We minimize* f *using a steepest descent method, starting from a point* z^0*. Let* $z^1, \ldots, z^j, \ldots$ *denote the iterates obtained by applying exact line search at each step. Denoting* $\mathcal{F} = \min_z f(z)$*, we have*

$$f\left(z^j\right) - \mathcal{F} \;\leq\; \left(1 - \frac{\alpha}{\omega}\right)^j \left[\, f\left(z^0\right) - \mathcal{F} \,\right]. \tag{18}$$

Remark. Similar results can be proven for the case when approximate minimizers are found in the line search procedures. See a discussion on Armijo's rule in [34].

Corollary. *Provided Theorem 1 is applicable to* $f(z) = -\overline{\rho}(z)$*, we can construct a fairly good improving vector in the column generation scheme. Namely, let* β $(0 < \beta \ll 1)$ *be given. Taking a finite (and moderate) number of steps with the steepest descent method, we find a vector* $\widehat{z}$ *satisfying*

$$\overline{\rho}\left(\widehat{z}\right) \;\geq\; (1-\beta)\,\overline{\mathcal{R}}.$$

Proof. Substituting $f(z) = -\overline{\rho}(z)$ and $z^0 = \overline{z}$ in (18), and introducing the notation $\rho = 1 - \alpha/\omega$, we get

$$\overline{\mathcal{R}} - \overline{\rho}\left(z^j\right) \;\leq\; \rho^j \left[\, \overline{\mathcal{R}} - \overline{\rho}\left(\overline{z}\right) \,\right]. \tag{19}$$

(We have $\mathcal{F} = -\overline{\mathcal{R}}$ by definition.) From $\phi_k(.) \geq \phi(.)$ and Observation 1, we get

$$\overline{\rho}\left(\overline{z}\right) \;=\; \overline{\vartheta} \,+\, \overline{u}^T\overline{z} - \phi\left(\overline{z}\right) \;\geq\; \overline{\vartheta} \,+\, \overline{u}^T\overline{z} - \phi_k\left(\overline{z}\right) \;=0$$

Due to non-negativity, $\overline{\rho}\left(\overline{z}\right)$ can be discarded in (19), and we get

$$\overline{\rho}\left(z^j\right) \;\geq\; \left(1-\rho^j\right)\overline{\mathcal{R}}.$$

Selecting j such that $\rho^j \leq \beta$ yields an appropriate $\widehat{z} = z^j$. □

Setting $j = 1$ always resulted in a good improving vector in our computational experiments. The above discussion is only meant as motivation for performing a single line search, showing that the procedure works in an ideal case. The condition (17) obviously does not hold for every z with $f(z) = -\overline{\rho}(z)$. However, in the case of normal distribution, there exists a bounded box $\mathcal{Z}$ such that the probability weight outside $\mathcal{Z}$ can be ignored. For the sake of simplicity let us assume that the polyhedron $\mathcal{T} = \{Tx | Ax \leq b\}$ is bounded, and that $\mathcal{T} \subset \mathcal{Z}$. Then we'll always have $\overline{z} \in \mathcal{Z}$, provided the primal model (14) has been properly initialized. Starting from $\overline{z} \in \mathcal{Z}$, we perform a single line search. Due to special characteristics of the function $\phi(z)$ and due to $\overline{u} \leq 0$ being boundable, this line search can be restricted to a bounded neighborhood of $\mathcal{Z}$. Such restriction would justify assumption (17). However, we implemented a simple approximate line search without restriction, and still found that iterates fell into a relatively small box.

4 Implementation issues

For the implementation of our method and computational study we used MATLAB with the IBM ILOG CPLEX (Version 12.6.3) optimization toolbox.

The master problem

We assume that the distribution is standard normal. Let r denote the number of the components of the random vector (equal to the number of the rows of the matrix T).

First we look for an appropriate $z_0 \in \mathbb{R}^r$ vector whose inclusion makes the primal model problem (14) feasible. This is done by solving the problem

$$\begin{array}{llll} \max & t & & \\ \text{such that} & 1t & -Tx & \leq 0, \\ & & Ax & \leq b, \end{array} \tag{20}$$

where $t \in \mathbb{R}$, and $1 \in \mathbb{R}^r$ denotes a vector consisting of ones. If (20) has no feasible solution then the original problem is also infeasible. On the other hand, if the objective value is not bounded then probability 1 can be achieved in the original problem. Let $z_0 = 1t^\star$, with $t^\star$ denoting an optimal solution of (20).

Let $\mathcal{Z} \subset \mathbb{R}^r$ denote a bounded box such that the probability weight outside $\mathcal{Z}$ can be ignored. In our case the distribution is standard normal, hence we consider an r-dimensional box $\mathcal{Z}$ that it is symmetrical with respect to the origo. In our experiments we worked with a box such that $\mathrm{P}(\mathcal{Z}) \approx 0.99$.

Let $z^{max} = (z_1^{max}, \ldots, z_r^{max})$ denote maximal vertex of $\mathcal{Z}$. To ease the solution of the primal model problem (14), we initialize it by adding the following vectors (besides z_0, above)

$$\begin{array}{lll} z_\ell & = \left(z_1^{max}, \ldots, z_{\ell-1}^{max}, 0, z_{\ell+1}^{max}, \ldots, z_r^{max}\right) & (\ell = 1, \ldots, r), \\ z_{r+1} & = 0, & \\ z_{r+2} & = z^{max}. & \end{array} \tag{21}$$

Consequently we have $k = r + 2$ in (14).

We solved the master problem with the CPLEX simplex solver, applying the optimality tolerance $1E-4$.

The oracle

In accordance with Section 3, our aim is to maximize the reduced cost (16). Since $\overline{\vartheta}$ is constant in a given iteration the oracle has to find an approximate solution to the problem $\max_z \left\{\overline{u}^T z - \phi(z)\right\}$. This problem can be reformulated as minimizing the function $\phi(z) - \overline{u}^T z$. Here $\phi(z) = -\log F(z)$ and $F(z)$ is the multidimensional normal distribution function. $\phi(z)$ is a convex function, due to the logconcavity of $F(z)$. We implemented the approximate form of the steepest descent method described in Section 3. We perform a single line search ($j = 1$) and even in this single line search, we stop with an approximate minimum. Namely, we apply the golden section ratio, see, e.g. [34]. We perform only 1 or 2 golden section ratio steps.

The steepest descent direction can be found by calculating the gradient vector of the function:

$$\nabla\left(\phi(z)-\overline{u}^T z\right)=\nabla\phi(z)-\overline{u}=-\nabla\log\left(F(z)\right)-\overline{u}=-\frac{\nabla F(z)}{F(z)}-\overline{u}. \tag{22}$$

Consequently we need to calculate the function value and gradient vector of the multidimensional normal distribution function $F(z)$. For this computation we use the formulas in section 6.6.4 of Prékopa's book [49]. By using these formulas the calculation of the gradient of a multidimensional probability distribution function can be reduced to computing conditional distribution function values.

The numerical computation of multivariate normal distribution values was performed with the QSIMVNV Matlab function implemented by Genz [18].

5 Computational study

Before describing test problems and discussing computational results, let us illustrate condition (17) with a small example.

Preliminary examinations

We illustrate the well-conditioned nature of the objective in case of a two-dimensional standard normal distribution with moderately dependent marginals (covariance 0.5).

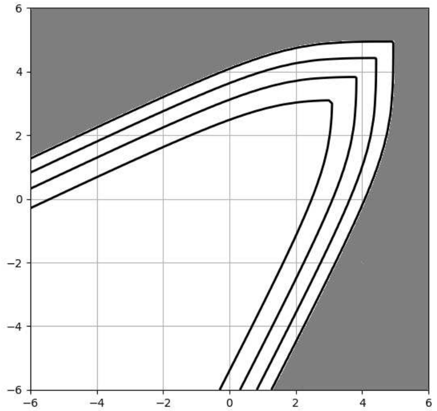


Figure 1
Smaller eigenvalue of the Hessian $\nabla^2\phi(z)$
$(-6 \le z_1, z_2 \le +6)$

We depict the eigenvalues of the Hessian matrix of $\phi(z)=-\log F(z)$, where $F(z)$ is the distribution function. We calculated the smaller and the larger of the two eigenvalues of the Hessian, while both components of z fall into the interval $[-6,+6]$.

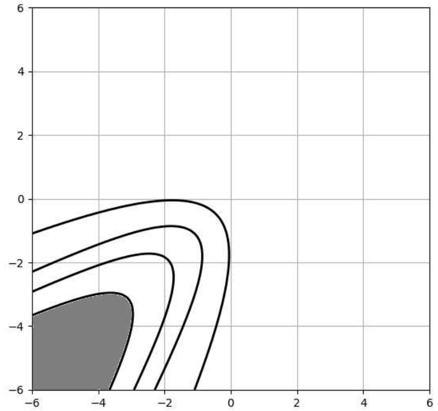


Figure 2
Larger eigenvalue of the Hessian $\nabla^2\phi(z)$
$(-6 \le z_1, z_2 \le +6)$

Figure 1 depicts the smaller eigenvalue. Contour lines from top right are $1e-5, 1e-4, 1e-3, 1e-2$. In the area not filled with gray, the smaller eigenvalue is above $1e-5$.

Figure 2 depicts the larger eigenvalue. Contour lines from top right are $1, 1.2, 1.4, 1.6$. In the area not filled with gray, the larger eigenvalue is below 1.6.

These experiments illustrate that there is a fairly large safe domain over which $\phi(z)$ is well-conditioned.

Test problems

First we considered eight test problems published in [62] by T. Szántai. These problems occur at a coffee company. The company is marketing three different blends of coffee. There is a rigid set of requirements for each of the blends according their acidity, caffeine content, liquoring value, hardness and aroma. On the first day of a particular month the company found that its available supply of green coffees was limited to 8 different types. These green coffees vary according to price, quantity available and the above mentioned five taste characteristics. The demands for the company's 3 blends during the coming month are random variables with given expected values, standard deviations and correlation coefficients. The company is confronted with the problem of determining an optimum combination of avaliable green coffees for next month's roasting operation. So they have to formulate a stochastic programming problem to satisfy all of the random demands with a prescribed (high) probability and pay the smallest possible price for the green coffees. All data and numerical results according to probability level 0.9 can be found in the paper [62]. In this paper we will call these problems 'Coffee1', ..., 'Coffee8'.

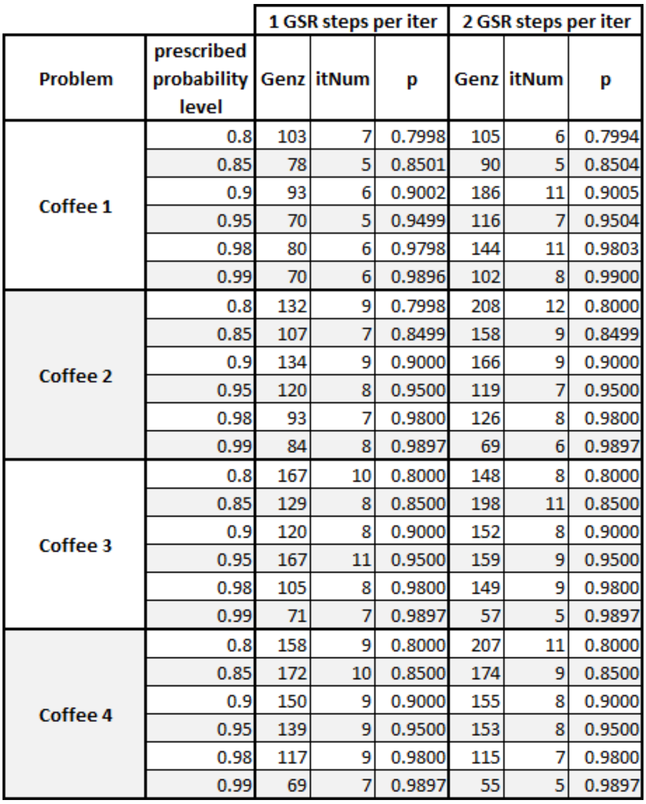
Secondly we considered an extended version of the coffee blending problem. In this extension the company is marketing five different blends of coffees and so the multivariate normal probability distribution is five dimensional. This problem will be called 'Coffee9' in this paper.

Finally we considered a cash matching problem with fifteen dimensional normal probability distribution. In this problem we are interested in investing a certain amount of cash on behalf of a pension fund that needs to make certain payments over the coming 15 years of time. Details of this problem can be found in [10] and [20]. This problem will be called 'CashMatching' in this paper.

Numerical results

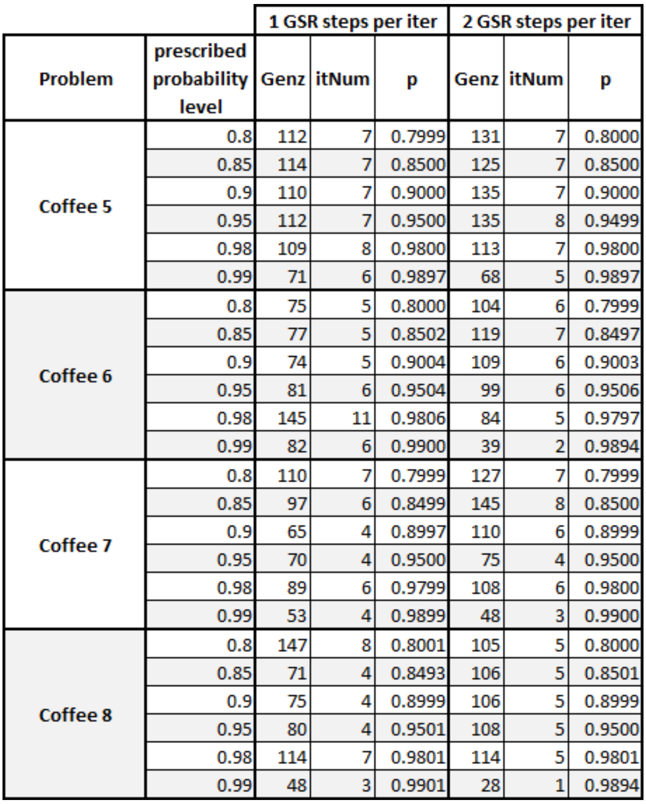
We solved each test problem with different right-hand sides of the cost constraint. Our computational results are reported in Figures 3 - 5.

Our test problems had originally been formulated as cost minimization under a probabilistic constraint. We converted the problems to probability maximization. The right hand-sides of the cost constraints had been set in such a way that the corresponding optimal probability levels would be those listed in the column 'prescribed probability level' of our tables. For these computations we used Szántai's computer code [61].



		1 GSR steps per iter			2 GSR steps per iter		
Problem	prescribed probability level	Genz	itNum	p	Genz	itNum	p
Coffee 1	0.8	103	7	0.7998	105	6	0.7994
	0.85	78	5	0.8501	90	5	0.8504
	0.9	93	6	0.9002	186	11	0.9005
	0.95	70	5	0.9499	116	7	0.9504
	0.98	80	6	0.9798	144	11	0.9803
	0.99	70	6	0.9896	102	8	0.9900
Coffee 2	0.8	132	9	0.7998	208	12	0.8000
	0.85	107	7	0.8499	158	9	0.8499
	0.9	134	9	0.9000	166	9	0.9000
	0.95	120	8	0.9500	119	7	0.9500
	0.98	93	7	0.9800	126	8	0.9800
	0.99	84	8	0.9897	69	6	0.9897
Coffee 3	0.8	167	10	0.8000	148	8	0.8000
	0.85	129	8	0.8500	198	11	0.8500
	0.9	120	8	0.9000	152	8	0.9000
	0.95	167	11	0.9500	159	9	0.9500
	0.98	105	8	0.9800	149	9	0.9800
	0.99	71	7	0.9897	57	5	0.9897
Coffee 4	0.8	158	9	0.8000	207	11	0.8000
	0.85	172	10	0.8500	174	9	0.8500
	0.9	150	9	0.9000	155	8	0.9000
	0.95	139	9	0.9500	153	8	0.9500
	0.98	117	9	0.9800	115	7	0.9800
	0.99	69	7	0.9897	55	5	0.9897

Figure 3
Computational results for problems 'Coffee1', ..., 'Coffee4'



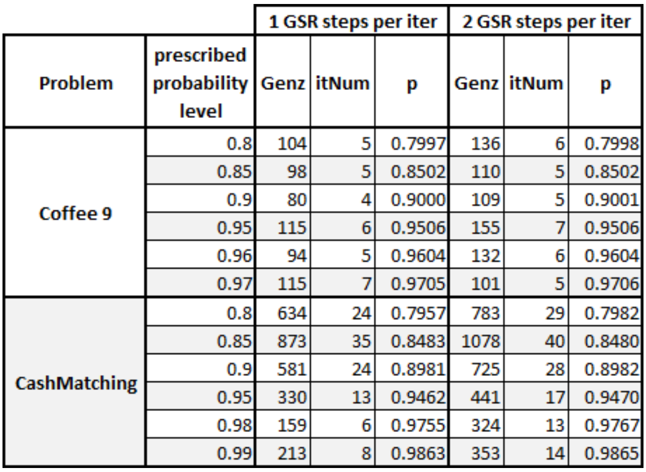
		1 GSR steps per iter			2 GSR steps per iter		
Problem	prescribed probability level	Genz	itNum	p	Genz	itNum	p
Coffee 5	0.8	112	7	0.7999	131	7	0.8000
	0.85	114	7	0.8500	125	7	0.8500
	0.9	110	7	0.9000	135	7	0.9000
	0.95	112	7	0.9500	135	8	0.9499
	0.98	109	8	0.9800	113	7	0.9800
	0.99	71	6	0.9897	68	5	0.9897
Coffee 6	0.8	75	5	0.8000	104	6	0.7999
	0.85	77	5	0.8502	119	7	0.8497
	0.9	74	5	0.9004	109	6	0.9003
	0.95	81	6	0.9504	99	6	0.9506
	0.98	145	11	0.9806	84	5	0.9797
	0.99	82	6	0.9900	39	2	0.9894
Coffee 7	0.8	110	7	0.7999	127	7	0.7999
	0.85	97	6	0.8499	145	8	0.8500
	0.9	65	4	0.8997	110	6	0.8999
	0.95	70	4	0.9500	75	4	0.9500
	0.98	89	6	0.9799	108	6	0.9800
	0.99	53	4	0.9899	48	3	0.9900
Coffee 8	0.8	147	8	0.8001	105	5	0.8000
	0.85	71	4	0.8493	106	5	0.8501
	0.9	75	4	0.8999	106	5	0.8999
	0.95	80	4	0.9501	108	5	0.9500
	0.98	114	7	0.9801	114	5	0.9801
	0.99	48	3	0.9901	28	1	0.9894

Figure 4
Computational results for problems 'Coffee5', ..., 'Coffee8'

We solved each problem with two settings of the oracle, performing either 1 or 2 Golden Section Ratio (GSR) steps in course of each line search. – The corresponding data are shown under the headers '1 GSR step per iter' and '2 GSR steps per iter'. In each case, we list the number of calls to the Genz subroutine (under the header 'Genz'), the number of oracle calls (under the header 'itNum'), and the optimum found (under the header 'p').

In each case, most of the computation time was spent in the Genz subroutines. In case of the 'Coffee' problems, performing 2 GSR steps per iteration resulted in slightly less calls to the Genz subroutine than 1 GSR step did. Interestingly, the 'CashMatching' problem was solved significantly faster when performing a single GSR step per iteration, instead of two steps. All these results indicate that approximate solution of the column generation problems is sufficient.

The $\widehat{z}$ vectors returned by the oracle always fell into a relatively small box, thereby remaining in the safe domains where the respective objective functions are well-conditioned.



		1 GSR steps per iter			2 GSR steps per iter		
Problem	prescribed probability level	Genz	itNum	p	Genz	itNum	p
Coffee 9	0.8	104	5	0.7997	136	6	0.7998
	0.85	98	5	0.8502	110	5	0.8502
	0.9	80	4	0.9000	109	5	0.9001
	0.95	115	6	0.9506	155	7	0.9506
	0.96	94	5	0.9604	132	6	0.9604
	0.97	115	7	0.9705	101	5	0.9706
CashMatching	0.8	634	24	0.7957	783	29	0.7982
	0.85	873	35	0.8483	1078	40	0.8480
	0.9	581	24	0.8981	725	28	0.8982
	0.95	330	13	0.9462	441	17	0.9470
	0.98	159	6	0.9755	324	13	0.9767
	0.99	213	8	0.9863	353	14	0.9865

Figure 5
Computational results for problems 'Coffee9' and 'CashMatching'

6 Conclusions

The proposed probability-maximization approach is based on a polyhedral approximation of the epigraph of the probabilistic function. Finding a new approximation point in the present scheme is easier than finding a p-efficient point in the classic scheme of Dentcheva, Prékopa and Ruszczyński [12]. In the present scheme, an approximation point is found by unconstrained optimization. In LP terms, this is a column generation scheme where new columns are found by maximizing reduced cost.

The inner approximating model of the epigraph is immune to noise in gradient computation, in the following sense. Suppose that at iteration k, the next iterate z_{k+1} is just a rough approximate solution of the relevant subproblem (reduced cost-maximization). As long as $\phi(z_{k+1})$ is computed with reasonable accuracy, the model remains a true inner approximation.

Our computational experiments indicate that rough approximate solution of the subproblems is sufficient for convergence. We also provide theoretical explanation of this observation. – A randomized version of the present algorithm is proposed with convergence proof in [15].

Acknowledgement

This research is supported by EFOP-3.6.1-16-2016-00006 "The development and enhancement of the research potential at Pallas Athena University" project. The Project is supported by the Hungarian Government and co-financed by the European Social Fund.

References

[1] T. Arnold, R. Henrion, A. Möller, and S. Vigerske. A mixed-integer stochastic nonlinear optimization problem with joint probabilistic constraints. *Pacific Journal of Optimization*, 10:5–20, 2014.

[2] M. Bertocchi, G. Consigli, and M.A.H. Dempster (eds). *Stochastic Optimization Methods in Finance and Energy: New Financial Products and Energy Market Strategies*. International Series in Operations Research and Management Science. Springer, 2012.

[3] C. Borell. Convex set functions in d-space. *Periodica Mathematica Hungarica*, 6:111–136, 1975.

[4] H.J. Brascamp and E.H. Lieb. On extensions of the Brunn-Minkowski and Prékopa-Leindler theorems, including inequalities for log-concave functions and with an application to the diffusion equations. *Journal of Functional Analysis*, 22:366–389, 1976.

[5] G. C. Calafiore and M. C. Campi. The scenario approach to robust control design. *IEEE Trans. Automat. Control*, 51:742–753, 2006.

[6] A. Charnes and W. Cooper. Deterministic equivalents for optimizing and satisficing under chance constraints. *Operations Research*, 11:18–39, 1963.

[7] A. Charnes, W.W. Cooper, and G.H. Symonds. Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Management Science*, 4:235–263, 1958.

[8] D. Dentcheva. Optimization models with probabilistic constraints. In G. Calafiore and F. Dabbene, editors, *Probabilistic and Randomized Methods for Design under Uncertainty*, pages 49–97. Springer, 1st edition, 2006.

[9] D. Dentcheva. *Optimisation Models with Probabilistic Constraints. Chapter 4 in [59]*. MPS-SIAM series on optimization. SIAM and MPS, Philadelphia, 2009.

[10] D. Dentcheva, B. Lai, and A. Ruszczyński. Dual methods for probabilistic optimization problems. *Mathematical Methods of Operations Research*, 60:331–346, 2004.

[11] D. Dentcheva and G. Martinez. Regularization methods for optimization problems with probabilistic constraints. *Mathematical Programming*, 138:223–251, 2013.

[12] D. Dentcheva, A. Prékopa, and A. Ruszczyński. Concavity and efficient points of discrete distributions in probabilistic programming. *Mathematical Programming*, 89:55–77, 2000.

[13] J. Elzinga and T.G. Moore. A central cutting plane method for the convex programming problem. *Mathematical Programming*, 8:134–145, 1975.

[14] Y.M. Ermoliev, T.Y. Ermolieva, G.J. Macdonald, and V.I. Norkin. Stochastic optimization of insurance portfolios for managing exposure to catastrophic risk. *Annals of Operations Research*, 99:207–225, 2000.

[15] C.I. Fábián and T. Szántai. A randomized method for smooth convex minimization, motivated by probability maximization. *Optimization Online*, 2017. (Posted at the Stochastic Programming area, in March).

[16] C.A. Floudas and P.M. Pardalos (Eds). *Encyclopedia of Optimization*. Springer - Verlag, 2nd edition, 2009.

[17] A. Genz. Numerical computation of multivariate normal probabilities. *J. Comp. Graph Stat.*, 1:141–149, 1992.

[18] A. Genz. Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1:141–150, 1992.

[19] A. Genz and F. Bretz. *Computation of multivariate normal and t probabilities.* Number 195 in Lecture Notes in Statistics. Springer, Dordrecht, 2009.

[20] R. Henrion. Introduction to chance constraint programming. Technical report, Weierstrass-Institut für Angewandte Analysis und Stochastik, 2004. www.wias-berlin.de/people/henrion/ccp.ps.

[21] R. Henrion and A. Möller. Optimization of a continuous distillation process under random inflow rate. *Computer & Mathematics with Applications*, 45:247–262, 2003.

[22] R. Henrion and A. Möller. A gradient formula for linear chance constraints under Gaussian distribution. *Mathematics of Operations Research*, 37:475–488, 2012.

[23] R. Henrion and C. Strugarek. Convexity of chance constraints with independent random variables. *Computational Optimization and Applications*, 41:263–276, 2008.

[24] R. Henrion and C. Strugarek. *Convexity of Chance Constraints with Dependent Random Variables: the use of Copulae. (Chapter 17 in [2]).* Springer New York, 2011.

[25] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms*. Springer-Verlag, 1993.

[26] P. Kall and J. Mayer. *Stochastic Linear Programming: Models, Theory, and Computation.* Springer's International Series in Operations Research and Management Science. Kluwer Academic Publishers, 2005.

[27] É. Komáromi. A dual method for probablistic constrained problems. *Mathematical Programming Study*, 28:94–112, 1986. (Special issue on stochastic programming, A. Prékopa and R.J.-B. Wets, editors).

[28] É. Komáromi. On properties of the probabilistic constrained linear programming problem and its dual. *Journal of Optimization Theory and Applications*, 55:377–390, 1987.

[29] M. A. Lejeune. Pattern-based modeling and solution of probabilistically constrained optimization problems. *Operations Research*, 60(6):13561372, 2012.

[30] M. A. Lejeune and F. Margot. Solving chance-constrained optimization problems with stochastic quadratic inequalities. *Operations Research*, page 139, 2016.

[31] X. Liu, S. Küçükyavuz, and J. Luedtke. Decomposition algorithms for two-stage chance-constrained programs. *Mathematical Programming*, 157(1):219–243, 2016.

[32] J. Luedtke and S. Ahmed. A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 19:674–699, 2008.

[33] J. Luedtke, S. Ahmed, and G.L. Nemhauser. An integer programming approach for linear programs with probabilistic constraints. *Mathematical Programming*, 122(2):247–272, 2010.

[34] D.G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. International Series in Operations Research and Management Science. Springer, 2008.

[35] K. Marti. Differentiation of probability functions : The transformation method. *Computers and Mathematics with Applications*, 30:361–382, 1995.

[36] K. Marti. Differentiation of probability functions : The transformation method. *Math. Programming*, 75(2):201–220, 1996.

[37] J. Mayer. *Stochastic Linear Programming Algorithms: A Comparison Based on a Model Management System*. Gordon and Breach Science Publishers, 1998.

[38] J. Mayer. *On the Numerical solution of jointly chance constrained problems. Chapter 12 in [68]*. Springer, 1st edition, 2000.

[39] B.L. Miller and H.M. Wagner. Chance constrained programming with joint constraints. *Operations Research*, 13:930–945, 1965.

[40] M. Minoux. *Programmation Mathématique: Théorie et Algorithmes*. Tec & Doc Lavoisier, 2nd edition, 2007.

[41] D.R. Morgan, J.W. Eheart, and A.J. Valocchi. Aquifer remediation design under uncertainty using a new chance constraint programming technique. *Water Resources Research*, 29:551–561, 1993.

[42] A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *SIAM Journal of Optimization*, 17(4):969–996, 2006.

[43] B. Pagnoncelli, S. Ahmed, and A. Shapiro. Sample average approximation method for chance constrained programming: Theory and applications. *J. Optim. Theory Appl*, 142:399–416, 2009.

[44] A. Prékopa. On probabilistic constrained programming. In H.W. Kuhn, editor, *Proceedings of the Princeton Symposium on Mathematical Programming*, pages 113–138. Princeton University Press, Princeton, New Jersey, 1970.

[45] A. Prékopa. Logarithmic concave measures with applications to stochastic programming. *Acta Scientiarium Mathematicarum (Szeged)*, 32:301–316, 1971.

[46] A. Prékopa. Contributions to the theory of stochastic programming. *Mathematical Programming*, 4:202–221, 1973.

[47] A. Prékopa. On logarithmic concave measures and functions. *Acta Scientiarium Mathematicarum (Szeged)*, 34:335–343, 1973.

[48] A. Prékopa. Dual method for a one-stage stochastic programming problem with random rhs obeying a discrete probabiltiy distribution. *Z. Operations Research*, 34:441–461, 1990.

[49] A. Prékopa. *Stochastic Programming*. Kluwer Academic Publishers, Dordrecht, 1995.

[50] A. Prékopa. *Probabilistic programming. In [58] (Chapter 5).* Elsevier, Amsterdam, 2003.

[51] A. Prékopa. On the relationship between probabilistic constrained, disjunctive and multiobjective programming. Technical Report 7-2007, Rutgers Center for Operations Research, Rutgers University, Piscataway, NJ, 2007. (RUTCOR Research Report).

[52] A. Prékopa, S. Ganczer, I. Deák, and K. Patyi. The STABIL stochastic programming model and its experimental application to the electrical energy sector of the Hungarian economy. In M.A.H. Dempster, editor, *Stochastic Programming*, pages 369–385. Academic Press, London, 1980.

[53] A. Prékopa and T. Szántai. Flood control reservoir system design using stochastic programming. *Math. Programming Study*, 9:138–151, 1978.

[54] A. Prékopa and T. Szántai. A new multivariate gamma distribution and its fitting to empirical streamflow data. *Water Resources Research*, 14:19–24, 1978.

[55] A. Prékopa and T. Szántai. On optimal regulation of a storage level with application to the water level regulation of a lake. *European Journal of Operations Research*, 3:175–189, 1979.

[56] A. Prékopa, B. Vizvári, and T. Badics. Programming under probabilistic constraint with discrete random variable. In F. Giannesi, T. Rapcsák, and S. Komlósi, editors, *New Trends in Mathematical Programming*, pages 235–255. Kluwer, Dordrecht, 1998.

[57] A. Ruszczyński. *Nonlinear Optmization*. Princeton University Press, 2006.

[58] A. Ruszczyński and A. Shapiro. *Stochastic Programming*, volume 10 of *Handbooks in Operations Research and Management Science*. Elsevier, Amsterdam, 2003.

[59] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming. Modeling and Theory*, volume 9 of *MPS-SIAM series on optimization*. SIAM and MPS, Philadelphia, 2009.

[60] T. Szántai. *Numerical evaluation of probabilities concerning multidimensional probability distributions*. PhD thesis, Hungarian Academy of Sciences, 1985.

[61] T. Szántai. A computer code for solution of probabilistic-constrained stochastic programming problems. In Y.M. Ermoliev and R.J.-B. Wets, editors, *Numerical Techniques for Stochastic Optimization*, pages 229–235. Springer-Verlag, Berlin, 1988.

[62] T. Szántai. Probabilistic constrained programming and distributions with given marginals. In J. Stepan V. Benes, editor, *Distributions with Given Marginals and Moment Problems*, pages 205–210, 1997.

[63] E. Tamm. On g-concave functions and probability measures (russian). *Eesti NSV Teaduste Akademia Toimetised, Füüsika-Matemaatika*, 28:17–24, 1977.

[64] D.M. Topkis and A.F. Veinott. On the convergence of some feasible direction algorithms for nonlinear programming. *SIAM Journal on Control*, 5(2):268–279, 1967.

[65] S. Uryas'ev. Derivatives of probability functions and integrals over sets given by inequalities. *Journal of Computational and Applied Mathematics*, 56(1-2):197–223, 1994.

[66] S. Uryas'ev. Derivatives of probability functions and some applications. *Annals of Operations Research*, 56:287–311, 1995.

[67] S. Uryas'ev. *Derivatives of probability and Integral functions: General Theory and Examples. Appearing in [16].* Springer - Verlag, 2nd edition, 2009.

[68] S. Uryas'ev (ed). *Probabilistic Constrained Optimization: Methodology and Applications*. Kluwer Academic Publishers, 2000.

[69] W. van Ackooij. Decomposition approaches for block-structured chance-constrained programs with application to hydro-thermal unit commitment. *Mathematical Methods of Operations Research*, 80:227253, 2014.

[70] W. van Ackooij. Eventual convexity of chance constrained feasible sets. *Optimization (A Journal of Math. Programming and Operations Research)*, 64:1263–1284, 2015.

[71] W. van Ackooij, V. Berge, W. de Oliveira, and C. Sagastizábal. Probabilistic optimization via approximate p-efficient points and bundle methods. *Computers & Operations Research*, 77:177–193, 2017.

[72] W. van Ackooij and W. de Oliveira. Level bundle methods for constrained convex optimization with various oracles. *Computation Optimization and Applications*, 57(3):555–597, 2014.

[73] W. van Ackooij and R. Henrion. (sub-)gradient formulae for probability functions of random inequality systems under gaussian distribution. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):63–87, 2017.

[74] W. van Ackooij, R. Henrion, A. Möller, and R. Zorgati. On probabilistic constraints induced by rectangular sets and multivariate normal distributions. *Mathematical Methods of Operations Research*, 71(3):535–549, 2010.

[75] W. van Ackooij, R. Henrion, A. Möller, and R. Zorgati. On joint probabilistic constriants with Gaussian Coefficient Matrix. *Operations Research Letters*, 39:99–102, 2011.

[76] W. van Ackooij, R. Henrion, A. Möller, and R. Zorgati. Joint chance constrained programming for hydro reservoir management. *Optimization and Engineering*, 15:509–531, 2014.

[77] W. van Ackooij and C. Sagastizábal. Constrained bundle methods for upper inexact oracles with application to joint chance constrained energy problems. *Siam Journal on Optimization*, 24(2):733–765, 2014.

[78] C. van de Panne and W. Popp. Minimum-cost cattle feed under probabilistic protein constraints. *Managment Science*, 9:405–430, 1963.

[79] A.F. Veinott. The supporting hyperplane method for unimodal programming. *Operations Research*, 15:147–152, 1967.

[80] B. Vízvári. The integer programming background of a stochastic integer programming algorithm of Dentcheva-Prékopa-Ruszczyński. *Optimization Methods and Software*, 17:543–559, 2002.

[81] G. Zoutendijk. *Methods of Feasible Directions: A Study in Linear and Non-Linear Programming*. Elsevier Publishing Co., Amsterdam, 1960.

Approximation of the Whole Pareto Optimal Set for the Vector Optimization Problem

In Memoriam András Prékopa (1929-2016)

Tibor Illés[1], **Gábor Lovics**[2]

[1] Optimization Research Group, Department of Differential Equations, Budapest University of Technology and Economics, Egry József u. 1., 1111 Budapest, Hungary,
illes@math.bme.hu
[2] Hungarian Central Statistical Office, Keleti Károly u. 5-7., 1024 Budapest, Hungary,
Gabor.Lovics@ksh.hu

In multi-objective optimization problems several objective functions have to be minimized simultaneously. In this work, we present a new computational method for the linearly constrained, convex multi-objective optimization problem. We propose some techniques to find joint decreasing directions for both the unconstrained and the linearly constrained case as well. Based on these results, we introduce a method using a subdivision technique to approximate the whole Pareto optimal set of the linearly constrained, convex multi-objective optimization problem. Finally, we illustrate our algorithm by solving the Markowitz model on real data.

Keywords: multi-objective optimization; joint decreasing direction; approximation of Pareto optimal set; Markowitz model
2000 MSC: 90C29

1 Introduction

In the literature of economics and finance the measure of risk has always been a very interesting topic, and nowadays it may be even more important than ever. One of the first idea to be taken into consideration is the risk in financial activities coming from Markowitz [26], who developed his famous model, where the investors make portfolios from different securities, and try to maximize their profit and minimize their risk at the same time. In this model, the profit was linear and the risk was defined as the variance of the securities. The Markowitz model can be formulated as a linearly constrained optimization problem with two objective (linear profit and quadratic risk) functions.

In the general case, the least risky portfolio is not the most profitable one; thus we could not optimize the two objectives at the same time. Therefore, we need to find portfolios, where one of the goals cannot be improved without worsening the other. This kind of solutions are called *Pareto optimal* or *Pareto efficient* solutions discussed by Pareto [30].

Single Pareto optimal solution of the Markowitz model can be computed by scalarization methods, see Luc [23] and Miettinen [28], where the weighted sum of the objective functions, serving as a new objective function, defines a single quadratic objective function. Therefore, after the scalarization of the Markowitz model, the optimization problem simplifies to a quadratic optimization problem over linear constraints [25]. The simplified problem's optimal solution is a single Pareto optimal solution of the original problem. The effect of the weights of the objective functions determine the computed Pareto efficient solution of the original problem. The weights might have unpredictable effects on the computed Pareto efficient solution in general. Weakness of this approach is that it restricts the Pareto efficient solution set to a single element and it's local neighborhood. In this way, we lose some information, like how much extra profit can be gained by accepting a larger risk. Finding - or at least approximating - the whole Pareto efficient solution set of the original, multi-objective problem, may lead to a better understanding of the modeled practical problem [27].

For some unconstrained multi-objective optimization problems there are research papers [7, 8, 13, 34] discussing algorithms applicable for approximating the Pareto efficient solution set. However, many multi-objective optimization problems - naturally - have constraints [13, 14]. A simple example for a constrained multi-objective optimization problem is the earlier mentioned Markowitz model. In this paper, we extend and generalize the algorithm of Dellnitz et al. [7] for approximating the Pareto efficient set of a linearly constrained convex multi-objective problem. Fliege and Svaiter [13] obtained some theoretical results, that are similar to our approach for finding joint decreasing directions.

In the next section, most important definitions and results of vector optimization problems are summarized. In the third section, we discuss some results about the unconstrained vector optimization problem. The method called subdivision technique introduced by Dellnitz et al. [7, 8] was developed to approximate the Pareto efficient solution set of an unconstrained vector optimization problems. The subdivision method uses some results described in [34]. An important ingredient of all methods, that can approximate the Pareto optimal set of a convex vector optimization problem, is the computation of a joint decreasing direction for all objective functions. We show that - using results from linear optimization - a joint decreasing direction for an unconstrained vector optimization problem can be computed.

In the fourth section, the computation of a feasible joint decreasing direction for linearly constrained convex vector optimization problem is discussed. The set of feasible joint decreasing directions forms a finitely generated cone and can be computed, as shown in Section 4. Interesting optimality conditions of Eichfelder and Ha [10] for multi-objective optimization problems show some similarities to those that are used in this paper during the computations of joint decreasing directions; however their results do not have practical, algorithmic applications yet.

Section 5 contains an algorithm, which is a generalization of the subdivision method for the linearly constrained convex vector optimization problem. In Section 6, we show some numerical results obtained on a real data set (securities from Budapest Stock Exchange) for the Markowitz model.

Comparing our method presented in Section 5, with the subdivision algorithm of Dellnitz et al. [7, 8], clearly, our method works for unconstrained vector optimization problems (UVOP), like that of Dellnitz et al. [7, 8], but we generate different joint decreasing directions. Furthermore, our method is applicable to vector optimization problems (VOP) with convex objective functions and linear constraints, keeping all advantageous properties of subdivision algorithm of Dellnitz et al. [7,8].

For the (VOP) there are some sophisticated scalarization methods reported in [15]. This method, as scalarization methods in general, finds a single Pareto optimal solution. The scalarization method introduced by [15] defines a weighted optimization problem (WOP) of (VOP) with fixed weights and a feasible solution set, that depends on the current feasible vector $\mathbf{x}$. Thus, in each iteration of the algorithm the actual feasible solution set is restricted to such a subset of the original feasible solution set, where some Pareto optimal solutions are located.

Subdivision techniques - including ours - find a cover set of the Pareto optimal solution set, formed by boxes used in the subdivision procedure (see Sections 5 and 6). The approximation of the whole Pareto optimal solution set is controlled by the diameter of the covering boxes computed in the subdivision procedure of the algorithm.

Although both the scalarization algorithm of Gianessi et al. [15] and our subdivision algorithm have some similarities (i.e., in each iteration the feasible solution set decreases), there are significant differences as well. The scalarization algorithm finds a single Pareto optimal solution under some quite general assumptions, while the subdivision algorithm approximates the whole Pareto optimal solution set for the (VOP) with convex objective functions and linear constraints.

We use the following notations throughout the paper: scalars and indices are denoted by lowercase Latin letters, column vectors by lowercase boldface Latin letters, matrices by capital Latin letters, and finally sets by capital calligraphic letters.

The vector, with all 1 elements is denoted by $\mathbf{e}$, i.e.

$\mathbf{e}^T := (1, 1, \ldots, 1) \in \mathbb{R}^n$,

for some $n \in \mathbb{N}$, where ^{T} stands for the transpose of a (column) vector (or a matrix). Vector $\mathbf{e}_i \in \mathbb{R}^n$ is the ith unit vector of the n dimensional Euclidean space.

2 Basic Definitions and Results in Vector Optimization

In this section, we discuss some notations, define the vector (or multi-objective) optimization problem and the concept of Pareto optimal solutions. Furthermore, we state two well known results of vector optimization, which are important for our approach.

We define the *unit simplex set*, as,

Definition 1. *Let $\mathscr{S}_k$ denote the unit simplex in the k dimensional vector space, and define it as follows:*

$$\mathscr{S}_k := \{\mathbf{w} \in \mathbb{R}^k : \ \mathbf{e}^T\mathbf{w} = 1, \ \ \mathbf{w} \geq \mathbf{0}\}.$$

Let $\mathscr{F} \subseteq \mathbb{R}^n$ be a set and $F : \mathscr{F} \to \mathbb{R}^k$ is a function defined as

$$F(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})]^T,$$

where $f_i : \mathscr{F} \to \mathbb{R}$ is a coordinate function for all i.

The *general vector optimization problem* $(GVOP)$ can be formulated as

$$(GVOP) \qquad\qquad \text{MIN}\, F(\mathbf{x}), \ \text{subject to} \ \mathbf{x} \in \mathscr{F}.$$

If the set $\mathscr{F}$ and the function F are convex, then $(GVOP)$ is a *convex vector optimization problem*. We assume that F is differentiable.

Usually, different objective functions of $(GVOP)$ describe conflicting goals, therefore such $\mathbf{x} \in \mathscr{F}$, that minimize all objective functions at the same time, is unlikely to exist. For this reason, the following definitions naturally extend the concept of an optimal solution for $(GVOP)$ settings.

Definition 2. *Let $(GVOP)$ be given. We say that $\mathbf{x}^* \in \mathscr{F}$ is a*

1. weakly Pareto optimal solution *of problem $(GVOP)$ if there does not exist a feasible solution $\mathbf{x} \in \mathscr{F}$ which satisfies the vector inequality $F(\mathbf{x}) < F(\mathbf{x}^*)$;*
2. Pareto optimal solution *if does not exist feasible solution $\mathbf{x} \in \mathscr{F}$ which satisfies the vector inequality $F(\mathbf{x}) \leq F(\mathbf{x}^*)$ and $F(\mathbf{x}) \neq F(\mathbf{x}^*)$.*

Furthermore, we call the set $\mathscr{F}^* \subseteq \mathscr{F}$ a weakly Pareto optimal set *if every $x^* \in \mathscr{F}^*$ is a weakly Pareto optimal solution of the $(GVOP)$.*

Our goal is to approximate the whole Pareto optimal or weakly Pareto optimal solution sets for different vector optimization problems. During the approximation procedure of the whole Pareto optimal solution set, we compute many Pareto optimal solutions and produce an outer approximation of the whole Pareto optimal solution set.

The literature contains several methods that, find one of the Pareto optimal solutions, see [9, 23, 28], but sometimes it is interesting to compute all of them, or at least as much as we can.

One of the frequently used method to compute a Pareto optimal solution uses a weighted sum of the objective functions to obtain a single objective optimization problem. Let $\mathbf{w} \in \mathscr{S}_k$ be a given vector of weights. From a vector optimization problem, using a vector of weights, we can define the weighted optimization problem as follows

$$(WOP) \qquad \min \mathbf{w}^T F(\mathbf{x}), \text{ subject to } \mathbf{x} \in \mathscr{F}.$$

We state without proof two well-known theorems, that describe the relationship between $(GVOP)$ and (WOP). The first theorem shows that the (WOP) can be used to find a Pareto optimal solution, see for instance [9,23,28].

Theorem 1. *Let a $(GVOP)$ and the corresponding (WOP) for a $\mathbf{w} \in \mathscr{S}_k$ be given. Assume that $\mathbf{x}^* \in \mathscr{F}$ is an optimal solution of the (WOP) problem; then $\mathbf{x}^*$ is a weak Pareto optimal solution for the $(GVOP)$.*

Next theorem needs a bit more complicated reasoning, but for the convex case each Pareto optimal solution of the $(GVOP)$ can be found through a (WOP) using the proper weights [9,23,28].

Theorem 2. *Let $(GVOP)$ be a convex vector optimization problem, and assume that $\mathbf{x}^* \in \mathscr{F}$ is a Pareto optimal solution of the $(GVOP)$; then there is a $\mathbf{w} \in \mathscr{S}_k$ weight vector, and a (WOP) problem, for which $\mathbf{x}^*$ is an optimal solution.*

The method, that will be described in section 5, decreases every coordinate function of F at the same time and always moves from a feasible solution to another feasible solution; hence we introduce the following useful definition.

Definition 3. *Let problem $(GVOP)$ and feasible point $\mathbf{x} \in \mathscr{F}$ be given. Vector $\mathbf{v} \in \mathbb{R}^n$, $\mathbf{v} \neq \mathbf{0}$ is called a*

1. joint decreasing direction *at point $\mathbf{x}$ iff there exists $h_0 > 0$ for every $h \in]0, h_0]$ satisfying that*

   $$F(\mathbf{x} + h\mathbf{v}) < F(\mathbf{x});$$

2. feasible joint decreasing direction *iff it is a joint decreasing direction and there exists $h_1 > 0$ such that, for every $h \in]0, h_1]$ we have $\mathbf{x} + h\mathbf{v} \in \mathscr{F}$.*

Example. Let the following unconstrained vector optimization problem

$$(GVOP_1) \qquad \text{MIN } F(x_1, x_2) = \begin{pmatrix} f_1(x_1, x_2) = x_1^2 + x_2^2 \\ f_2(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^2 \end{pmatrix},$$

be given alongside a point $\mathbf{x}^T = (x_1, x_2) = (0, 1)$ and direction $\mathbf{v}^T = (1, -1)$. Now we show that $\mathbf{v}$ is a joint decreasing direction for the objective function F at point $\mathbf{x}$.

It is easy to show that

$$f_1(\mathbf{x} + h\mathbf{v}) = f_2(\mathbf{x} + h\mathbf{v}) = h^2 + (1 - h)^2 = 2\left(h - \frac{1}{2}\right)^2 + \frac{1}{2}$$

From the last form of the coordinate functions, it is easy to see that the coordinate functions are decreasing on the $[0; \frac{1}{2}]$ interval; therefore $\mathbf{v}$ is a joint decreasing direction with $h_0 = \frac{1}{2}$.

If we add a single constraint to our example, then we obtain a new problem

$$(GVOP_2) \qquad \text{MIN}\, F(x_1,x_2) = \left(\begin{array}{l} f_1(x_1,x_2) = x_1^2 + x_2^2 \\ f_2(x_1,x_2) = (x_1-1)^2 + (x_2-1)^2 \end{array} \right),$$
$$\text{subject to } x_1 \leq \frac{1}{3}.$$

It is easy to see that **v** is a feasible joint decreasing direction for problem $(GVOP_2)$ too, with $h_1 = \frac{1}{3}$.

From now on, let us consider $(GVOP)$ with a convex, differentiable objective function F and let us denote the Jacobian-matrix of F at point **x** by $J(\mathbf{x})$. Then, $\mathbf{v} \in \mathbb{R}^n$ is a joint decreasing direction of function F at point **x**, if and only if

$$[J(\mathbf{x})]\mathbf{v} < 0, \qquad (1)$$

as **v** is a decreasing direction for the ith coordinate function f_i at point **x**, if and only if $[\nabla f_i(\mathbf{x})]^T \mathbf{v} < 0$.

3 Results for Unconstrained Vector Optimization

In this section, we review some results of unconstrained vector optimization, namely for $\mathscr{F} = \mathbb{R}^n$. We assume that F is a differentiable function. The unconstrained vector optimization problem is denoted by $(UVOP)$.

Before we show how a joint decreasing direction can be computed, we need a criterion to decide wether an **x** is a Pareto optimal solution or not, see Schäffler et al. [34].

Definition 4. *Let $J(\mathbf{x}) \in \mathbb{R}^{k\times n}$ be the Jacobian matrix of a differentiable function $F : \mathbb{R}^n \to \mathbb{R}^k$ at a point $\mathbf{x} \in \mathbb{R}^n$. An $\mathbf{x}^*$ is called* substationary point *of F iff there exist a $\mathbf{w} \in \mathscr{S}_k$, which fulfills the following equation:*

$$\mathbf{w}^T[J(\mathbf{x}^*)] = \mathbf{0}.$$

In the unconstrained case, point $\mathbf{x}^*$ is a substationary point of the objective function F of (GVOP), if it is a stationery point of the weighted objective function of (WOP). From Theorem 1 and Theorem 2 we can see that in convex case, substationary points are weak Pareto optimal solutions of the unconstrained vector optimization problem (GVOP).

We are ready to discuss two models to find joint decreasing directions. The first model has been discussed in [34] and uses a quadratic programming problem formulation to compute joint decreasing directions. Later we show that a joint decreasing direction can be computed in a simpler way by using a special linear programming problem.

Let us define the following quadratic programming problem $(QOP(\mathbf{x}))$ for any $\mathbf{x} \in \mathbb{R}^n$, with variable $\mathbf{w}$

$$(QOP(\mathbf{x})) \qquad \min \mathbf{w}^T \left(J(\mathbf{x})\,[J(\mathbf{x})]^T \right) \mathbf{w}, \text{ subject to } \mathbf{w} \in \mathscr{S}_k.$$

From the well known Weierstarss Theorem it follows that this problem always has an optimal solution, since the feasible set is compact and the function

$$g : \mathscr{S}_k \to \mathbb{R}, \quad g(\mathbf{w}) = \mathbf{w}^T \left(J(\mathbf{x})\,[J(\mathbf{x})]^T \right) \mathbf{w}$$

is a convex, quadratic, continuous function for any given $\mathbf{x} \in \mathbb{R}^n$.

Next theorem is an already known statement [34, Theorem 2.1], for which we give a new and shorter proof. This shows that using the $(QOP(\mathbf{x}))$ problem we can find a joint decreasing direction of F or a certificate that $\mathbf{x}$ is a Pareto optimal solution of problem $(UVOP)$.

Theorem 3. *Let a problem $(UVOP)$, a point $\mathbf{x} \in \mathbb{R}^n$ and the associated $(QOP(\mathbf{x}))$ be given. Let $\mathbf{w}^* \in \mathbb{R}^k$ denote the optimal solution of $(QOP(\mathbf{x}))$. We define vector $\mathbf{q} \in \mathbb{R}^n$ as $\mathbf{q} = [J(\mathbf{x})]^T \mathbf{w}^*$. If $\mathbf{q} = \mathbf{0}$, then $\mathbf{x}$ is a substationary point, otherwise $-\mathbf{q}$ is a joint decreasing direction for F at point $\mathbf{x}$.*

Proof. When $\mathbf{q} = \mathbf{0}$ then Definition 4 shows that $\mathbf{x}$ is substationary point. When $\mathbf{q} \neq \mathbf{0}$, we indirectly assume that $-\mathbf{q}$ is not a decreasing direction for the ith coordinate function, f_i of F and $[\nabla f_i(\mathbf{x})]^T \mathbf{q} \neq \mathbf{0}$. It means that $[\nabla f_i(\mathbf{x})]^T \mathbf{q} < 0$. Since $[\nabla f_i(\mathbf{x})]^T = \mathbf{e}_i^T J(\mathbf{x})$, so our indirect assumption means

$$[\nabla f_i(\mathbf{x})]^T \mathbf{q} = \mathbf{e}_i^T [J(\mathbf{x})][J(\mathbf{x})]^T \mathbf{w}^* < 0.$$

We show that $\mathbf{e}_i - \mathbf{w}^* \neq \mathbf{0}$ is a feasible decreasing direction of $g(\mathbf{w}^*)$ which contradicts the optimality of $\mathbf{w}^*$. The $\mathbf{e}_i = \mathbf{w}^*$ can not be fulfilled because it contradicts the indirect assumption, and it is easy to see, that $\mathbf{e}_i$ is a feasible solution of $(QOP(\mathbf{x}))$ so $\mathbf{e}_i - \mathbf{w}^*$ is a feasible direction at point $\mathbf{w}^*$.
Since

$$\nabla g(\mathbf{w}) = 2[J(\mathbf{x})][J(\mathbf{x})]^T \mathbf{w}$$

thus

$$\begin{aligned}[\nabla g(\mathbf{w}^*)]^T (\mathbf{e}_i - \mathbf{w}^*) &= 2\mathbf{w}^{*T} [J(\mathbf{x})][J(\mathbf{x})]^T (\mathbf{e}_i - \mathbf{w}^*) \\ &= 2\mathbf{w}^{*T} [J(\mathbf{x})][J(\mathbf{x})]^T \mathbf{e}_i - 2\mathbf{w}^{*T} [J(\mathbf{x})][J(\mathbf{x})]^T \mathbf{w}^* \\ &= 2\mathbf{q}^T [\nabla f_i(\mathbf{x})] - 2\mathbf{w}^{*T} [J(\mathbf{x})][J(\mathbf{x})]^T \mathbf{w}^* < 0,\end{aligned}$$

where the first term of the sum is negative because of the indirect assumption, and the second term is not positive, because $[J(\mathbf{x})][J(\mathbf{x})]^T$ is a positive semidefinite matrix. □

The previous result underline the importance of solving $(QOP(\mathbf{x}))$ problem efficiently. For solving smaller size linearly constrained convex quadratic problems pivot algorithms [1, 4–6, 16, 22] can be used. In case of larger size linearly constrained, convex quadratic problems, interior point algorithms can be used to solve the problem (see for instance [18, 20]).

Theorem 3 shows that a joint decreasing direction can be computed as the convex combination of the gradient vectors of coordinate functions of F. Following the ideas discussed above, we can formulate a linear programming problem such that any optimal solution of the linear program defines a joint decreasing direction of problem $(UVOP)$. Some similar results can be found in [13].

Let us define the linear optimization problem $(LP(\mathbf{x}))$ in the following way:

$$(LP(\mathbf{x})) \qquad \max\ q_0, \quad \text{subject to} \quad [J(\mathbf{x})]\mathbf{q} + q_0\mathbf{e} \le \mathbf{0},\ \ 0 \le q_0 \le 1,$$

where $\mathbf{q} \in \mathbb{R}^n$ and $q_0 \in \mathbb{R}$ are the decision variables of the problem $LP(\mathbf{x})$. Now we are ready to state and prove a theorem that discusses a connection between $(UVOP)$ and $(LP(\mathbf{x}))$.

Theorem 4. *Let a point $\mathbf{x} \in \mathbb{R}^n$, an $(UVOP)$ and an associated $(LP(\mathbf{x}))$ be given. Then the $(LP(\mathbf{x}))$ always has an optimal solution $(\mathbf{q}^*, q_0^*)$. There are two cases for the optimal value of the $(LP(\mathbf{x}))$, either $q_0 = 0$ thus $\mathbf{x}$ is a substationary point of the $(UVOP)$, or $q_0 = 1$ thus $\mathbf{q}^*$ is a joint decreasing direction for the function F at point* $\mathbf{x}$.

Proof. It is easy to see that $\mathbf{q} = \mathbf{0}$, $q_0 = 0$ is a feasible solution of problem $(LP(\mathbf{x}))$ and 1 is an upper bound of the objective function, which means $(LP(\mathbf{x}))$ should have an optimal solution.
Let us examine the case

$$[J(\mathbf{x})]\mathbf{q} + q_0\mathbf{e} \le \mathbf{0}, \quad q_0 > 0. \tag{2}$$

If system (2) has a solution, than $\left(\frac{1}{q_0}\mathbf{q}, 1\right)$ is a solution of the system, so the optimal value of the objective function is 1. This mean that

$$[J(\mathbf{x})]\mathbf{q} \le -\mathbf{e}$$

so the $\mathbf{q}$ is a joint decreasing direction of function F.
If the system (2) has no solution then the optimal value of the objective function is 0, and from the Farkas lemma ([11, 12, 17, 22, 29, 31, 32, 35]) we know that there exists a $\mathbf{w}$ which satisfies the following:

$$\mathbf{w}^T[J(\mathbf{x})] = \mathbf{0}, \quad \mathbf{e}^T\mathbf{w} = 1, \quad \mathbf{w} \ge \mathbf{0}. \tag{3}$$

It means that if the optimal value of the problem $(LP(\mathbf{x}))$ is 0, than $\mathbf{x}$ is a substationary point. □

A linear programming problem $(LP(\mathbf{x}))$ (and later on $(LPS(\mathbf{x}))$) can be solved by either pivot or interior point algorithms, see [21]. In case of applying pivot methods to solve linear programming problem, simplex algorithm is a natural choice, see [24, 29, 35]. A recent study on anti-cycling pivot rules for linear programming problem contains a numerical study on different pivot algorithms, see [6]. Sometimes, if the problem is well structured and small, criss-cross algorithm of T. Terlaky can be used for solving the linear programming problem as well, see [17, 36]. More about interior point algorithms for linear programming problems can be learnt from [19, 24, 33].

In this section two techniques were introduced to decide whether a point $\mathbf{x}$ is a weak Pareto optimal solution of problem $(UVOP)$ or to find a joint decreasing direction. Before we generalize this result to the linear constrained case let us compare this technique with some known procedures. The classical scalarization technique based on (WOP) finds a Pareto optimal solution $\mathbf{x}^*$ of $(UVOP)$. Due to the requirements defined by the concept of the weak Pareto optimal solution, it may happen that in some iterations of the scalarization algorithm such feasible solutions are computed for which some objective function's value increases. In our method this phenomena can not happen, because in each iteration we select a joint decreasing direction.

4 Vector Optimization with Linear Constraints

In this section, we show how we can find a feasible joint decreasing direction for linearly constrained vector optimization problems. First we find a feasible joint decreasing direction for a special problem, where we only have sign constraints on the variables. After that we generalize our results to general linearly constrained vector optimization problems. Our method can be considered as the generalization of the well known reduced gradient method to vector optimization problems. Some similar result can be found in [13], for the feasible direction method of Zountendijk.

First we define the vector optimization problem with sign constraints $(SVOP)$:

$$(SVOP) \qquad MIN\ F(\mathbf{x}), \quad \text{subject to} \quad \mathbf{x} \geq \mathbf{0},$$

where F is a convex function. From Theorem 1 we know that $\mathbf{x}^* \geq \mathbf{0}$ is a Pareto optimal solution if there exists a $\mathbf{w} \in \mathscr{S}_k$ vector such that $\mathbf{x}^*$ is an optimal solution of

$$(SWOP) \qquad \min\ \mathbf{w}^T F(\mathbf{x}), \quad \text{subject to} \quad \mathbf{x} \geq \mathbf{0}.$$

Since Slater regularity and convexity conditions hold, from the KKT theorem [24] we know that $\mathbf{x}^* \geq \mathbf{0}$ is an optimal solution of $(SWOP)$ iff it satisfies the following system:

$$\mathbf{w}^T[J(\mathbf{x}^*)] \geq \mathbf{0}, \quad \mathbf{w}^T[J(\mathbf{x}^*)]\mathbf{x}^* = 0. \tag{4}$$

Let the vector $\mathbf{x} \geq \mathbf{0}$ be given and we would like to decide wether it is an optimal solution of the (SWOP) problem or not. Let us define the index sets

$$I_+ = \{i : x_i > 0\}, \qquad \text{and} \qquad I_0 = \{i : x_i = 0\},$$

that depend on the selected vector $\mathbf{x}$. Using the index sets I_0, I_+, we partition the column vectors of matrix $J(\mathbf{x})$ into two parts. The two parts are denoted by $J(\mathbf{x})_{I_0}$ and $J(\mathbf{x})_{I_+}$. Taking into consideration the partition, the KKT conditions can be written in an equivalent form as

$$\mathbf{w}^T[J(\mathbf{x})]_{I_+} = \mathbf{0}, \quad \mathbf{w}^T[J(\mathbf{x})]_{I_0} \geq \mathbf{0}, \quad \mathbf{w} \in \mathscr{S}_k. \tag{5}$$

The inequality system (5) plays the same role for $(SVOP)$ as (3) for $(UVOP)$, namely $\mathbf{x}$ is a Pareto-optimal solution if (5) has a solution.

Now we can define a linear programming problem corresponding to $(SVOP)$ such that an optimal solution of the linear programming problem either defines a joint decreasing direction or gives a certificate that the solution $\mathbf{x}$ is a Pareto optimal solution of $(SVOP)$.

$$\begin{array}{lrl} (LPS(\mathbf{x})) & & \max\ z, \\ & \text{subject to} & [J(\mathbf{x})]_{I_+}\mathbf{u} + [J(\mathbf{x})]_{I_0}\mathbf{v} + z\mathbf{e} \leq \mathbf{0}, \\ & & \mathbf{v} \geq \mathbf{0}, \quad 0 \leq z \leq 1, \end{array}$$

where $\mathbf{u}, \mathbf{v}$ and z are the decision variables of problem $(LPS(\mathbf{x}))$. Now we are ready to prove the following theorem.

Theorem 5. *Let a $(SVOP)$ and an associated $(LPS(\mathbf{x}))$ be given, where $\mathbf{x} \in \mathscr{F}$ is a feasible point. The problem $(LPS(\mathbf{x}))$ always has an optimal solution $(\mathbf{u}^*, \mathbf{v}^*, z^*)$. There are two cases for the optimal value of problem $(LPS(\mathbf{x}))$, $z^* = 0$ which means that $\mathbf{x}$ is a Pareto optimal solution of the $(SVOP)$, or $z^* = 1$ which means that $\mathbf{q}^T = (\mathbf{u}^*, \mathbf{v}^*)$ is a feasible joint decreasing direction of function F.*

Proof. It is easy to see that $\mathbf{u} = \mathbf{0}$, $\mathbf{v} = \mathbf{0}$, $z = 0$ is a feasible solution of the problem $(LPS(\mathbf{x}))$ and 1 is an upper bound of the objective function, therefore $(LPS(\mathbf{x}))$ has an optimal solution.
Let us examine the following system

$$[J(\mathbf{x})]_{I_+}\mathbf{u} + [J(\mathbf{x})]_{I_0}\mathbf{v} + z\mathbf{e} \leq \mathbf{0}, \quad \mathbf{v} \geq \mathbf{0}, \quad z > 0. \tag{6}$$

If system (6) has a solution, then $\left(\frac{1}{z}\mathbf{u}, \frac{1}{z}\mathbf{v}, 1\right)$ is an optimal solution of the problem $(LPS(\mathbf{x}))$ with optimal value 1. Thus the vector $\mathbf{q}^T = (\mathbf{u}, \mathbf{v})$ satisfies

$$[J(\mathbf{x})]\mathbf{q} \leq -\mathbf{e} < \mathbf{0},$$

so the $\mathbf{q}$ is a feasible joint decreasing direction for function F at $\mathbf{x} \in \mathscr{F}$. Vector $\mathbf{q}$ is feasible because $\mathbf{q}_{I_0} = \mathbf{v} \geq \mathbf{0}$.
If the system (6) has no solution then the optimal value of the objective function is 0, and from a variant of the Farkas lemma, see [11, 12, 17, 22, 31, 32, 35] we know that there exists a $\mathbf{w}$ which satisfies the following system of inequalities:

$$[J(\mathbf{x})]_{I_+}^T\mathbf{w} = \mathbf{0}, \quad [J(\mathbf{x})]_{I_0}^T\mathbf{w} \geq \mathbf{0}, \quad \mathbf{e}^T\mathbf{w} = 1, \quad \mathbf{w} \geq \mathbf{0}. \tag{7}$$

It means that if the optimal value of problem $(LPS(\mathbf{x}))$ is 0, then point $\mathbf{x}$ is a Pareto optimal solution of (SVOP). □

We are ready to find feasible joint decreasing direction to a linearly constrained vector optimization problem at a feasible solution $\tilde{\mathbf{x}}$. Let the matrix $A \in \mathbb{R}^{m\times n}$ and vector $\mathbf{b} \in \mathbb{R}^m$ be given. Without loss of generality we may assume that $rank(A) = m$. Furthermore, let us assume the following non degeneracy assumption (for details see [2]): any m columns of A are linearly independent and every basic solution is non degenerate. We have a vector optimization problem with linear constraint in the following form

$$(LVOP) \qquad \text{MIN } F(\mathbf{x}), \quad \text{subject to} \quad A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}.$$

Like in the reduced gradient method, see [2], we can partition the matrix A into two parts $A = [B,N]$, where B is a basic and N the non-basic part of the matrix. Similarly every $\mathbf{v} \in \mathbb{R}^n$ vector can be partitioned as $\mathbf{v} = [\mathbf{v}_B, \mathbf{v}_N]$. We call $\mathbf{v}_B$ basic and $\mathbf{v}_N$ a nonbasic vector. We can chose the matrix B such that the $\tilde{\mathbf{x}}_B > 0$ is fulfilled. While $A\mathbf{x} = \mathbf{b}$ holds, we know that

$$B\mathbf{x}_B + N\mathbf{x}_N = \mathbf{b}, \qquad \text{and} \qquad \mathbf{x}_B = B^{-1}(\mathbf{b} - N\mathbf{x}_N).$$

We can redefine function F in a reduced form as

$$F_N(\mathbf{x}_N) = F(\mathbf{x}_B, \mathbf{x}_N) = F(B^{-1}(\mathbf{b} - N\mathbf{x}_N), \mathbf{x}_N).$$

Let us define using the partition (B,N) and the following sign constraint optimization problem

$$(SVOP_B(\tilde{\mathbf{x}})) \qquad \text{MIN } F_N(\mathbf{x}_N), \quad \text{subject to} \quad \mathbf{x}_N \geq \mathbf{0}.$$

Let $\mathbf{q}_N$ denote a feasible joint decreasing direction for $(SVOP_B(\tilde{\mathbf{x}}))$ at point $\tilde{\mathbf{x}}_N$, which can be found by applying Theorem 5. Let $\mathbf{q}_B = -B^{-1}N\mathbf{q}_N$; then we show that $\mathbf{q} = [\mathbf{q}_B, \mathbf{q}_N]$ is feasible joint decreasing direction for $(LVOP)$ at point $\tilde{\mathbf{x}}$. Let us notice that

$$A(\tilde{\mathbf{x}} + h\mathbf{q}) = A\tilde{\mathbf{x}} + h(B\mathbf{q}_B + N\mathbf{q}_N) = \mathbf{b} + h\left(-B(B^{-1}N\mathbf{q}_N) + N\mathbf{q}_N\right) = \mathbf{b},$$

for every $h \in \mathbb{R}$. So $F_N(\tilde{\mathbf{x}} + h\mathbf{q}_N) = F(\tilde{\mathbf{x}} + h\mathbf{q})$ and while $\mathbf{q}_N$ is a feasible joint decreasing direction with $h_1 > 0$ stepsize for $(SVOP_B(\tilde{\mathbf{x}}))$. We can show that $\mathbf{q}$ is a joint decreasing direction for problem $(LVOP)$ with the same $h_1 > 0$ step size. While $\tilde{\mathbf{x}}_B > 0$, there exists $h_2 > 0$, such that $\tilde{\mathbf{x}}_B + h_2\mathbf{q}_B \geq 0$, therefore $\mathbf{q}$ is a feasible joint decreasing direction for problem $(LVOP)$ with a step-size

$$h_3 = \min(h_1, h_2) > 0. \tag{8}$$

We can compute h_1 and h_2 using a ratio-test, since $\tilde{x}_i + hq_i \geq 0$ for all i is required, therefore

$$h_1 = \min\left\{-\frac{\tilde{x}_i}{q_i} : \quad q_i < 0, \quad i \in \mathscr{N}\right\} > 0, \text{ and}$$

$$h_2 = \min\left\{-\frac{\tilde{x}_i}{q_i} : \quad q_i < 0, \quad i \in \mathscr{B}\right\} > 0.$$

5 The Subdivision Algorithm for Linearly Constrained Vector Optimization Problem

In this section, we show how can we build a subdivision method to approximate the Pareto optimal set of a linearly constrained vector optimization problem. Our method is a generalization of the algorithm discussed in [7], where you can find some results about convergence of the subdivision technique. The original method can not handle linear constraints.

Our algorithm approximates the Pareto optimal solution set $\mathscr{F}^*$, using small boxes that each contains at least one computed Pareto optimal solution. The smaller the sets, the better approximation of the $\mathscr{F}^*$, therefore we define the following measure of sets involved in the approximation of $\mathscr{F}^*$.

Definition 5. *Let $\mathscr{H} \subseteq \mathbb{R}^n$ be given; the* diameter *of $\mathscr{H}$ is defined as*

$$diam(\mathscr{H}) := \sup_{x,y \in \mathscr{H}} ||x-y||.$$

Let $\mathbb{H}$ be a family of sets, which contains a finite number of sets from $\mathbb{R}^n$; then the diameter *of $\mathbb{H}$ is*

$$diam(\mathbb{H}) = \max_{\mathscr{H} \in \mathbb{H}} diam(\mathscr{H}).$$

Let us assume that the feasible set of our problem is nonempty, closed and bounded. In this case Pareto optimal solution set of the problem $(LVOP)$ is a nonempty set, defined as,

$$\mathscr{F}_L^* = \{\mathbf{x} \in \mathscr{F} : \mathbf{x} \text{ is Pareto optimal solution of } (LVOP)\}.$$

Based on our assumption, that $\mathscr{F}$ is bounded set, there exists

$$\mathscr{H}_0 = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{l} \le \mathbf{x} \le \mathbf{u}\},$$

where $\mathbf{l}, \mathbf{u} \in \mathbb{R}^n$ are given vectors and

$$\mathscr{F}_L^* \subseteq \mathscr{F} \subseteq \mathscr{H}_0 \cap \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{b}\}.$$

Our goal is to introduce such a subdivision algorithm for $(LVOP)$ problem, that iteratively defines better and better inner and outer approximation of the Pareto optimal solution set of a problem $(LVOP)$. The inner approximation will be an increasing sequence of sets $\mathscr{F}\mathscr{P}_i$, containing finitely many Pareto optimal solutions produced by the algorithm. The outer approximation will be a family of sets $\mathbb{H}_i$ produced in each iteration of the algorithm, that covers $\mathscr{F}_L^*$ with decreasing diameter, $diam(\mathbb{H}_i)$.

The input of our method are data of the (LVOP) problem, namely matrix $A \in \mathbb{R}^{m \times n}$, vector $\mathbf{b} \in \mathbb{R}^m$, function $F\colon \mathbb{R}^n \to \mathbb{R}^k$, set $\mathscr{H}_0$ and a constant $\varepsilon > 0$.

The output of our algorithm is a family of sets $\mathbb{H}$, such that $diam(\mathbb{H}) < \varepsilon$ and each $\mathscr{H} \in \mathbb{H}$ contains at least one Pareto optimal solution.

The algorithm uses some variables and subroutines, too. The $\mathscr{SP}$, $\mathscr{FP}$ are finite-element sets of points from $\mathbb{R}^n$, $\mathscr{H},\mathscr{G} \subseteq \mathscr{F}$, $\mathbb{H}',\mathbb{K},\mathbb{K}'$ and $\mathbb{A}$ are family of sets like $\mathbb{H}$.

Our algorithm in the first step defines the family of sets $\mathbb{H}$, which contains only the $\mathscr{H}_0$ set. The cycle in step 2 runs while the diameter of $\mathbb{H}$ is not small enough. The algorithm reaches this goal in a finite number of iteration, because as you will see in subroutine Newset($\mathbb{H}$) the diameter of $\mathbb{H}$ converges to zero. Nevertheless we show that after every execution of the cycle the family of sets $\mathbb{H}$ contains sets $\mathscr{H}$ which have Pareto optimal solutions. At the beginning it is trivial, because $\mathbb{H}$ contain the whole feasible set.

Subdivision algorithm for (LVOP)

```
1. $\mathbb{H} = \{\mathscr{H}_0\}$
2. **While** $diam(\mathbb{H}) \geq \varepsilon$ **do**
    (a) $\mathbb{H}'$=Newsets($\mathbb{H}$)
    (b) $\mathscr{SP} = \emptyset$
    (c) **While** $\mathbb{H} \neq \emptyset$ **do**
        i. $\mathscr{H} \in \mathbb{H}$
        ii. $\mathscr{SP} = \mathscr{SP} \cup$ Startpoint($\mathscr{H}$)
        iii. $\mathbb{H} = \mathbb{H} \setminus \{\mathscr{H}\}$
        **End While**
    (d) $\mathscr{FP}$ =Points($\mathscr{SP}$,$A$,$\mathbf{b}$,$F$)
    (e) **While** $\mathbb{H}' \neq \emptyset$ **do**
        i. $\mathscr{H} \in \mathbb{H}'$
        ii. **If** $\mathscr{H} \cap \mathscr{FP} \neq \emptyset$ **then** $\mathbb{H} = \mathbb{H} \cup \{\mathscr{H}\}$ **End If**
        iii. $\mathbb{H}' = \mathbb{H}' \setminus \{\mathscr{H}\}$
        **End While**
    **End While**
3. Output($\mathbb{H}$)
```

In step 2(a) we define a family of sets $\mathbb{H}'$ using the subroutine Newset($\mathbb{H}$). The sets from $\mathbb{H}'$ are smaller than sets form $\mathbb{H}$ and cover the same set. Therefore the result of this subroutine has two important properties:

1. $\cup_{\mathscr{H}\in\mathbb{H}'}(\mathscr{H} \cap \mathscr{F}) = \cup_{\mathscr{H}\in\mathbb{H}}(\mathscr{H} \cap \mathscr{F})$,
2. $diam(\mathbb{H}') = \frac{1}{K} diam(\mathbb{H})$,

where $K > 1$ is a constant.

The steps in cycle 2, from step 2(b), delete the sets from $\mathbb{H}'$ which do not contain any Pareto optimal points. Step 2(b) makes set $\mathscr{S}\mathscr{P}$ empty. The cycle in step 2(c) produces a finite number of random starting points in set $\mathscr{H} \cap \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{b}\}$ for every $\mathscr{H} \in \mathbb{H}$ using subroutine Startpoint($\mathscr{H}$), and puts the generated points into the set *SP*.

The main step of our algorithm 2(d) is the subroutine Points($\mathscr{S}\mathscr{P}, A, \mathbf{b}, F$) that produce a set $\mathscr{F}\mathscr{P}$ which contains Pareto optimal points. This subroutine uses our results from Section 4.

In cycle 2(e) we keep every set from $\mathbb{H}'$ which contains Pareto optimal solutions and add those to $\mathbb{H}$. Finally, we check the length of the diameter of $\mathbb{H}$ and repeat the cycle until the diameter is larger than the accuracy parameter ε.

Subroutine Points($\mathscr{S}\mathscr{P}$, A, $\mathbf{b}$, F)

```
1. While $\mathscr{S}\mathscr{P} \neq \emptyset$ do
    (a) $\mathbf{s} \in \mathscr{S}\mathscr{P}$
    (b) $\mathbf{x} = \mathbf{s}$, $z = 1$
    (c) While $z = 1$ do
        i. $(B, N) = A$
        ii. $(\mathbf{x}_B, \mathbf{x}_N) = \mathbf{x}$
        iii. $(\mathbf{q}, z)$=Solve($LPS(\mathbf{x}_N)$)
        iv. If $z = 1$ then
            A. $h_3$=stepsize($F$, $B$, $N$, $\mathbf{b}$, $\mathbf{x}_N$, $\mathbf{q}$)
            B. $\mathbf{x}_N = \mathbf{x}_N + h_3\mathbf{q}$
            C. $\mathbf{x}_B = \mathbf{b} - B^{-1}N\mathbf{x}_N$
            D. $\mathbf{x} = (\mathbf{x}_B, \mathbf{x}_N)$
           End If
       End While
    (d) $\mathscr{S}\mathscr{P} = \mathscr{S}\mathscr{P} \setminus \{\mathbf{s}\}$
    (e) $\mathscr{F}\mathscr{P} = \mathscr{F}\mathscr{P} \cup \{\mathbf{x}\}$
   End While
2. Output($\mathscr{F}\mathscr{P}$)
```

Subroutine Points uses a version of the reduced gradient method for computing Pareto optimal solutions or joint decreasing directions, as discussed in Section 4.

This subroutine works until from all points in $\mathscr{SP}$ search for a Pareto optimal point has been executed. The cycle 1(c) runs until it finds a Pareto optimal point. As we discussed in section 4 the cycle finishes when $z = 0$. In line 1(c)i the matrix A is partitioned into a basic and a non basic parts, denoted by B and N, respectively. The same partition is made with vector $\mathbf{x}$ according to 1(c)ii, and we choose the basis such that $\mathbf{x}_B > 0$ should be satisfied. The $LCP(\mathbf{x}_N)$ is solved in step 1(c)iii. If the variable $z = 0$ then $\mathbf{x}$ is a Pareto optimal solution and we select a new starting point from $\mathscr{SP}$, unless $\mathscr{SP}$ is empty. Otherwise $\mathbf{q}$ is a feasible joint decreasing direction for the reduced function F_N. In step 1(c)ivB we compute step-size h_3 which was defined in (8), and a new feasible solution $\mathbf{x}$ is computed.

Let us summarize the properties of our subdivision algorithm at the end of this section. In each iteration of the algorithm, a set of Pareto optimal solutions, $\mathscr{FP}_i$, as inner approximation of $\mathscr{F}_L^*$ and a family of box sets $\mathbb{H}_i$, with diameter $diam(\mathbb{H}_i)$, as outer approximation has been produced. The input data for inner and outer approximations of $\mathscr{F}_L^*$ are as follows

$$\mathscr{FP}_0 := \emptyset \qquad \textit{and} \qquad \mathbb{H}_0 := \{\mathscr{H}_0\}.$$

Proposition 1. *Let a problem $(LVOP)$ with nonempty, polytope $\mathscr{F}$ and differentiable, convex objective function F be given. Furthermore, let us assume that $\mathscr{F} \subseteq \mathscr{H}_0$ holds, where $\mathscr{H}_0 \subset \mathbb{R}^n$ is a box set (i.e. generalized interval). Our subdivision algorithm in iteration k produces two outputs:*

a) a subset of Pareto optimal solutions $\mathscr{FP}_k$, and

b) a family of box sets $\mathbb{H}_k$,

with the following properties

1. *$\mathscr{FP}_i \subseteq \mathscr{FP}_{i+1}$ for all $i = 0, 1, \ldots, k-1$,*
2. *$\cup_{\mathscr{H}\in\mathbb{H}_{i+1}}\mathscr{H} \subseteq \cup_{\mathscr{H}\in\mathbb{H}_i}\mathscr{H}$ for all $i = 0, 1, \ldots, k-1$,*
3. *$diam(\mathbb{H}_{i+1}) = \frac{1}{K}diam(\mathbb{H}_i)$, where $K > 1$ is a constant,*
4. *$\mathscr{FP}_i \subset \mathscr{F}_L^*$, for all $i = 0, 1, \ldots, k$,*
5. *$\mathscr{F}_L^* \subset \cup_{\mathscr{H}\in\mathbb{H}_i}\mathscr{H}$, for all $i = 0, 1, \ldots, k$.*

Dellnitz et al. [7] discussed two important issues related to subdivision methods: (i) convergence (see Section 3), and (ii) possibility of deleting box that contains Pareto optimal solution (see paragraph 4.2).

Convergence of subdivision methods according to Dellnitz and his coauthors follows under mild smoothness assumptions of the objective functions and compactness of their domains together with some useful properties of the iteration scheme. All these necessary properties of the objective functions and iteration scheme are satisfied in our case, too. Convergence of our subdivision method is based on similar arguments as in case discussed by Dellnitz et al. [7].

It may occur that a box containing Pareto optimal solution is deleted during the subdivision algorithm, as stated in [7]. This phenomenon is related to the discretization induced by the iteration scheme. Decision whether keep or delete a box during the course of the algorithm depends on whether we found Pareto optimal solution in that box or not. From each box, finite number of points are selected and tested whether those are Pareto optimal solutions or joint decreasing direction corresponds to them. From those test points that are not Pareto optimal solutions, using joint decreasing direction an iterative process is started that stops with founding a Pareto optimal solution. If all Pareto optimal solutions computed in this way lay out of the box, we may conclude that the box under consideration does not contain Pareto optimal solution. However, this conclusion is based only on finitely many test points thus there is a chance to delete a box even if it contains Pareto optimal solution. This situation, in practice, can be handled by applying different strategies. All these strategies decrease the opportunity of deleting a box containing Pareto optimal solution.

In [7] discussed a recovering algorithm that could be used after the diameter of boxes reached the prescribed $\varepsilon > 0$ accuracy. Their recovering algorithm finds all those boxes with the current diameter that are necessary to ensure that a cover set of the Pareto optimal solutions is obtained. For details, see [7], recovering algorithm (paragraph 4.2).

6 The Markowitz Model and Computational Results

Let us illustrate our method by solving the Markowitz model to find the most profitable and less risky portfolios. The standard way of solving the model is to find one of the Pareto optimal solution with an associated (WOP) see [26, 28]. The question is whether such single Pareto optimal solution is what we really need for decision making. Naturally, if we would like to make extra profit, we should accept larger risk. Therefore, a single Pareto optimal solution does not contain enough information for making a practical decision. If we produce or approximate the Pareto optimal solution set then we can make use of the additional information for making more established decision.

The analytical description of the whole Pareto optimal set for the Markowitz model is known [37]. Thus as a test problem, the Markowitz model has the following advantage: it is possible to derive its Pareto optimal solution set in an analytical way [37], therefore the result of our subdivision algorithm can be compared with the analytical description of the Pareto optimal solution set.

We are now ready to formulate the original Markowitz model. Our goal is to make a selection from n different securities. Let x_i denote how much percentage we spend from our budget on security i $(i = 1, 2, \dots, n)$, based on more approximate information than a single Pareto optimal solution. Therefore, our decision space is the n-dimensional unit simplex, $\mathscr{S}_n$.

Let $\mathbf{a} \in \mathbb{R}^n$ denote the expected return of the securities, while $C \in \mathbb{R}^{n \times n}$ denotes the covariant matrix of the securities return. It is known that the expected return of our portfolio is equal to $\mathbf{a}^T\mathbf{x}$. One of our goal is to maximize the expected return.

It is much harder to measure the risk of the portfolio, but in this model it is equal to the variance of the securities return, namely $\mathbf{x}^T C\mathbf{x}$. Our second goal is to minimize this value. Now we are ready to formulate our model

$$(MM) \qquad \text{MIN} \begin{pmatrix} -\mathbf{a}^T\mathbf{x} \\ \mathbf{x}^T C\mathbf{x} \end{pmatrix}, \quad \text{subject to} \quad \mathbf{x} \in \mathscr{S}_n.$$

For computational purposes we used data from the spot market [3] and daily prices of A category shares has been collected for a one year period from 01. 09. 2010. to 01. 09. 2011. Let $P_{i,d}$ denote the daily price of the i-th share on date d, then the i-th coordinate of the vector $\mathbf{a}$ is equal to $(P_{i,01.09.2011.} - P_{i,01.09.2010.})/P_{i,01.09.2010.}$. Thus we only work with the relative returns from the price change and do not deal with shares dividend. We compute the daily return of the shares for every day (d) from 01. 09. 2010. to 31. 08. 2011. as $(P_{i,d} - P_{i,d+1})/P_{i,d}$, and C is the covariant matrix of this daily return. To illustrate our method we use three shares ($i =$ MOL, MTELEKOM, OTP) that are usually selected into portfolios because these shares correspond to large and stable Hungarian companies. We used the following data:

$$\mathbf{a} = \begin{pmatrix} -0,1906 \\ -0,2556 \\ -0,1665 \end{pmatrix}$$

$$C = 10^{-5} \begin{pmatrix} 27,1024 & 7,5655 & 17,1768 \\ 7,5655 & 16,4816 & 8,1816 \\ 17,1768 & 8,1816 & 34,2139 \end{pmatrix}$$

The input data for the *Subdivision algorithm for (LVOP)* are: matrix $A = \mathbf{e} \in \mathbb{R}^3$, $\mathbf{b} = 1$ since we have a single constraint in our model, and the objective function $F(\mathbf{x}) = \begin{pmatrix} -\mathbf{a}^T\mathbf{x} \\ \mathbf{x}^T C\mathbf{x} \end{pmatrix}$. Let $\mathscr{H}_0 = \{\mathbf{0} \leq \mathbf{x} \leq \mathbf{e}\}$, $\varepsilon = \frac{1}{2^6}$ and $K = 2$.

At the beginning of the algorithm in step 1 the family of set $\mathbb{H}$ has been defined (see Figure 1).

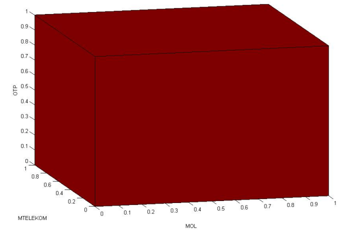


Figure 1

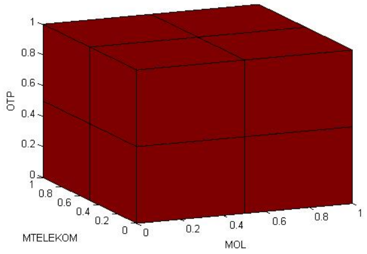


Figure 2

At the first iteration the procedure *Newsets* in step $2(a)$, defines $\mathbb{H}'$ in the following way. First, it cuts the set $\mathscr{H}_0$ into eight equal pieces as you see in Figure 2. After that

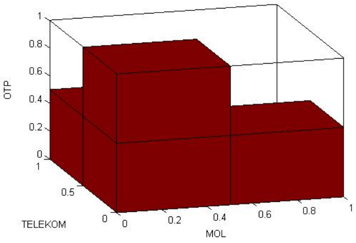


Figure 3

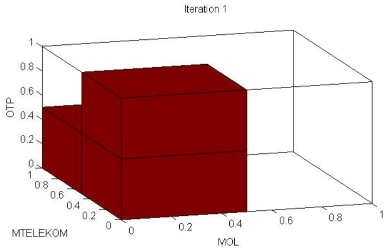


Figure 4

all those sets are deleted from $\mathbb{H}'$ that does not contain any point from the feasible solution set of the problem. Thus the result of the procedure *Newsets*, the family of $\mathbb{H}'$ covering the feasible solution set of the given problem has been shown in Figure 3.

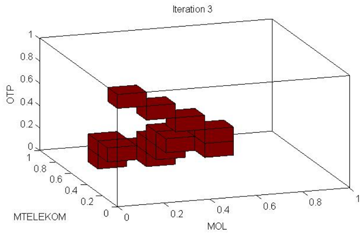


Figure 5

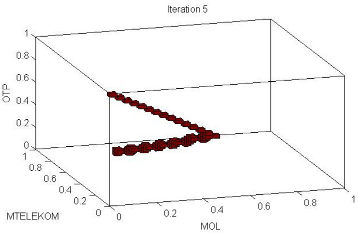


Figure 6

The main part of the algorithm starts at step $2(c)$. Two hundred random points are generated from the unit simplex (set $\mathscr{S}\mathscr{P}$). For each generated point either a joint decreasing direction is computed and after that a corresponding Pareto optimal solution has been identified through some iteration or it has been shown that the generated point itself is a Pareto optimal solution of the problem. After we obtained 200 Pareto optimal solutions in set $\mathscr{F}\mathscr{P}$ at step $2(d)$ we delete those boxes that does not contain any point from $\mathscr{F}\mathscr{P}$ at step $2(e)$. The result of the first iteration can be seen in Figure 4.

From the original eight boxes remains three. For these three boxes the procedure has been repeated in the second iteration. The results of iteration 3, 5 and 7 are illustrated in Figures 5, 6, and 7, respectively.

These figures illustrate the flow of our computations. Finally to illustrate the convergence of our method the whole Pareto optimal set was determined based on the result of [37], and compared to the result computed in the fifth iteration, see in Figure 8.

The summary of our computations are shown in the 1 where I stands for the iteration number; B_{in} and B_{out} denotes the number of boxes at the beginning and at the end

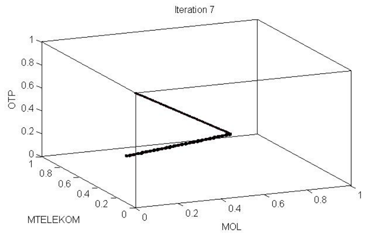


Figure 7

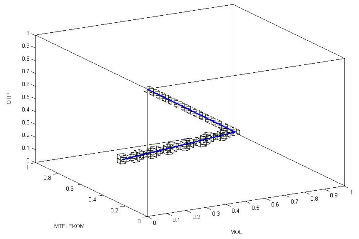


Figure 8

of iteration, respectively. Furthermore, $T(s)$ is the computational time of the I^{th} iteration in seconds, while d is the diameter of the family of sets, $\mathbb{H}$.

I	B_{in}	$T(s)$	B_{out}	d
1	1	8	3	2^{-3}
2	24	29	7	2^{-6}
3	56	70	15	2^{-9}
4	120	166	29	2^{-12}
5	232	312	56	2^{-15}
6	448	696	110	2^{-18}
7	880	1429	228	2^{-21}

Table 1
Computational results for Markowitz model using subdivision method.

The total computational time for our MATLAB implementation using a laptop with the following characteristics (processor: Intel(r) Core(TM) i3 [3.3 GHZ], RAM Memory: 4096 MB), took 2710 seconds for the subdivision algorithm for the given Markovitz model to approximate the whole Pareto optimal solution set with the accuracy $\varepsilon = 2.4\ 10^{-8}$.

Analyzing our approximation of the Pareto optimal solution set, we can conclude that our first option is to buy OTP shares only. From the data it can be understood that this share has the biggest return (smallest loss in the financial crisis), so this solution represents the strategy when someone does not care about the risk but only about the return. From that point a line starts which represents strategies related to portfolios based on OTP and MOL shares. Clearly, there exists a breaking point where a new line segment starts. From the braking point the line lies in the interior of the simplex suggesting a portfolio based on all three selected shares.

Acknowledgement

This research has been supported by the TÁMOP-4.2.2./B-10/1-2010-0009, Hungarian National Office of Research and Technology with the financial support of the European Union from the European Social Fund.

Tibor Illés acknowledges the research support obtained as a part time *John Anderson Research Lecturer* from the Management Science Department, Strathclyde University, Glasgow, UK. This research has been partially supported by the UK Engineering and Physical Sciences Research Council (grant no. EP/P005268/1).

References

[1] A.A. Akkeleş, L. Balogh and T. Illés, New variants of the criss-cross method for linearly constrained convex quadratic programming. *European Journal of Operational Research*, 157:74–86, 2004.

[2] M.S. Bazaraa H.D. Sherali and M.C. Shetty, *Nonlinear Programing Theory and Algorithms.* 3-rd edition, Jhon Wiley & Son Inc., New Jersey, 2006.

[3] Budapest Stock Exchange, `http://client.bse.hu/topmenu/trading_data/stat_hist_download/trading_summ_pages_dir.html`
Accessed: 2012. 06. 10.

[4] Zs. Csizmadia, *New pivot based methods in linear optimization, and an application in petroleum industry*, PhD Thesis. Eötvös Loránd University of Sciences, Budapest, Hungary, 2007.

[5] Zs. Csizmadia and T. Illés, New criss-cross type algorithms for linear complementarity problems with sufficient matrices. *Optimization Methods and Software*, 21(2):247–266, 2006.

[6] Zs. Csizmadia, T. Illés and A. Nagy, The s-monotone index selection rules for pivot algorithms of linear programming. *European Journal of Operation Research*, 221(3):491–500, 2012.

[7] M. Dellnitz, O. Schütze and T. Hestermeyer, Covering Pareto Sets by Multilevel Subdivision Techniques. *Journal of Optimization Theory and Applications*, 124(1):113–136, 2005.

[8] M. Dellnitz, O. Schütze and Q. Zheng, Locating all the zeros of an analytic function in one complex variable, *Journal of Computational and Applied Mathematics*, 138(2):325–333, 2002.

[9] G. Eichfelder, *Adaptive Scalarization Methods in Multiobjective Optimization*, Springer-Verlag, Berlin, Heidelberg, 2008.

[10] G. Eichfelder and T.X.D. Ha, Optimality conditions for vector optimization problems with variable ordering structures. *Optimization: A Journal of Mathematical Programming and Operations Research*, 62(5):597–627, 2013.

[11] Farkas Gy., A Fourier-féle mechanikai elv alkalmazásai, (The applications of the mechanical principle of Fourier [in Hungarian]). *Mathematikai és Természettudományi Értesítő*, 12:457–472, 1894.

[12] Gy. Farkas, Theorie der einfachen Ungleichungen. *Journal für die Reine und Angewandte Mathematik*, 124:1–27, 1902.

[13] J. Fliege, B.F. Svaiter, Steepest Descent Methods for Multicriteria Optimization, Mathematical Methods of Operations Research. *Mathematical Methods of Operations Research*, 51:479–494, 2000.

[14] J. Fliege, An Efficient Interior-Point Method for Convex Multicriteria Optimization Problems, *Mathematical Methods of Operations Research*, 31(4):825–845, 2006.

[15] F. Giannessi, G. Mastroeni and L. Pellegrini, On the Theory of Vector Optimization and Variational Inequalities. Image Space Analysis and Separation. In *Vector Variational Inequalities and Vector Equilibria: Mathematical Theories*, editor: F. Giannessi. Series on Nonconvex Optimization and its Applications, Vol.38, Kluwer, Dordrecht, 2000.

[16] D. den Hertog, C. Roos and T. Terlaky, The linear complementarity problem, sufficient matrices and the criss-cross method. *Linear Algebra and Its Applications*, 187:1–14, 1993.

[17] T. Illés and K. Mészáros, A New and Constructive Proof of Two Basic Results of Linear Programming. *Yugoslav Journal of Operations Research*, 11:15–30, 2001.

[18] T. Illés and M. Nagy, A new variant of the Mizuno-Todd-Ye predictor-corrector algorithm for sufficient matrix linear complementarity problem. *European Journal of Operational Research*, 181(3):1097–1111, 2007.

[19] T. Illés, M. Nagy and T. Terlaky, Interior Point Algorithms [in Hungarian] (Belsőpontos algoritmusok), pp. 1230–1297. Iványi Antal (alkotó szerkesztő), *Informatikai Algoritmusok 2*, ELTE Eötvös Kiadó, Budapest, 2005.

[20] T. Illés, C. Roos and T. Terlaky, Polynomial Affine–Scaling Algorithms for $P_*(\kappa)$ Linear Complementarity Problems. In P. Gritzmann, R. Horst, E. Sachs, R. Tichatschke, editors, *Recent Advances in Optimization*, Proceedings of the 8^{th} French-German Conference on Optimization, Trier, July 21-26, 1996, *Lecture Notes in Economics and Mathematical Systems 452*, pp. 119–137, Springer Verlag, 1997.

[21] T. Illés and T. Terlaky, Pivot Versus Interior Point Methods: Pros and Cons. *European Journal of Operations Research* 140:6–26, 2002.

[22] E. Klafszky and T. Terlaky, The role of pivoting in proving some fundamental theorems of linear algebra. *Linear Algebra and its Application* 151:97–118, 1991.

[23] D.T. Luc, *Theory of Vector Optimization*. Lecture Notes in Economics and Mathematical Systems, No. 319, Springer-Verlag, Berlin, 1989.

[24] G D. Luenberger and Y Ye, *Linerar and Nonlinear Programming*. International Series in Operations Research and Management Science, Springer-Verlag, New York, 2008.

[25] H. Markowitz, The Optimalization of Quadratic Function Subject to Linear Constarint. *Naval Reserch Logistic Quartely*, 3(1-2):111–133, 1956.

[26] H. Markowitz, Portfolio Selection. *The Journal of Finance*, 7(1):77–91, 1952.

[27] H. M. Markowitz, *Portfolio Selection: Efficient Diversification of Investment*, Jhon Wiley and Son Inc., New York, 1959.

[28] K. Miettinen, *Nonlinear Multiobjective Optimization*. International Series in Operations Research and Management Science, Springer US, 1998.

[29] K. G. Murty, *Linear Complementarity, Linear and Nonlinear Programming*. Heldermann Verlag, Berlin, 1988.

[30] V. Pareto, *Cours D'Économie Politique*. F. Rouge, Lausanne, 1896.

[31] Prékopa A., *Lineáris programozás I.*. Bolyai János Matematikai Társulat, Budapest, 1968.

[32] A. Prékopa, A Very Short Introduction to Linear Programming. *RUTCOR Lecture Notes* 2-92, 1992.

[33] C. Roos, T. Terlaky and J-Ph. Vial, *Theory and Algorithms for Linear Optimization: An Interior Point Approach*. Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons., 1997.

[34] S. Schäffler, R. Schultz and K. Weinzierl, A Stochastic Method for the Solution of Unconstrained Vector Optimization Problems. *Journal of Optimalization Theory and Application*, 114(1):112–128, 2002.

[35] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley and Sons Inc., New York, 1986.

[36] T. Terlaky, A convergent criss–cross method. *Math. Oper. und Stat. Ser. Optimization* 16(5):683–690, 1985.

[37] J. Vörös, Portfolio analysis – An analytic derivation of the efficient portfolio frontier. *European Journal of Operational Reserch*, 23(3):294–300, 1986.

Population Dynamic Models Leading to Logarithmic and Yule Distribution

János Izsák[1], **László Szeidl**[2]

[1]Department of Systematical Zoology and Ecology, Eötvös Loránd University, Pázmány Péter sétány 1/C, H-1117 Budapest, Hungary
E-mail: ijanos@caesar.elte.hu

[2]Institute of Applied Informatics, John von Neumann Faculty of Informatics, Óbuda University, Bécsi u. 96/B. H-1034 Budapest, Hungary
E-mail: szeidl@uni-obuda.hu

Abstract: A significant field of species abundance distribution (SAD) has a population dynamical character, in which it is supposed that the stochastic speciation process and the evolution of different species are determined by the same linear birth and death process. The distributions of the number of individuals after the speciation tend to a discrete limit distribution depending on some condition if the observation time increases. In the earlier publications, in general, the speciation process was supposed to be a homogeneous Poisson process. In a more realistic case, if the speciation process is inhomogeneous Poisson, the investigation of the model is obviously more difficult. In this paper we deal with the models, in which the birth and death intensities are identical, the speciation rate is bounded, locally integrable and has asymptotically power type behaviour. Limit parameters for these models, depending on the speciation rate, are proportional to a logarithmic or (exactly or asymptotically) Yule distribution. In connection with the sample statistics some results are derived in general and also in special cases (logarithmic and Yule distribution), which are related to the random choice of a species or an individual from the whole population of the system.

Keywords: population dynamic model; species abundance distribution; Kendall process; Poisson process; logarithmic distribution; Yule distribution

1 Introduction

A frequently cited field of species abundance models possesses population dynamical background. In these models continuous abundances are mostly assumed (Engen and Lande, 1996a, 1996b). Generally, it is supposed that the process which describes the entering time points of the new species in the system is Poisson process (Karlin és McGregor, 1967). In this paper we consider models in which the species abundance can take discrete values $0,1,2,...$, the evolution of the species entering the system is determined by a linear birth and death model (Kendall (1948a, 1948b) and as an essential enlargement of the population dynamical models, the speciation processes are assumed from a class of inhomogeneous Poisson processes. For the description of the model parameters, the Yule distribution plays an important role instead of logarithmic distribution.

It is worth noting that in case when the speciation rate is varying in time, i.e. the speciation process in the model is inhomogeneous Poisson, it is more difficult to reach concrete results. We mention here the results of Branson (1991, 2000), in which the models lead to logarithmic distribution under special inhomogeneity condition. In this paper we deal with a class of models which lead to distributions exactly or asymptotically proportional to the Yule distribution.

Note that the Yule distribution with parameter ρ (>0) is determined as $p_k = = \rho\Gamma(\rho+1)\frac{\Gamma(k)}{\Gamma(k+\rho+1)}$, $k=1,2,...$, for which the asymptotic relation $p_k = = \rho\Gamma(\rho+1)k^{-\rho-1}(1+o(1))$, $k\to\infty$ holds and it can be interpreted as a generalization of power type (Pareto type) distribution for a discrete case (see Simon (1955), Newman (2006)).

Let us consider a system of many species on the time interval $(t_0-T,t_0]$, where $t_0\le 0$, $T>0$ and the system is empty at the initial time t_0-T, i.e. the system does not contain any species. After passing T time we investigate the system at the observation time t_0. The time points, when the species enter in the system, are determined by the random jumping points of a homogeneous or inhomogeneous Poisson process Π, having intensity function $\lambda(t)$, $t\le t_0$, which is defined on the half line $(-\infty,t_0]$ (see 4.5.§., Kingman (1993)). The process Π defines a right continuous Poisson process $N_T(t)$, $t_0-T\le t\le t_0$ for each $T>0$ on the time interval $(t_0-T,t_0]$ satisfying the condition $N_T(t_0-t)=0$. We mention that the process $N_T(t)$ can be given by construction (see p. 50., Kingman (1993), p. 62., Lakatos et al. (2013)).

It is clear that the Poisson process $N_T(t)$, $t_0-T\le t\le t_0$ has rate function which equals $\lambda(t)$ on the interval $t_0-T\le t\le t_0$. The rate function (of formation of a new species) $\lambda(t)$ of the speciation process does not depend on the species entering the system, but it can depend on time t. Then for any pairwise disjoint intervals $(x_i,y_i]\subset(t_0-T,t_0]$, $i=1,2,...$ the increments $N_T(y_i)-N_T(x_i)$ are independent random variables with Poisson distribution of parameter $E(N_T(y_i)-N_T(x_i))=\int_{x_i}^{y_i}\lambda(s)ds$.

Note that if we investigate the abundance distribution for the case of homogeneous (i.e. $\lambda(t)\equiv\lambda_0$) speciation process then we have the same distribution for any observation time t_0 as $T\to\infty$, in contrast with the inhomogeneous cases, when the limit depends on the observation time t_0. Partly, this means that if there exists the limit abundance distribution in homogeneous cases as $T\to\infty$ then the limit is identical with the equilibrium (stationary) distribution, while in cases of inhomogeneous speciation process this property is no longer valid.

Assume that the number of individuals of a species entering the system equals 1. Moreover, the random fluctuation of the population size of a species does not depend on others and it is determined by a continuous-time Markov chain for all species with the same transition probabilities $P_{1,k}(s)$, $s \geq 0$, $k = 0,1,...$ The state 0 (i.e. the extinction of a species) means the absorption state. Then the number of species $S_{k,T}$, $k = 1,2,...$ having exactly k, $k = 1,2,...$ living individuals at the observation time t_0 are independent random variables with Poisson distribution of parameters $\mu_{k,T}$, $k = 1,2,...$ which can be given in the following general form (Karlin and McGregor, 1967)

$$\mu_{k,T} = \int\limits_{t_0-T}^{t_0} P_{1,k}(t_0 - t)\lambda(t)dt = \int\limits_{0}^{T} P_{1,k}(t)\lambda(-t+t_0)dt, \; k = 1,2,... \tag{1}$$

This formula plays an important role in the computation of the parameters $\mu_{k,T}$. In accordance with the Kendall population dynamical model, after a species enters the system, the random fluctuation of the population size of a species is determined by a linear birth and death model (Kendall, 1948a, 1948b), where the birth and death rates na and nc, respectively, depend on the actual population size n of the species and a and c ($a \leq c$) are positive constants.

It is known (see Karlin and McGregor (1967)) that if the speciation process $N_T(t)$ is homogeneous Poisson with intensity rate λ, then the random variables $S_{k,T}$, $k = 1,2,...$ (the number of species $S_{k,T}$, $k = 1,2,...$ having exactly k, $k = 1,2,...$ living individuals) are independent and have Poisson distribution with parameters $\mu_{k,T}$ (see also Engen and Lande (1996a, l996b), Watterson (1974), Lange (2010), Bowler and Kelly (2012)), where

$$\mu_{k,T} = \frac{\lambda}{a}\frac{1}{k}\rho^k \left(\frac{1 - e^{-(c-a)T}}{1 - \rho e^{-(c-a)T}} \right)^k \to \mu_k = \frac{\lambda}{a}\frac{1}{k}\rho^k, \; T \to \infty, \; \text{if } \rho = a/c < 1 \tag{2}$$

and

$$\mu_{k,T} = \frac{\lambda}{a}\frac{1}{k}\left(\frac{aT}{1+aT} \right)^k \to \mu_k = \frac{\lambda}{a}\frac{1}{k}, \; T \to \infty, \; \text{if } a = c. \tag{3}$$

From the formulas (2) and (3) it follows for the case $a < c$ $(\rho < 1)$ that the sequence $\mu_{k,T}$, $k = 1,2,...$ (i.e. the expected values of the number of species having exactly k individuals) is proportional to the logarithmic distribution with parameter $\rho\frac{1-e^{-(c-a)T}}{1-\rho e^{-(c-a)T}}$ (in the limit as $T \to \infty$ with parameter ρ). This distribution does not depend on the observation time t_0 if the speciation process is homogeneous, i.e. $\lambda(t) \equiv \lambda_0$.

In case $a = c$ the sequence $\mu_{k,T}$, $k = 1,2,...$ is proportional to a logarithmic distribution only if $T < \infty$. In that case the parameter of the logarithmic distribution equals

$\frac{aT}{1+aT}$. The parameters $\mu_{k,T}$ have the limit

$$\mu_k = \lim_{T\to\infty} \mu_{k,T} = \frac{\lambda_0}{a}\frac{1}{k}, \;\; k = 1,2,...,$$

however, they will be no longer proportional to a probability distribution because $\sum_{k=1}^{\infty} \frac{1}{k} = \infty$. From this it follows that if $T \to \infty$, then the expected value of the number of species having minimum one individual at time t_0, tends to ∞, at the same time the expected value of the number of species with exactly k individuals tends to value $\frac{\lambda_0}{a}\frac{1}{k}$. Thus the number of species with exactly k individuals has Poisson limit distribution with parameter $\frac{\lambda_0}{a}\frac{1}{k}$, $k = 1,2,...$
We note that in the remaining case under the condition $a > c$ for all $k = 1,2,...$ $\lim_{T\to\infty} ES_{k,T} = \lim_{T\to\infty} \mu_{k,T} = \infty$ is true.

2 Results

In the present section of the paper we study two problems, as follows.

1. We consider the birth and death process under the condition that the birth and death rates are equal ($a = c$), however, the rate $\lambda(t)$ of the Poisson speciation process $N(t)$ is inhomogeneous. The problem is to give exact and asymptotic formulas for the behaviour of the parameters $\mu_{k,T}$ as $T \to \infty$ under the condition

   $$\lambda(t) = \frac{\lambda_0}{(1+\alpha|t|)^{\beta}}, \; -\infty < t \le 0 \tag{4}$$

   or in more general setting, if $\lambda(t)$ satisfies the asymptotic condition

   $$\frac{(1+\alpha\,|t|)^{\beta}}{\lambda_0}\lambda(t) \to 1, \; t \to -\infty, \;\; \lambda_0, \alpha, \beta > 0, \; -\infty < t \le t_0 \le 0, \tag{5}$$

   where λ_0, α and β are arbitrary positive numbers. This model generalizes the above described models.

2. In connection to this model, we consider a random choice problem for Poissonian distributed abundances at observation time t_0. In this model, we investigate a species randomly chosen from the population or an individual from the whole population with which probability belongs to a species with k $(k \ge 1)$ individuals.

2.1 Exact and asymptotic results for the parameters $\mu_k = \lim_{T\to\infty} \mu_{k,T}$ when the speciation rate $\lambda(t)$ satisfies the conditions (4) and (5)

In case of inhomogeneous Poisson speciation process, the consideration at time t_0 of the parameters μ_k will be more difficult comparing to a homogeneous case, because the parameters $\mu_k = \int_{-\infty}^{t_0} P_{1k}(t_0 - t)\lambda(t)dt,\ k = 1,2,...$ may depend not only on speciation rate $\lambda(t)$ but also on the observation time t_0.

In this section we assume that the condition (4) or (5) holds, instead of the homogeneity of the speciation rate ($\lambda(t) \equiv \lambda_0$), which makes possible a more general framework for the modelling of the population dynamics. Here the observation time $t_0 \leq 0$ can be arbitrarily chosen. Note that under the condition (4) $\lambda(t)$ is a monotonically increasing function which realizes monotonically increasing speciation rate. The fact that the speciation rate can be increasing, from a biological point of view, is referred in the paper of Rolland et al. (2014). In special cases we give exact formulas for the parameters $\mu_k,\ k = 1,2,...$, and at the same time the asymptotic formulas will be valid for the class of bounded rate functions $\lambda(t)$ satisfying the more general condition (5), instead of (4).

In accordance with the model stated above, the dynamics (in time) of the number of individuals of a species is described by a linear birth and death process (Kendall process) for which the rate of birth and death are na and nc, respectively, depending on the population size n and on the given constants $a, c > 0$. The initial population size of a species is 1 and the state 0 is an absorbing one. The birth and death process is a continuous-time Markov chain, which determines the random fluctuation of the population size in time after speciation.

Denote the population size of species by $X_t,\ t \geq 0,\ X_0 = 1$, where t means the passing time after speciation and let $P_{1k}(t) = P(X_t = k \mid X_0 = 1)$, $k = 0,1,...$ be the transition probability function of the process. Since the initial state of the process is 1, thus $P_{11}(0) = 1$ and $P_{1k}(0) = 0,\ k \neq 1$.

The generating function of the time-dependent transition probabilities $P_{1k}(t)$, $k = 0,1,...$ of the Markov chain $X_t,\ t \geq 0$ can be determined by the Kolmogorov forward differential equations, from which the transition probabilities $P_{1k}(t)$ can be given in an explicit form (Kendall, 1948a):

$$P_{1,0}(t) = \frac{at}{1+at},\ \ P_{1,k}(t) = \frac{(at)^{k-1}}{(1+at)^{k+1}},\ k = 1,2,... \tag{6}$$

Theorem 1. If the birth and death intensities are equal ($a = c$) and the intensity function of the speciation process satisfies the condition (5), then
a) independently of the value t_0 the following asymptotic relation holds

$$\mu_k = \frac{\lambda_0}{a}\left(\frac{a}{\alpha}\right)^{\beta}\beta\Gamma(\beta+1)\frac{1}{k^{\beta+1}}(1+(1)),\ k \to \infty. \tag{7}$$

This means that for sufficiently large k, the elements of the sequence μ_k of expected values of the numbers of the species with k members are asymptotically proportional to the elements of a Yule distribution with parameter β.
b) Under the condition (4) an exact formula holds for the sequence $\mu_k,\ k = 1,2,...$if $a = \alpha > 0$, $\beta > 0$ and $t_0 = 0$ is the time of the observation. In this case the sequence $\mu_k,\ k = 1,2,...$ can be given with the help of the Yule distribution of parameter β

multiplying by the constant $\frac{\lambda_0}{a\beta}$ as follows:

$$\mu_k = \frac{\lambda_0}{a\beta}\beta\frac{\Gamma(k)\Gamma(\beta+1)}{\Gamma(k+\beta+1)},\ k=1,2,... \tag{8}$$

In the special case, for $\beta=1$ the equation $\mu_k = \frac{\lambda_0}{a}\frac{1}{k(k+1)}$, $k=1,2,...$, holds and for $\beta=2$ the equation $\mu_k = \frac{\lambda_0}{a}\frac{2}{k(k+1)(k+2)}$, $k=1,2,...$ is true.

Proof. For simplicity, define $\lambda(t)=\lambda(-t),\ t>0$. If the birth and death rates are equal, i.e. $a=c$, then the transition probabilities $P_{1k}(t)$ satisfy the relations (6), therefore by the formula (1) the numbers of species with $k\geq 1$ members at the observation time t_0 ($t_0 \leq 0$) are independent and have Poisson distribution with parameters (expected values) as follows

$$\mu_k = \int_{-\infty}^{t_0} P_{1k}(t_0-t)\lambda(t)dt = \int_{-\infty}^{0} P_{1k}(-t)\lambda(t_0+t)dt = \int_0^{\infty} P_{1k}(t)\lambda(t_0-t)dt =$$
$$= \int_0^{\infty} \frac{(at)^{k-1}}{(1+at)^{k+1}}\lambda(t-t_0)dt,\ k=1,2,... \tag{9}$$

These integrals are finite because the integrands are bounded, moreover, from the condition (5) $\frac{(1+\alpha|t|)^{\beta}}{\lambda_0}\lambda(t)\to 1,\ t\to\infty$ follows, then by (6) we have

$$P_{1,k}(t)\lambda(t) = \frac{\lambda_0}{\alpha^{\beta}}t^{-\beta-2}(1+o(1)),\ t\to\infty,$$

which means that $(P_{1,k}(t)\lambda(t))^{-1}\frac{\lambda_0}{\alpha^{\beta}}t^{-\beta-2} \to 1,\ t\to\infty$. The integral in (9) can be given in the form

$$\mu_k = \frac{\lambda_0}{a}\int_0^{\infty} f_k(t)g(t)dt,\ k\geq 1, \tag{10}$$

where

$$f_k(t) = \frac{t^{k-1}}{(1+t)^{k+1+\beta}},\ \ g(t) = \frac{1}{\lambda_0}(1+t)^{\beta}\lambda(t/a+|t_0|).$$

It is clear that from the condition (5) it follows that the function $g(t)$ satisfies the asymptotic relation

$$g(t) = \frac{(1+t)^{\beta}}{[1+\alpha(t/a+|t_0|)]^{\beta}}\frac{1}{\lambda_0}[1+\alpha(t/a+|t_0|)]^{\beta}\lambda(t/a+|t_0|) \to \left(\frac{a}{\alpha}\right)^{\beta},\ t\to\infty. \tag{11}$$

Let us consider the asymptotic behaviour of the parameters μ_k as $k\to\infty$. Firstly, we prove that the following convergence is true

$$\left(\int_0^{\infty} f_k(t)dt\right)^{-1}\mu_k \to \frac{\lambda_0}{a}\left(\frac{a}{\alpha}\right)^{\beta},\ k\to\infty. \tag{12}$$

Since the integral $\int_0^\infty f_k(t)dt,\ k=1,2,...$ in formula (12) can be determined by formula 2.2.4.24., p. 298., Prudnikov et al. (1986) and it equals the Yule distribution of parameter β as follows

$$\int_0^\infty f_k(t)dt = \int_0^\infty \frac{t^{k-1}}{(1+t)^{k+1+\beta}}dt = \frac{\Gamma(k)\Gamma(\beta+1)}{\Gamma(k+\beta+1)},\ k\geq 1, \tag{13}$$

therefore if we prove the relations (12) and (13) we immediately have the asymptotic relation (7) of the Theorem 1.
It is known that the gamma function has the following asymptotic property (see p. 257, Davis,.1972): for any fixed real numbers u,v

$$\frac{\Gamma(x+u)}{\Gamma(x+v)} = x^{u-v}(1+o(1)),\ x\to\infty, \tag{14}$$

consequently, by (13) and (14) we have

$$\int_0^\infty f_k(t)dt = \Gamma(\beta+1)\frac{1}{k^{\beta+1}}(1+o(1)),\ k\to\infty. \tag{15}$$

Now, we verify the relation (12). For arbitrary positive numbers γ, A and for any $0\leq t\leq A$

$$k^\gamma\left(\frac{t}{1+t}\right)^k \leq k^\gamma\left(\frac{A}{1+A}\right)^k = \exp\left\{k\log\frac{A}{1+A}+\gamma\log k\right\} =$$
$$= \exp\left\{-k\left[\log\left(1+\frac{1}{A}\right)-\frac{\gamma}{k}\log k\right]\right\}\to 0,\ \ k\to\infty$$

holds. It is obvious that

$$\int_0^A f_k(t)dt < \left(\frac{A}{1+A}\right)^{k-1}\int_0^A \frac{1}{(1+t)^{\beta+2}}dt < \left(\frac{A}{1+A}\right)^{k-1}\to 0,\ \ k\to\infty.$$

Since $\lambda(t)$ and consequently $g(t)$ are bounded functions, then for $\gamma=\beta+1$ we have

$$k^{\beta+1}\int_0^A f_k(t)g(t)dt < \max_{0\leq t\leq A} g(t)\cdot\left(\frac{A}{1+A}\right)^{k-1}\to 0,\ \ k\to\infty \tag{16}$$

and

$$k^{\beta+1}\int_0^A f_k(t)dt < k^{\beta+1}\left(\frac{A}{1+A}\right)^{k-1}\to 0,\ \ k\to\infty. \tag{17}$$

By virtue of the asymptotic relation (11) the convergence $g(t)\to\left(\frac{a}{\alpha}\right)^\beta,\ t\to\infty$ is true, therefore for arbitrarily chosen $\varepsilon>0$ there exists a constant A_ε such that

$$\left|g(t)-\left(\frac{a}{\alpha}\right)^\beta\right| < \varepsilon,\ \ t\geq A_e. \tag{18}$$

From this it follows that

$$\left| \int_{A_\varepsilon}^{\infty} f_k(t) \left(\frac{a}{\alpha}\right)^{\beta} dt - \int_{A_\varepsilon}^{\infty} f_k(t) g(t) dt \right| \le \varepsilon \int_{A_\varepsilon}^{\infty} f_k(t) dt. \tag{19}$$

In summary, on the basis of the relations (15), (16) and (17) from (19) it is clear that for every $\varepsilon > 0$ it holds

$$\limsup_{k\to\infty} \left| \left(\int_0^{\infty} f_k(t) dt \right)^{-1} \int_0^{\infty} f_k(t) g(t) dt - \left(\frac{a}{\alpha}\right)^{\beta} \right| < \varepsilon,$$

thus

$$\mu_k = \frac{\lambda_0}{a} \int\limits_0^{\infty} f_k(t) g(t) dt = \frac{\lambda_0}{a} \left(\frac{a}{\alpha}\right)^{\beta} \Gamma(\beta+1) \frac{1}{k^{\beta+1}} (1+o(1)),\ k \to \infty.$$

The result of the second part b) of the Theorem 1 is obtained directly from the formulas (9) and (13):

$$\mu_k = \frac{\lambda_0}{a} \int\limits_0^{\infty} \frac{t^{k-1}}{(1+t)^{k+1+\beta}} dt = \frac{\lambda_0}{a} \frac{\Gamma(k)\Gamma(\beta+1)}{\Gamma(k+\beta+1)}.$$

□

2.2 Theorems on the random choice of a species or an individual from the whole population related to the model considered above

Consider a population of various species. Assume in general that the number of species of the population is not necessarily bounded. Denote the number of species consisting of exactly k individuals by S_k, $k = 1,2,...$ and suppose that the random variables S_k are independent, having Poisson distribution with parameters μ_k, $k = 1,2,...$ and the condition $\mu = \sum_{k=1}^{\infty} \mu_k < \infty$ holds. For example, the random variables may be S_k, the number of species having k individuals at the time t_0 (see the model described earlier). Define the events A_k and B_k as follows

$A_k = \{$randomly chosen species from the population of species consists of k individuals$\}$,
$B_k = \{$randomly chosen individual from the population of individuals belongs to a species consisting of exactly k individuals$\}$.

Let us consider the probabilities $P(A_k)$, and $P(B_k)$, $k = 1,2,...$ of the events A_k, and B_k, respectively. Denote $\overline{S}_k = \sum_{i \neq k} S_i$ and $R_k = \frac{1}{k} \sum_{i \neq k} iS_i$, $k = 1,2,....$ Let $\mathscr{R}_k$ be the set of all possible values of the random variables $kR_k = \sum_{i \neq k} iS_i$, that is, for $k = 1,2,...$

$\mathscr{R}_k = \{ \sum\limits_{i \neq k} im_i :\ m_i$ are arbitrary natural numbers and the sum $\sum\limits_{i \neq k} im_i$ is finite$\}$.

The random choice of a species or an individual from the population considered above means that for all $n \geq 0,\ m \geq 0,\ n+m > 0$ and $r \in \mathscr{R}_k$ the following relations hold

$$P(A_k \mid S_k = n,\ \overline{S}_k = m) = \frac{n}{n+m},\ \ P(B_k \mid S_k = n,\ R_k = \frac{1}{k}r) = \frac{kn}{kn+r}.$$

Using the formula of total probability we get

$$\begin{aligned} P(A_k) &= \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} P(A_k \mid S_k = n,\ \overline{S}_k = m) P(S_k = n,\ \overline{S}_k = m) = \\ &= \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} \frac{n}{n+m} P(S_k = n,\ \overline{S}_k = m) = E \frac{S_k}{S_k + \overline{S}_k}. \end{aligned} \tag{20}$$

Taking into consideration that the random variables S_k and $\overline{S}_k$ are independent and have Poisson distribution with parameters μ_k and $\mu - \mu_k$ respectively, using the relation (20) it is easy to determine the well-known general formula (21) for the probability $P(A_k)$

$$P(A_k) = \frac{\mu_k}{\mu},\ k = 1, 2, \ldots \tag{21}$$

The computation of the probability $P(B_k),\ k = 1, 2, \ldots$ is more difficult and leads to an interesting formula determined by the parameters $\mu_k,\ \mu$ and the generating function (z-transform) $G(z)$ of the sequence $\mu_k,\ k = 1, 2, \ldots$ This formula makes the further consideration of the probability $P(B_k)$ as $k \to \infty$ possible.

The number of different species possessing the population is $S_k + \overline{S}_k = \sum_{i=1}^{\infty} S_i$ and the number of individuals in the population equals $kS_k + kR_k = \sum_{i=1}^{\infty} iS_i$. Using the formula of the total probability we have

$$\begin{aligned} P(B_k) &= \sum_{n=0}^{\infty} \sum_{r \in \mathscr{R}_k} P(B_k \mid S_k = n,\ kR_k = r) P(S_k = n,\ kR_k = r) = \\ &= \sum_{n=1}^{\infty} \sum_{r \in \mathscr{R}_k} \frac{kn}{kn+r} P(S_k = n,\ kR_k = r) = E \frac{S_k}{S_k + R_k}. \end{aligned} \tag{22}$$

It will be noted that $\sum\limits_{n=1}^{\infty} P(B_k) = 1$, because

$$\sum_{n=1}^{\infty} E \frac{S_k}{S_k + R_k} = E \sum_{n=1}^{\infty} \frac{kS_k}{kS_k + kR_k} = E \frac{kS_k + kR_k}{kS_k + kR_k} = 1.$$

Let us define the generating function of $G(z)$ of the sequence $\mu_k,\ k = 1, 2, ..$ as follows

$$G(z) = \sum_{k=1}^{\infty} \mu_k z^k,\ |z| \leq 1.$$

Theorem 2. If $S_i, i = 1,2,...$ denote the number of species containing i individuals and the random variables $S_1, S_2, ...$ are independent and they have Poisson distribution function with parameters $\mu_1, \mu_2, ...$, then the following relation holds

$$P(B_k) = \mu_k \int_0^1 \exp\left\{-\sum_{i=1}^{\infty} \mu_i (1 - x^{i/k})\right\} dx = \mu_k \int_0^1 \exp\left\{-\mu + G(x^{1/k})\right\} dx \tag{23}$$

and the probabilities $P(B_k)$ satisfy the asymptotic relation

$$P(B_k) = \mu_k (1 + o(1)), \;\; k \to \infty. \tag{24}$$

Proof. Since the random variables S_k and R_k are independent, then using the formula (22) the probability $P(B_k)$ can be given in the form

$$P(B_k) = E\frac{S_k}{S_k + R_k} = E\left(E(\frac{S_k}{S_k + R_k} \mid R_k)\right) = E\left(\sum_{n=1}^{\infty} \frac{n}{n + R_k} \frac{\mu_k^n}{n!} e^{-\mu_k}\right).$$

It is clear that $P(B_k) = 0$, when $\mu_k = 0$ and for $\mu_k > 0$

$$\frac{\mu_k^n}{n + R_k} = \mu_k^{-R_k} \int_0^{\mu_k} x^{R_k + n - 1} dx.$$

The order of summation and integration, as well as the order of integration and expectation can be changed in the following relation, thus we have

$$\begin{aligned} P(B_k) &= e^{-\mu_k} E\left(\sum_{n=1}^{\infty} \frac{1}{(n-1)!} \mu_k^{-R_k} \int_0^{\mu_k} x^{R_k + n - 1} dx\right) = \\ &= e^{-\mu_k} E\left(\int_0^{\mu_k} \mu_k^{-R_k} x^{R_k} \sum_{n=0}^{\infty} \frac{x^n}{n!} dx\right) = e^{-\mu_k} E\left(\int_0^{\mu_k} \left(\frac{x}{\mu_k}\right)^{R_k} e^x dx\right) = \\ &= e^{-\mu_k} \mu_k E\left(\int_0^1 x^{R_k} e^{\mu_k x} dx\right) = e^{-\mu_k} \mu_k \int_0^1 E\left(x^{R_k}\right) e^{\mu_k x} dx. \end{aligned} \tag{25}$$

The expected value Ex^{R_k} equals the generating function of random variable $R_k = \sum_{i \neq k} \frac{i}{k} S_i$ in the place $x^{1/k}$, which is easy to compute. Since the random variables S_i $(i = 1,2,...)$ are independent and they have Poisson distribution with parameters μ_i, $i = 1,2,...$, moreover, the generating function of random variable S_i has the form

$$Ex^{S_i} = e^{\mu_i (x-1)}, \; 0 < x \leq 1,$$

then

$$\begin{aligned} Ex^{R_k} = E(x^{1/k})^{\sum_{i \neq k} i S_i} = E \prod_{i \neq k} \left(x^{i/k}\right)^{S_i} &= \exp\left\{\sum_{i \neq k} \mu_i (x^{i/k} - 1)\right\} \\ &= \exp\left\{-(\mu - \mu_k) + \sum_{i \neq k} \mu_i x^{i/k}\right\}. \end{aligned}$$

From the formula (25) we get

$$P(B_k) = e^{-\mu_k}\mu_k \int_0^1 \exp\left\{-(\mu-\mu_k) + \sum_{i\neq k}\mu_i x^{i/k}\right\} e^{\mu_k x} dx =$$
$$= \mu_k \int_0^1 \exp\left\{-\mu + \sum_{i=1}^{\infty}\mu_i x^{i/k}\right\} dx = \mu_k \int_0^1 \exp\left\{-\mu + G(x^{1/k})\right\} dx,$$

which is the statement (23) of the Theorem.

We now prove that the asymptotic relation (24) holds. Using the formula (23) it is enough to verify that the following convergence holds

$$\int_0^1 \exp\left\{-\mu + G(x^{1/k})\right\} dx \to 1, \text{ if } k \to \infty. \tag{26}$$

On the one hand, the generating function $G(x)$ is continuous, monotonically increasing on the interval $[0,1]$ and has the limit value μ from left in the point 1, then $0 \leq \mu - G(x^{1/k}) \leq \mu - G(\varepsilon^{1/k})$, $0 \leq \varepsilon \leq x \leq 1$. On the other hand, for every fixed constant ε, $0 < \varepsilon < 1$ the convergence $\varepsilon^{1/k} \to 1$ holds as $k \to \infty$, then $G(\varepsilon^{1/k}) \to \mu$, $k \to \infty$ and

$$1 \geq \int_0^1 \exp\left\{-\mu + G(x^{1/k})\right\} dx =$$
$$= \int_0^{\varepsilon} \exp\left\{-\mu + G(x^{1/k})\right\} dx + \int_{\varepsilon}^1 \exp\left\{-\mu + G(\varepsilon^{1/k})\right\} dx \geq$$
$$\geq \varepsilon e^{-\mu} + (1-\varepsilon)\exp\left\{-\mu + G(\varepsilon^{1/k})\right\}.$$

Since the constant ε, $0 < \varepsilon < 1$ can be arbitrarily chosen and

$$\varepsilon e^{-\mu} + (1-\varepsilon)\exp\left\{-\mu + G(\varepsilon^{1/k})\right\} \to \varepsilon e^{-\mu} + (1-\varepsilon),\ k \to \infty,$$

then the statement (26) is true, which verifies the asymptotic relation (24) of the Theorem 2. □

Remark. It is worth mentioning that the asymptotic relation $P(B_k) = P(A_k)(1+o(1))$ holds if k tends to infinity, which is a direct consequence of the connections (21) and (24).

Remark. In special cases the formula (23) of the Theorem 2. may be computationally applicable for the numerical investigation of the probabilities $P(B_k)$ depending on k, when the generating function of the sequence μ_k has known form.

For instance, if the sequence μ_k equals the Fisher's logarithmic series (Fisher et al., 1943), which is given by $\mu_k = (\alpha/k)\rho^k$, where μ_k is the expected number of species with k individuals, ρ is a positive number less than 1, and Fisher's α is a positive constant and it is often used as a measure of biodiversity. In this case we have

$$G(z) = \sum_{i=1}^{\infty}(\alpha/k)\rho^k z^{1/k} = -\alpha\log(1-\rho z^{1/k}), \quad \mu = \sum_{i=1}^{\infty}(\alpha/k)\rho^k = -\alpha\log(1-\rho)$$

and consequently

$$P(B_k) = (\alpha/k)\rho^k \int_0^1 \exp\left\{-\alpha(\log(1-\rho)+\log(1-\rho x^{1/k}))\right\} dx =$$

$$= (\alpha/k)\rho^k \int_0^1 \exp\left\{\alpha\log\frac{1-\rho x^{1/k}}{1-\rho}\right\} dx = (\alpha/k)\rho^k \int_0^1 \left(\frac{1-\rho x^{1/k}}{1-\rho}\right)^{\alpha} dx.$$

Another example is the case when the members of the sequence μ_k in the Theorem 2. are proportional to that of a Yule distribution with parameter $\beta > 0$, instead of logarithmic distribution. Let $\mu_k = \alpha\beta\frac{\Gamma(\beta)\Gamma(k)}{\Gamma(k+\beta+1)}$, $k = 1,2,...$ for some $\alpha > 0$. Applying the formula of generating functions of the Yule distributions (see p. 287, Johnson, 2005), then the sequence of probabilities $P(B_k)$ can be formulated as follows

$$P(B_k = \mu_k \int_0^1 \exp\left\{-\alpha + \frac{\alpha\beta}{\beta+1}\, {}_2F_1[1,1;\beta+2;z^{1/k}]z^{1/k}\right\} dx,$$

where ${}_2F_1$ denotes the generalized hypergeometric function.

Conclusions

We have dealt with the model in which a Kendall process describes the evolution of the species after entering the system. The birth and death intensities are assumed to be identical. We have considered inhomogeneous speciation process, for which the speciation rate is bounded, locally integrable and has an asymptotically power type behaviour. This model led (exactly or asymptotically) to Yule abundance distributions instead of a logarithmic one, arising in the homogeneous cases. More precisely, in the inhomogeneous cases the parameters of the models, depending on the speciation rate, are proportional (exactly or asymptotically) to the members of the Yule distribution. This means an enlargement of the class of the possible limit distributions, which can arise for the discrete population dynamical models.

In connection with the sample statistics some results are derived in general and also in special cases (for the logarithmic and Yule distribution), which are related to the random choice of a species or an individual from the whole population of models considered above.

Acknowledgment

The authors are indebted to reviewers for their valuable comments and suggested corrections.

References

[1] Bowler, M.G. and Kelly, C.K. (2012) On the statistical mechanics of species abundance distributions. Theoretical Population Biology 82: 85-91.

[2] Branson, D. (1991) Inhomogeneous birth-death and birth-death-immigration processes and the logarithmic series distribution. Stochastic Processes and their Applications 39, 131-137.

[3] Branson, D. (2000) Inhomogeneous birth-death and birth-death-immigration processes and the logarithmic series distribution, Part 2. Stochastic Processes and their Applications, 86, 183-191.

[4] Davis, Ph.J. (1972) Gamma Functions and Related Functions. In: Handbook of Mathematical Functions, Abramovitz, M. and Stegun, I.A. (Eds.), National Bureau of Standards, USA, Tenth Printing, 253-266.

[5] Engen, S. and Lande, R. (1996a) Population dynamic models generating the lognormal species abundance distribution. Mathematical Biosciences, 132, 169-183.

[6] Engen, S. and Lande, R. (1996b) Population dynamic models generating species abundance distributions of the gamma type. Journal of Theoretical Biology, 178, 325-331.

[7] Fisher, R. A., Corbet, A. S., Williams, C. B., (1943) The relation between the number of species and the number of individuals in a random sample of an animal population. J. Animal Ecology, 12, 42-58.

[8] Johnson, N.L. Kemp, A.W. Kotz, S. (2005) Univariate Discrete Distribution. Third Edition, Wiley & Sohns, New Jersey.

[9] Karlin, S. and McGregor , J. (1967) The number of mutant forms maintained in a population. Proc. Fifth Berkeley Symp. Math. Stat. Probab., 4, 415-438.

[10] Kendall, D.G. (1948a) On some models of population growth leading to R. A. Fisher's logarithmic series distribution. Biometrika, 35, 6-15.

[11] Kendall, D.G. (1948b) On the generalized ”birth-and-death” process. Ann Math Stat 19, 6-15.

[12] Kingman, J.F.C. (1993) Poisson Processes, Clarendon Press, Oxford University.

[13] Lakatos, L., Szeidl, L. and Telek, M. (2013) Introduction to Queueing Systems with Telecommunication Applications, Springer, New York, Heidelberg, Dordrecht, London.

[14] Lange, K. (2010) Applied Probability (Second Edition), Springer, New York, Dordrecht, Heidelberg, London.

[15] Newman, M. E. J. Power laws, Pareto distributions and Zipf's law. arXiv:cond-mat/0412004v3, 29 May, 2006, 1-28.

[16] Prudnikov, A.P., Brychkov, Y.A. and Marichev, O.I. (1998) Integrals and Series, Vol. 1., Elementary Functions. Gordon and Breach Sciences Publishers, New York.

[17] Rolland, J., Condamine, F.L., Jiguet, F., Morlon, H. (2014) Faster speciation and reduced extinction in the tropics contribute to the mammanial latitudinal diversity gradient. PLOS Biology, 12, 1, 1-11.

[18] Simon, H. A. (1955) On a class of skew distribution functions. Biometrika 42, 425–440 .

[19] Watterson, G.A. (1974) Models for the logarithmic species abundance distributions. Theoret. Population Biol. 6, 217-250.

A bounding scheme for proving the Wright conjecture on delay differential equations

Balázs Bánhelyi[1], Tibor Csendes[1*], Tibor Krisztin[2], and Arnold Neumaier[3]

[1] Institute of Informatics, University of Szeged, Árpád tér 2, 6720 Szeged, Hungary, banhelyi@inf.szte.hu, csendes@inf.szte.hu

[2] Bolyai Institute, University of Szeged, Aradi vértanúk tere 1, 6720 Szeged, Hungary, krisztin@math.u-szeged.hu

[3] Department of Mathematics, University of Vienna, A-1090 Vienna, Austria, arnold.neumaier@univie.ac.at

[*] Corresponding author

Abstract: We provide here an elementary derivation of the bounding scheme applied for proving the Wright conjecture on delay differential equations. We also report a minor extension of the parameter range where the conjecture was proven, to $\alpha \in [1.5, 1.57065]$.

Keywords: bounding scheme; delay differential equation; interval arithmetic; validated computation; Wright conjecture

1 Introduction

In this paper we give an elementary derivation of a bounding scheme to prove Wright's conjecture [6] on the delay differential equation

$$\dot{u}(t) = -\alpha u(t-1)[1+u(t)], \quad \alpha > 0. \tag{1}$$

That bounding scheme is then applied in a verified computational algorithm for systematic checking the α values in question. If we consider only those solutions of equation (1) which have values in $(-1,\infty)$, the transformation $x = \log(1+u)$ leads to the equation

$$\dot{x}(t) = f_\alpha(x(t-1)) \tag{2}$$

with $f_\alpha(\xi) = -\alpha(e^\xi - 1)$, $\xi \in \mathbb{R}$. Throughout this paper (2) will also be called Wright's equation.

In [2] we proved

Theorem 1. *If* $\alpha \in [1.5, 1.5706]$*, then the zero solution of equation (2) is globally attractive.*

We used the following statement in the proof:

Theorem 2. *The zero solution of (2) is globally attracting if and only if (2) has no slowly oscillating periodic solution.*

Recall that a solution $x : R \to R$ oscillates slowly if $|z_1 - z_2| > 1$ for any two different zeros of x. In [2] a theoretical proof was given for

Corollary 1. *If* $0 < \alpha < \frac{\pi}{2}$ *and* $p^\alpha : \mathbb{R} \to \mathbb{R}$ *is a slowly oscillating periodic solution of equation (2) then*

$$\max_{t\in\mathbb{R}} p^\alpha(t) \geq \log \frac{\pi}{2\alpha} > 1 - \frac{2\alpha}{\pi}.$$

The computational part of the proof of Theorem 1 proves

Theorem 3. *If* $\alpha \in [1.5, 1.5706]$ *and* $y : \mathbb{R} \to \mathbb{R}$ *is a slowly oscillating periodic solution of (2), then* $\max_{t\in\mathbb{R}} |y(t)| \leq 1 - \frac{2\alpha}{\pi}$.

Now, a combination of Theorem 2, Corollary 1, and Theorem 3 proves Theorem 1.

In an earlier paper [1], the first author investigated the problem with traditional verified differential equation solver algorithms [4, 5]. He found that a proof of the conjecture along these lines would require an enormous amount of computation time with the present technological conditions (compilers, algorithms and computer capacities). He was able to prove only that for all α values within the tiny interval $[1.5, 1.5 + 10^{-22}]$ the trajectories of the solutions will reach a phase when the absolute value of the solution remain below 0.075 for a time interval of a unit length. For wider parameter intervals, or for values closer to $\pi/2$ the required CPU times exploded. Thus traditional computer-assisted techniques involving general, inclusion monotone iterative techniques for differential equations appear not suitable for settling the conjecture.

2 The bounding scheme

Let $p : \mathbb{R} \to \mathbb{R}$ be a nontrivial periodic solution of (2). Set $M = \max_{t\in\mathbb{R}} p(t)$ and $-m = \min_{t\in\mathbb{R}} p(t)$. We skip here the technical details from Wright's paper, and just give the conditions obtained by him:

$$M \leq -\alpha\left(e^{-m} - 1\right) + (-m)\frac{e^{-m}}{e^{-m} - 1} - 1 \quad \text{if } \alpha\left(e^{-m} - 1\right) \leq -m, \tag{3}$$

$$M \leq \alpha - \frac{1 - e^{\alpha\left(e^{-m}-1\right)}}{\left(1 - e^{-m}\right)}, \tag{4}$$

$$m \leq \alpha\left(e^M - 1\right) - M\frac{e^M}{e^M - 1} + 1. \tag{5}$$

The present approach follows another line of thought, still it is a kind of direct extension of that of Wright. Denote three subsequent zeroes of the trajectory by 0, z_1, and z_2. We may assume that $y(t) > 0$ for $t \in (0, z_1)$, and $y(t) < 0$ for $t \in (z_1, z_2)$. Let us define the following functions bounding the trajectories (see Figure 1):

$y_{(inc,1)}^{(upper)}(t)$: an upper bounding function for the time interval $0 \leq t \leq 1$,

$y_{(inc,1)}^{(lower)}(t)$: a lower bounding function for the time interval $0 \leq t \leq 1$,

$y_{(dec,n)}^{(upper)}(t)$: an upper bounding function for the time interval $1 \leq t \leq z_1$,

$y_{(dec,1)}^{(lower)}(t)$: a lower bounding function for the time interval $z_1 \leq t \leq z_1 + 1$,

$y_{(dec,1)}^{(upper)}(t)$: an upper bounding function for the time interval $z_1 \leq t \leq z_1 + 1$,

$y_{(inc,n)}^{(lower)}(t)$: a lower bounding function for the time interval $z_1 + 1 \leq t \leq z_2$.

The trajectory bounding functions are illustrated by dashed lines on Figure 1. Here four consecutive time intervals will be considered defined by the zeros and by the extremal values of the trajectory denoted by $(inc,1), (dec,n), (dec,1)$, and (inc,n), respectively. The length of the time intervals $(inc,1)$ and $(dec,1)$ are known to be one. On the other hand the length of (dec,n), denoted as $p_M = z_1 - 1$ and that of (inc,n), $p_m = z_2 - z_1 - 1$ are unknown, it is even unclear whether these are larger than one.

The trajectory bounding functions will be sharpened sequentially, in an iterative way, i.e. the bounding functions of the time interval $(inc,1)$ will be used to improve the bounding function on the interval (dec,n), etc. Then, the bounding function of the last interval, (inc,n) will be used to make the inequalities for the interval $(inc,1)$ sharper, and so on. Those bounding function improvements that are based on a single bounding function of the earlier time interval are basically similar to the original technique used by Wright. The sharpening steps using two bounding functions on the argument interval apply a new, Taylor series based method to be described later in this paper. At start we set the upper bounding functions to constant M, the lower bounding functions to $-m$ with the exceptions of $y_{(inc,1)}^{(lower)} = 0$ and $y_{(dec,1)}^{(upper)} = 0$.

We iterate only on such cases, when the conditions (3) to (5) and that of Corollary 1 are fulfilled. The conditions we check at the end of each iteration cycle of the

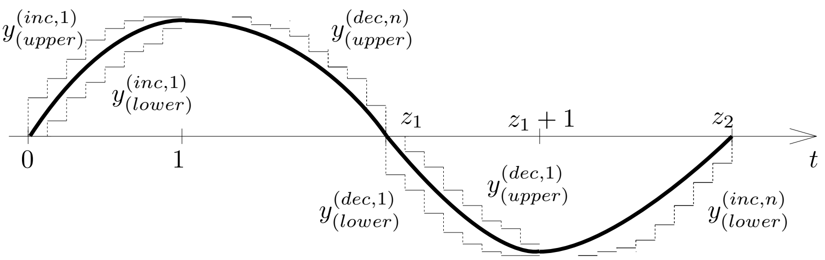


Figure 1
The trajectory bounding functions shown as dashed lines for a full period

bounding function sharpening procedure are

$$y^{(upper)}_{(inc,1)}(0+1) < M \quad \text{and} \quad -m < y^{(lower)}_{(dec,1)}(z_1+1). \tag{6}$$

In case at least one of these conditions are satisfied then the solution of the investigated delay differential equation cannot have a periodic solution with a maximal value of M and the minimal value of m as assumed for the given α parameter.

3 Improved bounds for the unit width intervals

First we show how to obtain an upper bound on the periodic trajectory on the interval $(inc,1)$ based on the $y^{(lower)}_{(inc,n)}(t)$ function. Since $y^{(lower)}_{(inc,n)}(t)$ is a lower bounding function, so $y^{(lower)}_{(inc,n)}(t) \leq y(t)$ holds for all $t \leq 0$. Now integrate y' from 0 to t $(0 \leq t \leq 1)$:

$y(t) = y(t) - y(0) =$

$$-\alpha \int_0^t e^{y(x-1)} - 1\,dx = -\alpha \int_{0-1}^{t-1} e^{y(x)} - 1\,dx \leq -\alpha \int_{0-1}^{t-1} e^{y^{(lower)}_{(inc,n)}(x)} - 1\,dx.$$

We can obtain a new, stronger bounding function from this bound and from the old one for the $t \geq 0$ case:

$$y^{(upper)}_{(inc,1)}(t) = \min \left\{ \begin{array}{c} y^{(upper)}_{(inc,1)}(t) \\ \\ -\alpha \int_{0-1}^{t-1} e^{y^{(lower)}_{(inc,n)}(x)} - 1\,dx \end{array} \right\},\ t \in [0,1]. \tag{7}$$

We suppress the iteration number in the bounding function, the new one on the left hand side of the defining equation is calculated from the old function on the right hand side as it is usual in computer programs. We can get a new bounding function for the lower bounding function in $(dec,1)$ in a similar way:

$$y^{(lower)}_{(dec,1)}(t) = \max \left\{ \begin{array}{c} y^{(lower)}_{(dec,1)}(t) \\ \\ -\alpha \int\limits_{z_1-1}^{t-1} e^{y^{(upper)}_{(dec,n)}(x)} - 1\, dx \end{array} \right\}, \ t \in [z_1, z_1+1]. \tag{8}$$

We can obtain an improved lower bound for the trajectory on the interval $(inc,1)$ by $y(1)-y(t) = M - y(t) =$

$$-\alpha \int\limits_{t}^{1} e^{y(x-1)} - 1\, dx = -\alpha \int\limits_{t-1}^{0} e^{y(x)} - 1\, dx \le -\alpha \int\limits_{t-1}^{0} e^{y^{(lower)}_{(inc,n)}(x)} - 1\, dx.$$

The new lower bounding function is then

$$y^{(lower)}_{(inc,1)}(t) = \max \left\{ \begin{array}{c} y^{(lower)}_{(inc,1)}(t) \\ \\ M + \alpha \int\limits_{t-1}^{0} e^{y^{(lower)}_{(inc,n)}(x)} - 1\, dx \end{array} \right\} \text{ if } t \in [0,1]. \tag{9}$$

We can build an improved upper bound also for the time interval $(dec,1)$ in a similar way:

$$y^{(upper)}_{(dec,1)}(t) = \min \left\{ \begin{array}{c} y^{(upper)}_{(dec,1)}(t) \\ \\ -m + \alpha \int\limits_{t-1}^{0} e^{y^{(upper)}_{(dec,n)}(x)} - 1\, dx \end{array} \right\} \text{ if } t \in [0,1]. \tag{10}$$

By that we have completed the description of the improved bounding functions for the unit width time intervals.

4 Bounds for the period length

A sharp enclosure of the period length is very important for the success of the proof for the conjecture, especially for α values close to $\pi/2$. To calculate bounds on the period length and as a part of that bounds for the not unit length time intervals we apply an Euler type differential equation solution method

$$Y(x) = Y(x_0) + Y^{(1)}([x_0,x])(x - x_0),$$

$$Y([x_0,x]) = Y(x_0) + Y^{(1)}([x_0,x])([0, x - x_0])$$

customized for delay equations. In these equations we used the notions of interval calculations [5], i.e. capitals denote interval values. The implementation details will

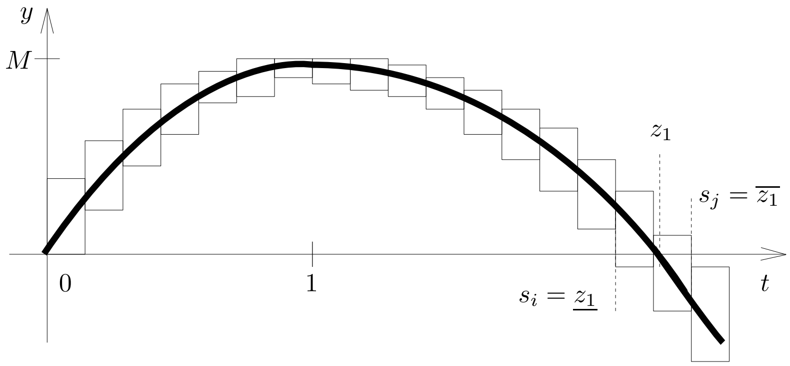


Figure 2
Illustration of the bounding procedure for the z_1 zero of the trajectory

be discussed in the next section. To use this method we need an enclosure $Y(x_0)$ of the trajectory in the start point, and bounds on a given number of time intervals covering together unit length time intervals.

For these calculations we need lower and upper bounds for the trajectory on the unit length time intervals before the investigated (dec,n) and (inc,n) phases. These are available due to the previous subsection. The lower and upper bounds for the zeros z_1 and z_2 of the trajectory will be determined using the interval enclosures obtained on time intervals for the trajectory. Consider first the case when we follow the trajectory from 1 to find z_1, i.e. we want to find bounds for p_M. Assume that as a part of the verified integration the first interval that contains zero is $Y(t_i,t_i+h)$, where h is the step size of the numerical integration. Then there may follow some integration steps for which the respective Y enclosures contain zero. Let the last such be $Y(t_j,t_j+h)$ (in some cases it is possible that $i=j$). Then $[t_i,t_j+h]$ is obviously a verified enclosing interval for z_1. The same technique that is illustrated on Figure 2 is also applicable for the bounding of p_m.

Denote the enclosures of p_M and p_m to be calculated from the above bounds of the zeros by P_M and P_m, respectively. The lower and upper bounds of these intervals are denoted as usual in interval calculation, with underline and overline, e.g. $P_M = [\underline{P}_M, \overline{P}_M]$.

5 Improved bounds for the not unit width intervals

As we could see in the previous subsection, it is not easy to determine z_1, as the zero of the investigated trajectory. In the present subsection we build a valid upper bound for the trajectory on the intervals $(inc,1)$ and (dec,n) that can be applied as needed also until the point z_1 for calculating further improving bounds on the interval $(dec,1)$.

Consider the trajectory on $[0,1+\overline{P}_M]$, i.e. on the intervals $(inc,1)$ and (dec,n). The bounds on the trajectory are at this point obtained by the new bounds of (9) and (10) on $(inc,1)$, and by the verified solution of the differential equation, as described in Section 4 on (dec,n). Let us call this complete bounding function as Y, and its upper bound as $\overline{Y}$. For a monotonically increasing $y(t)$ function we have

$$y(t) \geq y(t-\Delta t) \text{ if } \Delta t \geq 0$$

and for a monotonically decreasing $y(t)$ function

$$y(t) \geq y(t-\Delta t) \text{ if } \Delta t \leq 0.$$

The trajectory is known to be strictly monotonically increasing on $(inc,1)$, while strictly monotonically decreasing on (dec,n).

Consider first the $(inc,1)$ time interval, here the $y^{(upper)}_{(inc,1)}$ gives an upper bounding function, $\overline{Y}$ for the periodic trajectory. Since $p_M \leq \overline{P}_M$, the relation

$$\Delta t = \left(1+\overline{P}_M\right) - z_1 = \overline{P}_M - p_M \geq 0$$

holds. Now these imply

$$\overline{Y}(t) \geq y(t) \geq y(t-\Delta t) = y\left(t - \left(\left(1+\overline{P}_M\right) - z_1\right)\right).$$

These relations can be interpreted as $\overline{Y}$ is an upper bounding function also for $y(t-\Delta t)$, i.e. for the trajectory shifted by Δt on the interval

$$\left[-\left(\left(1+\overline{P}_M\right)-z_1\right), 1-\left(\left(1+\overline{P}_M\right)-z_1\right)\right] =$$

$$\left[-(\overline{P}_M - p_M), 1-(\overline{P}_M - p_M)\right] = \left[z_1 - \overline{P}_M - 1, z_1 - \overline{P}_M\right].$$

Consider now the (dec,n) phase, the verified solution will give an upper bound for $y(t)$ on $[1,1+\underline{P}_M]$. Here $y(t)$ is strictly monotonically decreasing, thus due to $\underline{P}_M \leq p_M$ the relations

$$\overline{Y}(t) \geq y(t) \geq y(t-\Delta t) = y\left(t-\left(\left(1+\underline{P}_M\right)-z_1\right)\right)$$

hold with $\Delta t = \underline{P}_M - p_M \leq 0$. Here again $\overline{Y}$ is an upper bounding function also for $y(t-\Delta t)$, i.e. for the trajectory shifted by Δt on the interval

$$\left[1-\left(\underline{P}_M - p_M\right), 1+\underline{P}_M - \left(\underline{P}_M - p_M\right)\right] =$$

$$\left[z_1 - \underline{P}_M, z_1\right].$$

The explanation for the above bounding technique is illustrated on Figure 3. The first case can be understood as if the original periodic solution would be shifted in such a way that the original z_1 zero coincides with $1+\overline{P}_M$. Since $y(t)$ is monotonically increasing on the interval $(inc,1)$, thus the upper bounding function $\overline{Y}(t)$ remains an upper bound of the shifted function too (upper picture of Figure 3). The highlighted upper bounding functions parts are presented as bounds of the $y(t)$ trajectory.

In the second case the original trajectory is shifted in such a way that the zero z_1 coincides with $(1+\underline{P}_M)$. The monotonically decreasing $y(t)$ will then remain below $\overline{Y}(t)$ on the given time interval (see the second picture of Figure 3). As it can be seen on this figure, in the gap between the two highlighted function we consider the constant M value. With the above considerations we have provided a bounding function that can be used also until the unknown z_1 time point.

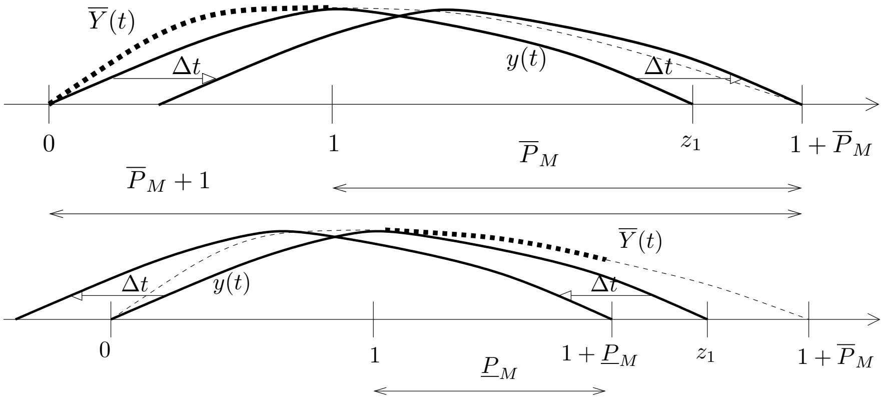


Figure 3
Illustrations of how the bounds can be obtained for the cases when the shifted z_1 coincides with $1+\overline{P}_M$ and with $1+\underline{P}_M$, respectively

The same technique can be applied to establish such a valid lower bound for the trajectory on the intervals $(dec,1)$ and (inc,n), that can be applied for further bound improvements even in the case when the necessary integration should start from the z_2 zero.

Let us see now how can we produce stronger bounds on the intervals (dec,n) and (inc,n) before the z_1-1, and z_2-1 time points, respectively – on the basis of the bounds discussed earlier in the present subsection. Consider first the (dec,n) case, then for the present upper bounding function

$$y^{(upper)}_{(dec,n)} \geq y(t).$$

Integrate the derivative function y' from t to z_1, where $z_1-1 \leq t \leq z_1$:

$$-y(t) = y(z_1) - y(t) = -\alpha \int_{t}^{z_1} e^{y(x-1)} - 1\, dx = -\alpha \int_{t-1}^{z_1-1} e^{y(x)} - 1\, dx.$$

In other terms

$$y(t) \leq \alpha \int_{t-1}^{z_1-1} e^{y^{(upper)}_{(dec,n)}(x)} - 1\, dx.$$

This bounding function can be use to update the old one:

$$y_{(dec,n)}^{(upper)}(t) = \min \left\{ \begin{array}{c} y_{(dec,n)}^{(upper)}(t) \\ \\ \alpha \int\limits_{t-1}^{z_1-1} e^{y_{(dec,n)}^{(upper)}(x)} - 1 \, dx \end{array} \right\} \text{ if } t \in [z_1 - 1, z_1]. \tag{11}$$

In a similar way we can calculate a new lower bounding function on the interval (inc, n):

$$y(t) \geq \alpha \int\limits_{t-1}^{z_2-1} e^{y_{(inc,n)}^{(lower)}} - 1 \, dx,$$

that implies the update

$$y_{(inc,n)}^{(lower)}(t) = \min \left\{ \begin{array}{c} y_{(inc,n)}^{(lower)}(t) \\ \\ \alpha \int\limits_{t-1}^{z_2-1} e^{y_{(inc,n)}^{(lower)}(x)} - 1 \, dx \end{array} \right\} \text{ if } t \in [z_2 - 1, z_2]. \tag{12}$$

Notice that in both cases the new, improved bound utilizes earlier bound values also from more than 1 time unit distance to the actual right end zero of the trajectory. This gives an explanation how improvements made at the first part of the present subsection can improve our bounds at a much later time point.

6 The iterative improvement of the bounding functions

The lower and upper bounds derived in the earlier subsections will be applied in an iterative procedure to make them even sharper that possibly allows to conclude that for a given pair of M and m values the delay differential equation (1) with the investigated interval of α parameter leads to a contradiction. The iteration cycle begins with the time interval $(inc, 1)$, and with the integration of the right hand side of the differential equation we update the earlier upper bound on (dec, n). This new upper bound will then be used to improve the lower and upper bounding functions on $(dec, 1)$, and finally the latter help us to make $y_{(inc,n)}^{(lower)}$ sharper.

Now the bounding functions $y_{(inc,1)}^{(lower)}$, $y_{(dec,1)}^{(upper)}$, $y_{(inc,1)}^{(upper)}$, and $y_{(dec,1)}^{(lower)}$ are defined on unit length time intervals, on $[0, 1]$ and $[z_1, z_1 + 1]$, respectively. In contrast to these, in the case of $y_{(inc,n)}^{(lower)}$ and $y_{(dec,n)}^{(upper)}$ we must also calculate with their values over wider time intervals. To be able to handle the delayed terms, we have to save bounding function values for a unit length interval in the first case, and for two width intervals otherwise (this later figure proved to be satisfactory for our investigation).

Due to the computer representation of reals, it is advantageous to subdivide these time intervals into 2^l, and 2^{l+1} subintervals for a natural number l, respectively.

Denote these subintervals by t_i, where $i \in (1,\ldots,2^l)$, and for the (dec,n) and (inc,n) time intervals $i \in (1,\ldots,2^{l+1})$ in increasing order as they depart from the zero. It is intentional that the order of the numeration for the unit length intervals is the opposite of that for (dec,n) and (inc,n). Within such a subinterval, the respective bounding function will be represented by a real number, i.e. we use a bounding step function for the saved bounding functions. This step function is denoted by Y, as also in Section 4. The right hand side of the differential equation can then easily be bounded using the step functions both at t_j and at the same time at $t_j - 1$. The updated value of $Y^{(upper)}_{(inc,1)}(t_i)$ $(i = 1,\ldots,2^l)$ can be calculated applying $Y^{(lower)}_{(inc,n)}$ according to (7):

$$Y^{(upper)}_{(inc,1)}(t_i) = \min\left\{-\alpha\sum_{j=1}^{i}\left(e^{Y^{(lower)}_{(inc,n)}(t_{2^l-j+1})}-1\right)/2^l \;;\; Y^{(upper)}_{(inc,1)}(t_i)\right\}. \tag{13}$$

In a similar way, we can obtain the other bounding functions updated using the stronger bounds given as (8) to (10):

$$\begin{aligned}
Y^{(lower)}_{(dec,1)}(t_i) &= \max\left\{-\alpha\sum_{j=1}^{i}\left(e^{Y^{(upper)}_{(dec,n)}(t_{2^l-j+1})}-1\right)/2^l \;;\; Y^{(lower)}_{(dec,1)}(t_i)\right\}, & (14)\\
Y^{(lower)}_{(inc,1)}(t_i) &= \max\left\{M+\alpha\sum_{j=i}^{2^l}\left(e^{Y^{(lower)}_{(inc,n)}(t_{2^l-j+1})}-1\right)/2^l \;;\; Y^{(lower)}_{(inc,1)}(t_i)\right\}, & (15)\\
Y^{(upper)}_{(dec,1)}(t_i) &= \min\left\{-m+\alpha\sum_{j=i}^{2^l}\left(e^{Y^{(upper)}_{(dec,n)}(t_{2^l-j+1})}-1\right)/2^l \;;\; Y^{(upper)}_{(dec,1)}(t_i)\right\}. & (16)
\end{aligned}$$

On the basis of these bounding functions, we can calculate bounds on the trajectory for the next, not unit length time intervals. The bounds on the trajectory will provide lower and upper bounds on the next zero, as discussed in Section 4. Thus we obtain lower and upper bounds on the trajectory on the time intervals $[0,1+\underline{P}_M]$, and $[0,1+\overline{P}_M]$, respectively. The formal description of the algorithm for the determination of the bounds of zeros is given as Algorithm 1. Here we bound the trajectory after the time 1, or z_1+1, and check whether the respective $Y(t_j)$ interval contains zero. The algorithm is able to identify lower and upper bounds within length 2 intervals, this was satisfactory for our investigation. The reordering of the 2^{-l} size subintervals mentioned in Section 7 must be made after Algorithm 1 was run.

Consider now how these bounding functions can be used to improve $y^{(upper)}_{(dec,n)}$. The integration of the step function $Y(t_i)$, $i \in \left(1,\ldots,2^l\right)$ gives with (11) and (12) the updated upper and lower bounding functions

$$Y^{(upper)}_{(dec,n)}(t_i) = \max\left\{\alpha\sum_{j=i}^{2^l}\left(e^{Y^{(upper)}_{(dec,n)}(t_{j-2^l})}-1\right)/2^l \;;\; Y^{(upper)}_{(dec,n)}(t_i)\right\}, \tag{17}$$

Algorithm 1 Determination of $\underline{P}_M$ and $\overline{P}_M$ for the bounds for the period length

```
Input:   – s: $M$ or $-m$ as an extremal value of the periodic trajectory,
         – $\alpha$: a parameter of the studied delay differential equation,
         – $2^l$: the number of equal width subintervals in the unit length time
           interval,
         – $L, U$: lower and upper bound functions on the unit length time interval.
Output:  – An enclosure of the length for the not unit width interval,
           bounding of the trajectory from 1 and $z_1 + 1$, respectively.

Step 1. Compute $Y(t_i)$ $(i = 1, \dots, 2^l)$ as the enclosures of the periodic solution on
        subintervals of the unit length time period by using the $U$ and $L$ functions on
        the $(inc, 1)$ and $(dec, 1)$ intervals.
Step 2. Set $j = (2^l + 1)$ and $Y_{last} = [s, s]$.
Step 3. Enclose $Y(t_j)$ with the expression $\left(Y_{last} + \left(-\alpha\left(e^{Y(t_{j-2^l})} - 1\right)\right) \cdot [0, 1/2^l]\right)$.
Step 4. Set $Y_{last} = Y_{last} + \left(-\alpha\left(e^{Y(t_{j-2^l})} - 1\right)\right)/2^l$.
Step 5. If $0 \notin Y(t_{j-1})$ and $0 \in Y(t_j)$, then calculate the new lower bound for the
        length of the not unit width interval: $\underline{P}_M = (j-1)/2^l$.
Step 6. If $0 \in Y(t_{j-1})$ and $0 \notin Y(t_j)$, then calculate the new upper bound for the
        length of the not unit width interval: $\overline{P}_M = (j-1)/2^l$ and STOP.
Step 7. Set $j = j + 1$.
Step 8. If $j < 2^{l+2}$, then continue with Step 3, otherwise STOP.
```

and

$$Y^{(lower)}_{(inc,n)}(t_i) = \min\left\{\alpha\sum_{j=i}^{2^l}\left(e^{Y^{(lower)}_{(inc,n)}(t_{j-2^l})} - 1\right)/2^l\ ;\ Y^{(lower)}_{(inc,n)}(t_i)\right\}. \tag{18}$$

This completes the description of the iterative procedure to improve bounding functions on the periodic solutions of the delay differential equation (1). The periodic solution should reach at the time point 1 the maximal value of M, while at the end of $(dec, 1)$ the value $-m$. We can use this fact as a condition to be checked, whether to the given M, m pair a periodic solution belongs for the actual α differential equation parameter. The corresponding inequalities are (cf. (6)):

$$Y^{(upper)}_{(inc,1)}(t_{2^n}) \geq M \quad \text{and} \quad Y^{(lower)}_{(dec,1)}(t_{2^n}) \leq -m.$$

The checking algorithm is also able to decide on these conditions when the M values are given as intervals. To exclude such possible intervals of M we apply the above conditions for the upper bounds of the respective intervals:

$$Y^{(upper)}_{(inc,1)}(t_{2^n}) < \underline{M}. \tag{19}$$

By this condition we can delete all points of the respective subintervals.

7 Extension of the parameter range

In [2] the Wright conjecture was proven for α values between 1.5 and 1.5706. We continued the computational part of the proof with unchanged theoretical background. The computational environment was a blade server with 12 cores and 24 threads, we set the algorithm parameters in the same way for all checked new subintervals. In this way, the computation times in Table 1 reflect well the necessary increasing computational complexity.

Table 1
The CPU time requirements of the proven α intervals.

Interval	CPU time in hours
$[1.57060, 1.57061]$	56.9
$[1.57061, 1.57062]$	64.9
$[1.57062, 1.57063]$	83.7
$[1.57063, 1.57064]$	119.4
$[1.57064, 1.57065]$	141.2

Seeing the data in Table 1 we can draw the conclusion that the necessary computation times for proving new subintervals with unchanged algorithm parameters grows in a highly nonlinear way. That confirms our earlier conclusion drawn in [2] that additional theoretical insight should be utilized to achieve a substantial progress in the proven α values. After submitting our manuscript, J. Bouwe van den Berg and J. Jaquette published their theoretical proof on the remaining part of Wright's conjecture [3], that was based on our earlier computational result [2]. It confirms indirectly, that our bounding scheme approach is justified for the larger part of the α parameter interval in the conjecture.

Acknowledgement. The research work was partially supported by the EFOP-3.6.3-VEKOP-16-2017-00002 project ”Integrated program for young research associates in the fields of informatics and computer science”. The authors are grateful for the coding and testing support given by Zsolt Bagóczki.

References

[1] B. Bánhelyi, *The investigation of a delay differential equation by a verified computer procedure* (in Hungarian), Alkalmazott Matematikai Lapok 24(2007) 131–150.

[2] B. Bánhelyi, T. Csendes, T. Krisztin, and A. Neumaier, *Global attractivity of the zero solution for Wright's equation*, SIAM J. on Applied Dynamical Systems 13(2014) 537-563.

[3] J. Bouwe van den Berg and J. Jaquette, A proof of Wright's conjecture. arXiv:1704.00029.

[4] N.S. NEDIALKOV, K.R. JACKSON, AND G.F. CORLISS, *Validated Solutions of Initial Value Problems for Ordinary Differential Equations*, Applied Mathematics and Computation, 105(1999) 21-68.

[5] H. RATSCHEK AND J. ROKNE, *Computer Methods for the Range of Functions.* Ellis Horwood, Chichester, 1984.

[6] E.M. WRIGHT, *A non-linear difference-differential equation*, J. für die Reine und Angewandte Mathematik 194 (1955) 66–87.

An always convergent algorithm for global minimization of multivariable continuous functions

J. Abaffy, A. Galántai

Óbuda University
John von Neumann Faculty of Informatics
1034 Budapest, Bécsi u. 96/b, Hungary
abaffy.jozsef@nik.uni-obuda.hu
galantai.aurel@nik.uni-obuda.hu

Abstract: We develop and test a Bolzano or bisection type global optimization algorithm for continuous real functions over a rectangle. The suggested method combines the branch and bound technique with an always convergent solver of underdetermined nonlinear equations. The numerical testing of the algorithm is discussed in detail.

Keywords: global optimum, nonlinear equation, always convergent method, Newton method, branch and bound algorithms, Lipschitz functions

1 Introduction

In this paper we study the minimization problem

$$f(x) \to \min \quad \left(f: \mathbb{R}^n \to \mathbb{R},\ x \in X = \times_{i=1}^{n} [l_i, u_i]\right) \tag{1}$$

with $f \in C(X)$, and develop a method to find its global minimum. Assume that

$$[x_{sol}, iflag] = \texttt{equation\_solve}(f, c) \tag{2}$$

denotes a solution algorithm for the single multivariate equation

$$f(x) = c \quad (x \in X) \tag{3}$$

such that $iflag = 1$, if a true solution $x_{sol} \in X$ exists (that is $f(x_{sol}) = c$), and $iflag = -1$, otherwise.

Let $f_{\min} = \min\{f(x) \,|x \in X\}$ be the global minimum of f, and let $b_1 \in \mathbb{R}$ any lower bound of f such that $f_{\min} \geq b_1$. Let $z_0 \in D_f$ be any initial approximation to the global minimum point ($f(z_0) \geq b_1$). The suggested algorithm then takes the form:

```
Data: $a_1 = f(z_1)$, $b_1$, $i = 1$
while $a_i - b_i > tol$ do
    $c_i = (a_i + b_i)/2$
    $[\xi, iflag] = \texttt{equation\_solve}(f, c_i)$;
    if $iflag = 1$ then
        $z_{i+1} = \xi$, $a_{i+1} = f(\xi)$, $b_{i+1} = b_i$;
    else
        $z_{i+1} = z_i$, $a_{i+1} = a_i$, $b_{i+1} = c_i$;
    end
    $i = i + 1$
end
```

Algorithm 1.

Using the idea of Algorithm 1 we can also determine a lower bound of f, if such a bound is not known a priori (see later or [1]). Algorithm 1 has certain conceptual similarities with the bisection algorithms of Shary [30], [31] and Wood [40], [41].

Theorem 1. *Assume that $f : \mathbb{R}^n \to \mathbb{R}$ is continuous and bounded from below by b_1. Then Algorithm 1 is globally convergent in the sense that $f(z_i) \to f_{\min}$.*

Proof. At the start we have z_1 and the lower bound b_1 such that $f(z_1) \geq b_1$. Then we take the midpoint of this interval, i.e. $c_1 = (f(z_1) + b_1)/2$. If a solution ξ exists such that $f(\xi) = c_1$ $(iflag = 1)$, then $c_1 = f(z_2) \geq f_{\min} \geq b_1$ holds by the initial assumptions. If there is no solution of $f(\xi) = c_1$ (i.e. $iflag = -1$), then $c_1 < f_{\min}$. By continuing this way we always halve the inclusion interval $(b_i, f(z_i))$ for $f_{\min}$. Hence the method is convergent in the sense that $f(z_i) \to f_{\min}$. ■

Note that sequence $\{z_i\}$ is not necessarily convergent.

The performance of Algorithm 1 clearly depends on the equation solver, which for $n > 1$, has to solve a sequence of underdetermined equations of the form (3).

In paper [1] we tested a version of Algorithm 1 that used a locally convergent nonlinear Kaczmarz method [38], [23], [24], [22] and a local minimizer for acceleration as well. The algorithm showed fast convergence in most of the test problems, but in some cases it also showed numerical instability, when $\|\nabla f(z_k)\|$ was close to zero. This and later experiments indicated that only ”globally convergent” and gradient free solvers are useful in the above scheme at the price of loosing speed.

Hence in [2], for one dimensional Lipschitz functions, we developed and successfully tested a version of Algorithm 1 that is based on an always convergent iteration method of Szabó [36], [37].

Here we investigate two versions of Algorithm 1 that use an always convergent iteration method (Galántai [14]) for solving equations of the form (3). This solver is based on continuous space-filling curves lying in the rectangle X and it has a kind of monotone convergence to the nearest zero on the given curve, if it exists, or the iterations leave the region in a finite number of steps.

Definition 1. *Let $r : [0,1] \to [0,1]^n$ $(n \geq 2)$ be a continuous mapping. The curve*

$r = r(t)$ *(*$t \in [0,1]$*) is space-filling if* r *is surjective.*

Given a space-filling curve $r : [0,1] \to [0,1]^n$ and the rectangle $X = \times_{i=1}^{n} [l_i, u_i]$, the mapping

$$h_i(t) = (u_i - l_i) r_i(t) + l_i, \quad i = 1, \ldots, n$$

clearly fills up the whole rectangle X.

The use of space-filling curves in optimization was first suggested by Butz [5], [6], and later by Strongin and others (see, e.g. [34], [35], [32]).

These methods reduce problem (1) to the one dimensional problem

$$f(h(t)) \to \min \quad (t \in [0,1])$$

using mainly the Hilbert space filling function and one dimensional global minimizers. We note that Butz [8] suggested the use of Hilbert's space-filling functions for solving nonlinear systems as well (see also [14]). However these dimension reduction type minimization methods are criticized by various authors pointing out the limited use, speed and other matters (see, e.g. Törn and Zilinskas [39] or Pintér [28]). Using complexity results of Nemirovksy and Yudin [27] Goertzel [16] argues in favour of such methods if f is Lipschitz. For the global minimization of Lipschitz functions, see, e.g. Hansen, Jaumard, Lu [18], [19], [20], [21] and Pintér [28].

Our aim here is only to assess the feasibility and reliability of Algorithm 1 using space-filling based equation solvers, which seems to be a new approach.

Instead of space-filling curves we can also use α-dense curves introduced by Cherruault and Guillez (see, e.g. [9], [17] or [10]).

Definition 2. *Let* $I = [a,b] \subset \mathbb{R}$ *be an interval and* $X = \times_{i=1}^{n} [l_i, u_i] \subset \mathbb{R}^n$ *be a rectangle. The map* $x : I \to X$ *is an* α*-dense curve, if for every* $x \in X$*, there exists a* $t \in I$ *such that* $\|x(t) - x\| \leq \alpha$*.*

The α-dense curves are not space-filling functions. Note that the practical approximations of space-filling curves are also α-dense curves for some α. For 2D, the kth approximating polygon of the Hilbert curve is α-dense with $\alpha \leq \sqrt{2}/2^{2k}$ (see, e.g. Sagan [29]). Recently Mora [25] characterized the connection of space-filling and α-dense curves.

In the rest of the paper we define the class of always convergent methods for solving nonlinear equations in Section 2. Details and the results of numerical testing will be given in Section 3. The numerical testing was performed on a set of 2D Lipschitz continuous problems.

We close the paper with conclusions and the appendix of test problems.

2 Always convergent methods for nonlinear equations

Consider nonlinear equations of the form

$$f(x) = 0 \quad (f : \mathbb{R}^n \to \mathbb{R}^m, x \in X = \times_{i=1}^{n} [l_i, u_i]), \tag{4}$$

where f is continuous on the rectangle X.

Assume that a continuous curve $\Gamma = \{r(t) : 0 \le t \le 1\} \subset X$ is given. We seek for the solution of $f(x) = 0$ on the curve Γ, that is the solution of equation

$$f(r(t)) = 0 \quad (t \in [0,1]), \tag{5}$$

which is equivalent to the real equation

$$\|f(r(t))\| = 0 \quad (t \in [0,1]). \tag{6}$$

Theorem 2. *(Galántai [14]). Assume that $f : \mathbb{R}^n \to \mathbb{R}^m$ is continuous on the rectangle $X = \times_{i=1}^{n} [l_i, u_i]$ and $\Gamma = \{r(t) : 0 \le t \le 1\} \subset X$ is a continuous curve. Let ω_f and ω_r be the modulus of continuity of f on X and Γ on $[0,1]$, respectively. Assume that $\rho_f, \rho_r : [0,\infty) \to [0,\infty)$ are continuous and strictly monotone increasing functions so that*

$$\rho_f(0) = 0, \quad \rho_f(\delta) \ge \omega_f(\delta) \quad (\delta \in [0, diam(X)]), \quad \lim_{\delta \to \infty} \rho_f(\delta) = \infty \tag{7}$$

and

$$\rho_r(0) = 0, \quad \rho_r(\delta) \ge \omega_r(\delta) \quad (\delta \in [0,\tau]), \quad \lim_{\delta \to \infty} \rho_r(\delta) = \infty \tag{8}$$

hold, respectively. Furthermore assume that
(a) $F(x,y)$ is continuous in $[0,1] \times [0,\infty)$;
(b) $x \ge 0$, $F(x,y) = x \Leftrightarrow y = 0$;
(c) $F(x,y) < x$ $(x \in [0,1]$, $y > 0)$;
(d) For $x > \xi$ $(x, \xi \in [0,1])$ and $0 \le y \le x - \xi$, $F(x,y) \ge \xi$.
(e) $F(x,y)$ is strictly monotone increasing in x, and strictly monotone decreasing in y;
Define $\varphi(t) = \rho_r^{-1}\left(\rho_f^{-1}(\|f(r(t))\|)\right)$ $(t \in [0,1])$. Let $t_0 = 1$ and assume that $\varphi(1) > 0$. Define

$$t_{i+1} = F(t_i, \varphi(t_i)) \quad (i = 0,1,2,\ldots). \tag{9}$$

Then $\{t_i\}$ is a strictly monotone decreasing sequence that converges to $\xi_{\max}$ if a root ξ of $\|f(r(t))\| = 0$ exists in $[0,1]$. If no root exists, then the sequence $\{t_i\}$ leaves the interval $[0,1]$ in a finite number of steps.

For the proof of theorem, see [14] or [15]. If Γ is a space-filling curve, then the method clearly always convergent in the sense that it either converges to a solution (if exists) or it leaves the region in a finite number of iterations (if no solution exist). If one selects a curve Γ that is not space-filling, the algorithm may fail to find a zero. Note however that the space-filling functions used in practice are only approximations to the true ones.

A function f is said to be Lipschitz β $(0 < \beta \le 1$) with the Lipschitz constant L, that is $f \in \mathrm{Lip}_L \beta$, if

$$\|f(x) - f(y)\| \le L \|x - y\|^{\beta} \quad \left(x, y \in D_f\right). \tag{10}$$

Assume that $f \in \mathrm{Lip}_{L_f}\beta$ $(0 < \beta \le 1)$. Then $\omega_f(\delta) \le L_f\delta^\beta$ and we can select $\rho_f(\delta) = L_f\delta^\beta$ and $\rho_f^{-1}(\delta) = \left(\frac{\delta}{L_f}\right)^{1/\beta}$. Similarly, if curve Γ is $\mathrm{Lip}_{L_\Gamma}\mu$ $(\mu \in (0,1])$, that is

$$\|r(s) - r(t)\| \le L_\Gamma |s-t|^\mu \quad (t,s \in [0,\tau]), \tag{11}$$

then $\omega_r(\delta) \le L_\Gamma\delta^\mu$ and so we can take $\rho_r(\delta) = L_\Gamma\delta^\mu$ and $\rho_r^{-1}(\delta) = \left(\frac{\delta}{L_\Gamma}\right)^{1/\mu}$. Thus

$$\varphi(t) = \rho_r^{-1}\rho_f^{-1}(\|f(r(t))\|) = \frac{1}{L_\Gamma^{\frac{1}{\mu}}}\left(\frac{\|f(r(t))\|}{L_f}\right)^{\frac{1}{\mu\beta}}. \tag{12}$$

Based upon the numerical testing [14] we select $F(x,y) = x - y$, and the method

$$t_{i+1} = t_i - \varphi(t_i) \quad (i = 0,1,\ldots). \tag{13}$$

Here we use the Hilbert space filling curve (see, e.g. [33], Butz [5], [7], [29], [3], [35], [32]).

Lemma 1. *The Hilbert mapping $r_H : [0,1] \to [0,1]^n$ is space-filling, nowhere differentiable and $Lip_K\mu$ with $L_\Gamma = 2\sqrt{n+3}$ and $\mu = 1/n$:*

$$\|r_H(s) - r_H(t)\| \le L_\Gamma |s-t|^{1/n} \quad (s,t \in [0,1]). \tag{14}$$

For a proof, see, e.g. [42]. For $n = 2$, the Lipschitz constant $L_\Gamma = 2\sqrt{5}$ can be replaced by the sharper value $L_\Gamma = \sqrt{6}$ (Bauman [4]). The following figure shows the recursive kth approximation of the Hilbert curve for $k = 6$.

Similarly to space-filling functions there are many α-dense curves (see, e.g. [10]). Here we use the α-dense curve of Cherruault [10] given by

$$x_i(t) = \frac{1}{2}(1 - \cos(\omega_i 2\pi t)), \quad i = 1,\ldots,n \tag{15}$$

with $\omega_i = \sigma^i$ (for reasons, see [14]). For $n = 2$ and $\sigma = 1000$, $\alpha \approx 0.0044$. This curve is smooth $(\mu = 1)$ unlike the Hilbert curve, but it has a huge Lipschitz constant (see, e.g. [14]). The following figure shows the Cherruault curve for $\sigma = 100$ in 777 points.

3 The numerical experiments

We tested two algorithms. Namely, Algorithm 1 with given lower estimates for the global minimum and the following modification of Algorithm1 that constructs a lower bound for $f_{\min}$.

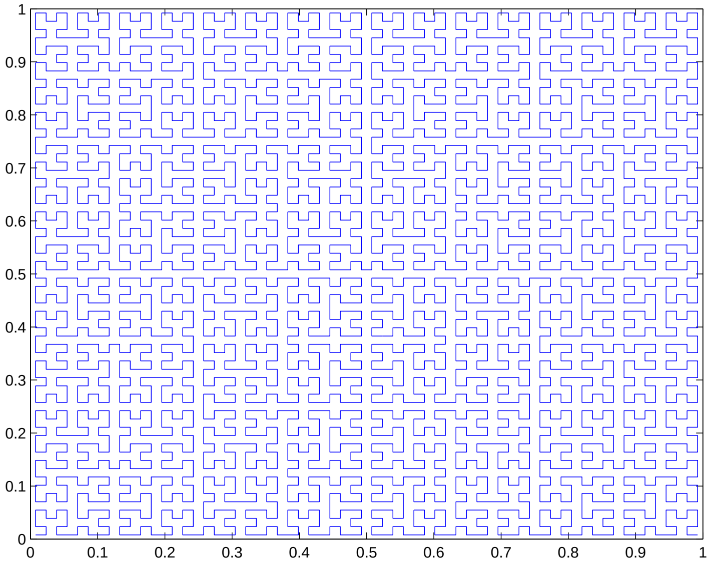


Figure 1
Hilbert curve approximation for $k = 6$.

```
Data: $a_1 = f(z_1)$, $iflag = 1$, $d = 1$, $c = a_1 - d$
while $iflag = 1$ do
    $[\xi, iflag] = \texttt{equation\_solve}(f, c)$;
    if $iflag = 1$ then
        $a_1 = f(\xi)$, $z_1 = \xi$, $d = 2d$, $c = a_1 - d$;
    else
        $b_1 = c$;
    end
end
Data: $i = 1$
while $a_i - b_i > tol$ do
    $c_i = (a_i + b_i)/2$
    $[\xi, iflag] = \texttt{equation\_solve}(f, c_i)$;
    if $iflag = 1$ then
        $z_{i+1} = \xi$, $a_{i+1} = f(\xi)$, $b_{i+1} = b_i$;
    else
        $z_{i+1} = z_i$, $a_{i+1} = a_i$, $b_{i+1} = c_i$;
    end
    $i = i + 1$
end
```

Algorithm 2.

We used the numerical solver (13) with the exit condition

$$\|f(r(t_i))\| \leq tol \vee i = itmax. \tag{16}$$

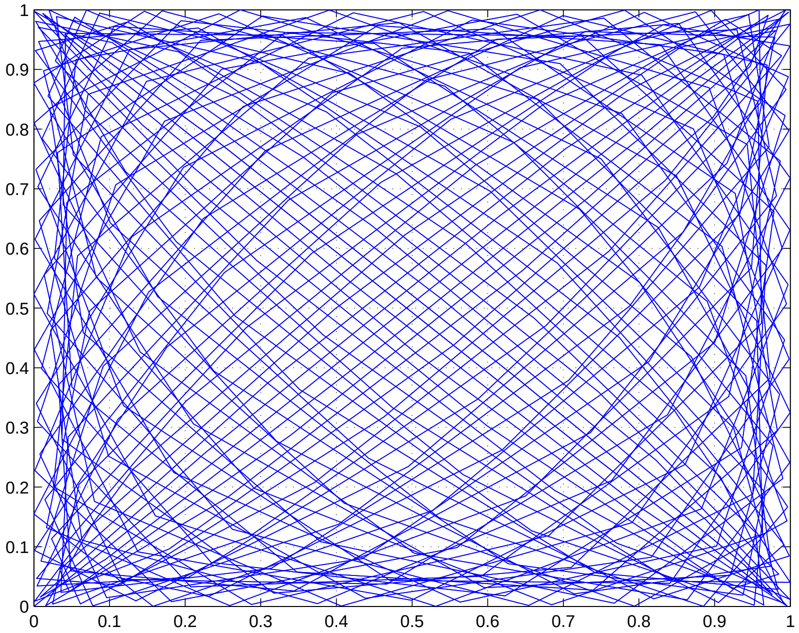


Figure 2
Cherruault 2D curve for $\sigma = 100$.

We selected a set of two dimensional Lipschitz 1 test problems whose Lipschitz constants were numerically estimated using standard techniques (see, e.g. [19], [28]).

We used two versions of equation solver (13): one that is based on Hilbert's space-filling curve and a second one that is based on Cherruault's α-dense curve (15).

For the computation of the 2D Hilbert curve we used the algorithm of page 52 of Bader [3] with $depth = 54$, that computes the points of the curve with an error proportional to $2^{-54} = 5.5511 \times 10^{-17}$.

Since the stepsize $\varphi(t_i)$ can be arbitrarily small, it is reasonable to impose the lower bound $\varphi(t_i) \geq \varepsilon_{machine}$ on the iterates t_i. For $f \in \mathrm{Lip}_{L_f} 1$ and $r \in \mathrm{Lip}_{L_\Gamma} \frac{1}{2}$, this holds if and only if $\|f(r(t_i))\| \geq L_f L_\Gamma \varepsilon_{machine}^{1/2} \approx 6.67 \times 10^{-8} L_f$ and we have the lower bound $tol \geq 6.67 \times 10^{-8} L_f$ for the tol parameter. The computer experiments of [14] and also of Butz [8] indicate that tol can not be to small. Here we selected $tol = 1e-3$ and $itmax = 1e+6$ for the Hilbert's curve based solver and $itmax = 1e+5$ for the α-dense based solver.

The computations were carried out in Matlab R2011b (64 bit) on a PC with Windows 7 operating system and Intel I7 processor.

The CPU times and absolute errors of Algorithms 1 and 2 using Hilbert's curve based solver (Bolzano-v1H, Bolzano-v2H) are shown on the following two figures.

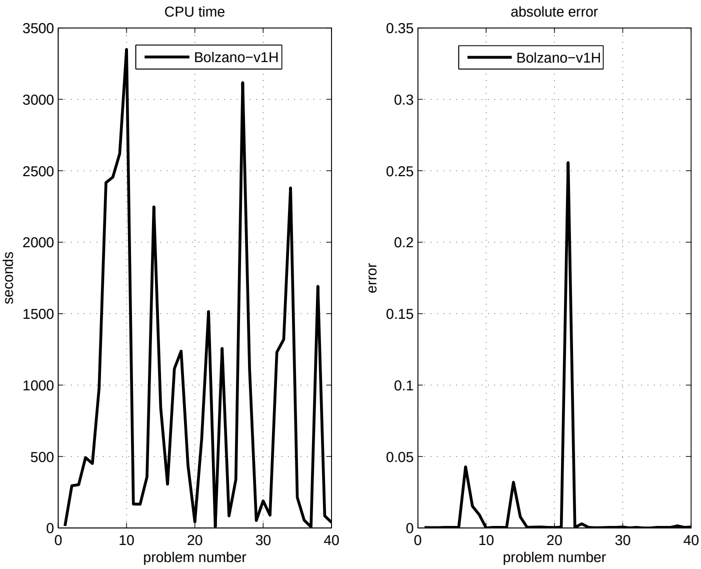


Figure 3
CPU time and absolute error of Algorithm 1 using Hilbert's curve based solver.

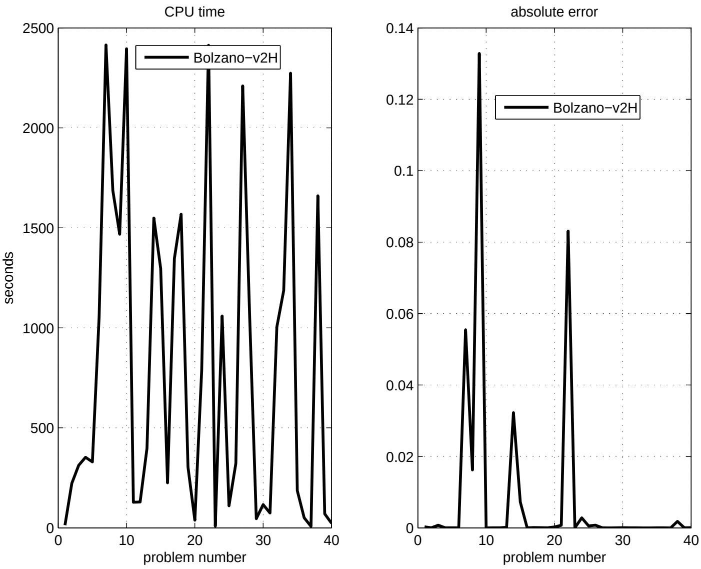


Figure 4
CPU time and absolute error of Algorithm 2 using Hilbert's curve based solver.

Here we can observe extremely big computational times for both algorithms (as expected) and only a few absolute errors greater than $10 \times tol$. The computational times of Algorithm 2 are somewhat less than in the case of Algorithm 1 (the average

CPU time of Algorithm 2 is 797.89 sec. in opposition to the average CPU time of Algorithm 1, which is 892.49 sec.). The absolute errors for Algorithm 1 exceed $10 \times tol = 1e-2$ for the test problems number 7, 8, 14 and 22 while for Algorithm 2 the corresponding cases are the test problems number 7,8,9,14 and 22. A close inspection of these cases reveals that the stepsize $\varphi(t_i)$ of algorithm (13) become less than $\varepsilon_{machine}$, while t_i was much bigger (only for cases $c \approx f_{\min}$). Hence $t_{i+1} = t_i$ was repeated due to the floating point arithmetic and it was stopped only by *itmax*. This problem can be overcome using multiple precision arithmetic.

The CPU times and absolute errors of Algorithms 1 and 2 using α-dense curve based solver (Bolzano-v1C, Bolzano-v2C) are shown on Figures 5 and 6.

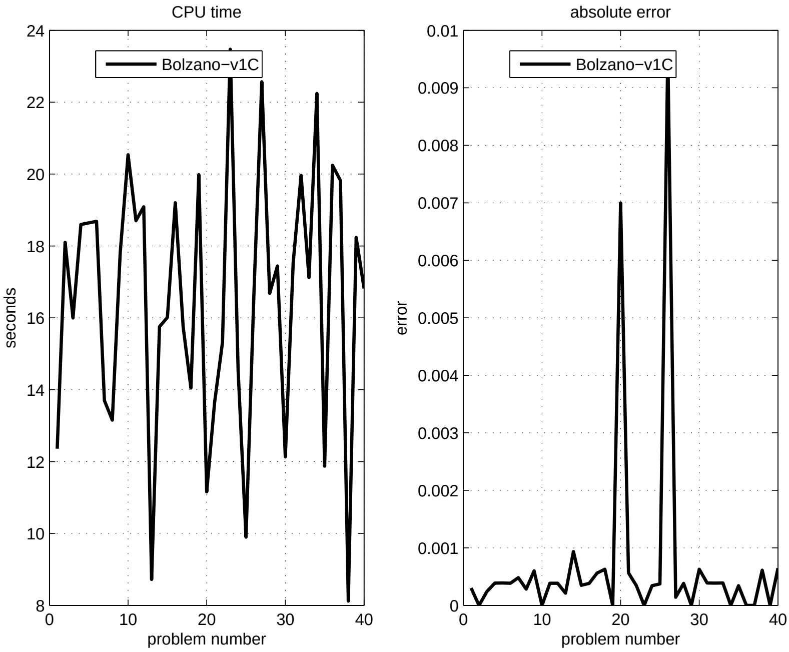


Figure 5
CPU time and absolute error of Algorithm 1 using α-dense curve based solver

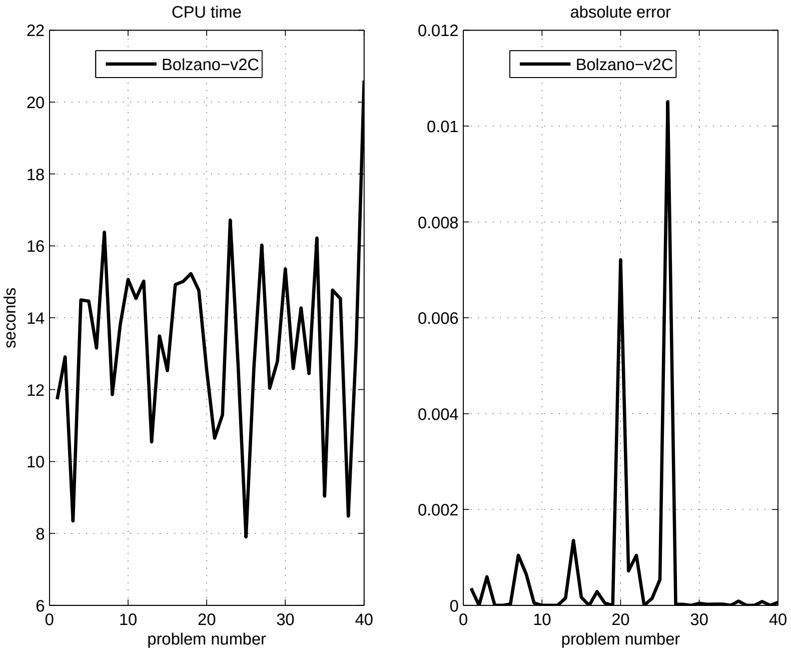


Figure 6
CPU time and absolute error of Algorithm 2 using α-dense curve based solver.

For these versions of Algorithms 1 and 2, the computational times are significant less, while the achieved precision is also better. For Algorithm 1, none of the absolute errors exceeds $10 \times tol = 1e-2$, while for Algorithm 2, there is only one case, test number 26, when the error exceed $1e-2$. It is, in fact, 0.010507.

A comparison of the four versions using the performance profile of Moré et al. [13], [26] clearly shows the ranking of the Algorithms (see Figure 7).

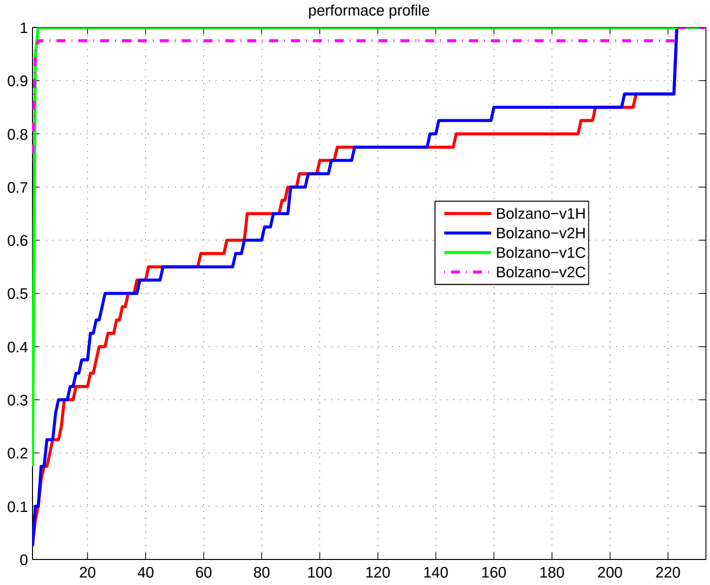


Figure 7
performance profile

4 Conclusions

In the paper we described two general algorithms to find a global minimum of a continuous function in an n-dimensional rectangle. The key point of our algorithms is to solve underdetermined nonlinear equations of the form $f(x) = c$ with such a method that gives unambiguously if the solution exists or not. For this purpose we used a method that exploits the Hilbert space filling function and Cherruault's α dense function. We tested the algorithms for 40 two dimensional well-known test problems. The experimental results clearly indicate that the solutions obtained by the α-dense function based algorithms are much more accurate and require much less execution time than the corresponding algorithms using the Hilbert space filling function. The obtained results show also the reliability of the algorithms in the numerical implementations too. Finally we have to mention that we still need to analyze the cases when $n > 2$.

5 Appendix

Here we enlist the test problems.

1. Adjiman function

$$f(x) = \cos(x_1)\sin(x_2) - \frac{x_1}{x_2^2+1} \qquad (x \in [-1,2] \times [-1,1]).$$

2. Alpine 1 function

$$f(x) = \sum_{i=1}^{n} |x_i \sin(x_i) + 0.1x_i| \quad (x \in [-10, 10]^n).$$

3. Alpine 2 function

$$f(x) = \prod_{i=1}^{n} \sqrt{x_i} \sin(x_i), \quad (x \in [0, 10]^n).$$

4. Bohachevsky 1 function

$$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7 \quad \left(x \in [-1, 1]^2\right).$$

5. Bohachevsky 2 function

$$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)\cos(4\pi x_2) + 0.3 \quad \left(x \in [-1, 1]^2\right).$$

6. Bohachevsky 3 function

$$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3 \quad \left(x \in [-1, 1]^2\right).$$

7. Booth function

$$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2 \quad \left(x \in [-10, 10]^2\right).$$

8. Branin function

$$f(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10,$$

where $x \in [-5, 10] \times [0, 15]$.

9. Brown almost linear function

$$f(x) = \sum_{i=1}^{n} f_i^2(x), \quad (x \in [-1, 2]^n),$$

$$f_i(x) = x_i + \sum_{j=1}^{n} x_j - (n+1), \quad 1 \le i \le n-1,$$

$$f_n(x) = \left(\prod_{j=1}^{n} x_j\right) - 1.$$

10. Bukin 12 function

$$f(x) = 1000(|x_1 + 5 - \rho\cos(\rho)| + |x_2 + 5 - \rho\sin\rho|) + \rho,$$

$$\rho = \sqrt{(x_1 + 5)^2 + (x_2 + 5)^2},\ x \in [-10, 0]^2.$$

11. Chained crescent function 1

$$f(x) = \max\left\{\sum_{i=1}^{n-1}\left(x_i^2 + (x_{i+1}-1)^2 + x_{i+1} - 1\right),\right.$$
$$\left.\sum_{i=1}^{n-1}\left(-x_i^2 - (x_{i+1}-1)^2 + x_{i+1} + 1\right)\right\},$$

where $x \in [-1,1]^n$.

12. Chained crescent function 2

$$f(x) = \sum_{i=1}^{n-1}\max\left\{x_i^2 + (x_{i+1}-1)^2 + x_{i+1} - 1, -x_i^2 - (x_{i+1}-1)^2 + x_{i+1} + 1\right\}$$
$$x \in [-1,1]^n.$$

13. Chained LQ function

$$f(x) = \sum_{i=1}^{n-1}\max\left\{-x_i - x_{i+1}, -x_i - x_{i+1} + \left(x_i^2 + x_{i+1}^2 - 1\right)\right\} \quad \left(x \in [-1,1]^n\right).$$

14. Chained Mifflin function

$$f(x) = \sum_{i=1}^{n-1}\left(-x_i + 2\left(x_i^2 + x_{i+1}^2 - 1\right) + 1.75\left|x_i^2 + x_{i+1}^2 - 1\right|\right) \quad \left(x \in [-1,4]^n\right).$$

15. Chichinadze function

$$f(x) = x_1^2 - 12x_1 + 11 + 10\cos\left(\frac{\pi}{2}x_1\right) + 8\sin(\pi x_1) - \frac{1}{\sqrt{2\pi}}e^{-\frac{(x_2-0.5)^2}{2}}$$
$$x \in [0,10]\times[0,5].$$

16. Cosine mixture function

$$f(x) = -0.1\sum_{i=1}^{n}\cos(5\pi x_i) + \sum_{i=1}^{n}x_i^2 \quad \left(x \in [-1,1]^n,\ -0.1 < 0 < 5\pi\right).$$

17. Cross in tray function

$$f(x) = -0.0001\left(\left|\sin(x_1)\sin(x_2)e^{\left|100-\frac{\sqrt{x_1^2+x_2^2}}{\pi}\right|}\right|+1\right)^{0.1} \quad \left(x \in [-10,10]^2\right).$$

18. Deb function

$$f(x) = -\frac{1}{n}\sum_{i=1}^{n}\sin^6(5\pi x_i) \quad \left(x \in [-1,1]^n\right).$$

19. Egg crate function

$$f(x)=x_1^2+x_2^2+25\left(\sin^2(x_1)+\sin^2(x_2)\right)\quad\left(x\in[-5,5]^2\right).$$

20. El-Attar-Vidyasagar-Dutta function

$$f(x)=\left|x_1^2+x_2-10\right|+\left|x_1+x_2^2-7\right|+\left|x_1^2-x_2^3-1\right|\quad\left(x\in[-5,5]^2\right).$$

21. Hosaki function

$$f(x)=\left(1-8x_1+7x_1^2-\frac{7}{3}x_1^3+\frac{1}{4}x_1^4\right)x_2^2e^{-x_2}\quad(x\in[0,5]\times[0,6]).$$

22. Levy function

$$f(x)=\sin^2(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2\left(1+10\sin^2(\pi y_i+1)\right)+$$
$$(y_n-1)^2\left(1+10\sin^2(2\pi y_n)\right),$$

where

$$y_i=1+\frac{x_i-1}{4}\ (i=1,\ldots,n),\quad x\in[-10,10]^n.$$

23. MAXHILB function

$$f(x)=\max_{1\le i\le n}\left|\sum_{j=1}^{n}\frac{x_j}{i+j-1}\right|\quad(x\in[-1,1]^n).$$

24. McCormick function

$$f(x)=\sin(x_1+x_2)+(x_1-x_2)^2-\frac{3}{2}x_1+\frac{5}{2}x_2+1\quad(x\in[-1.5,4]\times[-3,3])$$

25. Michalewicz function

$$f(x)=-\sum_{i=1}^{n}\sin(x_i)\sin^{2m}\left(\frac{ix_i^2}{\pi}\right)\quad(x\in[0,\pi]^n,\ m=10).$$

26. Mishra 2 function

$$f(x)=\left(1+n-\frac{1}{2}\sum_{i=1}^{n-1}(x_i+x_{i+1})\right)^{n-\frac{1}{2}\sum_{i=1}^{n-1}(x_i+x_{i+1})}\quad(x\in[0,1]^n).$$

27. Multimod function

$$f(x)=\sum_{i=1}^{n}|x_i|\prod_{j=1}^{n}\left|x_j\right|\quad(x\in[-10,10]^n).$$

28. Nesterov 2 function

$$f(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} \left| x_{i+1} - 2x_i^2 + 1 \right| \quad (x \in [0,2]^n).$$

29. Nesterov 3 function

$$f(x) = \frac{1}{4}|x_1 - 1| + \sum_{i=1}^{n-1} |x_{i+1} - 2|x_i| + 1| \quad (x \in [0,2]^n).$$

30. Parsopoulos function

$$f(x) = \cos(x_1)^2 + \sin(x_2)^2 \quad \left(x \in [-5,5]^2\right).$$

31. Pathological function

$$f(x) = \sum_{i=1}^{n-1} \left(0.5 + \frac{\sin\left(\sqrt{100x_i^2 + x_{i+1}^2}\right)^2 - 0.5}{1 + 0.001\left(x_i^2 - 2x_i x_{i+1} + x_{i+1}^2\right)^2} \right) \quad (x \in [-100,100]^n).$$

32. Pintér's function

$$\begin{aligned} f(x) = {} & \sum_{i=1}^{n} i x_i^2 + \sum_{i=1}^{n} i \sin^2(x_{i-1}\sin x_i - x_i + \sin x_{i+1}) \\ & + \sum_{i=1}^{n} i \ln\left(1 + i\left(x_{i-1}^2 - 2x_i + 3x_{i+1} - \cos x_i + 1\right)^2\right), \\ & x_0 = x_n,\ x_{n+1} = x_1,\ x \in [-1,1]^n. \end{aligned}$$

33. Powell sum function

$$f(x) = \sum_{i=1}^{n} |x_i|^{i+1} \quad (x \in [-1,1]^n).$$

34. Trigonometric function

$$\begin{aligned} f(x) &= \sum_{i=1}^{n} f_i^2(x), \quad (x \in [-1,1]^n), \\ f_i(x) &= n - \sum_{j=1}^{n} \cos(x_j) + i(1 - \cos(x_i)) - \sin(x_i). \end{aligned}$$

35. Ursem F1 function

$$f(x) = -\sin(2x_1 - 0.5\pi) - 3\cos(x_2) - 0.5x_1 \quad (x \in [-2.5,3] \times [-2,2]).$$

36. Ursem F3 function

$$f(x) = -\sin(2.2\pi x_1 + 0.5\pi)\frac{(3-|x_1|)(2-|x_2|)}{4} \\ - \sin\left(0.5\pi x_2^2 + 0.5\pi\right)\frac{(2-|x_1|)(2-|x_2|)}{4},$$

where $x \in [-2,2] \times [-1.5,1.5]$.

37. Ursem F4 function

$$f(x) = -3\sin(0.5\pi x_1 + 0.5\pi)\frac{2-\sqrt{x_1^2+x_2^2}}{4} \quad \left(x \in [-2,2]^2\right).$$

38. Vincent function

$$f(x) = -\sum_{i=1}^{n}\sin(10\log(x_i)) \quad (x \in [0.25,10]^n).$$

39. W function

$$f(x) = 1 - \frac{1}{n}\sum_{i=1}^{n}\cos(kx_i)e^{-\frac{x_i^2}{2}} \quad (x \in [-\pi,\pi]^n),$$

where k is a parameter. $k = 10$ for $n = 2$.

40. Yang function 1

$$f(x) = \left(e^{-\sum_{i=1}^{n}\left(\frac{x_i}{\beta}\right)^{2m}} - 2e^{-\sum_{i=1}^{n}(x_i-\pi)^2}\right)\prod_{i=1}^{n}\cos^2(x_i), \\ m = 5,\ \beta = 15,\ x \in [-1,4]^n$$

References

[1] Abaffy J., Galántai A.: A globally convergent branch and bound algorithm for global minimization, in LINDI 2011 3rd IEEE International Symposium on Logistics and Industrial Informatics, August 25–27, 2011, Budapest, Hungary, IEEE, 2011, pp. 205-207, ISBN: 978-1-4577-1842-7, DOI: 10.1109/LINDI.2011.6031148

[2] Abaffy J., Galántai A.: An always convergent algorithm for global minimization of univariate Lipschitz functions, Acta Polytechnica Hungarica, vol. 10, No. 7, 2013, 21–39

[3] Bader, M.: Space-Filling Curves An Introduction with Applications in Scientific Computing, Springer, 2013

[4] Bauman, K.E.: The Dilation Factor of the Peano-Hilbert curve, Mathematical Notes, 2006, 80, 5, 609-620

[5] Butz, A.R.: Space Filling Curves and Mathematical Programming, Information and Control, 1968, 12, 314–330

[6] Butz, A.R.: Convergence with Hilbert's Space Filling Curve, Journal of Computer and System Sciences, 1969, 3, 128–146

[7] Butz, A.R.: Alternative algorithms for Hilbert's space-filling curve, IEEE Transactions on Computers, April, 1971, 424–426

[8] Butz, A.R.: Solutions of Nonlinear Equations with Space Filling Curves, Journal of Mathematical Analysis and Applications, 1972, 37, 351–383

[9] Cherruault, Y.: Mathematical Modelling in Biomedicine, D. Reidel Publishing Company, Dordrecht, Holland, 1986

[10] Cherruault, Y., Mora, G.: Optimisation Globale. Théorie des courbes α-denses, Economica, Paris, 2005, ISBN 2-7178-5065-1

[11] Csendes T.: Nonlinear parameter estimation by global optimization - efficiency and reliability, Acta Cybernetica 8, 1988, 361–370

[12] Csendes T., Pál L., Sendín, J.-Ó. H., Banga, J.R.: The GLOBAL optimization method revisited, Optimization Letters, 2, 2008, 445–454

[13] Dolan, E.D., Moré, J.J.: Benchmarking optimizations software with performance profiles, Mathematical Programming, Series A 91, 2002, 201–213

[14] Galántai, A.: Always convergent methods for solving nonlinear equations, Journal of Computational and Applied Mechanics, Vol. 10., No. 2., 2015, pp. 183-208

[15] Galántai, A.: Always convergent methods for solving nonlinear equations of several variables, Numerical Algorithms, DOI 10.1007/s11075-017-0392-z

[16] Goertzel, B.: Global Optimizations with Space-Filling Curves, Applied Mathematics Letters, 12, 1999, 133-135

[17] Guillez, A.: Alienor, fractal algorithm for multivariable problems, Mathl Comput. Modelling, 1990, 14, 245–247

[18] Hansen, P., Jaumard, B., Lu, S.H.: On the number of iterations of Piyavskii's global optimization algorithm, Mathematics of Operations Research, 16, 1991, 334–350

[19] Hansen, P., Jaumard, B., Lu, S.H.: On using estimates of Lipschitz constants in global optimization, JOTA, 75, 1, 1992, 195–200

[20] Hansen, P., Jaumard, B., Lu, S.H.: Global optimization of univariate Lipschitz functions: I. Survey and properties, Mathematical Programming, 55, 1992, 251–272

[21] Hansen, P., Jaumard, B., Lu, S.H.: Global optimization of univariate Lipschitz functions: II. New algorithms and computational comparison, Mathematical Programming, 55, 1992, 273–292

[22] Levin, Y., Ben-Israel, A.: Directional Newton method in n variables, Mathematics of Computation, 71, 2001, 251–262

[23] McCormick, S.: An iterative procedure for the solution of constrained nonlinear equations with application to optimization problems, Numerische Mathematik, 23, 1975, 371–385

[24] Meyn, K.-H.: Solution of underdetermined nonlinear equations by stationary iteration methods, Numerische Mathematik, 42, 1983, 161–172

[25] Mora, G.: The Peano curves as limit of α-dense curves, Rev. R. Acad. Cien. Serie A. Mat. 2005, 99, 1, 23–28

[26] Moré, J.J, Wild, S.M.: Benchmarking derivative-free optimization algorithms, SIAM J. Optimization, 20,1, 2009, 172–191

[27] Nemirovsky, S., Yudin, V.: Problem Complexity and Method Efficiency in Optimization, Wiley, 1983

[28] Pintér, J.D.: Global Optimization in Action, Kluwer, 1996

[29] Sagan, H.: Space-filling Curves, Springer, 1994

[30] Shary, S.P.: A surprising approach in interval global optimization, Reliable Computing, 7, 2001, 497–505

[31] Shary, S.P.: Graph subdivision methods in interval global optimization, in M. Ceberio, V. Kreinovich (eds.) Contraint Programming and Decision Making, Studies in Computational Intelligence 539, Springer, 2014, 153–170

[32] Sergeyev, Y.D., Strongin, R.G., Lera, D.: Introduction to Global Optimization Exploiting Space-Filling Curves, Springer, 2013

[33] Singh, A.N.: The Theory and Construction of Non-Differentiable Functions, Lucknow University Studies, No. I., Lucknow, India, 1935

[34] Strongin, R.G.: On the convergence of an algorithm for finding a global extremum, Engineering Cybernetics, 1973, 11, 4, 549–555

[35] Strongin, R.G., Sergeyev, Y.D.: Global Optimization with Non-Convex Constraints, Springer, 2000

[36] Szabó Z.: Über gleichungslösende Iterationen ohne Divergenzpunkt I-III, Publ. Math. Debrecen, 20 (1973) 222-233, 21 (1974) 285–293, 27 (1980) 185-200

[37] Szabó Z.: Ein Erveiterungsversuch des divergenzpunkfreien Verfahrens der Berührungsprabeln zur Lösung nichtlinearer Gleichungen in normierten Vektorverbänden, Rostock. Math. Kolloq., 22, 1983, 89–107

[38] Tompkins, C.: Projection methods in calculation, in: H. Antosiewicz (ed.): Proc. Second Symposium on Linear Programming, Washington, D.C., 1955, 425–448

[39] Törn, A., Zilinskas, A.: Global Optimization, Lecture Notes in Computer Science 350, Springer, 1987

[40] Wood, G.R.: The bisection method in higher dimensions, Mathematical Programming, 55, 1992, 319–337

[41] Wood, G.: Bisection global optimization methods, in: C.A. Floudas, P.M. Pardalos (eds.): Encyclopedia of Optimization, 2nd ed., Springer, 2009, pp. 294–297

[42] Zumbusch, G.: Parallel Multilevel Methods: Adaptive Mesh Refinement and Loadbalancing, B.G. Teubner, Stuttgart-Leipzig-Wiesbaden, 2003

The Epsilon Probability Distribution and its Application in Reliability Theory

József Dombi[1], Tamás Jónás[2], Zsuzsanna Eszter Tóth[2]

[1]Department of Computer Algorithms and Artificial Intelligence
University of Szeged
Árpád tér 2, H-6720 Szeged, Hungary
dombi@inf.u-szeged.hu

[2]Institute of Business Economics
Eötvös Loránd University
Egyetem tér 1-3, H-1053 Budapest, Hungary
jonas@gti.elte.hu, tothzs@gti.elte.hu

Abstract: This paper elaborates a new probability distribution, namely, the epsilon probability distribution with implications for reliability theory and management. This probability distribution is founded on the so-called epsilon function that is introduced here. It is also shown that the asymptotic epsilon function is just an exponential function. The properties of this probability distribution suggest that it may serve as a viable alternative to the exponential probability distribution. As the epsilon probability distribution function is a power function, it is more convenient than the exponential probability distribution function from a computational point of view. The main findings and a practical example indicate that the new probability distribution can be utilized to describe the probability distribution of the time to first failure random variable both in the second and third phases of the hazard function.

Keywords: exponential probability distribution; epsilon probability distribution; hazard function, failure rate modeling

1 Introduction

The exponential probability distribution as one of the key distributions in the theory and practice of reliability management [3] [4] plays a significant role in analyzing many data sets obtained from life-tests, and in the use of order statistics. This distribution also appears frequently in lifetime and reaction time studies. It has several remarkable statistical properties, most notably, its characterization through the lack of memory property. Furthermore, it provides mathematical traceability [5]. Consequently, there is extensive literature on the theory and applications of the exponential distribution from the 1930s (e.g. [6], [7], [8]). Weibull [9] considered an extension of the exponential distribution referred to as Weibull distributions, including the exponential distribution as a special case where the shape parameter equals one. Davis [10] also discussed the analysis of failure rate data using the exponential distribution. Characterizations of the exponential distribution originate from mathematicians [11] [12]. Some researchers derived characterizations of the exponential distribution which are modifications of characterizations of the normal distribution [13] [14]. Since then, the characterization results for the exponential distribution have been paid significant attention [15].

In reliability analysis the negative exponential model provides simple, closed-form solutions to many problems [16]. Weibull's classic generalization is usually applied for modelling systems with monotone failure rates [9]. However, according to data in reliability analysis, the operation of a device population can generally be divided into three distinct periods called infancy, useful life, and wear-out periods with each region corresponding to a specific type of failure. This hazard-rate curve, which typically maps the failure rate versus time, has been verified by experience for many types of products. In reliability theory this bathtub-shape is widely used to describe the failure patterns of different products. The relevant literature regarding the bathtub curve is quite diverse. An overview of bathtub-shaped failure rate distributions are provided by [17] and [18]. The exponential distribution is widely applied for modeling the bathtub-shaped failure rate, mainly the useful life period [19], [20], [22].

Models which allow only monotone failure rates might not be appropriate or adequate for modeling the populations that give rise to such data. There have been several attempts to address the need for a family of distributions which allow flexibility in modeling. In this paper, a new probability distribution, namely, the epsilon probability distribution is introduced and its application in reliability theory is discussed. This novel probability distribution is founded on the so-called epsilon function which, just like the exponential function, may be deduced from the n^{th} order epsilon differential equation. The solution of the zero order epsilon differential equation is the exponential function, while the solution of the first order epsilon differential equation is the epsilon function. The epsilon probability distribution has two parameters; namely a λ parameter that has the same meaning as the λ parameter in the exponential probability distribution, and a d parameter that determines the domain $(0,d)$ where the epsilon probability distribution is defined $(d > 0)$. Next, it is shown that the asymptotic epsilon probability distribution is just the exponential probability distribution, which means in practice that the exponential probability distribution with a parameter λ can be substituted by the epsilon probability distribution that has parameters λ and d. Besides the connection between the epsilon and the exponential functions, the relationship between continuous-valued logic and the epsilon function is also highlighted. Namely, the generator function of the Dombi operators [20] may be considered as a special case of the epsilon function.

The remaining part of the paper is organized as follows. In Section 2 the epsilon function, its basic properties and its connection with the exponential function are introduced. In Section 3 we define the epsilon probability distribution and introduce its asymptotic properties. In Section 4 its application in reliability theory is demonstrated through a practical example. Finally, key conclusions are drawn related to the new probability distribution.

2 Epsilon Function

Here we introduce the n^{th} order epsilon differential equation.

Definition 1. *We define the n^{th} order epsilon differential equation as*

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = \lambda \left(\frac{d^2}{d^2 - x^2} \right)^n f(x), \tag{1}$$

where $\lambda \in \mathbb{R}$, $\lambda \neq 0$, $d \in \mathbb{R}$, $d > 0$, $x \in \mathbb{R}$, $x \neq d$, $f(x) > 0$, $n \in \mathbb{N}$.

Lemma 1. *If $n = 0$ and $x \in \mathbb{R}$, then the solution of the n^{th} order epsilon differential equation is*

$$f(x) = \mathrm{e}^{\lambda x + C}, \tag{2}$$

where $C \in \mathbb{R}$.

Proof. If $n = 0$, then the differential equation in (1) may be written as

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = \lambda f(x). \tag{3}$$

Separating the variables in (3) results in

$$\int \frac{1}{f(x)} \mathrm{d}f(x) = \lambda \int \mathrm{d}x. \tag{4}$$

When we integrate both sides of this equation, we get

$$\ln |f(x)| = \lambda x + C, \tag{5}$$

and utilizing the fact, that $f(x) > 0$ means that

$$f(x) = \mathrm{e}^{\lambda x + C}, \tag{6}$$

where $C \in \mathbb{R}$. □

Note that if we wish $f(x)$ to satisfy the condition $f(0) = 1$, then parameter C in (6) needs to be set to 0.

Lemma 2. *If $n = 1$ and $x \in (-d, +d)$, then the solution of the n^{th} order epsilon differential equation is*

$$f(x) = C \left(\frac{x + d}{d - x} \right)^{\lambda \frac{d}{2}}, \tag{7}$$

where $C > 0$.

Proof. If $n = 1$, then the differential equation in (1) has the form

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = \lambda \frac{d^2}{d^2 - x^2} f(x). \tag{8}$$

This equation may be written as

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = \lambda \frac{d^2}{(d+x)(d-x)} f(x). \tag{9}$$

Since

$$\frac{1}{(d+x)(d-x)} = \frac{1}{2d}\left(\frac{1}{x+d} + \frac{1}{d-x}\right), \tag{10}$$

Equation (9) may be written as

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = \lambda \frac{d}{2}\left(\frac{1}{x+d} + \frac{1}{d-x}\right) f(x), \tag{11}$$

and separating the variables in (11) results in

$$\int \frac{1}{f(x)} \mathrm{d}f(x) = \lambda \frac{d}{2} \int \left(\frac{1}{x+d} + \frac{1}{d-x}\right) \mathrm{d}x. \tag{12}$$

After integrating both sides of this equation, we get

$$\ln|f(x)| = \lambda \frac{d}{2} \ln|x+d| - \lambda \frac{d}{2} \ln|d-x| + \ln C, \tag{13}$$

where $C > 0$. Utilizing the fact, that $x \in (-d, +d)$ and $f(x) > 0$ means that

$$f(x) = C \left(\frac{x+d}{d-x}\right)^{\lambda \frac{d}{2}}. \tag{14}$$

□

Note that if we wish $f(x)$ to satisfy the condition $f(0) = 1$, then parameter C in (14) needs to be set to 1. In this case, we call the solution of the first order epsilon differential equation the epsilon function.

Definition 2. *The epsilon function $\varepsilon_{\lambda,d}(x)$ is given by*

$$\varepsilon_{\lambda,d}(x) = \left(\frac{x+d}{d-x}\right)^{\lambda \frac{d}{2}}, \tag{15}$$

where $\lambda \in \mathbb{R}$, $\lambda \neq 0$, $d \in \mathbb{R}$, $d > 0$, $x \in (-d, +d)$.

2.1 Some Basic Properties of the Epsilon Function

Here, we state the most important properties of the epsilon function; namely, continuity, monotonity, limits and convexity.

Continuity. $\varepsilon_{\lambda,d}(x)$ is a continuous function in $(-d,+d)$.

Monotonicity.

- If $\lambda > 0$, then $\varepsilon_{\lambda,d}(x)$ is strictly monotonously increasing
- If $\lambda < 0$, then $\varepsilon_{\lambda,d}(x)$ is strictly monotonously decreasing
- If $\lambda = 0$, then $\varepsilon_{\lambda,d}(x)$ has a constant value of 1

in the interval $(-d,+d)$.

Limits.

$$\lim_{x\to -d^+} \varepsilon_{\lambda,d}(x) = \begin{cases} 0, & \text{if } \lambda > 0 \\ \infty, & \text{if } \lambda < 0 \end{cases} \tag{16}$$

$$\lim_{x\to +d^-} \varepsilon_{\lambda,d}(x) = \begin{cases} \infty, & \text{if } \lambda > 0 \\ 0, & \text{if } \lambda < 0. \end{cases} \tag{17}$$

Note that if $\lambda > 0$, then $\varepsilon_{\lambda,d}(-d) = 0$, and if $\lambda < 0$, then $\varepsilon_{\lambda,d}(d) = 0$.

Convexity. The second derivative of $\varepsilon_{\lambda,d}(x)$ is

$$\frac{\mathrm{d}^2\varepsilon_{\lambda,d}(x)}{\mathrm{d}x^2} = \lambda \frac{d^2}{\left(d^2-x^2\right)^2} \varepsilon_{\lambda,d}(x) \left(2x+\lambda d^2\right), \tag{18}$$

whose sign depends on λ and $2x+\lambda d^2$. Thus,

- if $\lambda > 0$ and $x < -\frac{\lambda d^2}{2}$, then $\varepsilon_{\lambda,d}(x)$ is concave,
- if $\lambda > 0$ and $x > -\frac{\lambda d^2}{2}$, then $\varepsilon_{\lambda,d}(x)$ is convex,
- if $\lambda < 0$ and $x < -\frac{\lambda d^2}{2}$, then $\varepsilon_{\lambda,d}(x)$ is convex,
- if $\lambda < 0$ and $x > -\frac{\lambda d^2}{2}$, then $\varepsilon_{\lambda,d}(x)$ is concave.

That is, $\varepsilon_{\lambda,d}(x)$ has a single inflection point at $-\frac{\lambda d^2}{2}$. Notice that $-\frac{\lambda d^2}{2}$ lies in the interval $(-d,d)$, only if $|\lambda|d < 2$. Hence, $\varepsilon_{\lambda,d}(x)$ is strictly convex in $(-d,d)$, if $|\lambda|d \geq 2$.

Figure 1 shows two examples of the epsilon function curve with positive and negative λ parameter values.

2.2 Connection with the Exponential Function

Lemma 1 and Lemma 2 suggest an important connection between the exponential function and the epsilon function. Namely, the solution of the zero order epsilon differential equation is the exponential function, while the solution of the first order

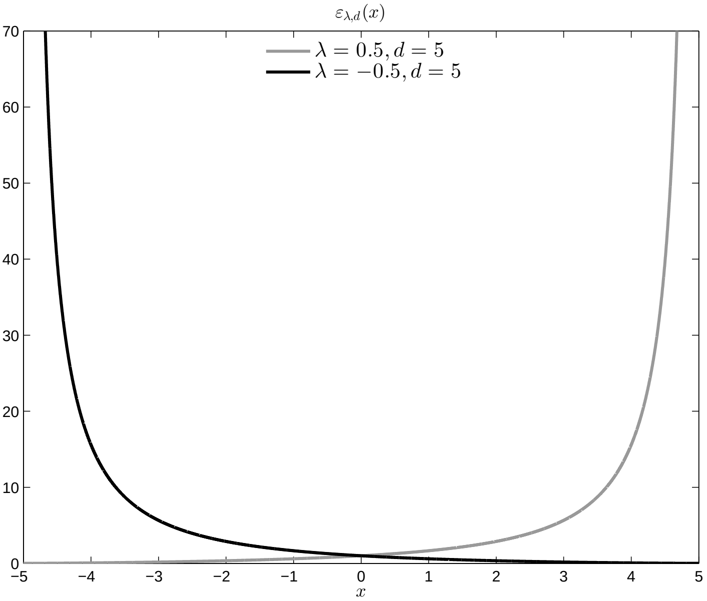


Figure 1
Examples of epsilon function curves

epsilon differential equation is the epsilon function. Table 1 summarizes how the exponential function and the epsilon function can be derived from the the n^{th} order epsilon differential equation.

Table 1
The exponential and epsilon functions derived from the n^{th} order epsilon differential equation

Domain	n	Condition	C	$f(x)$
$(-\infty,\infty)$	0	$f(0) = 1$	0	$e^{\lambda x}$
$(-d,+d)$	1	$f(0) = 1$	1	$\varepsilon_{\lambda,d}(x)$

Beyond the fact that both the exponential and the epsilon functions can be derived from the n^{th} order epsilon differential equation, the following theorem highlights an additional connection between these two functions.

Theorem 1. *For any $x \in (-d,+d)$, if $d \to \infty$, then*

$$\varepsilon_{\lambda,d}(x) \to e^{\lambda x}. \tag{19}$$

Proof. Let x have a fixed value, $x \in (-d, +d)$.

$$\lim_{d\to\infty} \varepsilon_{\lambda,d}(x) = \lim_{d\to\infty} \left(\frac{x+d}{d-x}\right)^{\lambda\frac{d}{2}} = \lim_{d\to\infty} \left(\left(\frac{d-x+2x}{d-x}\right)^{d}\right)^{\frac{\lambda}{2}} = \\ = \lim_{d\to\infty} \left(\left(1+\frac{2x}{d-x}\right)^{d}\right)^{\frac{\lambda}{2}}. \tag{20}$$

Since x is fixed, if $d \to \infty$, then $\Delta = d - x \to \infty$ and so the previous equation can be continued as follows.

$$\lim_{d\to\infty} \left(\left(1+\frac{2x}{d-x}\right)^{d}\right)^{\frac{\lambda}{2}} = \lim_{\Delta\to\infty} \left(\left(1+\frac{2x}{\Delta}\right)^{\Delta+x}\right)^{\frac{\lambda}{2}} = \\ = \left(\lim_{\Delta\to\infty}\left(1+\frac{2x}{\Delta}\right)^{\Delta} \lim_{\Delta\to\infty}\left(1+\frac{2x}{\Delta}\right)^{x}\right)^{\frac{\lambda}{2}} = \left(e^{2x}\right)^{\frac{\lambda}{2}} \cdot 1^{\frac{\lambda}{2}} = e^{\lambda x}. \tag{21}$$

□

Based on Theorem 1, we may state that with respect to parameter d $(d \to \infty)$, the asymptotic epsilon function is just the exponential function. Actually, if $x \ll d$, then $\varepsilon_{\lambda,d} \approx e^{\lambda x}$; that is, if d is sufficiently large, then the epsilon function suitably approximates the exponential function.

2.3 Connection with Dombi Operators in Continuous-valued Logic

The Dombi operator [2] in continuous-valued logic is given by

$$o_\alpha(x_1, x_2, \ldots, x_n) = \frac{1}{1+\left(\sum\limits_{i=1}^{n}\left(\frac{1-x_i}{x_i}\right)^{\alpha}\right)^{1/\alpha}}, \tag{22}$$

where $x_1, x_2, \ldots, x_n$ are continuous-valued logic variables. If $\alpha \geq 0$, then the Dombi operator is a conjunction operator; if $\alpha \leq 0$, then the Dombi operator is a disjunction operator. The generator function $g_\alpha(x)$ of the Dombi operators in continuous-valued logic is given by

$$g_\alpha(x) = \left(\frac{1-x}{x}\right)^{\alpha}. \tag{23}$$

Lemma 3. *The generator function $g_\alpha(x)$ can be derived from the epsilon function $\varepsilon_{\lambda,d}(x)$ by a linear function transformation.*

Proof. Let us apply the $x' = (x+d)/(2d)$ linear transformation to the variable x, where $x \in (-d,d)$, $d > 0$. After this transformation, the domain of x' is the interval $(0,1)$, $x = 2dx' - d$, and

$$\begin{aligned} \varepsilon_{\lambda,d}(x) &= \left(\frac{x+d}{d-x}\right)^{\lambda\frac{d}{2}} = \left(\frac{2dx'-d+d}{d-2dx'+d}\right)^{\lambda\frac{d}{2}} = \left(\frac{x'}{1-x'}\right)^{\lambda\frac{d}{2}} = \\ &= \left(\frac{1-x'}{x'}\right)^{-\lambda\frac{d}{2}} = g_{\alpha}(x'), \end{aligned} \tag{24}$$

where $\alpha = -\lambda d/2$. □

Based on this result, the generator function of the Dombi operators may be viewed as a special case of the epsilon function.

3 Epsilon Probability Distribution

Here, we will define the epsilon probability distribution and show how it is connected with the exponential distribution.

Definition 3. *The continuous random variable ξ has an epsilon probability distribution with the parameters $\lambda > 0$ and $d > 0$, if the probability density function $f_{\lambda,d}(x)$ of ξ is given by*

$$f_{\lambda,d}(x) = \begin{cases} 0, & \textit{if } x \leq 0 \\ \lambda \frac{d^2}{d^2-x^2}\varepsilon_{-\lambda,d}(x), & \textit{if } 0 < x < d \\ 0, & \textit{if } x \geq d, \end{cases} \tag{25}$$

where

$$\varepsilon_{-\lambda,d}(x) = \left(\frac{x+d}{d-x}\right)^{-\lambda\frac{d}{2}}. \tag{26}$$

In order to show that $f_{\lambda,d}(x)$ is in fact a probability density function, we will prove the following lemma.

Lemma 4. *The function $f_{\lambda,d}(x)$ has the following properties.*

1. $f_{\lambda,d}(x) \geq 0$ *for any* $x \in \mathbb{R}$
2. $\int\limits_{-\infty}^{\infty} f_{\lambda,d}(x)\mathrm{d}x = 1.$

Proof. The first property of $f_{\lambda,d}(x)$ trivially follows. The second property of $f_{\lambda,d}(x)$ can be demonstrated by using the definition of $\varepsilon_{-\lambda,d}(x)$ and the Lemma 2:

$$\int\limits_{-\infty}^{\infty} f_{\lambda,d}(x)\mathrm{d}x = \int\limits_{0}^{d} \lambda \frac{d^2}{d^2-x^2}\varepsilon_{-\lambda,d}(x)\mathrm{d}x = \left[-\varepsilon_{-\lambda,d}(x)\right]_0^d = 0-(-1) = 1. \tag{27}$$

□

If the random variable η has an exponential probability distribution with the parameter $\lambda > 0$, then the probability density function $f_\lambda(x)$ of η is given by

$$f_\lambda(x) = \begin{cases} 0, & \text{if } x \leq 0 \\ \lambda \mathrm{e}^{-\lambda x}, & \text{if } x > 0. \end{cases} \tag{28}$$

The next theorem tells us how the epsilon probability distribution is connected with the exponential probability distribution.

Theorem 2. *For any $x \in \mathbb{R}$ and $\lambda > 0$, if $d \to \infty$, then*

$$f_{\lambda,d}(x) \to f_\lambda(x). \tag{29}$$

Proof. Let x be fixed and let $x \in \mathbb{R}$. We will distinguish the following cases.

- If $x \leq 0$, then $f_{\lambda,d}(x) = f_\lambda(x) = 0$ holds by definition.
- If $x \in (0,d)$, $d > 0$, then

  $$f_{\lambda,d}(x) = \lambda \frac{d^2}{d^2 - x^2} \varepsilon_{-\lambda,d}(x). \tag{30}$$

  If $d \to \infty$, then

  $$\frac{d^2}{d^2 - x^2} \to 1, \tag{31}$$

  and following Theorem 1,

  $$\varepsilon_{-\lambda,d}(x) \to \mathrm{e}^{-\lambda x}. \tag{32}$$

  That is, if $d \to \infty$, then

  $$f_{\lambda,d}(x) \to \lambda \mathrm{e}^{-\lambda x} = f_\lambda(x). \tag{33}$$

□

The probability distribution function $F_{\lambda,d}(x)$ of the random variable ξ that has an epsilon probability distribution with parameters $\lambda > 0$ and $d > 0$ can be derived from the epsilon probability density function in the following way.

$$F_{\lambda,d}(x) = \int_{-\infty}^{x} f_{\lambda,d}(t)\mathrm{d}t =$$

$$= \begin{cases} \int_{-\infty}^{x} f_{\lambda,d}(t)\mathrm{d}t = \int_{-\infty}^{x} 0\mathrm{d}t = 0, & \text{if } x \leq 0 \\ \int_{-\infty}^{0} 0\mathrm{d}t + \int_{0}^{x} \lambda \frac{d^2}{d^2-t^2} \varepsilon_{-\lambda,d}(t)\mathrm{d}t, & \text{if } 0 < x < d \\ \int_{-\infty}^{0} 0\mathrm{d}t + \int_{0}^{d} \lambda \frac{d^2}{d^2-t^2} \varepsilon_{-\lambda,d}(t) + \int_{d}^{x} 0\mathrm{d}t, & \text{if } x \geq d. \end{cases} \tag{34}$$

As

$$\begin{aligned}&\int_0^x \lambda \frac{d^2}{d^2-t^2}\varepsilon_{-\lambda,d}(t)\mathrm{d}t = \left[-\varepsilon_{-\lambda,d}(t)\right]_0^x = \\ &= \left[-\left(\frac{t+d}{d-t}\right)^{-\lambda\frac{d}{2}}\right]_0^x = -\left(\frac{x+d}{d-x}\right)^{-\lambda\frac{d}{2}} - (-1) = 1-\varepsilon_{-\lambda,d}(x)\end{aligned} \tag{35}$$

and

$$\int_0^d \lambda \frac{d^2}{d^2-t^2}\varepsilon_{-\lambda,d}(t)\mathrm{d}t = 1, \tag{36}$$

$F_{\lambda,d}(x)$ may be written as

$$F_{\lambda,d}(x) = \begin{cases} 0, & \text{if } x \le 0 \\ 1-\varepsilon_{-\lambda,d}(x), & \text{if } 0 < x < d \\ 1, & \text{if } x \ge d. \end{cases} \tag{37}$$

Note that if the random variable η has an exponential probability distribution with the parameter $\lambda > 0$, then the probability distribution function $F_\lambda(x)$ of η is given by

$$F_\lambda(x) = \begin{cases} 0, & \text{if } x \le 0 \\ 1-\mathrm{e}^{-\lambda x}, & \text{if } x > 0. \end{cases} \tag{38}$$

Theorem 3. *For any $x \in \mathbb{R}$, $\lambda > 0$ and $d > 0$, if the random variable ξ has an epsilon probability distribution with the parameters λ and d and the η random variable has an exponential distribution with parameter λ, then*

$$\lim_{d\to\infty} P(\xi < x) = P(\eta < x). \tag{39}$$

Proof. Since $F_{\lambda,d}(x) = P(\xi < x)$ and $F_\lambda(x) = P(\eta < x)$ for any $x \in \mathbb{R}$, using the definitions of $F_{\lambda,d}(x)$ and $F_\lambda(x)$, this theorem follows from Theorem 1. □

Figure 2 shows some examples of how the density function curve of epsilon probability distribution can match the density function curve of exponential probability distribution. In each subplot of Figure 2, the left hand side scale belongs to functions $f_\lambda(x)$ and $f_{\lambda,d}(x)$, while the right hand side scale is connected with the difference function $f_\lambda(x) - f_{\lambda,d}(x)$. We can see that the goodness of approximation improves as d increases. Similar to Figure 2, Figure 3 shows some examples of how the epsilon probability distribution function curve can match the exponential probability distribution function curve.

Based on Theorem 2, we may state that the asymptotic epsilon probability distribution is just the exponential probability distribution. Thus, in practical applications, the exponential probability distribution with a parameter λ can be substituted by the

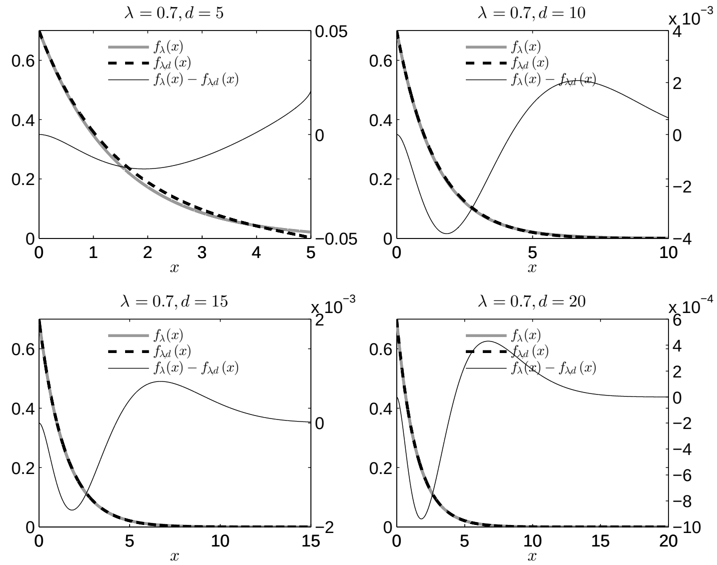


Figure 2
Examples of density function curves

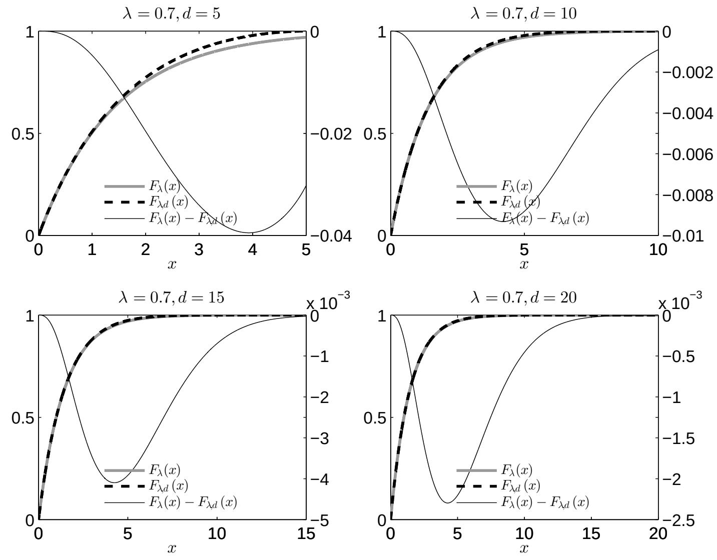


Figure 3
Examples of probability distribution function curves

epsilon probability distribution that has the parameters λ and d, if $x \ll d$. It is worth mentioning that while the exponential probability distribution function is a transcendent function, the epsilon probability distribution function is a power function. This means that from a computational point of view, the epsilon probability distribution function is more convenient than the exponential probability distribution function. This feature of the epsilon probability distribution further enhances its applicability in problems where computation time is a critical factor.

From here on, we will use the notations $\xi \sim \varepsilon(\lambda,d)$ and $\eta \sim \exp(\lambda)$ to indicate that ξ has an epsilon probability distribution with the parameters $\lambda > 0$ and $d > 0$, and η has an exponential distribution with parameter λ, respectively.

3.1 Some Asymptotic Properties of the Epsilon Probability Distribution

Here, we will show some asymptotic properties of the random variable ξ, where $\xi \sim \varepsilon(\lambda,d)$. Namely, we will show the asymptotic expected value and asymptotic standard deviation of ξ, and demonstrate the asymptotic memoryless property of ξ.

Asymptotic expected value

Theorem 4. *If $\xi \sim \varepsilon(\lambda,d)$, then*

$$\lim_{d\to\infty} E(\xi) = \frac{1}{\lambda}, \tag{40}$$

where $E(\xi)$ is the expected value of ξ.

Proof. Here, using the definition of the expected value and Theorem 2

$$\begin{aligned}\lim_{d\to\infty} E(\xi) = \lim_{d\to\infty} \int\limits_0^\infty x f_{\lambda,d}(x)\mathrm{d}x = \\ = \int\limits_0^\infty x \left(\lim_{d\to\infty} f_{\lambda,d}(x)\right)\mathrm{d}x = \int\limits_0^\infty x\lambda \mathrm{e}^{-\lambda x}\mathrm{d}x.\end{aligned} \tag{41}$$

The last integral is the expected value of the random variable $\eta \sim \exp(\lambda)$. It is known that the expected value $E(\eta)$ of η is $1/\lambda$ and so

$$\lim_{d\to\infty} E(\xi) = \frac{1}{\lambda}. \tag{42}$$

□

Asymptotic standard deviation

Theorem 5. *If $\xi \sim \varepsilon(\lambda,d)$, then*

$$\lim_{d\to\infty} D(\xi) = \frac{1}{\lambda}, \tag{43}$$

where $D(\xi)$ is the standard deviation of ξ.

Proof. Since $D^2(\xi) = E(\xi^2) - E^2(\xi)$,

$$\lim_{d\to\infty} D^2(\xi) = \lim_{d\to\infty} E(\xi^2) - \lim_{d\to\infty} E^2(\xi). \tag{44}$$

Using Theorem 2,

$$\begin{aligned} \lim_{d\to\infty} E(\xi^2) &= \lim_{d\to\infty} \int_0^\infty x^2 f_{\lambda,d}(x)\mathrm{d}x = \\ &= \int_0^\infty x^2 \left(\lim_{d\to\infty} f_{\lambda,d}(x)\right)\mathrm{d}x = \int_0^\infty x^2\lambda \mathrm{e}^{-\lambda x}\mathrm{d}x. \end{aligned} \tag{45}$$

The last integral can be calculated as follows.

$$\begin{aligned} \int_0^\infty x^2\lambda\mathrm{e}^{-\lambda x}\mathrm{d}x &= \left[x^2\left(-\mathrm{e}^{-\lambda x}\right)\right]_0^\infty - \int_0^\infty 2x(-\mathrm{e}^{-\lambda x})\mathrm{d}x = \\ &= \frac{2}{\lambda}\int_0^\infty x\lambda\mathrm{e}^{-\lambda x}\mathrm{d}x = \frac{2}{\lambda}\frac{1}{\lambda} = \frac{2}{\lambda^2}. \end{aligned} \tag{46}$$

Next, using Theorem 4,

$$\lim_{d\to\infty} E^2(\xi) = \frac{1}{\lambda^2} \tag{47}$$

and so

$$\lim_{d\to\infty} D^2(\xi) = \lim_{d\to\infty} E(\xi^2) - \lim_{d\to\infty} E^2(\xi) = \frac{2}{\lambda^2} - \frac{1}{\lambda^2} = \frac{1}{\lambda^2}, \tag{48}$$

from which after taking into account the fact that the λ parameter always has a positive value, means that

$$\lim_{d\to\infty} D(\xi) = \frac{1}{\lambda}. \tag{49}$$

□

Asymptotic memoryless property

Definition 4. *The random variable ξ is memoryless with respect to t for all $\Delta t > 0$, if*

$$P(\xi > t+\Delta t|\xi > t) = P(\xi > \Delta t). \tag{50}$$

It is known that the exponential distribution is the only continuous distribution that is memoryless. Here, we will show that $\xi \sim \varepsilon(\lambda, d)$ is an asymptotically memoryless random variable, if $d \to \infty$.

Theorem 6. *If* $\xi \sim \varepsilon(\lambda, d)$*, then*

$$\lim_{d\to\infty} P(\xi > t + \Delta t | \xi > t) = \lim_{d\to\infty} P(\xi > \Delta t). \tag{51}$$

Proof. Utilizing the definition of the conditional probability and the assumption that $\xi \sim \varepsilon(\lambda, d)$

$$\begin{aligned} P(\xi > t + \Delta t | \xi > t) &= \frac{P(\xi > t + \Delta t, \xi > t)}{P(\xi > t)} = \frac{P(\xi > t + \Delta t)}{P(\xi > t)} = \\ &= \frac{1 - F_{\lambda,d}(t + \Delta t)}{1 - F_{\lambda,d}(t)}, \end{aligned} \tag{52}$$

and using (37)

$$\frac{1 - F_{\lambda,d}(t + \Delta t)}{1 - F_{\lambda,d}(t)} = \frac{\varepsilon_{-\lambda,d}(t + \Delta t)}{\varepsilon_{-\lambda,d}(t)}. \tag{53}$$

Furthermore, utilizing Theorem 1, equations (52) and (53),

$$\lim_{d\to\infty} P(\xi > t + \Delta t | \xi > t) = \lim_{d\to\infty} \frac{\varepsilon_{-\lambda,d}(t + \Delta t)}{\varepsilon_{-\lambda,d}(t)} = \frac{\mathrm{e}^{-\lambda(t+\Delta t)}}{\mathrm{e}^{-\lambda t}} = \mathrm{e}^{-\lambda \Delta t}. \tag{54}$$

Similarly, the right hand side of (51) may be written as

$$\lim_{d\to\infty} P(\xi > \Delta t) = \lim_{d\to\infty} \varepsilon_{-\lambda,d}(\Delta t) = \mathrm{e}^{-\lambda \Delta t}. \tag{55}$$

That is, both the left and right hand sides of (51) are equal to $\mathrm{e}^{-\lambda \Delta t}$. □

4 Application in Reliability Theory

Let the continuous random variable τ be the time to first failure of a component or system. The conditional probability that this component or system will fail the first time in the time interval $(t, t + \Delta t]$, given that it has survived up to time t, can be calculated as follows:

$$\begin{aligned} P(\tau \leq t + \Delta t | \tau > t) &= \frac{P(t < \tau \leq t + \Delta t)}{P(\tau > t)} = \\ &= \frac{F(t + \Delta t) - F(t)}{1 - F(t)} = \frac{F(t + \Delta t) - F(t)}{R(t)}, \end{aligned} \tag{56}$$

where $F(t)$ is the probability distribution function of τ and $R(t) = 1 - F(t)$ is the survival function of τ. In reliability theory, the failure rate function $h(t)$ for τ is given by

$$h(t) = \lim_{\Delta t \to 0} \frac{F(t + \Delta t) - F(t)}{\Delta t R(t)} = \frac{f(t)}{R(t)}, \tag{57}$$

where $f(t)$ is the probability density function of τ. The hazard function $h(t)$ is also called the failure rate function. In practice, based on (56) and (57), the quantity $h(t)\Delta t$ represents the conditional probability that a component or a system will fail in the time interval $(t, t+\Delta t]$, given that it has survived up to time t.

A typical hazard function curve of a component or a system is "bathtub shaped"; that is, it can be divided into three distinct phases called the infant mortality period, useful life, and wear-out period. During the early life (infant mortality), the failure rate function decreases with respect to time. In this phase, the failures are mostly caused by initial weaknesses or defects in the material, defective design, poor quality control, poor workmanship, and damaged or missing parts in the assembly phase. In the second phase, which is also known as the useful life, the failure rate function is constant with respect to time. In this period only random failures occur. These unexpected failures are caused by over-stress conditions and they cannot be eliminated by maintenance practices. Even the best design fabrication and screening techniques cannot completely eliminate the effect of such failures. Watson [21] gives reasons for the assumption of a constant failure rate. The third, wear-out phase can be described by an increasing failure rate function. The wear-out period can be postponed by introducing replacement technologies.

It is typical that the probability distribution of τ is different in the three characteristic phases of the "bathtub shaped" hazard function. If τ has an exponential distribution with parameter λ, then using (57), the hazard function $h_\lambda(t)$ for τ is

$$h_\lambda(t) = \frac{f_\lambda(t)}{1-F_\lambda(t)} = \frac{\lambda \mathrm{e}^{-\lambda t}}{\mathrm{e}^{-\lambda t}} = \lambda. \tag{58}$$

That is, if $\tau \sim exp(\lambda)$, then the failure rate function is constant with respect to time. Most commonly, the exponential probability distribution is utilized to describe the probability distribution of τ in the second, constant phase of the hazard curve, while probability distributions that result in decreasing or increasing hazard functions are used to model the probability distribution of τ in the first and third phases of the hazard function curve.

Now let us assume that τ has an epsilon distribution with the parameters λ and d. In this case, the hazard function $h_{\lambda,d}(t)$ is

$$h_{\lambda,d}(t) = \frac{f_{\lambda,d}(t)}{1-F_{\lambda,d}(t)} = \frac{\lambda \frac{d^2}{d^2-t^2}\varepsilon_{-\lambda,d}(t)}{\varepsilon_{-\lambda,d}(t)} = \lambda \frac{d^2}{d^2-t^2}, \tag{59}$$

if $0 < t < d$. The hazard function $h_{\lambda,d}(t)$ has some important properties that are worth emphasizing here.

- If $t \in (0,d)$ is fixed, then $h_{\lambda,d}(t)$ tends to λ as d approaches infinity. In practice, it means that if t is small compared to d, then $h_{\lambda,d}(t) \approx \lambda$. That is, if d is sufficiently large, then the probability distribution of τ may be described by the epsilon distribution in the second phase of the hazard function.
- The hazard function $h_{\lambda,d}(t)$ is increasing with respect to t. If t is small compared to d, then $h_{\lambda,d}(t)$ is increasing slowly, and $h_{\lambda,d}(t)$ tends to infinity as t approaches d (from the left hand side).

Based on the above properties of the hazard function $h_{\lambda,d}(t)$, we may conclude that the epsilon probability distribution can be utilized to describe the probability distribution of the time to first failure random variable both in the second and in the third phases of the hazard function. Moreover, if the second phase of a failure rate function is slightly increasing instead of being constant with respect to time, then the probability distribution of τ in this phase of the hazard function can be better described by the epsilon probability distribution than by the exponential probability distribution.

4.1 A practical example

Now we will show how the epsilon probability distribution can be utilized to model the probability distribution of the time to first failure random variable in the second and third phases of the hazard function.

The failure rate function $h(t)$ can be estimated from empirical data by

$$h(t) \approx \frac{N(t) - N(t+\Delta t)}{N(t)\Delta t}, \tag{60}$$

where $N(t)$ is the number of components or systems that have survived up to time t from the number of components or systems $N(0)$ that were initially put into operation. If $\Delta t = 1$, then the estimated failure rate h_i for period i may be given by

$$h_i = \frac{N(i) - N(i+1)}{N(i)}, \tag{61}$$

$i = 0, 1, \ldots, n$, and so $h_0, h_1, \ldots, h_n$ may be viewed as an empirical failure rate time series.

Here, we examined the empirical failure rate times series corresponding to the second and third phases of the hazard function curve of an electronic product. The investigated time series, which is shown in Figure 4, contained 120 failure rates that had been computed from empirical data using Equation (61).

The hazard function $h_{\lambda,d}(t)$ was fitted to the first 30, 60, 90 and 120 data values of the empirical failure rate time series. In each case, the estimations λ^* and d^* of the parameters λ and d, respectively, were identified by solving the following minimization problem:

$$\begin{gathered}\mathscr{F}(\lambda, d) = \sum_{i=0}^{k} \left(h_i - \lambda \frac{d^2}{d^2 - i^2} \right)^2 \to \min \\ \lambda > 0, d > k.\end{gathered} \tag{62}$$

We solved this minimization problem by applying an interior point algorithm [1]. In order to find the global minimum of the target function $\mathscr{F}(\lambda, d)$, the initial values for λ_0 and d_0 of λ and d, respectively, were set as follows.

$$\begin{gathered}\lambda_0 = \frac{1}{k} \sum_{i=0}^{k} h_i \\ d_0 = k + \delta,\end{gathered} \tag{63}$$

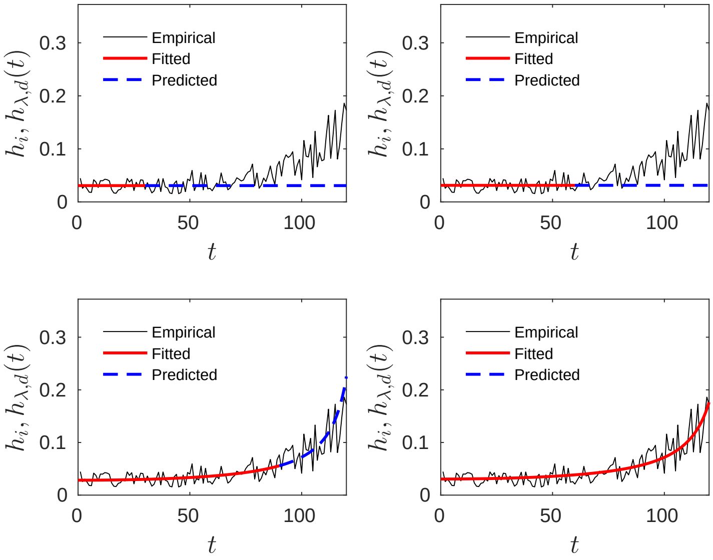


Figure 4
Hazard functions fitted to empirical failure rate time series

where $\delta > 0$ is a number close to zero. In our implementation $\delta = 10^{-3}$.

In each subplot of Figure 4, the continuous red line indicates the fitted $h_{\lambda,d}(t)$ values, while the dashed blue line shows the predicted failure rates that were based on the function $h_{\lambda,d}(t)$. Table 2 summarizes the mathematical modeling results. It contains the estimated model parameters and the Mean Square Error (MSE) values for the fits and predictions.

Table 2
Hazard function fitting results

Input range	λ^*	d^*	MSE (Fitted)	MSE (Predicted)
0–29	0.0307	$1.8729 \cdot 10^{152}$	$8.7506 \cdot 10^{-5}$	0.0022
0–59	0.0312	$3.1758 \cdot 10^{152}$	$1.1538 \cdot 10^{-4}$	0.0032
0–89	0.0283	128.3420	$1.2907 \cdot 10^{-4}$	0.0010
0–119	0.0305	131.7570	$2.9315 \cdot 10^{-4}$	———

Based on Figure 4 and on Table 2, the following properties of our modeling should be mentioned here. The volatility of the analyzed failure rate time series increases with respect to time; that is, it displays a lower variability in its quasi constant phase than in its increasing phase. This observation is in line with the result that the MSE of the fitted curve slightly increases as the number of input data values increases. In the case of the first two input ranges, where the empirical hazard rate curve is in its quasi constant phase, the estimated values for the d parameter are very large, while the estimates for the λ parameter are very close. These results are in line with the previously discussed theoretical ones; namely, if d is large and $t \ll d$, then the epsilon probability distribution is suitable for describing the time to first failure random variable in the quasi constant phase of the hazard function. When the empirical failure rate curve is in its increasing phase, then the estimates of parameter d are close to the total length of the time series (120 periods), while the two estimates for the λ parameter are similar. This empirical finding supports our theoretical results that the epsilon probability distribution can be utilized to describe the probability distribution of the time to first failure random variable in the third, increasing phase of the hazard function as well.

Conclusions

Here, we presented the epsilon probability distribution as a new distribution and suggested its possible applications in reliability theory and management. This new distribution is derived from the n^{th} order epsilon differential equation. The solution of the zero order epsilon differential equation is the exponential, while the solution of the first order one is the epsilon function. Some basic properties including continuity, monotonity, limits and convexity were then stated. We also established an important connection between the exponential and the epsilon function; namely, the asymptotic epsilon function is just the exponential function. In practice it means that if $x \ll d$, then $\varepsilon_{\lambda,d} \approx \mathrm{e}^{\lambda x}$; that is, if d is sufficiently large, then the epsilon function approximates the exponential function quite well. The revealed connection between the epsilon function and the Dombi operators in continuous-valued logic leads us to think that the generator function of the Dombi operators may be viewed as a special case of the epsilon function.

It should also be mentioned here that the epsilon probability distribution function is a power function when compared to the transcendent exponential probability distribution function. This feature is advantageous in cases where computation time is a competitive factor.

Focusing on the application of epsilon probability distribution in reliability management, this distribution can be utilized to describe the mortality and useful life period, assuming a typical bathtub-shaped failure rate. Our practical example also suggests that if the second phase of a failure rate function is slightly increasing rather than being constant with respect to time, the epsilon probability distribution will better describe the time to first failure random variable.

As for suggestions for possible future research, it might be interesting to examine the higher order $(n \geq 1)$ epsilon differential equations and their connections with other probability distributions.

Acknowledgement

This study was supported by Flextronics International Ltd, Budapest, Hungary.

References

[1] Waltz, R., Morales, J., Nocedal, J., Orban, D.,: An interior algorithm for nonlinear optimization that combines line search and trust region steps, Mathematical Programming, 2006, Vol. 9, pp. 391-408

[2] Dombi, J.: General class of fuzzy operators, the demorgan class of fuzzy operators and fuzziness included by fuzzy operators, Fuzzy Sets and Systems, 1982, Vol. 8, pp. 149-168

[3] Yuge, T., Maruyama, M., Yanagi, S.: Reliability of a k-out-of-n system with common-cause failures using multivariate exponential distribution, Procedia Computer Science, 2016, Vol. 96, pp. 968–976

[4] Wang, L., Shi, Y.: Reliability analysis of a class of exponential distribution under record values, Journal of Computational and Applied Mathematics, 2013, Vol. 239, pp. 367–379

[5] Balakrishnan, N., Basu, A. P.: The exponential distribution. Theory, Methods and Applications. Gordon and Breach Publishers, Amsterdam, 1995

[6] Kondo, T.: Theory of sampling distribution of standard deviations, Biometrika, 1931, Vol. 22, pp. 31–64

[7] Steffensen, J. F.: Some recent research in the theory of statistics and actuarial science, Cambridge University Press, Cambridge, England, 1930

[8] Weibull, W.: The phenomenon of rupture in solids, Stockholm : Generalstabens Litografiska Anstalts Forlag, 1939

[9] Weibull, W.: A statistical distribution function of wide applicability, Journal of Applied Mechanics, 1951, Vol. 18, pp. 293–297

[10] Davis, D. J.: An analysis of some failure data, Journal of the American Statistical Association, 1952, Vol. 47, pp. 113–150

[11] Rossberg, H. J.: Über die Verteilungsfunktionen der Differenzen und Quotienten von Ranggrössen, Matematische Nachrichten, 1960, Vol. 21, pp. 37–79

[12] Rényi, A.: A characterization of the Poisson process, Magyar Tudományos Akadémia Matematika Kutató Intézet Közleménye, 1956, Vol. 1, pp. 519–527 Translated into English in Selected papers of Alfréd Rényi, vol 1, Akadémiai Kiadó, 1976

[13] Ghurye, S. G.: Characterization of some location and scale parameter families of distributions, In Contributions to Probability and Statistics, 1960, pp. 202–215, Stanford University Press, Stanford, California

[14] Teicher, H.: Maximum likelihood characterization of distributions, Annals of Mathematical Statistics, 1961, Vol. 32, pp. 1214–1222

[15] Galambos, J., Kotz, S.: Characterizations of probability distributions, Lecture Notes in Mathematics, 1978, No. 675., Springer-Verlag, New York

[16] Mudholkar, G.S., Srivastava, D.K.: Exponentiated Weibull family for analyzing bathtub failure-rate data, IEEE Transactions on Reliability, 1993, Vol. 42, pp. 299–302

[17] Rajarshi, S., Rajarshi M. B.: Bathtub distributions: A review, Communications in Statistics Theory and Methods, 1987, Vol. 17, No. 8, pp. 2597–2621

[18] Lai C.D., Xie, M., Murthy, D.N.P.: Bathtub-shaped failure rate life distributions, Handbook of Statistics, 2001, Vol.20, pp. 69-104

[19] Lemonte, A. J.: A new exponential-type distribution with constant, decreasing, increasing, upside-down bathtub and bathtub-shaped failure rate function, Computational Statistics and Data Analysis, 2013, Vol. 62, pp. 149–170

[20] Silva, R. B., Barreto-Souza, W., Cordeiro, G. M.: A new distribution with decreasing, increasing and upside-down bathtub failure rate, Computational Statistics and Data Analysis, 2010, Vol. 54, No. 4, pp. 935–944

[21] Watson, G. F.: MIL reliability: A new approach, IEEE Spectrum, 1992, Vol. 29, pp. 46–49

[22] O'Conor, P. D. T. (Ed.): Practical Reliability Engineering. Wiley, Chichester, England, 2006

NO PAGE NUMBERS

Markowitz Portfolio Selection Using Various Estimators of Expected Returns and Filtering Techniques for Correlation Matrices

András London, Imre Gera, Balázs Bánhelyi

University of Szeged, Institute of Informatics, 6720 Szeged, Árpád tér 2, Hungary
london@inf.u-szeged.hu

Abstract: In this study we examine the performance of the Markowitz portfolio optimization model using stock time series data of various stock exchanges and investment period intervals. Several methods are used to estimate expected returns, then different "noise" filtering techniques are applied on the correlation matrix containing the pairwise correlations of the time series. The performance of the methods is compared using the estimated and realized returns and risks, respectively. The results show that the estimated risk is closer to the realized risk using filtering methods in general. Bootstrap analysis shows that ratio between the realized return and the estimated risk (Sharpe ratio) is also improved by filtering. In terms of the expected return estimation results show that the James-Stein estimator improves the reliability of the portfolio, which means that the realized risk is closer to the estimated risk in this case.

Keywords: Portfolio optimization; Markowitz model; Random matrix theory; Hierarchical clustering

1 Introduction

The portfolio optimization is one of the fundamental problems in asset management that aims to reduce the risk of an investment by diversifying it into assets expected to fluctuate independently [7]. In his seminal work [17], Markowitz formulated the problem as a quadratic programming task: given the expected return of the portfolio, the risk, a quadratic function that is measured via the covariances of the asset time series, has to be minimized. Recently, the investigation of the correlation coefficient matrix, that is a normalization of the covariance matrix appears in the objective function of the model, has received a big amount of attention, see, without being exhaustive, e.g. [4, 6, 12, 13, 22, 24]. The question of quantifying the degree of statistical uncertainty, called "noise" especially in the statistical physics community, present in the correlation matrix and filter the part of information which is robust against this uncertainty has been addressed and tested [4, 10, 12, 13, 14]. Filtered correlation matrices have been successfully used in portfolio optimization for risk reduction [13, 22, 24]. However, most of these studies assumed that the investor has perfect knowledge on the future returns at the time of optimization.

In this study we investigate and test the portfolio optimization problem by using several filtering procedures applied to the correlation matrix given by the pairwise asset correlations. The performance of the procedures is simply measured by comparing the predicted and realized risk and return they provide, respectively. For more details on performance analysis of portfolio selection, see [26] for example. In this work the we assume that future returns are not known at the time of the investment. Moreover, besides the maximum likelihood estimator (i.e. the average of daily returns) we try other methods to calculate the expected returns.

The structure of this paper is the following. In Section 2 we describe the Markowitz portfolio optimization problem together with some possible estimations of the expected returns (Sec. 2.1) and several filtering procedures that can be performed on the correlation matrices (Sec. 2.2). In Section 3 we present our results with the detailed description of data sets, the experimental setup and evaluation metrics we used (Sec. 3.1, Sec. 3.2 and Sec. 3.3). Finally we draw some conclusions and indicate potential future work.

2 Markowitz portfolio optimization model

Given n risky assets, a portfolio composition is determined by the weights p_i $(i = 1,\dots,n)$, such that $\sum_i^n p_i = 1$, indicating the fraction of wealth invested in asset i. The expected return and the variance of the portfolio $\mathbf{p}$ are calculated as

$$r_p = \mathbf{p}\mathbf{r}^T = \sum_{i=1}^{n} p_i r_i \tag{1}$$

and

$$\sigma_p^2 = \mathbf{p}\Sigma\mathbf{p}^T = \sum_{i=1}^{n}\sum_{j=1}^{n} \sigma_{ij} p_i p_j, \tag{2}$$

respectively, where r_i is the expected return of asset i, σ_{ij} is the covariance between asset i and j and Σ is the covariance matrix. Vectors are considered as row vectors in this paper. We should point out that only the proportions $p_1,\dots,p_n$ are needed to determine the performance of the portfolio. It means that the values r_p and σ_p^2 are the same for any investment volume if the weights are the same.

In the classical Markowitz model [17] the risk is measured by the variance providing a quadratic optimization problem consists in finding vector $\mathbf{p}$, such that $\sum_{i=1}^{n} p_i = 1$ which minimizes σ_p^2 for a given "minimal expected return" value of r_p. Here, we assume that short selling is allowed and therefore p_i can be negative. The solution of this problem, found by Markowitz, is

$$\mathbf{p}^* = \lambda\Sigma^{-1}\mathbf{1}^T + \gamma\Sigma^{-1}\mathbf{r}^T, \tag{3}$$

where $\mathbf{1} = (1, \ldots, 1)$ while the other parameters are

$\lambda = (C - r_p B)/D$ and $\gamma = (r_p A - B)/D$

using the notations

$A = \mathbf{1}\Sigma^{-1}\mathbf{1}^T, B = \mathbf{1}\Sigma^{-1}\mathbf{r}^T, C = \mathbf{r}\Sigma^{-1}\mathbf{r}^T$ and $D = AC - B^2$.

However, Eq. 3 is rarely used to solve the Markowitz portfolio optimization problem due to numerical stability problems with matrix inversion [5]. Instead, we used the Lagrange multiplier method for optimization (see Sec. 3.3). Next we will describe three possible methods to calculate the expected stock returns in a given period.

2.1 Estimators for the expected returns

Considering the price time series of n assets and denoting the closure price of asset i in time t $(t = 0, 1, \ldots, T)$ by $P_i(t)$, the daily logarithmic return of i is defined as

$$r_i(t+1) = \log \frac{P_i(t+1)}{P_i(t)} = \log P_i(t+1) - \log P_i(t). \tag{4}$$

In case of stationary independent normal returns (as random variables) the maximum likelihood estimator is the sample mean of the past observations of r_i as it was defined by

$$\hat{r}_i^{ML} = \frac{1}{T} \sum_{t=1}^{T} r_i(t). \tag{5}$$

Hence, for the portfolio we define

$$\hat{\mathbf{r}}_{ML} = (\hat{r}_1^{ML}, \ldots, \hat{r}_n^{ML}), \tag{6}$$

The maximum likelihood return estimation can be highly inefficient since assets with high past returns are likely to contain more positive estimation errors than others. The positive part trimming could further reduce the risk and the James-Stein estimator [11] provides a constructive shrinkage estimator in order to do it. The James-Stein estimation for the expected return for asset i is

$$\hat{\mathbf{r}}_{JS} = (1 - w)\hat{\mathbf{r}}_{ML} + w r_0 \mathbf{1}, \tag{7}$$

where

$$r_0 = \frac{\mathbf{1}\Sigma^{-1}\hat{\mathbf{r}}_{ML}^T}{\mathbf{1}\Sigma^{-1}\mathbf{1}^T}, w = \frac{\lambda}{\lambda + T} \text{ and } \lambda = \frac{(n+2)(T-1)}{(\hat{\mathbf{r}}_{ML} - r_0\mathbf{1})\Sigma^{-1}(\hat{\mathbf{r}}_{ML} - r_0\mathbf{1})^T}.$$

In this calculation, each sample mean is shrunk toward the average return of the minimum variance portfolio r_0.

For small sample size, usually below 50, it was observed that there is no evidence that common asset expected returns are different. If all expected returns are assumed to be equal, the minimum-variance portfolio is efficient and

$$\hat{\mathbf{r}}_{MV} = r_0\mathbf{1}. \tag{8}$$

Finally, the covariance between asset i and j is estimated by the formula

$$\hat{\sigma}^2_{i,j} = \frac{1}{T-1}\sum_{t=1}^{T}(r_i(t)-\hat{r}_i)(r_j(t)-\hat{r}_j), \tag{9}$$

where $\hat{r}_i$ is denotes the estimated value of the with respect the estimator used.

2.2 Filtering the statistical uncertainty

Random matrix theory

The correlation coefficient between asset i and j is defined as $\rho_{ij} = \sigma_{ij}/\sqrt{\sigma_i\sigma_j}$, where $\sigma_i = \sigma_{ii}$ is the standard deviation (often called average volatility) of asset i. A simple random matrix is a matrix whose elements are random numbers from a given distribution [19]. In context of asset portfolios random matrix theory (RMT) can be useful to investigate the effect of statistical uncertainty in the estimation of the correlation matrix [24]. Given the time series of length T of the returns of n assets and assuming that the returns are independent Gaussian random variables with zero mean and unit variance (σ^2=1), in the limit $n \to \infty$, $T \to \infty$ such that $Q = T/n$ is fixed, the distribution $\mathscr{P}_{rm}(\lambda)$ of the eigenvalues of a random correlation matrix ($\mathbf{C}_{rm}$) is given by

$$\mathscr{P}_{rm}(\lambda) = \frac{Q}{2\pi\sigma^2}\frac{\sqrt{(\lambda-\lambda_{\min})(\lambda_{\max}-\lambda)}}{\lambda}, \tag{10}$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are the minimum and maximum eigenvalues, respectively [23], given in the form

$$\lambda_{\max,\min} = \sigma^2(1+\frac{1}{Q}\pm 2\sqrt{\frac{1}{Q}}). \tag{11}$$

Previous studies have pointed out that the largest eigenvalue of correlation matrices from returns of financial assets is completely inconsistent with Eq. 10 and refers the common behavior of the stocks in the portfolio, i.e. the behavior of the market itself. [12, 20]. Since Eq. 10 is strictly valid only for $n \to \infty$, $T \to \infty$, we constructed random matrices for the certain n and T values of the data sets we used and we compared the largest eigenvalues and the spectrum $\mathbf{C}$ and $\mathbf{C}_{rm}$. Since $\text{Trace}(\mathbf{C}) = n$ the variance of the part not explained by the largest eigenvalue can be quantified as $\sigma^2 = 1-\lambda_{\text{largest}}/n$. We can recalculate $\lambda_{\min}$ and $\lambda_{\max}$ in Eq. 11 and construct a filtered diagonal matrix get by setting to zero all eigenvalues of $\mathbf{C}$ smaller than $\lambda_{\max}$ and transform it to the basis of $\mathbf{C}$ with setting the diagonal elements to one. A possible RMT approach for portfolio optimization, following [22], is to use Σ_{rm} (that can be easily calculated form $\mathbf{C}_{rm}$) instead of Σ in the Markowitz model.

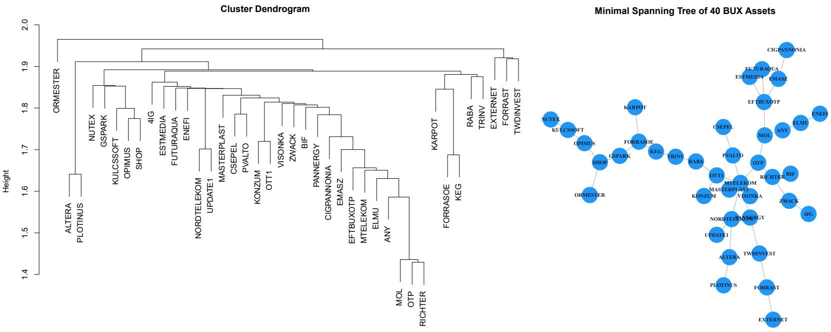


Figure 1
Indexed hierarchical tree - obtained by the single linkage clustering algorithm - and the associated MST of the correlation matrix of 40 assets of the Budapest Stock Exchange

Hierarchical clustering

Correlation based clustering can be considered as a filtering procedure transforming the correlation matrix such that a smaller number of distinct elements retains. The correlation matrix $\mathbf{C}$ has $n(n-1)/2 \sim n^2$ element therefore it contains a large amount of information even for a small number of assets considered in the portfolio. Mantegna and others showed that the single linkage hierarchical clustering algorithm (closely related to minimal spanning trees (MST) of graphs) provide meaningful economic information using only $n-1$ elements of the correlation matrix [15]. The effectiveness of clustering methods have been shown in many studies, e.g. in [2, 9, 18, 25]. To construct the MST, the correlation matrix $\mathbf{C}$ is converted into a distance matrix $\mathbf{D}$, for instance following [15, 16], using $d_{ij} = \sqrt{2(1-\rho_{ij})}$ ultrametric distance. Ultrametric distances are such distances that satisfy the inequality $d_{ij} \leq \max\{d_{ik}, d_{kj}\}$, which is a stronger assumption that the standard triangular inequality. The distance matrix $\mathbf{D}$ can be seen as representing a fully connected graph of the assets with edge weights d_{ij} representing a similarity of the time series of assets i and j. For this graph (i.e. a distance matrix) one can use the Kruskal algorithm in order to obtain the MST of $n-1$ elements and then construct the filtered correlation matrix $\mathbf{C}_{sl}$ using just the $n-1$ correlation coefficients converted back from the $n-1$ distances in the MST. Figure 1 shows an illustrative example for hierarchical clustering and the associated spanning tree obtained using the Budapest Stock Exchange data set. For portfolio optimization, we used Σ_{sl} instead of Σ in the Markowitz model.

In [24] the authors proposed a new portfolio optimization method using another widespread hierarchical clustering procedure, namely the average linkage algorithm. While the single linkage clustering procedure basically follows the greedy Kruskal MST method, the average linkage algorithm, in an iteration step, defines the distance between an element and a cluster as the average distance between the element and each element in the cluster. For detailed description, see e.g. [1]. For portfolio optimization, we can use Σ_{al} constructed using average linkage clustering in the Markowitz model.

3 Results

3.1 Data description

To compare the performance of the methods we analyzed two different data sets. The first data set consists of $n = 40$ stocks traded in the Budapest Stock Exchange (BSE) in the period 1995-2016, using 5145 records of daily returns per stock. The second data set contains the stock time series of $n = 48$ companies of the Information Technology sector (Hardware + Software) that are available on Yahoo Finance (YF) (https://finance.yahoo.com/) in almost the same period as the BSE data with 5395 records of daily returns of each stock.

We consider $t = t_0$ as the time when the optimization is performed. Since the covariance matrix has $\sim n^2$ distinct elements while the number of records used in the estimation is nT, the length of the time series need to be $T >> n$ to get small errors on the covariance. On the other hand, for large T the non-stationarity of the time series more likely appears. The problem is known as the "curse of dimensionality" [27]. To handle this, we computed the covariance matrix and expected returns using the $[-T, 0]$ interval with $T = 50 \approx n$, $T = 100 > n$ and $T = 500 >> n$ days preceding $t = 0$. The calculation of the expected returns, the covariance matrix and filtered covariance matrices was performed using the time series data of this interval. Then, the realized returns and realized risk (for each method) were calculated using the data on the $[0, T]$ interval. To quantify and compare the different methods considered, we used the measures described in the next section.

We should also mention here, that solving the Markowitz portfolio selection method as a quadratic programming problem is particularly simple when Σ (in. Eq. 2) is positive semi-definite and the constraints are equalities (as in Eq. 1). It is not difficult to see that the positive semi-definiteness is true for the original covariance matrix and also for the filtered matrix obtained by the RMT method. In [1] it was proved that the filtered correlation matrix obtained by the single linkage clustering procedure is always positive definite if all the elements of the obtained filtered correlation matrix are positive. This is usually the case for correlations of stock time series and it has been observed for all the matrices we have used. Moreover, it was proved in the same paper that the filtered correlation matrix obtained by using the average linkage clustering method is also positive definite under the same conditions as in the case of the single linkage procedure.

3.2 Performance evaluation

To measure the performance of a portfolio selected by the different models, we use the following measures to investigate how the estimated and the realized quantities relate to each other. For portfolio p, the Sharpe ratio measures the excess return (realized) per unit of risk (estimated):

$$S_p = \frac{r_p - r_f}{\hat{\sigma}_p^2} \tag{12}$$

The portfolio risk, due to the estimation of the correlation matrix is calculated as

$$R_p = \frac{|\hat{\sigma}_p^2 - \sigma_p^2|}{\hat{\sigma}_p^2} \tag{13}$$

where $\hat{\sigma}_p^2$ is the predicted risk, while σ_p^2 is the realized risk of the portfolio.

3.3 Simulation setup and results

We implemented our simulation environment in R [21]. We are given a data set of stock time series and the input parameters the `timeInverval` T, vector of `startingTimes` $\mathbf{t_0} = (t_0^1, \dots, t_0^k)$ and $\mathbf{r_p} = (r_p^1, \dots, r_p^\ell)$ vector of `expectedReturns` (equal steps between the average return and the maximal return over all asset by default). The simulation procedure is done via the following steps:

1. For each starting time t_0^j the `asset.solve.Complete.SelectTimes()` subroutine checks whether the portfolio optimization can be done for that starting time on interval $[-T, t_0^j]$
    - if yes, it calculates the optimal portfolio using `asset.solve.Complete.R()`
    - if not[1], it goes to the next starting time t_0^{j+1}
2. The subroutine stores portfolio weights and the data required for performance evaluation

The subroutine `asset.solve.Complete.R()` works as follows:

1. Determines the expected returns using maximum likelihood, James-Stein and minimum variance estimations
2. Determines the covariance matrix of stock time series
3. Calculates the filtered covariance matrices using the RMT, the single linkage and average linkage procedures
4. Portfolio optimization is performed for each return estimation
    - using the Lagrange multipliers method of the 'Rsolnp' package [8] calculates the optimal weights for each covariance matrix
    - calculates the portfolio risk according to the optimal weights

[1] Usually, data with lots of missing (NA) values results in a singular covariance matrix and optimization cannot be performed

- determines the realized risk and Sharpe-ratio

In order to improve the running times the 'doParallel' R package [3] was used (here we do not describe the details of parallelization).[2]

To check the robustness of the methods, a standard bootstrap experiment was performed. We considered 50 starting times randomly and solved the optimization problem using the time series on the intervals $[-T, t_0^j]$ $(T = 50, 100, 500,\ j = 1, \dots, 50)$. For each portfolio, the predicted risk was calculated according to Eq. 2 for fixed expected returns from the average $\sum_{i=1}^{n} r_i/n$ to the maximum expected return $\max\{r_i : i = 1, \dots, n\}$ with equal spans. The Lagrange multiplier method, that is available in 'Rsolnp' R package, was used for the optimization. In each case, the portfolios with realized returns in the top and bottom 10% were dropped. The realized risk using the determined stock weights at t_0^j, the realized covariance matrix and realized returns were calculated on $[t_0^j, T]$.

Fig. 2 and Fig. 3 show the ratio of the ratio of the realized risk σ_p^2 (continuous line) and the predicted risk $\hat{\sigma}_p^2$ (dashed line) as the function of the expected return r_p obtained by the different procedures for the BSE data set and Yahoo data set, respectively. For each T, the time of the investment t_0^j $(j = 1, \dots, 50)$ and the set of stocks were the same.

For the BSE data set, the classic method and the RMT method provide similar realized returns that are always higher using hierarchical clustering (single and average linkage). On the other hand, the risk ratio R_p (i.e. the reliability of the portfolio) is also significantly decreased (see Fig. 2, and Tab. 1 "Risk Ratio" column), but the deviations of the realized returns were increased. The Sharpe ratio of the hierarchical clustering methods were smaller than using the other methods, since the estimated risk was often higher than the risk obtained when using the classic and the RMT methods. It can be observed that each method provided better expected returns and smaller risk ratio (i.e. better reliability) for the smaller values of T $(T = 50, 100$, see Tab. 1). The results show that the James-Stein return estimation, although it increases the deviation of the realized returns, provides smaller risk ratios and improvements on the Sharpe ratio. The Sharpe-ratio of the minimum variance portfolio (see Tab. 1 last four column) was the highest due to very small expected risk the method estimated, while its reliability is significantly smaller than using the other return estimators.

For the Yahoo data set similar is true for the realized returns as in the case of BSE data set. Here, the smallest risk ratio was obtained when $T = 100$ days (Fig. 3 middle left and right). It can also be observed, that usage the James-Stein return estimator provided better results (realized returns, Sharpe ratio), while the usage minimum variance estimator decreased the risk ratio in some cases.

[2] We used 'doParallel' and its dependencies to create a parallel back end for the loop construction provided by the 'forEach' package.

Conclusions

In this paper, we investigated the Markowitz portfolio selection problem using filtered correlation matrices obtained using different filtering procedures, namely a random matrix theory approach and hierarchical clustering approaches. Furthermore, we used several estimators to determine the expected return of a portfolio. A large set of experiments have shown that using filtered covariance matrices the classic Markowitz solution can be outperformed in terms of realized returns and reliability, meaning that the realized risk and the estimated risk are closer to each other in case of filtering. Our simulations show that the different filtering procedures provide different portfolio optimization results: the most useful method can be different depending on the risk level of the portfolio, the investment period size and reliabilty of the risk and return estimation. We think that other filtering procedures combined with different return estimators could also provide interesting or better results for different parameters (e.g. expected returns, portfolio size, investment period length) of the optimization problem.

References

[1] ANDERBERG, M. R. Cluster analysis for applications. monographs and textbooks on probability and mathematical statistics, 1973.

[2] BASALTO, N., BELLOTTI, R., DE CARLO, F., FACCHI, P., AND PASCAZIO, S. Clustering stock market companies via chaotic map synchronization. *Physica A: Statistical Mechanics and its Applications 345*, 1 (2005), 196–206.

[3] CALAWAY, RICH, W. S., AND TENENBAUM, D. *Foreach Parallel Adaptor for the 'parallel' Package*, 2015. R package version 2.14.

[4] CONLON, T., RUSKIN, H. J., AND CRANE, M. Random matrix theory and fund of funds portfolio optimisation. *Physica A: Statistical Mechanics and its applications 382*, 2 (2007), 565–576.

[5] DU CROZ, J. J., AND HIGHAM, N. J. Stability of methods for matrix inversion. *IMA Journal of Numerical Analysis 12*, 1 (1992), 1–19.

[6] EL ALAOUI, M. Random matrix theory and portfolio optimization in moroccan stock exchange. *Physica A: Statistical Mechanics and its Applications 433* (2015), 92–99.

[7] ELTON, E. J., GRUBER, M. J., BROWN, S. J., AND GOETZMANN, W. N. *Modern portfolio theory and investment analysis*. John Wiley & Sons, 2009.

[8] GHALANOS, A., AND THEUSSL, S. *Rsolnp: General Non-linear Optimization Using Augmented Lagrange Multiplier Method*, 2015. R package version 1.16.

[9] GIADA, L., AND MARSILI, M. Data clustering and noise undressing of correlation matrices. *Physical Review E 63*, 6 (2001), 061101.

[10] GUHR, T., AND KÄLBER, B. A new method to estimate the noise in financial correlation matrices. *Journal of Physics A: Mathematical and General 36*, 12 (2003), 3009.

[11] JAMES, W., AND STEIN, C. Estimation with quadratic loss. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability* (1961), vol. 1, pp. 361–379.

[12] LALOUX, L., CIZEAU, P., BOUCHAUD, J.-P., AND POTTERS, M. Noise dressing of financial correlation matrices. *Physical review letters 83*, 7 (1999), 1467.

[13] LALOUX, L., CIZEAU, P., POTTERS, M., AND BOUCHAUD, J.-P. Random matrix theory and financial correlations. *International Journal of Theoretical and Applied Finance 3*, 03 (2000), 391–397.

[14] MALEVERGNE, Y., AND SORNETTE, D. Collective origin of the coexistence of apparent random matrix theory noise and of factors in large sample correlation matrices. *Physica A: Statistical Mechanics and its Applications 331*, 3 (2004), 660–668.

[15] MANTEGNA, R. N. Hierarchical structure in financial markets. *The European Physical Journal B-Condensed Matter and Complex Systems 11*, 1 (1999), 193–197.

[16] MANTEGNA, R. N., AND STANLEY, H. E. *Introduction to econophysics: correlations and complexity in finance*. Cambridge university press, 1999.

[17] MARKOWITZ, H. Portfolio selection: Efficient diversification of investments. cowles foundation monograph no. 16, 1959.

[18] MASLOV, S. Measures of globalization based on cross-correlations of world financial indices. *Physica A: Statistical Mechanics and its Applications 301*, 1 (2001), 397–406.

[19] MEHTA, M. L. *Random matrices*, vol. 142. Academic press, 2004.

[20] PLEROU, V., GOPIKRISHNAN, P., ROSENOW, B., AMARAL, L. A. N., AND STANLEY, H. E. Universal and nonuniversal properties of cross correlations in financial time series. *Physical Review Letters 83*, 7 (1999), 1471.

[21] R CORE TEAM. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[22] ROSENOW, B., PLEROU, V., GOPIKRISHNAN, P., AND STANLEY, H. E. Portfolio optimization and the random magnet problem. *EPL (Europhysics Letters) 59*, 4 (2002), 500.

[23] SENGUPTA, A. M., AND MITRA, P. P. Distributions of singular values for some random matrices. *Physical Review E 60*, 3 (1999), 3389.

[24] TOLA, V., LILLO, F., GALLEGATI, M., AND MANTEGNA, R. N. Cluster analysis for portfolio optimization. *Journal of Economic Dynamics and Control 32*, 1 (2008), 235–258.

[25] TUMMINELLO, M., ASTE, T., DI MATTEO, T., AND MANTEGNA, R. N. A tool for filtering information in complex systems. *Proceedings of the National Academy of Sciences of the United States of America 102*, 30 (2005), 10421–10426.

[26] URBÁN, A., AND ORMOS, M. Performance analysis of equally weighted portfolios: Usa and hungary. *Acta Polytechnica Hungarica 9*, 2 (2012), 155–168.

[27] ZIMEK, A., SCHUBERT, E., AND KRIEGEL, H.-P. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining 5*, 5 (2012), 363–387.

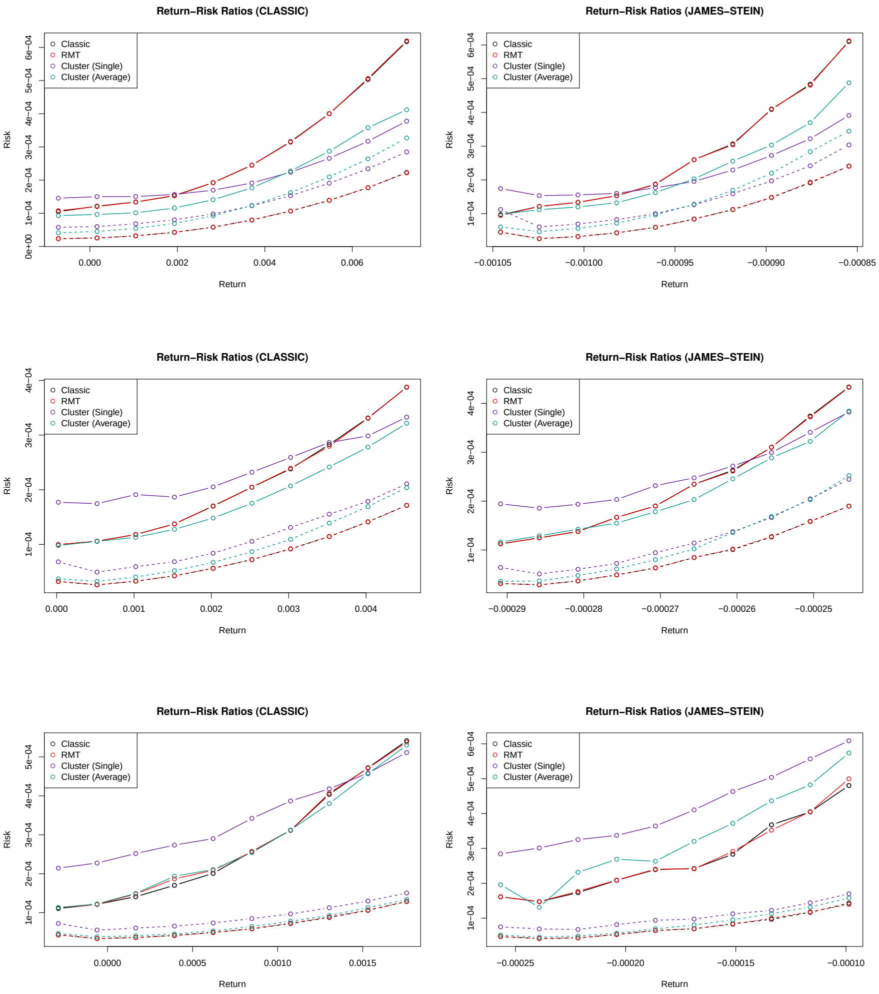


Figure 2

The ratio of the realized risk σ_p^2 and the predicted risk $\hat{\sigma}_p^2$ as the function of expected portfolio return (continuous line) and realized return (dashed line) for the different procedures as $T = 50, 100, 500$ (top-down) using the maximum likelihood estimator (left panels) and the James-Stein estimator (right panel). The data set contains 40 BSE stocks in the period 1995-2016.

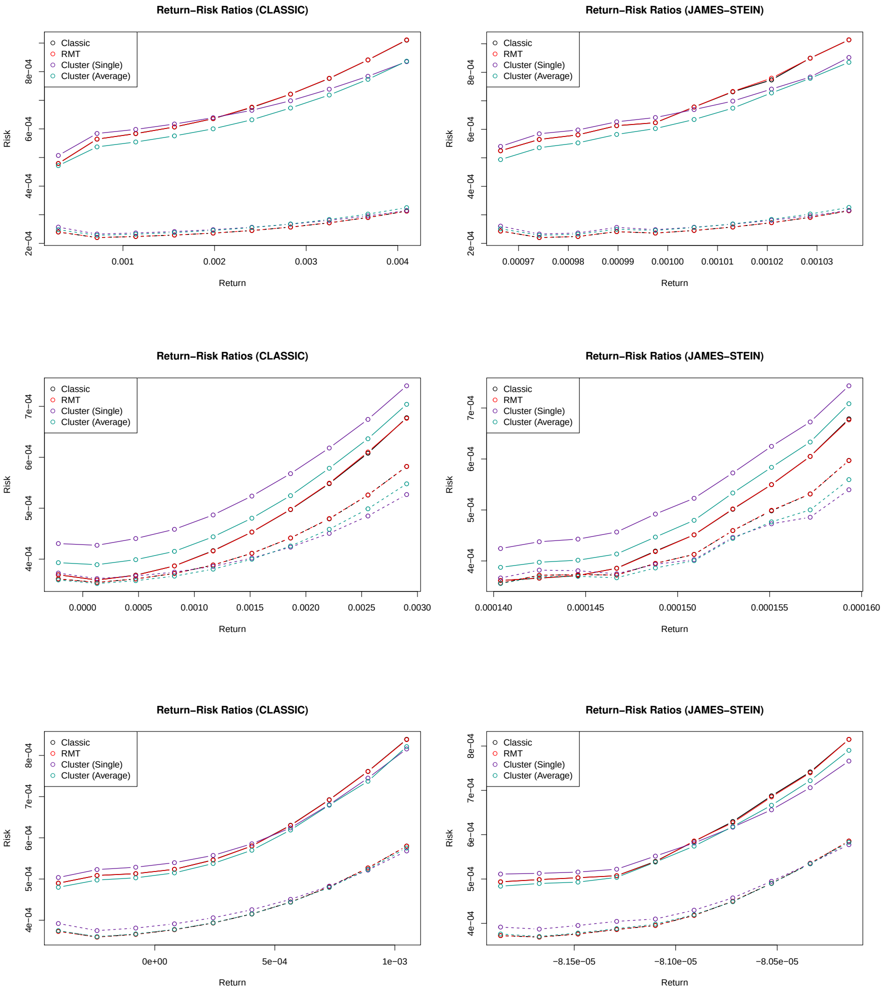


Figure 3
The ratio of the realized risk σ_p^2 and the predicted risk $\hat{\sigma}_p^2$ as the function of expected portfolio return (continuous line) and realized return (dashed line) for the different procedures as $T = 50, 100, 500$ (top-down) using the maximum likelihood estimator (left panels) and the James-Stein estimator (right panel). The data set contains 48 IT sector companies with available historical time series data in the Yahoo finance page in the period 1995-2016

BSE data set		Average return estimator				James-Stein estimator				Min variance estimator			
	Filtering	Realized Return	Realized Return (sd)	Sharpe ratio	Risk Ratio	Realized Return	Realized Return (sd)	Sharpe ratio	Risk Ratio	Realized Return	Realized Return (sd)	Sharpe ratio	Risk Ratio
$T = 50$	Classic	0.00123	0.00465	8.18922	2.30546	0.00117	0.00466	6.85066	2.14136	0.00084	0.00461	298776	3.32719
	RMT	0.00124	0.00465	7.92646	2.31340	0.00118	0.00466	7.14867	2.16424	0.00084	0.00461	214031	3.32716
	Single linkage	0.00121	0.00304	4.77064	0.73042	0.00192	0.00502	11.82008	0.76601	0.00107	0.00452	155939	1.17548
	Average linkage	0.00073	0.00189	0.51529	0.52383	0.00185	0.00475	10.52615	0.68557	0.00100	0.00447	266844	1.34028
$T = 100$	Classic	0.00013	0.00153	7.66392	2.01878	0.00101	0.00328	12.57921	1.99415	0.00070	0.00280	520868	3.52232
	RMT	0.00015	0.00154	7.87615	2.02137	0.00099	0.00325	12.34801	2.00979	0.00069	0.00280	508169	3.52259
	Single linkage	0.00119	0.00294	12.90469	1.44612	0.00164	0.00352	13.56478	1.45191	0.00080	0.00282	406972	2.65070
	Average linkage	0.00017	0.00114	8.95616	1.20554	0.00133	0.00339	13.76679	1.22242	0.00072	0.00280	422477	3.28602
$T = 500$	Classic	-0.00047	0.00121	-11.41798	3.06961	-0.00015	0.00115	-1.61974	2.70563	0.00026	0.00080	23937	4.75457
	RMT	-0.00052	0.00127	-14.16135	3.29680	-0.00016	0.00115	-1.86695	2.72909	0.00026	0.00080	24031	4.75513
	Single linkage	-0.00011	0.00121	-2.19925	2.94493	0.00013	0.00132	1.26614	2.88015	0.00034	0.00077	20362	3.17707
	Average linkage	-0.00053	0.00127	-13.73743	2.96084	-0.00009	0.00108	-0.25504	2.79321	0.00028	0.00075	22544	4.24691
Yahoo data set													
$T = 50$	Classic	-0.00058	0.00335	-3.76019	1.69646	-0.00055	0.00331	-3.46238	1.64100	-0.00075	0.00318	-9.20659	1.44308
	RMT	-0.00057	0.00334	-3.62685	1.69695	-0.00055	0.00331	-3.45527	1.64236	-0.00075	0.00317	-9.19923	1.44350
	Single linkage	-0.00048	0.00335	-0.55588	1.57894	-0.00049	0.00335	-0.55640	1.58649	-0.00074	0.00331	-4.67897	1.45398
	Average linkage	-0.00045	0.00328	0.37105	1.44763	-0.00045	0.00328	0.53901	1.45527	-0.00068	0.00318	-3.80231	1.28830
$T = 100$	Classic	0.00030	0.00223	1.58302	0.07236	0.00033	0.00226	1.64966	0.06094	0.00024	0.00199	0.10471	0.01050
	RMT	0.00030	0.00223	1.59425	0.07074	0.00034	0.00226	1.66608	0.05837	0.00024	0.00199	0.09566	0.01217
	Single linkage	0.00012	0.00213	0.62113	0.26152	0.00014	0.00219	0.55639	0.25039	0.00009	0.00187	-0.46180	0.15097
	Average linkage	0.00022	0.00218	1.08339	0.16704	0.00024	0.00222	1.11380	0.15679	0.00017	0.00194	-0.33963	0.08565
$T = 500$	Classic	-0.00030	0.00070	-1.11800	0.38939	-0.00029	0.00070	-1.05358	0.36550	-0.00024	0.00062	-0.69456	0.38257
	RMT	-0.00030	0.00070	-1.11424	0.38804	-0.00029	0.00070	-1.05042	0.36223	-0.00024	0.00061	-0.69082	0.38035
	Single linkage	-0.00032	0.00069	-1.05861	0.37295	-0.00031	0.00070	-1.03993	0.34634	-0.00027	0.00061	-0.67158	0.36608
	Average linkage	-0.00030	0.00068	-1.08412	0.36562	-0.00029	0.00069	-1.03587	0.35142	-0.00024	0.00059	-0.69585	0.35530

Table 1
Bootstrap experiments using 50 random samples for each value of T when the return is the mean of the average expected return of the portfolio and the maximal expected return over all stocks

On Measures of Dependence Between Possibility Distributions

Robert Fullér[1], István Á. Harmati[2], Péter Várlaki[3]

[1]Department of Informatics
Széchenyi István University, Egyetem tér 1, H-9026, Győr, Hungary
e-mail: rfuller@sze.hu

[2]Department of Mathematics and Computational Sciences
Széchenyi István University, Egyetem tér 1, H-9026, Győr, Hungary
e-mail: harmati@sze.hu

[3]System Theory Lab
Széchenyi István University, Egyetem tér 1, H-9026, Győr, Hungary
e-mail: varlaki@sze.hu

Abstract: A measure of possibilistic correlation between marginal possibility distributions of a joint possibility distribution can be defined as (see Fullér, Mezei and Várlaki, An improved index of interactivity for fuzzy numbers, Fuzzy Sets and Systems*, 165(2011), pp. 56-66) the weighted average of probabilistic correlations between marginal probability distributions whose joint probability distribution is defined to be uniform on the level sets of their joint possibility distribution. Using the averaging technique we shall discuss three quantities (correlation coefficient, correlation ratio and informational coefficient of correlation) which are used to measure the strength of dependence between two possibility distributions. We discuss the inverse problem, as we introduce a method to construct a joint possibility distribution for a given value of possibilistic correlation coefficient. We also discuss a special case when the joint possibility distribution is defined by the so-called weak t-norm and based on these results, we make a conjecture as an open problem for the range of the possibilistic correlation coefficient of any t-norm based joint distribution.*

Keywords: possibility theory, fuzzy numbers, possibilistic correlation, possibilistic dependence.

1 Introduction

Random variables, probability distributions are widely used models of incomplete information [23], and measuring dependence between random variables and random sequences is one of the main tasks of applied probabilty and statistics. There are plenty of measures of dependence, for example correlation coefficients, correlation ration, distance correlation etc.

Possibility distributions are used to model human judgments and preferences and in this way they are models of non-statistical uncertainties. Measuring the strength of dependence between these non-statistical uncertain quantities is quite important, also from theoretical and practical point of view. In probability theory, measures of dependence are usually defined by using the expected value of an appropriate function of the random variables. In possibility theory a measure of possibilistic correlation between marginal possibility distributions of a joint possibility distribution can be defined as the weighted average of probabilistic measures of dependence between marginal probability distributions (fuzzy numbers) whose joint probability distribution is defined to be uniform on the γ-level sets (a.k.a α-cuts) of their joint possibility distribution. This approach gives us a straightforward way to adopt the notions of probability theory to possibility distributions.

The rest of this paper is organized as follows. In Section 2 we recall the basic notions of possibility correlation, in Section 2 we survey some measures of possibilistic dependence. In Section 4 we discuss the inverse problem, i.e we construct a joint possibility distribution for a given correlation coefficient, in Section 5 we discuss the case when the joint possibility distribution is defined by the weak t-norm.

2 Basic Notions of Possibilistic Correlation

Definition 2.1. *A fuzzy number A is a fuzzy set of $\mathbb{R}$ with a normal, fuzzy convex and continuous membership function of bounded support.*

Fuzzy numbers can be viewed as possibility distributions. The concept and some basic properties of joint possibility distribution were introduced in [36].

Definition 2.2. *If $A_1, \ldots, A_n$ are fuzzy numbers, then C is their joint possibility distribution if*

$$A_i(x_i) = \max\{C(x_1, \ldots, x_n) \mid x_j \in \mathbb{R}, j \neq i\} \tag{1}$$

holds for all $x_i \in \mathbb{R}$, $i = 1, \ldots, n$. Furthermore, A_i is called the i-th marginal possibility distribution of C.

As a special case we define the joint possibility distribution of two fuzzy numbers (see Fig. 1), because we investigate the measures of dependence between pairs of fuzzy numbers.

Definition 2.3. *A fuzzy set C in $\mathbb{R}^2$ is said to be a joint possibility distribution of fuzzy numbers A, B, if it satisfies the relationships*

$$A(x) = \max\{C(x, y) \mid y \in \mathbb{R}\}, \qquad \text{and} \qquad B(y) = \max\{C(x, y) \mid x \in \mathbb{R}\}, \tag{2}$$

for all $x, y \in \mathbb{R}$. Furthermore, A and B are called the marginal possibility distributions of C.

Fuzzy numbers $A_1, \ldots, A_n$ are said to be non-interactive if their joint possibility distribution C satisfies the relationship

$$C(x_1, \ldots, x_n) = \min\{A_1(x_1), \ldots, A_n(x_n)\},$$

for all $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ (see Fig. 2).

Definition 2.4. *A γ-level set (or γ-cut) of a possibility distribution C is a non-fuzzy set denoted by $[C]^\gamma$ and defined by*

$$[C]^\gamma = \begin{cases} \{(x,y) \in \mathbb{R}^2 \mid C(x,y) \geq \gamma\} & \text{if } \gamma > 0 \\ \text{cl}(\text{supp}C) & \text{if } \gamma = 0 \end{cases} \tag{3}$$

where $\text{cl}(\text{supp}C)$ *denotes the closure of the support of* C.

3 Measures of Possibilistic Dependence

3.1 Possibilistic Correlation

Carlsson and Fullér introduced a definition of possibilistic mean and variance [2], and then Fullér and Majlander gave the definition of weighted possibilistic mean and variance [9]. Fullér, Mezei and Várlaki introduced a new definition of possibilistic correlation coefficient [10] between marginal distributions of the joint possibility distribution that improves the earlier definition introduced by Carlsson, Fullér and Majlender [3].

Definition 3.1 (see [10])**.** *Let $f\colon [0,1] \to \mathbb{R}$ a non-negative, monotone increasing function with the normalization property $\int_0^1 f(\gamma)\mathrm{d}\gamma = 1$. The f-weighted possibilistic correlation coefficient of fuzzy numbers A and B (with respect to their joint distribution C) is defined by*

$$\rho_f(A,B) = \int_0^1 \rho(X_\gamma, Y_\gamma) f(\gamma)\mathrm{d}\gamma, \tag{4}$$

where

$$\rho(X_\gamma, Y_\gamma) = \frac{\text{cov}(X_\gamma, Y_\gamma)}{\sqrt{\text{var}(X_\gamma)}\sqrt{\text{var}(Y_\gamma)}},$$

and, where X_γ and Y_γ are random variables whose joint distribution is uniform on $[C]^\gamma$ for all $\gamma \in [0,1]$, and $\text{cov}(X_\gamma, Y_\gamma)$ *denotes their probabilistic covariance.*

As we can see, the f-weighted possibilistic correlation coefficient is the f-weighted average of the probabilistic correlation coefficients $\rho(X_\gamma, Y_\gamma)$ for all $\gamma \in [0,1]$. Since f is an increasing function, it gives less importance to the lower levels of the possibility distribution. For detailed and illustrated examples see [8][11] and [12].

The range of the f-weighted possibilistic correlation coefficient when the marginal possibility distribution have the same membership function was discussed in [19] and [17].

Fuzzy numbers A and B are in perfect correlation [3], if their joint distribution is concentrated along a line (see Fig. 3 and Fig. 4), i.e. if there exist $a, b \in \mathbb{R}$, $a \neq 0$ such that their joint possibility distribution is

$$C(x_1, x_2) = \begin{cases} A(x_1) & \text{if } x_2 = ax_1 + b \\ 0 & \text{otherwise} \end{cases}$$

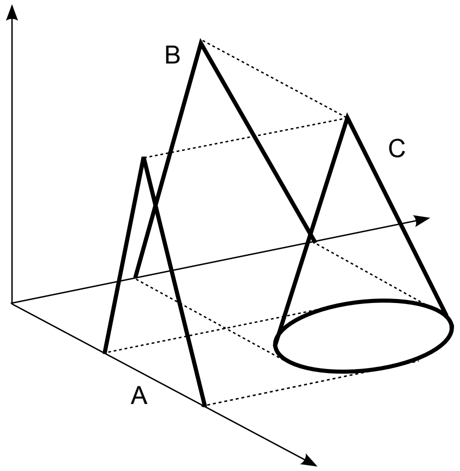


Figure 1: Joint possibility distribution C and its marginal possibility distributions (i.e. projections) fuzzy numbers A and B.

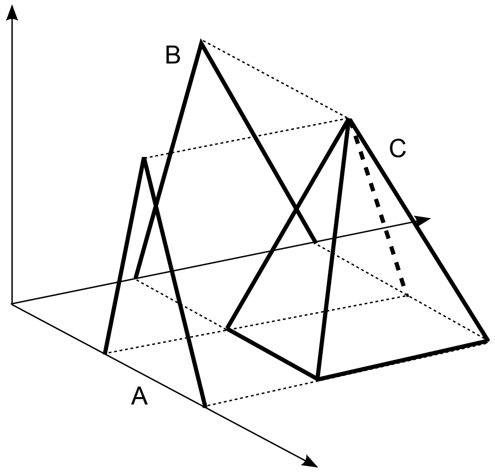


Figure 2: Joint possibility distribution C and its marginal possibility distributions fuzzy numbers A and B when the joint distribution is defined by $\min(A, B)$. In this case A and B are non-interactive which implies $\rho_f(A, B) = 0$.

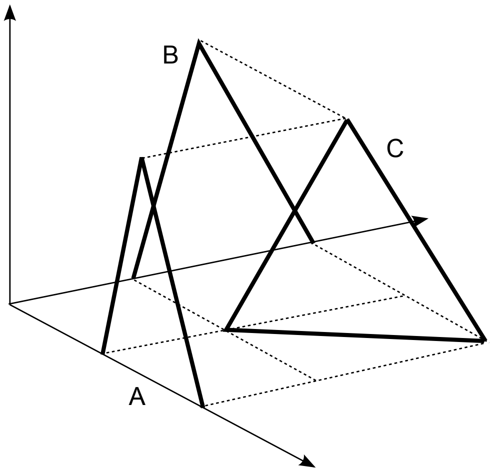


Figure 3: Joint possibility distribution C and its marginal possibility distributions A and B when the joint possibility distribution is defined along a line with positive steepness. This is the case of perfect positive correlation, which implies $\rho_f(A,B) = 1$.

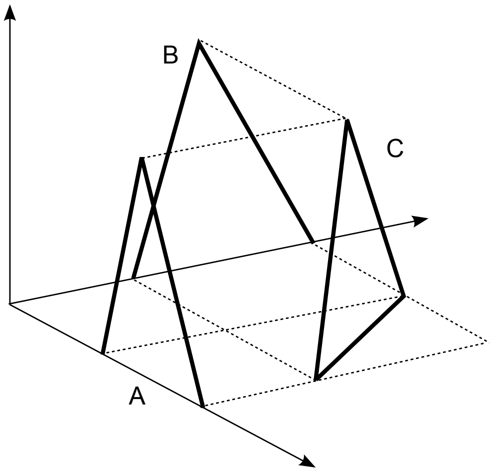


Figure 4: Joint possibility distribution C and its marginal possibility distributions A and B when the joint possibility distribution is defined along a line with negative steepness. This is the case of perfect negative correlation, which implies $\rho_f(A,B) = -1$.

If A and B have a perfect positive (negative) correlation then from $\rho(X_\gamma, Y_\gamma) = 1$ ($\rho(X_\gamma, Y_\gamma) = -1$) (see [3] for details), for all $\gamma \in [0, 1]$, we get $\rho_f(A, B) = 1$ ($\rho_f(A, B) = -1$) for any weighting function f.

We should note here that while non-interactivity implies zero correlation, the reverse direction is not necesseraly true, if the value of possibilistic correlation coefficient is zero then this not means automatically non-interactivity. For example, if for every γ, $[C]^\gamma$ is symmetrical to an axes which parallel with one the coordinate axis then $\mathrm{cov}(X_\gamma, Y_\gamma) = 0$ and $\rho_f(A, B) = 0$ for any weighting function f (see [6]).

3.2 Correlation Ratio Between Fuzzy Numbers

The correlation ratio η was firstly introduced by Karl Pearson [32] as a statistical tool and it was defined to random variables by Kolmogorov [24] as,

$$\eta^2(X|Y) = \frac{D^2[E(X|Y)]}{D^2(X)},$$

where X and Y are random variables. It measures not only a linear, but in general a functional dependence between random variables X and Y. If X and Y have a joint probability density function, denoted by $f(x, y)$, then we can compute $\eta^2(X|Y)$ using the following formulas

$$E(X|Y = y) = \int_{-\infty}^{\infty} x f(x|y)\mathrm{d}x$$

and

$$D^2[E(X|Y)] = E(E(X|y) - E(X))^2,$$

where,

$$f(x|y) = \frac{f(x, y)}{f(y)}.$$

In 2010 Fullér, Mezei and Várlaki introduced the definition of possibilistic correlation ratio for marginal possibility distributions (see [7]).

Definition 3.2. *Let us denote A and B the marginal possibility distributions of a given joint possibility distribution C. Then the* f-weighted possibilistic correlation ratio $\eta_f(A|B)$ *of marginal possibility distribution A with respect to marginal possibility distribution B is defined*

$$\eta_f^2(A|B) = \int_0^1 \eta^2(X_\gamma|Y_\gamma) f(\gamma)\mathrm{d}\gamma$$

where X_γ and Y_γ are random variables whose joint distribution is uniform on $[C]^\gamma$ for all $\gamma \in [0, 1]$, and $\eta(X_\gamma|Y_\gamma)$ denotes their probabilistic correlation ratio.

3.3 Informational Coefficient of Correlation

Definition 3.3. *For any two continous random variables X and Y (admitting a joint probability density), their mutual information is given by*

$$I(X,Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\ln\frac{f(x,y)}{f_1(x)\cdot f_2(y)}\mathrm{d}x\mathrm{d}y$$

where $f(x,y)$ is the joint probability density function of X and Y, and $f_1(x)$ and $f_2(y)$ are the marginal density functions of X and Y, respectively.

Definition 3.4. *[27] For two random variables X and Y, let denote $I(X,Y)$ the mutual information between X and Y. Their informational coefficient of correlation is given by*

$$L(X,Y) = \sqrt{1 - e^{-2I(X,Y)}}\,.$$

Based on the definition above, we can define the following [13][14]:

Definition 3.5. *Let us denote A and B the marginal possibility distributions of a given joint possibility distribution C. Then the f*-weighted possibilistic informational coefficient of correlation *of marginal possibility distributions A and B is defined by*

$$L(A,B) = \int_0^1 L(X_\gamma, Y_\gamma)f(\gamma)\mathrm{d}\gamma$$

where X_γ and Y_γ are random variables whose joint distribution is uniform on $[C]^\gamma$ for all $\gamma \in [0,1]$, and $L(X_\gamma, Y_\gamma)$ denotes informational coefficient of correlation, and f is a weighting function.

There are several other ways to translate the fundamental notions of probability theory to fuzzy numbers (or possibilistic variables), so there are different interpretations for the mean, variance and covariance of fuzzy numbers. Fuzzy random variables are discussed in [26][34] and [33], the variance of fuzzy random variables in [25][31], variance and covariance studied in [5].

Mean value of fuzzy numbers was defined in [4] and [20], the notion of independence is studied in [1], [21] and [35], and with applications in [29], [30].

Liu and Kao [28] used fuzzy measures to define a fuzzy correlation coefficient of fuzzy numbers and they formulated a pair of nonlinear programs to find the α-cut of this fuzzy correlation coefficient, then, in a special case, Hong [22] showed an exact calculation formula for this fuzzy correlation coefficient.

In [15] Fullér et al. introduced a method as a generalization of the concept described in [10]. Here the γ-level sets are equipped with non-uniform probability distribution, whose density function is derived from the joint possibility distribution.

4 Joint Possibility Distribution for Given Correlation

In this section we show a simple way to construct a joint possibility distribution (and in this way marginal possibility distributions) for a given value of the possibilistic correlation coefficient. We recall the fact in probability theory that for any value between -1 and 1 there exists a 2-dimensional Gaussian distribution whose marginal distributions has this value as correlation coefficient between them (for other types of distributions it is not necesseraly true).

Let the required value of the possibilistic correlation coefficient be ρ. Define the joint possibilistic distribution as follows:

$$C(x,y) = \exp\left(\frac{-1}{2(1-\rho^2)} \cdot (x^2 - 2\rho xy + y^2)\right) \tag{5}$$

The γ-level set (remember that $0 < \gamma \leq 1$, so $\ln\gamma \leq 0$):

$$[C]^\gamma = \left\{(x,y) \in \mathbb{R}^2 |\ x^2 - 2\rho xy + y^2 \leq -2(1-\rho^2) \cdot \ln\gamma\right\} \tag{6}$$

The γ-level set is a (maybe skew) ellipse, whose upper and lower curves are

$$y_1 = \rho x + \sqrt{1-\rho^2} \cdot \sqrt{-2\ln\gamma - x^2} \tag{7}$$

$$y_2 = \rho x - \sqrt{1-\rho^2} \cdot \sqrt{-2\ln\gamma - x^2} \tag{8}$$

The area of the γ-levels set is $T_\gamma = -2\pi\sqrt{1-\rho^2} \cdot \ln\gamma$. According to the definition of possibilistic correlation coefficient, we define a two dimensional uniform distribution on the γ-level set, so its density function is

$$f(x,y) = \begin{cases} \dfrac{1}{T_\gamma} & \text{if } (x,y) \in [C]^\gamma \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

X_γ and Y_γ are its marginal random variables. The marginal density function of X_γ (Y_γ has the same one):

$$f_1(x) = \begin{cases} \dfrac{-\sqrt{-2\ln\gamma - x^2}}{\pi \cdot \ln\gamma} & \text{if } -\sqrt{-2\ln\gamma} < x < \sqrt{-2\ln\gamma} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

The expected values are

$$M(X_\gamma) = M(Y_\gamma) = 0 \tag{11}$$

$$M(X_\gamma^2) = M(Y_\gamma^2) = \frac{-\ln\gamma}{2} \tag{12}$$

$$M(X_\gamma \cdot Y_\gamma) = \frac{-\rho \cdot \ln\gamma}{2} \tag{13}$$

So the correlation coefficient at level γ:

$$\rho(X_\gamma, Y_\gamma) = \frac{\mathrm{cov}(X_\gamma, Y_\gamma)}{\sqrt{\mathrm{var}(X_\gamma)}\sqrt{\mathrm{var}(Y_\gamma)}} = \frac{-\rho \cdot \ln \gamma/2}{-\ln \gamma/2} = \rho \tag{14}$$

Since the value of ρ not depends on γ, the value of possibilistic correlation equals this value:

$$\rho_f(A, B) = \int_0^1 \rho(X_\gamma, Y_\gamma) f(\gamma) \mathrm{d}\gamma = \rho \int_0^1 f(\gamma) \mathrm{d}\gamma = \rho \tag{15}$$

Note 4.1. *In fact the starting point was a two dimensional Gaussian probability density function:*

$$f(x, y) = \frac{1}{2\pi\sqrt{1-\rho^2}} \cdot \exp\left(\frac{-1}{2(1-\rho^2)} \cdot (x^2 - 2\rho xy + y^2)\right) \tag{16}$$

, where ρ is the correlation coefficient between the marginal random variables. So the result we get tells us that the possibilistic and probabilistic correlation coefficient could be the same for certain cases.

5 Correlation Coefficient for t-norm Defined Joint Distributions

An interestinq question is the range or behaviour of the possibilistic correlation coefficient when the joint possibility distribution has a special structure, i.e. it is defined by a t-norm. According to our best knowledge there are no simple general results to this problem. For the most widely used t-norm, the minimum t-norm the answer is straightforward, since this is the case when the marginal distributions (fuzzy numbers) are in non-interactive relation and this fact ensures zero correlation coefficient.

The case when the joint possibility distribution is defined by the product t-norm was discussed in [12], where the authors pointed out that the value of the possibilistic correlation falls between $-1/2$ and $1/2$, including the limits.

Well-known that the following inequality holds for any t-norm:

$$T_w(a, b) \leq t(a, b) \leq \min(a, b) \tag{17}$$

where T_w denotes the weak (or drastic) t-norm:

$$T_w(a, b) = \begin{cases} \min(a, b) & \text{if } \max(a, b) = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

In the following we give strict bounds for the possibilistic correlation coefficients when the joint distribution $C(x, y) = T_w(A(x), B(y))$, where A and B are the marginal distributions.

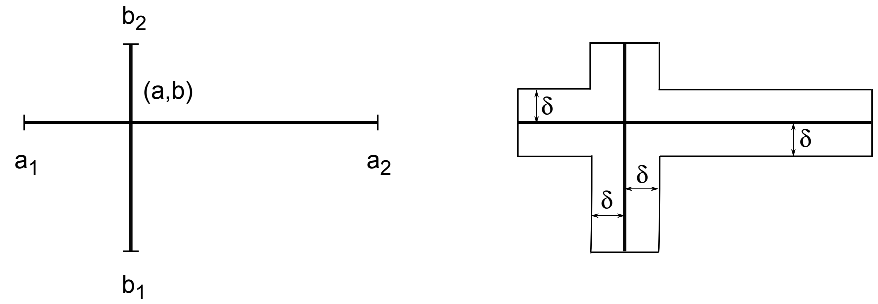


Figure 5: The γ level set of the joint distribution, when $C(x,y) = T_w(A(x), B(y))$ (left), and its δ neighborhood (right).

Let us denote the core of fuzzy number A by a, the core of B by b, the γ level sets by $[a_1(\gamma), a_2(\gamma)]$ and $[b_1(\gamma), b_2(\gamma)]$, respectively. For simplicity we use the notations: $a_1 = a_1(\gamma)$, $a_2 = a_2(\gamma)$, $b_1 = b_1(\gamma)$ and $b_2 = b_2(\gamma)$. The γ -level sets ($[C]^\gamma$s) of the joint distribution are (not necessarily symmetric) cross-shaped domains (see Fig. 5). The correlation coefficient for this domain is determined as the limit of the correlation coefficient computed for the δ neighborhood ($[C]^\gamma_\delta$) (see Fig. 5). Since the correlation coefficient is invariant under shifting and scaling (multiplying by a positive constant) of the marginal distributions, without loss of generality we can assume that $a = b = 0$, and $-1 \leq a_1, b_1 \leq 0$, $0 \leq a_2, b_2 \leq 1$, such that at least one of the following conditions hold: $a_1 = -1$, $b_1 = -1$, $a_1 = -1$, $b_2 = 1$, $a_2 = 1$, $b_2 = 1$ or $a_2 = 1$, $b_1 = -1$. These conditions are always feasible: we shift the cores to the origin, then rescale A by $\max\{|a_1|, a_2\}$ and B by $\max\{|b_1|, b_2\}$.

X_γ and Y_γ are random variables, whose joint distribution is uniform on $[C]^\gamma_\delta$, the corresponding random variables for $[C]^\gamma$ are X'_γ and Y'_γ.

The probability density function of X_γ (Y_γ has the same with appropriate modification of the parameters):

$$f_1(x) = \begin{cases} \dfrac{1}{T} \cdot 2\delta & \text{, if } a_1 < x < -\delta; \\ \dfrac{1}{T} \cdot (b_2 - b_1) & \text{, if } -\delta < x < \delta; \\ \dfrac{1}{T} \cdot 2\delta & \text{, if } \delta < x < a_2. \end{cases}$$

where $T = (a_2 - a_1) \cdot 2\delta + (b_2 - b_1) \cdot 2\delta - 4\delta^2$ denotes the area of $[C]^\gamma_\delta$.

Computing the expected values and after some simplifications we get:

$$M(X_\gamma) = \frac{a_2^2 - a_1^2}{2(a_2 - a_1 + b_2 - b_1) - 4\delta} \tag{19}$$

$$M(X'_\gamma) = \lim_{\delta \to 0} M(X_\gamma) = \frac{a_2^2 - a_1^2}{2(a_2 - a_1 + b_2 - b_1)} \tag{20}$$

$$M(X_\gamma^2) = \frac{2}{3} \cdot \frac{a_2^3 - a_1^3 + (b_2 - b_1)\delta^2 - 2\delta^3}{2(a_2 - a_1 + b_2 - b_1) - 4\delta} \tag{21}$$

$$M(X_\gamma'^2) = \lim_{\delta \to 0} M(X_\gamma^2) = \frac{2}{3} \cdot \frac{a_2^3 - a_1^3}{2(a_2 - a_1 + b_2 - b_1)} \tag{22}$$

Similar expressions hold for Y_γ with appropriate modification of the parameters, of course. The expected value of the product:

$$M(X_\gamma \cdot Y_\gamma) = 0 \quad \Rightarrow \quad M(X'_\gamma \cdot Y'_\gamma) = 0 \tag{23}$$

The correlation coefficient between X'_γ and Y'_γ:

$$\rho(X'_\gamma, Y'_\gamma) = \frac{\operatorname{cov}(X'_\gamma, Y'_\gamma)}{\sqrt{\operatorname{var}(X'_\gamma)} \cdot \sqrt{\operatorname{var}(Y'_\gamma)}} \tag{24}$$

where

$$\operatorname{cov}(X'_\gamma, Y'_\gamma) = M(X'_\gamma \cdot Y'_\gamma) - M(X'_\gamma) \cdot M(Y'_\gamma) = \frac{-(a_2^2 - a_1^2)(b_2^2 - b_1^2)}{4(a_2 - a_1 + b_2 - b_1)^2} \tag{25}$$

$$\operatorname{var}(X'_\gamma) = \frac{\frac{4}{3} \cdot (a_2^3 - a_1^3) \cdot (a_2 - a_1 + b_2 - b_1) - (a_2^2 - a_1^2)^2}{4(a_2 - a_1 + b_2 - b_1)^2} \tag{26}$$

$$\operatorname{var}(Y'_\gamma) = \frac{\frac{4}{3} \cdot (b_2^3 - b_1^3) \cdot (a_2 - a_1 + b_2 - b_1) - (b_2^2 - b_1^2)^2}{4(a_2 - a_1 + b_2 - b_1)^2} \tag{27}$$

We prove that the value of the above correlation coefficient always falls between $-3/5$ and $3/5$. Let's consider the case when $a_2 = 1$ and $b_2 = 1$ (the estimation works quite similarly for the other three cases). We give a lower estimation for the variances using the fact that $-1 \le a_1 \le 0$ and $-1 \le b_1 \le 0$, so we get an upper estimation for the correlation coefficient. The numerator of $\mathrm{var}(X'_\gamma)$:

$$\frac{4}{3}\cdot(1-a_1^3)\cdot(2-a_1-b_1)-(1-a_1^2)^2 \tag{28}$$

$$\geq \frac{4}{3}\cdot(1-a_1^2)^2\cdot(2-a_1-b_1)-(1-a_1^2)^2 \tag{29}$$

$$=(1-a_1^2)^2\cdot\left[\frac{4}{3}\cdot(2-a_1-b_1)-1\right] \tag{30}$$

$$\geq (1-a_1^2)^2\cdot\left[\frac{4}{3}\cdot 2-1\right]=(1-a_1^2)^2\cdot\frac{5}{3} \tag{31}$$

Applying this result we get that

$$\mathrm{var}(X'_\gamma)\geq\frac{(1-a_1^2)^2\cdot\frac{5}{3}}{4(a_2-a_1+b_2-b_1)^2} \tag{32}$$

So we get the following bounds for the square of the correlation coefficient:

$$\rho^2\leq\frac{(1-a_1^2)^2(1-b_1^2)^2}{(1-a_1^2)^2\cdot\frac{5}{3}\cdot(1-b_1^2)^2\cdot\frac{5}{3}}=\frac{9}{25} \tag{33}$$

which yields:

$$-3/5\leq\rho\leq 3/5 \tag{34}$$

These bounds are strict, since

- if $a_2=b_2=0$ and $a_1=b_1=-1$, then $\rho=-3/5$;
- if $a_2=b_1=0$ and $a_1=-1$, $b_2=1$, then $\rho=3/5$.

Remember that the possibilistic correlation coefficient was defined as the weighted average of probabilistic correlation coefficients over the γ levels. We proved that for every γ level set $-3/5\leq\rho(X_\gamma,Y_\gamma)\leq 3/5$, so these inequality holds for the possibilistic correlation coefficient for any weighting function f:

$$-3/5\leq\rho_f(A,B)\leq 3/5 \tag{35}$$

Our numerical and theoretical investigations done so far led us to a conjecture that the weak t-norm has a kind of boundary role here, which is still an open problem:

Question 5.1. *Is it true that for any joint possibility distribution defined by a t-norm, the possibilistic correlation coefficient falls between $-3/5$ and $3/5$?*

Conclusions

We briefly surveyed the developments of probability related γ-level based possibilistic measures of dependence. This level-based approach gives a useful tool to directly generalize the notions of probability theory to possibilistic variables and it may make a bridge between possibilistic and probabilistic ways of thinking. Although this connection gaves us a chance to adopt the results of probability theory, there are still many open questions.

We gave a short general solution to the inverse problem, namely we showed a family of joint possibility distributions to any given value of correlation. We determined the range of possibilistic correlation coefficient when the joint distribution is defined by the weak t-norm. Finally, we stated an open problem for the family of t-norm defined joint possibility distribution.

Acknowledgement

This work was supported by FIEK program (Center for Cooperation between Higher Education and the Industries at the Széchenyi István University, GINOP-2.3.4-15-2016-00003).

References

[1] L. M. de Campos, J. F. Huete, Independence concepts in possibility theory: Part I, *Fuzzy Sets and Systems* 103 (1999), pp. 127-152.
Part II. *Fuzzy Sets and Systems* 103 (1999), pp. 487-505.

[2] C. Carlsson, R. Fullér, On possibilistic mean and variance of fuzzy numbers, *Fuzzy Sets and Systems* 122 (2001), pp. 315-326.

[3] C. Carlsson, R. Fullér and P. Majlender, On possibilistic correlation, *Fuzzy Sets and Systems*, 155(2005) 425-445. doi: 10.1016/j.fss.2005.04.014

[4] D. Dubois and H. Prade, The mean value of a fuzzy number, *Fuzzy Sets and Systems*, 24(1987), pp. 279-300. doi: 10.1016/0165-0114(87)90028-5

[5] Y. Feng, L. Hu, H. Shu, The variance and covariance of fuzzy random variables and their applications, *Fuzzy Sets and Systems* 120 (2001), pp. 487-497.

[6] R. Fullér and P. Majlender, On interactive possibility distributions, in: V.A. Niskanen and J. Kortelainen eds., *On the Edge of Fuzziness, Studies in Honor of Jorma K. Mattila on His Sixtieth Birthday*, Acta universitas Lappeenrantaensis, No. 179, 2004 61-69.

[7] R. Fullér, J. Mezei and P. Várlaki, A Correlation Ratio for Possibility Distributions, in: E. Hüllermeier, R. Kruse, and F. Hoffmann (Eds.): *Computational*

Intelligence for Knowledge-Based Systems Design, Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2010), June 28 - July 2, 2010, Dortmund, Germany, Lecture Notes in Artificial Intelligence, vol. 6178(2010), Springer-Verlag, Berlin Heidelberg, pp. 178-187. doi: 10.1007/978-3-642-14049-5_19

[8] R. Fullér, J. Mezei and P. Várlaki, Some Examples of Computing the Possibilistic Correlation Coefficient from Joint Possibility Distributions, in: Imre J. Rudas, János Fodor, Janusz Kacprzyk eds., *Computational Intelligence in Engineering*, Studies in Computational Intelligence Series, vol. 313/2010, Springer Verlag, [ISBN 978-3-642-15219-1], pp. 153-169. doi: 10.1007/978-3-642-15220-7_13

[9] R. Fullér, P. Majlender, On weighted possibilistic mean and variance of fuzzy numbers, *Fuzzy Sets and Systems* 136 (2003), pp. 363-374.

[10] R. Fullér, J. Mezei, P. Várlaki, An improved index of interactivity for fuzzy numbers, *Fuzzy Sets and Systems* 165 (2011), pp. 50-60.

[11] R. Fullér, I. Á. Harmati, J.Mezei, P. Várlaki, On Possibilistic Correlation Coefficient and Ratio for Fuzzy Numbers, in: *Recent Researches in Artificial Intelligence, Knowledge Engineering & Data Bases, 10th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, February 20-22, 2011, Cambridge, UK, WSEAS Press, [ISBN 978-960-474-237-8], pp. 263-268.

[12] R. Fullér, I. Á. Harmati, P. Várlaki, On Possibilistic Correlation Coefficient and Ratio for Triangular Fuzzy Numbers with Multiplicative Joint Distribution, in: Proceedings of the Eleventh IEEE International Symposium on Computational Intelligence and Informatics (CINTI 2010), November 18-20, 2010, Budapest, Hungary, [ISBN 978-1-4244-9278-7], pp. 103-108. DOI 10.1109/CINTI.2010.5672266

[13] R. Fullér, I. Á. Harmati, P. Várlaki, I. Rudas, On Informational Coefficient of Correlation for Possibility Distributions, Recent Researches in Artificial Intelligence and Database Management, Proceedings of the 11th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases (AIKED '12), February 22-24, 2012, Cambridge, UK, [ISBN 978-1-61804-068-8], pp. 15-20.

[14] R. Fullér, I. Á. Harmati, P. Várlaki, I. Rudas, On Weighted Possibilistic Informational Coefficient of Correlation, International Journal of Mathematical Models and Methods in Applied Sciences, 6(2012), issue 4, pp. 592-599.

[15] R. Fullér, I. Á. Harmati, P. Várlaki, Probabilistic Correlation Coefficients for Possibility Distributions, Fifteenth IEEE International Conference on Intelligent Engineering Systems 2011 (INES 2011), June 23-25, 2011, Poprad, Slovakia, [ISBN 978-1-4244-8954-1], pp. 153-158. DOI 10.1109/INES.2011.5954737

[16] R. Fullér, I. Á. Harmati, P. Várlaki, On Probabilistic Correlation Coefficients for Fuzzy Numbers, in: Aspects of Computational Intelligence: Theory and Applications: Revised and Selected Papers of the 15th IEEE International Conference on Intelligent Engineering Systems 2011, INES 2011, Topics in Intelligent Engineering and Informatics series, vol. 2/2013, Springer Verlag, [ISBN:978-3-642-30667-9], 2013. pp. 249-263. DOI 10.1007/978-3-642-30668-6_17

[17] R. Fullér, I. Á. Harmati, On the lower limit of possibilistic correlation coefficient for identical marginal distributions, 8th European Symposium on Computational Intelligence and Mathematics (ESCIM 2016), 2016, Cádiz, pp. 37-42.

[18] H. Gebelein, Das satistische Problem der Korrelation als Variations- und Eigenwertproblem und sein Zusammenhang mit der Ausgleichungsrechnung, *Zeitschrift fr angew. Math. und Mech.*, 21 (1941), pp. 364-379.

[19] I. Á. Harmati, A note on f-weighted possibilistic correlation for identical marginal possibility distributions, *Fuzzy Sets and Systems*, 165(2011), pp. 106-110. doi: 10.1016/j.fss.2010.11.005

[20] S. Heilpern, The expected value of a fuzzy number, *Fuzzy Sets and Systems* 47 (1992), pp. 81-86.

[21] E. Hisdal, Conditional possibilities independence and noninteraction, *Fuzzy Sets and Systems* 1 (1978), pp. 283-297.

[22] D.H. Hong, Fuzzy measures for a correlation coefficient of fuzzy numbers under T_W (the weakest t-norm)-based fuzzy arithmetic operations, *Information Sciences*, 176(2006), pp. 150-160.

[23] E.T. Jaynes, *Probability Theory : The Logic of Science*, Cambridge University Press, 2003.

[24] A.N. Kolmogorov, Grundbegriffe der Wahrscheinlichkeitsrechnung, Julius Springer, Berlin, 1933, 62 pp.

[25] R. K. Körner, On the variance of fuzzy random variables, *Fuzzy Sets and Systems* 92 (1997), pp. 8393.

[26] H. Kwakernaak, Fuzzy random variables–I. Definitions and theorems, *Information Sciences* 15 (1978), pp. 1-29.

[27] E. H. Linfoot, An informational measure of correlation, *Information and Control*, Vol.1, No. 1 (1957), pp. 85-89.

[28] S.T. Liu, C. Kao, Fuzzy measures for correlation coefficient of fuzzy numbers, *Fuzzy Sets and Systems*, 128(2002), pp. 267-275.

[29] X. Li, B. Liu, The independence of fuzzy variables with applications, *International Journal of Natural Sciences & Technology* 1 (2006), pp. 95-100.

[30] Y. K. Liu, J. Gao, The independence of fuzzy variables with applications to fuzzy random optimization, *International Journal of Uncertainty, Fuzziness & Knowledge-Based Systems* 15 (2007), pp. 1-19.

[31] W. Näther, A. Wünsche, On the Conditional Variance of Fuzzy Random Variables, *Metrika* 65 (2007), pp. 109-122.

[32] K. Pearson, On a New Method of Determining Correlation, when One Variable is Given by Alternative and the Other by Multiple Categories, Biometrika, Vol. 7, No. 3 (Apr., 1910), pp. 248-257.

[33] M. L. Puri, D. A. Ralescu, Fuzzy random variables, *Journal of Mathematical Analysis and Applications* 114 (1986), pp. 409-422.

[34] A. F. Shapiro, Fuzzy random variables, *Insurance: Mathematics and Economics* 44 (2009), pp. 307314.

[35] S. Wang, J. Watada, Some properties of T-independent fuzzy variables, *Mathematical and Computer Modelling* 53 (2011), pp. 970-984.

[36] L. A. Zadeh, Concept of a linguistic variable and its application to approximate reasoning I, II, III *Information Sciences* 8 (1975), pp. 199-249, pp. 301-357;
L. A. Zadeh, Concept of a linguistic variable and its application to approximate reasoning I, II, III *Information Sciences* 9 (1975), pp. 43-80.