Production Systems and Information Engineering
Volume 4 (2006), pp. 3–19

COOPERATIVE PLANNING BY COORDINATING THE SUPPLY CHANNEL

PÉTER EGRI, JÓZSEF VÁNCZA
Computer and Automation Research Institute
Hungarian Academy of Sciences
H-1111 Budapest, Kende u. 13-17 HUNGARY
{egri,vancza}@sztaki.hu

[Received January 2006 and Accepted May 2006]

Abstract. Cooperation between manufacturing enterprises has recently become widespread in order to cope with newly emerged challenges, such as growing customer expectations, increasing product variety and decreasing product lifecycles. Nowadays the response to these challenges seems to be the formation of tight relations in supply chains and networks, which enables joint handling of market risks and involves sharing benefits and mutual growth. Our current work studies this situation in case of high manufacturing setup costs, which inspire enterprises to produce in large sized lots, which could in turn, lead to obsolete inventories on unstable markets. We propose a method for coordinating supply channels on such markets and a framework for cooperative planning.

Keywords: Coordination, cooperation, supply chain management

1. Introduction and motivation

Today's mass customized production (typically of consumer goods like low-tech electronics, mobile phones, light bulbs, cosmetics, etc.) aims at providing large product diversity and relatively cheap and simple manufacturing—both in small and large quantities—from standard components. Unfortunately, this policy usually induces either longer throughput times, lower service levels or higher inventories. On the other hand, customer expectations are permanently growing (e.g., acceptable delivery times are shorter and shorter), therefore a trade-off has to be found between the conflicting objectives. Shorter product life-cycles also follow from customization, which causes further problems [13].

Such markets are typically served by competing *supply network*s which consist of autonomous entities, most of them being engaged in more network

relations. In a particular network, hardly predictable customer demand must be anticipated and satisfied directly by a manufacturer of end-products that works in the focal point of the network, while other members supply the manufacturer with necessary components including packaging materials. In order to cope with the above mentioned challenges, standard, autonomous ways of improving competitiveness—such as reducing inventory levels and raising resource utilization—should be extended to the network level and enhanced by cooperation [11, 16].

Though the ultimate goal of production is to satisfy actual customer orders, all partners are forced to apply also make-to-stock strategies so that they can

1. meet demand in time,
2. satisfy some constraints of mass production technology, and
3. exploit economics of scale.

Hence, it is inevitable to produce even customized products on the basis of forecasts and to keep inventories both from products and components to hedge against uncertainties of demand. However, just due to the very nature of the market, from certain products or components obsolete inventories may easily remain, which cannot be sold or used any more. An alternative way is to sustain capacity buffers, but this certainly incurs extra equipment and labor costs, which in most cases exceed the cost of holding inventory.

The motivation of this work comes from a large-scale national industrial-academical R&D project aimed at realizing real-time, cooperative enterprises [12]. Our particular aim is to develop planning methods that improve the overall logistic and production performance of a supply network involved in mass production of customized consumer goods. Though, the solution should be generic: we focus on the problem of how a focal network as a whole can guarantee short delivery time and high service level while keeping its logistics costs as low as possible. We do not tackle the issue of making forecasts on the market of customized mass products. When modelling autonomous partners, details of how they organize their own production is not dealt with, either. We assume that each partner does its best when planning and scheduling its internal operations, and takes also responsibility for the quality and execution of its plans. Even so, there is an inevitable need to coordinate their logistics and production related decisions. For reasons discussed below, we prefer coordination models that facilitate and sustain *cooperation* among network members.

The proposed method is based not on the extension of local planning and control processes but rather on the extension of information access and decision rights of the partners. This is accompanied by extended responsibilities, too.

In general terms, the flow of information, commodity and currency between the partners is regulated by *contracts*. Our interest is in designing such protocols and decision models that are applicable under realistic conditions and help to find a common, acceptable performance trade-off for all members of focal supply networks.

In the sequel we give a short overview of related works, and then specify the requirements towards a cooperation mechanism. Section 4 presents an analytical model for coordinating the channel between the manufacturer and its suppliers. Next, this model is embedded into a cooperative planning mechanism. Section 6 discusses our industrial case study and summarizes simulation results on historical data sets. Finally, subsequent steps of our research program are presented and conclusions are drawn.

2. Related work

There exists a number of approaches that provide technology for information sharing in a supply network. However, these *supply chain management* (SCM) systems are mostly transactional and do not really support coordinated decision making [7, 14]. So-called *advanced planning systems* (APS) are already applicable to solve—even in a close-to optimal way—production planning and scheduling problems locally, at the nodes of a network [17], but still there is a lack of comprehension on how to coordinate local, distributed planning processes in case of firms whose primary objective is their own profit [14]. Since performance criteria are conflicting both at the individual partners and at the network level, local optimization may even adversely affect the system's performance—a phenomenon know for long as *double marginalization* [15].

When the manufacturer and its supplier make plans and decisions independently, the system can deviate from optimum and effect poor performance. So-called *channel coordination* is achieved when the manufacturer and its supplier make local decisions so that their joint profit is maximized. This is what could be produced by a centralized system (e.g., a so-called *virtual enterprise*), but it can be carried out also in case of autonomous enterprises that contract on the payment, if each firm's objective is aligned with the supply chain's overall objective [3]. Contracts that associate decision rights with appropriate incentives are just for accommodating different and disparate objectives. There is a variety of contracts both in the theory and practice of supply chains that strive to achieve good system performance while keeping the manufacturer-supplier relation flexible. While contracts in the practice are usually too complex for analytical modelling, most theoretical models work in a time-invariant, single-period setting. A general coordination framework based on *option*s is presented in [1], where several contract types (e.g., quantity flexibility) are

proved to be special cases of options. However, this model is applicable only for short-term coordination, because its horizon includes no more than two periods.

In any case, an integral part of coordination is to decide how much to produce from particular products and components at a given moment. The planning of *lot sizes* is well studied in the literature. Lot sizing problems (LSP) can be classified according to several criteria: granularity of time (continuous or discrete), number of production levels (single or hierarchical), length of production horizon (finite or infinite), capacity constraint (capacitated or uncapacitated), objective function (total or average cost), inventory limits, etc. For example, the widely used Economic Order Quantity (EOQ) model is continuous, uncapacitated, minimizes average cost and can be computed easily. On the other hand, the (improved) uncapacitated Wagner–Whitin method considers a finite and discrete horizon, minimizes the total cost and the optimal lot sizes can be computed in $\mathcal{O}(n \log n)$ time, where n is the length of the horizon. However, realistic variants of LSPs are usually NP-hard problems [2].

Stochastic inventory policies can handle uncertainty in case of demand volatility (such as the (R, Q) policy) and one-period uncertain demand (*newsvendor* model) [3], but the unexpected termination of the demand is still missing. In [4] a lot sizing model is introduced with imperfect demand forecasts, on a rolling horizon basis. In this situation, the decision is related only to the period right after the decision time, and then the horizon is rolled forward with the updated demand. It is shown, that this usually leads to "system nervousness": the altered demand in a later period could cause additional costs. This model includes multiple items, capacity limits and setup costs too. However, because of the generality of the model, even the approximate version has high computational complexity and works only on small-sized problems.

A common example of cooperation is *Vendor Managed Inventory* (VMI), which is a special *one-point-inventory* system, in which the supplier decides the appropriate inventory policies to manage the manufacturer's inventory, based on the manufacturer's forecasts. In a focal supply network, a manufacturer may not maintain inventory at all. Instead, it gives only forecasts and suppliers have to decide the production quantities and store the goods until the manufacturer needs and calls them off—this is the so-called *consignment VMI* [9]. In [5] a VMI implementation is described through a case study of a household electrical appliance manufacturer. As it was observed, VMI could operate much better than the traditional replenishment system, even if demand was highly unpredictable. However, it requires organizational changes, mighty trust and advanced information sharing between enterprises.

Relations between enterprises can be represented on a range of colors: from cold (competitive auctions, single business relations), through warm (cooperative planning), to hot (full integration). Although relations between manufacturers and end customers are usually "cold", the relationships with suppliers (*upstream firms*) are usually richer and more complex [15]. In a conventional, non-cooperative manufacturer-supplier relation, the manufacturer orders components and pays proportionally to the quantity delivered. It can be shown, that in such a situation, uncertainty is amplified due to safety stocks as we traverse upwards the chain (the so-called *bullwhip effect*) [10]. This deteriorates competitiveness of the net, therefore it is inadequate.

3. Requirements towards cooperative planning

We depart from a focal network, where market demand is transmitted to the manufacturer by distribution centers. All the partners are *autonomous*. The network is reconfigured time and again, but we consider the stable periods of its operation. In such periods, suppliers are contracted for producing particular components. There is no overlap between the channels—hence suppliers are not in a competitive situation. (They do compete at reconfiguration time, but this network design problem is out of the scope of the paper.) The suppliers may serve several manufacturers acting on different markets. In fact, a particular firm may fill in both manufacturer's and supplier's role in different nets. We allow also for lateral cooperation—when suppliers mutually help each other in critical shortage situations (see Figure 1).

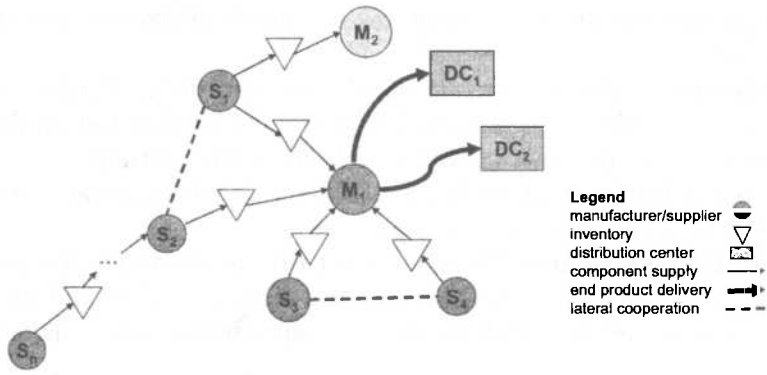


Figure 1. Typical elements and connections of the supply network.

The question all members of the network have to answer time and again can be put simply as how much and when to produce so that they can satisfy demand; neither more, nor less, neither earlier, nor later, just in the required quality.

A network-wide solution emerges from the interaction of local decisions. This is essentially a *distributed planning problem*: network members would like to exercise control over some future events based on information what they know at the moment for certain (about products, technologies, resource capabilities, sales histories) and only anticipate (demand, resource and material availability). Hence, partners—at least along the various supply channels—need to *coordinate*; i.e. to take into account some of the other's decisions. Further on, they can enhance even more their relations by *information exchange* and *cooperation.*

The basic requirements towards a cooperative planning mechanism are as follows:

- **Autonomy** of the network partners has to be respected. Partners are considered independent, rational entities, with their own resources, performance objectives and internal decision mechanisms. Though, together with the distribution of decisions rights, the mechanism has to align responsibilities too.
- **Service level** The mechanism has to guarantee that the overall network can operate at a predefined, arbitrarily high service level.
- **Channel coordination** The mechanism has to facilitate that on the long run, local decisions lead to the emergence of coordinated channels.
- **Profit and risk sharing** Acknowledging the opportunities and risks of the markets of customized mass products, the mechanism should allow the division of the profit and risk between the manufacturer and the suppliers.
- **Aggregation** Information at several levels of aggregation has to be handled.
- **Adequacy** The mechanism should be able utilize (quasi-) standard production planning and scheduling related information available from the local planner and scheduler systems of the partners.
- **Rolling horizon planning** is necessary to accommodate on a regular basis to changes and disturbances.
- **Solution efficiency** Decisions are to be made under the pressure of time, typically in interactive settings. Hence models and appropriate solution techniques with reasonable response time are sought for.

The above generic requirements are unequivocally reasonable for each member of a supply network, let it be either in the role of the manufacturer or the supplier. Note that channel coordination on the long run requires the sharing of medium-term production plans and risk-related information about the future of products. Solution efficiency, on the other hand, calls for *symmetric* information between the manufacturer and the suppliers (otherwise the

decisions models would be too complex). Hence, *cooperation* and *trust* between the partners is a prerequisite for meeting the above requirements. In relatively stable focal networks, partners are willing to cooperate and to share such private business information.

We defined the following steps for designing, setting up and running a cooperation mechanism with the above properties:

1. Disregarding the borders between network members—i.e., handling them as a virtual enterprise—one has to determine coordinated channels.
2. Assuming autonomous, self-interested network members, interaction rules—so-called *mechanisms*—have to be designed that provide network members both with sufficient information and incentives to cooperate. Mechanism design involves also the sharing of risks and benefits.
3. Implementation and integration of information sharing protocols, existing databases and decision support systems, as well as legal instruments.

4. Coordination of a supply channel

Planning tasks of enterprises are usually categorized according to their horizons into three levels: long term, medium term and short term [7, 14]. Consequently, a coordination model should cover all of these levels. The purchasing of raw materials can be planned on the long term, by exploiting economies of scale, forasmuch the bulk of them are standard materials and the demand of the end products can be aggregated. The production related decisions (plans, lot sizes) have to be made on medium term, by aligning the conflicting aims of flexibility and economic efficiency. On short term, the challenge is to organize smooth operation of the network, i.e., production should not stop anywhere due to material shortage. In this model we focus only on medium-term problems and assume, that raw material procurement is working effectively[1]

An acceptable coordination model should provide optimal or quasi-optimal trade-off between

1. inventory holding, obsolete inventory and setup cost on medium term, and
2. feasible, economical production and high service level on short term.

Our proposed model considers single components, discrete, finite (medium-term), rolling horizon component forecast and no inventory limits. We also assume uncapacitated production, and that throughput time of components (manufacturing plus shipment) fits into one time unit—a week in our case.

[1]We suggest also a protocol for short-term coordination, whose detailed elaboration and analysis is part of our future work.

Although the model is discrete, the proposed solution method is continuous, i.e., the lot size can cover arbitrary front fragment of the forecast horizon. The model does not assume the so-called Wagner–Whitin property, which says that within a time unit one can either satisfy demand from inventory or from production, but never from both. The partners are *risk neutral*: their objective is to minimize the expected average cost.

It is an exogenous property of the market, that the demand for a product can suddenly cease and this *run-out* produces obsolete inventory. It happens more frequently in case of the non-standardizable (customized) packaging materials, where *design change*s are also possible. This situation is different from the newsvendor problem, since we do not know anything about the length of the demand period. Run-out must be taken into account, because obsolete products cause significant loss in the network. All in all, we have identified two types of demand uncertainty:

1. quantity fluctuation, and
2. unexpected run-out.

To the best of our knowledge this latter one has not been studied yet, therefore it is a novelty in our model. To measure the loss in case of a run-out, the production cost of the obsolete inventory should be included into the total cost. The production cost may represent both material and labor costs and could be reduced with salvage value, etc.

We assume one-point-inventory between the manufacturer and the supplier. The manufacturer generates in each period a new *master plan* (MP), that determines on a medium-term horizon the output of finished goods in each time unit. The component forecast, which is derived from this MP, is the basic input for the supplier's lot sizing problem.

This forecast is uncertain, but must represent the best knowledge of the manufacturer. Concrete orders (*call-off*s) can be given only for one time unit ahead, therefore must be satisfied with *Just-In-Time delivery* from stock (with 100% service level—after similar considerations as for the "zero defects" principle of *Total Quality Management* [8]).

Since the component forecasts are derived from the MP, they do not provide valid statistical information (such as *standard deviation*), thus we cannot include it into our model. Nevertheless, the demand can neither be considered deterministic. Hence, we propose an easily implementable heuristic policy, which minimizes the expected average cost—either by the length of the expected consumption period or by the produced quantity. The model uses the *probability of run-out* that demand can cease in any time unit of the planning horizon with a specific probability.

This version of the model considers only one product. Thus there are no "speculative motives", i.e., it is always preferable to produce at a later period rather than producing earlier and holding stock. The model can be extended to more components, where setup cost depends on the set of manufactured products (changeover cost). In this case speculative motives can occur, which leads to a combinatorial optimization problem.

The basics of the model can be seen on Figure 2, whose parameters and variables are the following:

- n length of the horizon,
- F_i forecast for the ith week,
- h inventory holding cost per piece per time unit,
- c_s setup cost,
- c_p production cost per piece,
- p probability of run-out in an arbitrary time unit[2],
- x length of the period for which demand should be produced (decision variable).

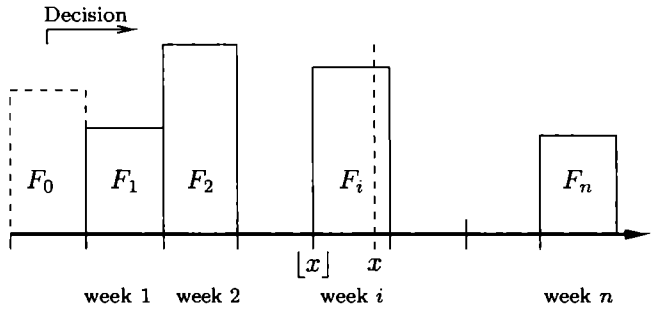


Figure 2. Planning horizon

Decision is made on week 0. In absence of speculative motives, at planning time the stock is below the safety stock level—practically considered to be zero. Since the lead time equals to the time unit, $x \geq 1$, (because later we will not have time to produce the next week's demand) and $F_1 > 0$ (no speculative motives). We also assume, that the call-off (F_0) can be satisfied from the stock (including safety stock).

We use some further notations: $S_k := \sum_{l=1}^{k} F_l$ is the accumulated forecast of the first k weeks and $q(x) := S_{i-1} + yF_i$ is the production quantity, where $i := \lfloor x \rfloor + 1$ and $y := \{x\}$ (here $\lfloor x \rfloor$ means the integer, and $\{x\}$ the fractional part of x). This expresses, that we produce all quantities of the first $(i-1)$

[2]For special cases, one can use a different p_i for every time unit.

weeks, and the y proportion of the ith week's demand. The expected decrease of the inventory level can be seen on Figure 3.

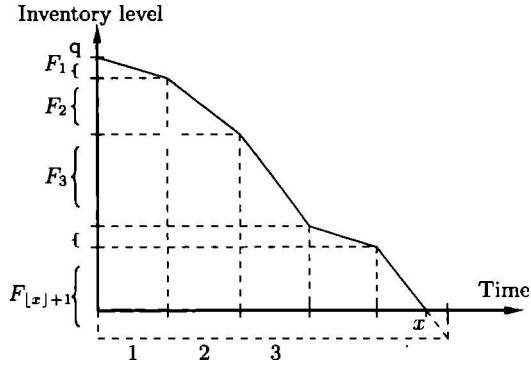


Figure 3. Expected inventory level

If we do not consider run-out, and assume linearly decreasing inventory within a time unit, then the expected storage cost in the first l ($l < i$) time unit is:

$$SC(l,x) = h \sum_{k=1}^{l} \left(q(x) - S_{k-1} - \frac{F_k}{2} \right), \tag{4.1}$$

where $q(x) - S_{k-1}$ is the opening inventory of the time unit k, and $\frac{F_k}{2}$ expresses the linearly consumption within the time unit. Hence, the expected storage cost with run-out can be expressed as:

$$\begin{aligned} SC(x) = &\sum_{k=1}^{i} \left(p(1-p)^{k-1} SC(k-1,x) \right) \\ &+(1-p)^i \left(SC(i-1,x) + h\frac{y^2 F_i}{2} \right) \end{aligned} \tag{4.2}$$

where $p(1-p)^{k-1}$ is the probability that the product runs out in the kth time unit, and with probability $(1-p)^i$ it is still saleable in the ith time unit. In this latter case, both the storage cost of the first $(i-1)$ time units and the storage cost of the remaining fraction[3] incur. The cost of the obsolete inventory can be computed in a similar manner:

$$OC(x) = c_p \sum_{k=1}^{i-1} \left(p(1-p)^{k-1} (q(x) - S_{k-1}) \right) + c_p p (1-p)^{i-1} y F_i. \tag{4.3}$$

[3]The quantity yF_i is consumed only during y time unit.

The model assumes, that if run-out happens at any time, the obsolete inventory is immediately thrown away—so no further storage cost must be paid. Thus we obtain piecewise continuously differentiable average cost functions $AC_x(x) = \frac{c_s+SC(x)+OC(x)}{x}$ and $AC_q(x) = \frac{c_s+SC(x)+OC(x)}{q(x)}$. They can be minimized by searching through the roots of the their derivative and the borders of the intervals.

Note that the above model is *hybrid*: continuous material flow (x, q) is controlled in discrete time unit, by discrete forecasts and actions. This property greatly reduces the computational complexity of the solution and makes the method practically applicable.

5. Cooperative planning in the network

The above channel coordination model gives the core for cooperative planning between the manufacturer and the supplier. Volatile markets call for flexible supply nets—hence suppliers provide not only components but also flexibility as part of their *service*. We suggest the following main rules for regulating this service (see also Figure 4):

The manufacturer is responsible for anticipating market demand, doing its local production planning and scheduling activities as well as for producing the end-products and delivering them to the customers. Planning and scheduling are performed on different horizons and with different time units (e.g., on weekly vs. daily basis). Specifically, the manufacturer:

- Generates master plans periodically, that determine its output on the medium term. Departing from the MP, it makes the F_i component forecasts (e.g., by using traditional Material Requirement Planning (MRP) methods).
- Schedules in detail its production on the short term. It generates component requests based on the schedules in form of daily call-offs towards the supplier.
- Provides and updates information about the probability of run-out.

The supplier's main responsibility is to satisfy the call-offs requested by the manufacturer. Consequently, it has to handle the one-point-inventory. In particular, the supplier:

- Acknowledges and guarantees the instant delivery of call-offs.
- Maintains the inventory: in each time unit it determines whether to produce or not to produce, as well as determines the lot size.
- Plans and schedules its own production according to its own objectives and additional demand.

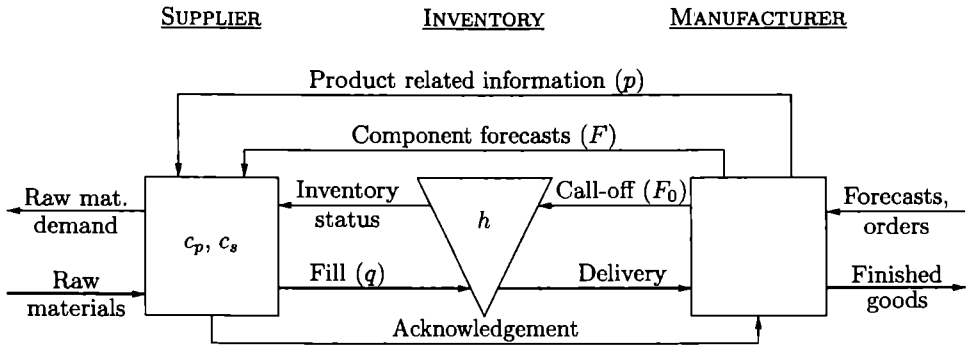


Figure 4. Information and material flow of cooperative planning

Bottom of the service is that production of end-products at the manufacturer must not be stalled by material shortage. Short-term production schedules may change frequently, and the total of call-offs for the actual planning period (F_0) may be more or less than the amount forecasted before. Hence, as part of the inventory, *safety stock* is needed to avoid short-term stock-outs of components. In fact, the safety stock de-couples the medium-term planning and short-term scheduling aspects of the cooperation problem. The safety stock level can be adjusted by at least three strategies:

1. Looking backward, based on the length of throughput time of components and the standard deviation of historical forecasts.
2. Looking backward, considering the forecasted demand of the next few time units.
3. Using the combination of the previous two methods.

The protocol and conditions of the above service are to be laid down by a contract that regulates the flow of information and material between the partners. As for the monetary terms, we suggest to introduce the *cost of flexibility* that has two components:

- The cost of operating on a risky market can be measured by the difference between the optimal expected average cost in risky and risk-free (i.e., where $p = 0$) markets (see also Section 6). This extra cost must be shared by the manufacturer and the supplier.
- Uncertainty in demand quantity can be measured by the variability of the series of component forecasts generated at subsequent planning times. As advancing in time, the effects of differences should be discounted.

Since a component forecast is created in every time unit, at the moment of the call-off we have an n dimensional, non-negative, real-valued *forecast history vector* for the actual week: $H_0 \in \left(\mathbb{R}_0^+\right)^n$, where $H_{0,j}$ $(j \in \{1,\ldots,n\})$ was made j weeks before. The difference between call-off and a forecasted quantity measures the fluctuation in the demand. An average fluctuation can be computed as a convex combination for the forecast history:

$$\sum_{j=1}^{n} \alpha_j \left|F_0 - H_{0,j}\right| \tag{5.1}$$

such that $\alpha_j \geq 0$ $(j \in \{1,\ldots,n\})$ and $\sum_{j=1}^{n} \alpha_j = 1$. It might be constant $(\alpha_j = \frac{1}{n})$, linearly decreasing discount $(\alpha_j = \frac{2}{n+1} - (j-1)\frac{2}{n^2+n})$, exponentially decreasing discount or even more complex functions. The proper approach may differ according to various product classes and needs further research.

The cost of flexibility must be shared on a regular basis. Note that this way the manufacturer has an incentive to make reliable master plans (and component forecasts) while the supplier is concerned in producing lots that coordinate the channel. No partner should be interested in the unilateral deviation from this mode of operation. This is the key of avoiding double marginalization and running the network in a cost-efficient way.

6. Case study and simulation experiments

The coordination model has been developed together with industrial partners, who form a complete focal network. Some typical characteristics of the focal manufacturer in the studied network are as follows: it produces altogether several million units/week from a mix of thousands of products. The ratio of the customization follows the *80/20 Pareto-principle*: they give 80% of the product spectrum, but only 20% of the volume. The setup costs are significant: 10-20% of the total costs, depending on the lot sizes. Since customized products are consumed slower, their smaller lot sizes involve higher average setup costs. Service level requirements are extremely high: some retailers suddenly demand products in large quantities even within 24 hours, and if it is not fulfilled on time, they cancel the order (i.e., backorders are not possible). This causes not only lost sales, but also of goodwill. All in all, making larger lots and maintaining inventories is a must, but it incurs not only the usual inventory handling costs, but the risk of obsolete inventory.

Since many product differences are only due to packaging, furthermore it is just the design of packaging that changes most often, the coordination of production and packaging material supply is of crucial importance. Hence, we started the simulation experiments with coordinating various channels of

packaging material supply. For solving the model and running simulations we have used the *Mathematica 5.2* system.

While forecasts and most of the parameters are easily accessible in the databases of the partners (so-called enterprise data warehouses), the probabilities of run-out are hard to estimate in general. Fortunately, on typical MP forecasts—where planned manufacturing of a product is sparse and involves large volumes—quantity is almost everywhere zero and the formula seems to be not too sensible to the uncertainty. According to our observations, optimal lot size will be a (decreasing) step function of the run-out probability (see Figure 5. for some representative results).

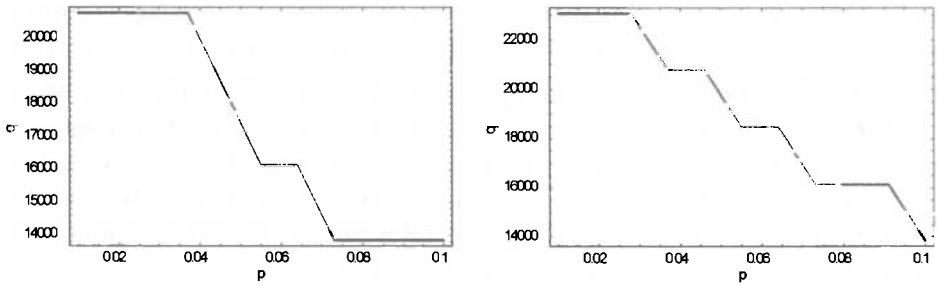


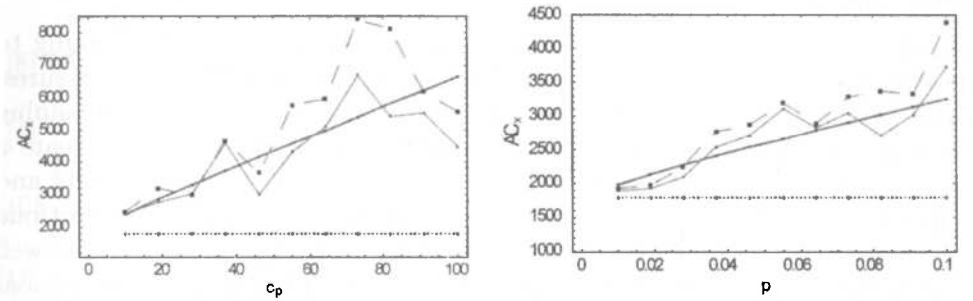
Figure 5. Example lot sizes, which minimize average costs (a) by the expected consumption period and (b) by quantity

We have computed lot sizes with both heuristics and some representative p values from the $[0.00005, 0.15]$ interval. They have been usually similar to each other and not far from those created by simple *rules-of-thumb* by human experts. Then we have simulated the run-outs and computed the average of accumulated costs. This second series of experiments have shown that our methods did not conflict with inventory handling rationale. So as to present the simulation results in a concise way, we have characterized products by two aspects: average forecasted volume and production frequency. After having classified products by deviation from the mean, we had altogether four clusters: high volume-high frequency (HVHF), high volume-low frequency (HVLF), low volume-high frequency (LVHF) and low volume-low frequency (LVLF) products. In Table 1. the minimum and maximum improvements on average costs can be seen in case of 1% probability of run-out as well as the cardinalities of clusters. The proposed lot sizes have effected lower average costs in 99.4% of the simulations (the table also contains the minimal negative value).

Table 1. Percental improvement on average costs

Improvement		on AC_x		on AC_q	
Category	#	min	max	min	max
HVHF	5	0	11	9	23
HVLF	2	2	4	12	18
LVHF	2	1	4	12	28
LVLF	21	-3	35	15	61

For the sake of generality, we have also tested the coordination model with large series of random forecasts and made sensitivity analysis by all its parameters. Two example diagrams can be seen on Figure 6, where each point represents a mean made on 1000 simulation runs on 1000 forecasts on a 12 weeks horizon.



(a) (b)

Figure 6. Change in average costs in function of (a) production cost and (b) probability of run-out

The constant dotted (blue) lines express, that on a risk-free market (i.e., $p = 0$) the average logistic cost would be independent from the production cost. The almost-linear thick (red) lines mean the expected average cost on risky markets. Thin (purple) curves, which oscillate around the thick (red) ones, are the costs that arose in simulations. If the probability of run-out had been disregarded, then the cost would have been usually higher, as indicated by the dashed (green) curve. The diagrams can be interpreted in the following way: the gap between the dotted (blue) and the thick (red) lines is the theoretical difference of the costs of operating in a risk-free and a risky market, while the gap between the thin (purple) and the dashed (green) curves expresses the cost of inconvenient lot sizing.

7. Conclusions and future work

Our model couples basic factors of supply chain performance like service level as well as production, setup, inventory and expected obsolete inventory costs. The proposed channel coordination model can be solved efficiently, and comparative simulation experiments on historic data sets have led to decreased expected average costs. At the same time, simulation has shown that the results are sensitive to some model parameters, in particular to the cost of setups, as well as to the probability of run-out. Controlling these costs parameter are the responsibilities of local planning and scheduling, just as the provision of component forecasts that are directly generated from the MP of the manufacturer. Certainly, adjustment of the main model parameters needs an *adaptive* control approach.

Operating on a market of customized mass products, inventory—and even obsolete inventory—is inevitable, but with coordination and cooperation their amount and incurred costs can be decreased without violating the service level of the supply network.

Based on our coordination model and solution method, we are planning to develop cooperation mechanisms to divide costs and benefits, which assures, that every enterprise (or decision maker) is responsible for its own planning decisions. Since there is no "one-size-fits-all" solution, we intend to create a *portfolio* of cooperation mechanisms. It is essential to classify products and components, e.g., by risk levels or by inter-enterprise relations. The portfolio may contain several standard protocols (Make-To-Order, VMI, etc.) as well as customized ones. An interesting further possibility is to introduce probability of run-out into the discrete Wagner – Whitin model and compare it with the approach presented in this paper. Finally, we are going to validate the suggested cooperative planning mechanism by multiagent simulation tailored to the actual focal supply network of our industrial partners [6].

Acknowledgments

This work has been supported by the VITAL NKFP grant No. 2/010/2004 and the OTKA grant No. T046509. The authors would like to thank Gábor Erdős for all his useful help and advice concerning *Mathematica*, Ferenc Erdélyi, András Kovács and the anonymous reviewers for their valuable remarks. The authors are indebted to the industrial partners for their helpful collaboration.

REFERENCES

[1] Anupindi, R.: Coordination and Flexibility in Supply Contracts with Options. *Manufacturing Services Operations Management*, 4(3), pp. 171-207, 2002.

[2] Brahimi, N., Dauzere-Peres, S., Najid, N. M., Nordli, A.: Single Item Lot Sizing Problems. *European Journal of Operational Research*, 168, pp. 1-16, 2006.

[3] Cachon, G. P.: Supply Chain Coordination with Contracts. In de Kok, A. G., Graves, S. C. (eds): *Supply Chain Management: Design, Coordination and Cooperation. Handbooks in Op. Res. and Man. Sci.*, 11, Elsevier, pp. 229-339, 2003.

[4] Clark, A. R., Clark, S. J.: Rolling Horizon Lot-sizing when Set-up Times are Sequence Dependent. *International Journal of Production Research*, 38(10), pp. 2287-2308, 2000.

[5] De Toni, A. F., Zamolo, E.: From a Traditional Replenishment System to Vendor-Managed Inventory: A Case Study from the Household Electrical Appliances Sector. *International Journal of Production Economics*, 96, pp. 63-79, 2005.

[6] Egri P. Váncza, J.: Cooperative Planning in the Supply Network – A Multiagent Organization Model. In Pechoucek, M., Petta, P., Varga, L. Zs. (eds.): *Multi-Agent Systems and Applications IV*, Springer LNAI 3690, pp. 346-356, 2005.

[7] Fleischmann, B., Meyr, H.: Planning Hierarchy, Modeling and Advanced Planning Systems. In de Kok, A. G., Graves, S. C. (eds): *Supply Chain Management: Design, Coordination and Cooperation. Handbooks in Op. Res. and Man. Sci.*, 11, Elsevier, pp. 457-523, 2003.

[8] Hopp, W. J., Spearman, M. L.: *Factory Physics – Foundations of Manufacturing Management.* McGraw Hill, 1996.

[9] Lee, C. C., Chu, W. H. J.: Who Should Control Inventory in a Supply Chain? *European Journal of Operational Research*, 164, pp. 158-172, 2005.

[10] Lee, H. L., Padmanabhan, V., Whang, S.: Information Distorsion in a Supply Chain: The Bullwhip Effect. *Management Science*, 43, pp. 546-558, 1997.

[11] Liker, J. K., Choi, T. Y.: Building Deep Supplier Relationships. *Harvard Business Review*, 82(12), pp. 104-113, 2004.

[12] Monostori, L., Fornasiero, R., Váncza, J.: Organizing and Running Real-time, Cooperative Enterprises. In Taisch, M. Thoben, K-D. (eds.), *Advanced Manufacturing: An ICT and Systems Perspective*, IMS, 2005, pp. 144-157, 2005.

[13] Selladurai, R. S.: Mass Customization in Operations Management: Oxymoron or Reality? *Omega*, 32, pp. 295-300, 2004.

[14] Stadtler, H.: Supply Chain Management and Advanced Planning Basics, Overview and Challenges. *European Journal of Operational Research*, 163, pp. 575-588, 2005.

[15] Tirole, J.: *The Theory of Industrial Organization.* MIT Press, 1988.

[16] Tseng, M. M, Lei, M., Su, C.: A Collaborative Control System for Mass Customization Manufacturing. *Annals of the CIRP*, 46(1), pp. 373-376, 1997.

[17] Váncza, J., Kis, T., Kovács, A.: Aggregation – The Key to Integrating Production Planning and Scheduling. *Annals of the CIRP*, 53(1), pp. 377-380, 2004.

Production Systems and Information Engineering
Volume 4 (2006), pp. 21–32

VISUAL WORKFLOW EDITORS: A CRITICAL REVIEW FROM USERS' PERSPECTIVE

FLORIAN URMETZER, ASHISH THANDAVAN, VASSIL N. ALEXANDROV
Center for Advanced Computing & Emerging Technologies
University of Reading, PO Box 225, Reading RG6 6AY
[f.urmetzer,a.thandavan,v.n.alexandrov]@rdg.ac.uk

ROB ALLAN
e-Science Center, Daresbury Laboratory
Daresbury, Warrington WA4 4AD
r.j.allan@dl.ac.uk

[Received December 2005 and Accepted March 2006]

Abstract. The aim of this paper is to discuss the state-of-the-art in visual workflow editing tools for scientific applications in distributed and grid computing for the e-Sciences. The structure of the research behind this paper was a large-scale review of literature on several workflow editing tools. These tools were then installed and used to be able to contrast the literature with a user experience. The outcomes are recommendations towards bettering workflow editor interfaces and indications for further research.

Keywords: workflows, visual workflow editors, grid computing

1. Introduction

Workflow management tools support the user in designing, creating and managing the execution of workflows [12] / [13]. They enable the users to describe and perform experimental procedures in an organized, replicable and, most importantly, provable way. The tools are needed for the definition and for the visualization of the processes involved in a computational experiment [15]. There are several projects involved in the development of such visual workflow editors (Kepler, Taverna, Triana, P-GRADE). The main endeavour of these projects is to enable the non-computing specialist to handle distributed computing resources in a user-friendly way through these interfaces. The computing resources are mainly defined as web services and/or Grid technology.

There are several user groups that are making substantial use of distributed technology, for example, Astronomy, Physics and Biology. The Bioinformatics community, for instance, has the need to access different specialized large data-sets and databases, which may be related to one particular disease and compare these to another data-set [8]. These resources are highly specialized databases, which are very expensive to maintain. Therefore, one of the data-sets maybe stored in Germany and another in Japan. When bound together through processors, they can be used virtually as one data-set. Workflow management tools are helping the e-scientist to use such external resources in a flexible way and independent from the IT specialist [12].

The major challenge to these tools is that the user-base for workflow editors is mostly not specialized in computing or in the use of such complex IT systems [14]. Therefore some authors state that there may be a trade off between a highly powerful tool and a target audience that is able to handle it e.g. [7].

This paper details the important outcomes of a wider study. It shows an outline of the arguments presented in the study and concludes in recommendations to better workflow editors and further the research in the field.

2. The tools in detail

This paper will look at four tools in detail. They are Triana, Taverna, Kepler and the P-GRADE Grid portal. These tools are all visual workflow editors and three of them have been created by research projects needing such tools. They differ in the system type - where Triana, Taverna and Kepler are Java applications, the P-GRADE Grid portal is server-based. The visual representation of the workflow is different from tool to tool, as is the quality and method of user interaction.

2.1. Taverna

Taverna is a workflow editing tool which is available from [21]. This tool is a component of the myGrid project which was funded by the Engineering and Physical Sciences Research Council (EPSRC). The development of the tool was mainly driven by the requirements of biologists from the UK's life sciences community [21].

The format of storage of workflows is SCUFL (Simple Conceptual Unified Flow Language). SCUFL is an XML based workflow language. It has been specially developed because the use of a generic, standard language would have not given the opportunity to investigate key aspects and needs for a workflow language in the bio-sciences. The interface of Taverna is based on three main windows: The SCUFL Diagram window, the XSCUFL Window

and the SCUFL Model Explorer (see fig. 1. The SCUFL Diagram window displays an overview of the present workflow; this window is a display-only facility and therefore not editable. The graphical display consists of nodes and links.

The nodes can be processors, inputs or outputs. Processors are a transformation entity that take data and process it. These processors can be of six different types as described in detail by [7].

1. WSDL processor - can call a web service defined in WSDL.
2. Soap lab processor - can call a complete Soap lab process.
3. Talisman processors - enables a Talisman task to be processed.
4. Nested workflow processors - are needed to implement child workflows.
5. String constant processor returns a string to an output port; for instance to a processor that needs a constant value for processing.
6. Local processor - enables the user to add local functionality like Java programs.

Figure 1. The Taverna interface with the toolbar on top, the hierarchy tree populated with services on the left and the editing area on the right.

The links are either data links (which show the direction of the flow of data) and coordination constraints (which control the execution, for example, of two processors). The interface supports three different visual displays - showing all ports, showing no ports or showing only those ports that have connectivity [8]. This is intended to show details of the services that are available through the tools. Normally the ports differ by the type of input and the type of

processing done on them. The visualization is always organized from the top to the bottom, from the input to the outputs [20]. There is also a text window displaying a read-only version of the current workflow.

Finally, there is a reporting facility, which can handle failure reporting and the collection of provenance data. The provenance tool is based on XML document format where the details are presented to the user in a tabular format [8]. This tool has been seen to be very useful throughout the tests performed during the research.

The workflows are edited in the workflow model explorer, which is a hierarchy tree. To add a service, the service is either dragged and dropped into the workflow model explorer or right-clicked on and added. The resource hierarchy tree has a search function at the top of the window, to search for resources in highly populated trees. The linking of the services is accomplished by choosing the output of a service to be linked in the model explorer, right-clicking on it and choosing one of the possible connections that are displayed. The visualization is automatically updated to include the connection and the connection is shown in the workflow model explorer under the Connections tab.

2.2. Triana

Triana is a visual programming environment that enables the user to create workflow graphs from the connection of programming units or components [6][11]. The tool is available from the Triana project [22]. The tool was developed by Cardiff University and is a part of the GridLab project [17]. The interface consists of a tool bar, a resource hierarchy tree and a visual display area (see fig. 2). The toolbar has the main functions (like copy, paste and save) as buttons. The resource hierarchy tree can be automatically populated with web services. The hierarchy tree is at all times organized alphabetically and has six ways to organize the resources via a drop-down menu above the hierarchy tree. The default case, where the default packages are displayed, shows the 'All packages' option, where all resource packages are shown. A 'Show all tools' option displays all the tools and finally tree options show only the data, the input or the output tools. Triana's source recovery tree interface has been described as limiting. It is argued that users may want an alternative or a range of different ways to discover resources [14]. The authors found that the non-availability of a search function for the hierarchy tree slowed down the assembly processes.

It was found that only web services type processors can be displayed. This is supported by White, Jones et al. [14] stating that Triana only processes a limited number of data types.

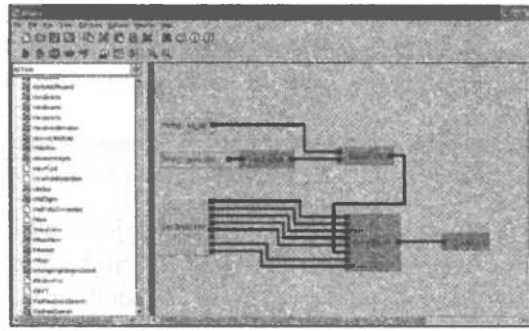


Figure 2. The Triana interface with the three interaction windows, the resources, the workflow builder and the workflow display.

These processors can be dragged and dropped into the workspace and connected together [10]. The connections are done in the display and editing area, by clicking an output and dragging and dropping the output onto an input. The workflow language used to save the workflow and to communicate is the Business Process Execution Language for Web Services (BPEL4WS).

The features of Triana include four major points [6]:

1. Simplified construction of Web Services, which details in a simplified discovery, composition, invocation and publication of services.
2. Execution of composite services on distributed systems.
3. Sensitivity analysis tool which enables the user to do a what-if? analysis.
4. Recording of workflows as well as automated provenance related information.

To have full functionality, Triana has some pre-requisites which are based on web services. These are service discovery methods, service composition methods, transparent execution methods and transparent publishing methods [6]. The system works on the basis of interacting components which are pluggable and modularized [10]. There are two tutorials, titled "Running a Wave Unit Remotely" and "Distributing Units Amongst OCL Servers" supplied with the installation files. The authors tried to follow these tutorials and get then to work but were not successful. One of the authors then found a letter in the users' mailing-list archive stating that the tutorials are out of date and do not work. It is therefore not obvious which features work, apart from the tested web services execution.

When executing a basic workflow, Triana shows the progress through little boxes in the workflow processors that turn black when active. Therefore the

user can check on progress. There is no prevenance collection or metadata entry like in Taverna. The Triana XML file has been found to be much longer compared to SCUFL as well.

2.3. Kepler

Kepler was built by a collaboration of different projects which included SEEK (Science Environment for Ecological Knowledge), SPA Center (Scientific Process Automation), Ptolemy II (Heterogeneous Modelling and Design), GEON (Cyber infrastructure for the Geosciences) and ROADNet (Real-time Observatories, Applications, and Data Management Network). These projects found the similar need for the development of an open source tool to create, edit and manage scientific workflows. Kepler is available free of charge from the project's home page [18].

Kepler is a workflow enactment tool that has been built on top of Ptolemy II, which is a software tool supporting heterogeneous, concurrent modeling and design [16].

Kepler's interface is structured in a toolbar with the most important functions in button form, a resource recovery tool on the left hand side of the screen, including a search option and finally an editing area on the right hand side of the screen (see fig. 3).

Keplers strength are described in three parts [2]

1. It enables the user to define models of computation precisely, including the process networks model, which is dataflow oriented.
2. It has a modular programming approach that is oriented towards the production of reusable components.
3. It is described as an easy-to-use graphical user interface that allows the user to create complicated workflows in an easy manner.

The application can define a row of different processors. For example, Kepler is able to handle database queries to major database types, it handles Globus jobs, web service definition language and finally XSLT & Xquery which are both XML editing types.

Kepler is able to process different plug-ins, called Actors. These define the flow of information or the process of the workflow. The modularity of the interface is based on the hierarchical abstraction. Therefore complex models maybe shown in one block to make the model visually more structured. These blocks may be internal or external processes [3]. The Actors and other resources for workflows are stored on the left hand side of the workflow editor interface in the form of a hierarchical tree. The parts needed are dragged and dropped

onto the workflow area where they can be linked up to each other by dragging the output of one actor or processor to the input of another one. All processors can be defined in detail through the interface.

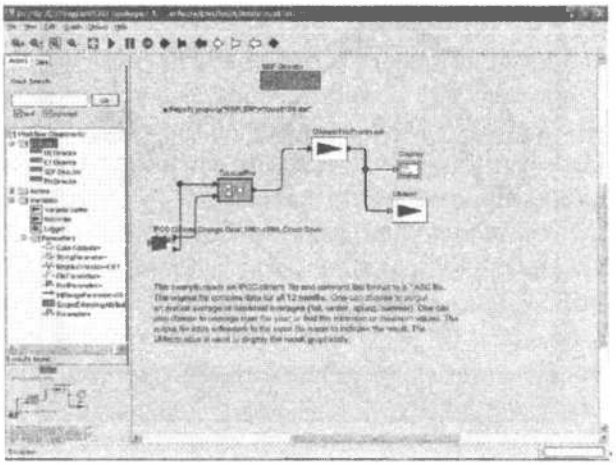


Figure 3. The Kepler interface with the tool bar on top, the hierarchy tree populated with services on the left and the editing area on the right.

The workflows are saved using an XML language called Modeling Mark-up Language (MoML). This XML language does not however include versioning and indexing of information and is described to be the only way of provenance as well [1].

2.4. P-GRADE Grid Portal

The Laboratory of Parallel and Distributed Systems in the MTA-SZTAKI Computer and Automation Research Institute, Hungarian Academy of Sciences, Budapest, Hungary has developed P-GRADE, a workflow solution for complex grid applications. Their tool is intended to help the user in designing, executing, monitoring the different stages of execution and visualizing the progress. While P-GRADE was originally a stand-alone application, the P-GRADE Grid Portal is a portal-based solution and therefore runs on a web server. The portal version will be discussed in this paper. The files required to install the portal are available from MTA-SZTAKI [23]. The installation is however very complicated, because of operating system requirements and some rather specific pre-requisite software - Condor v6.4.2 (or v6.4.7), Globus Toolkit v3.2, GridSphere v1.0.1, Apache Ant v1.6.1, Java 2 Platform SDK v1.4.2_04, Apache Tomcat v4.1.27 and C libraries (libpng, libgd) [19]. The

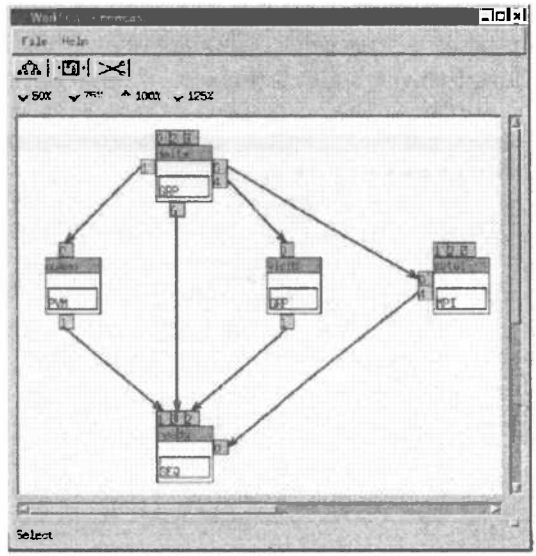


Figure 4. The P-GRADE interface showing the Petri net based workflow representation. Each processor has input and output ports. The green are input ports and the gray are output ports.

portal's interface is based on the P-GRADE graphical programming environment [5] (see fig. 4). The components of a workflow can be either sequential (C / Fortran) or parallel (PVM (Parallel Virtual Machine) / MPI (Message Passing Interface) / P-GRADE job). P-GRADE uses the hybrid GRAphical Process NEt Language (GRAPNEL) internally.

Communication between jobs is expressed via input and output ports which are defined when creating the port on the processor. The ports are then linked up to each other by dragging the output port to the input port.

The P-GRADE Grid Portal has a monitoring and visualisation facility as well as interoperability with different systems on heterogeneous Grid platforms, for example, Condor or Globus [5]. They can however be stored on the server site as well as the user end.

As the P-GRADE Grid portal is portal-based, the user only has to install a version of Java Webstart on his/her computer. The client accesses the portal via a standard web browser and logs in. Java Webstart will then download and start the application enabling the user to work with the system. All the proxy credentials for grid sites are managed via a MyProxy server through the portal. Therefore the portal-based system can be seen as very much user-friendly and very advantageous for grid systems.

3. The tools compared and analyzed

In this section, the authors would like to look at the tools from a comparative point of view. Therefore they would like to mention the major points encountered throughout the wider study and briefly discuss them.

One issue that is discussed in the literature from the workflow tools' projects, is that of the representation of the workflow. Direct Acyclic Graph (DAG) is used in Taverna and Direct Cyclic Graphs is used in Triana; these are discussed throughout the tools' literature and have been briefly touched upon in this document. A pictorial representation of single workflow entities has been discussed by other authors like Hernandez and Bangalore [4]. The arguments presented in the literature were not found to be supportive by the authors for the definition of a good representation. This was mainly due to the lack of publications detailing with comparative user testing and user opinions on the graphical representation of workflows. The authors agree that there may not be only one solution and therefore there may be the need to implement multiple interface representations of workflows to choose from.

The author strongly recommends testing workflow interfaces with potential users of the e-Science community in a structured way. This would then enable a further definition of interface needs and preferences of users.

The storage and presentation of services and processors available to the user was found to be in the form of a hierarchy tree in all tools. This form of organization was seen as usable by the authors, but only if the hierarchy tree is not overfilled with services. Additionally the organization in multiple layers and most important search mechanisms for the hierarchy tree are highly recommended by the authors.

However there should be another form of service retrieval and keeping, because the hierarchy tree has been discussed as not ideal by the research community [14]. There is therefore scope for further research into visualising the retrieval and keeping of services available to the user.

The editing mechanisms range from editing in a hierarchy tree that displays the workflow entities, like in Taverna, to editing the entities together graphically. There is however no evidence in the literature that supports any of the editing methods. The authors argue that a multiple approach to editing facilities is probably the best solution, leaving the decision to the user. A multiple way of editing workflows is however not implemented in any of the workflow tools. This means that there should be other ways of linking processes than those explored by the projects.

There are several features that must be included in e-Science workflow tools to support their user community.

1. There is a need to have meta data to describe the workflow to other users. This may include outcomes and links to publications and search words to be used in repositories.
2. This meta data should be included in the workflow language, because of the danger of a disconnection of the description and the workflow script.
3. The use of provenance collection tools was confirmed to be useful. Therefore information of the workflow execution is collected during the runtime of the workflow and later presented to the user for storage. This provenance information can then be used to validate the experiment at some later date.
4. Server-based tools are found to be advantageous, because proxy and networking problems can be overcome and access to computing resources can be managed from the server side by computing specialists rather than by users on their individual computers. In addition, deployment problems when updating versions of the interface and changing of the computer on the client side are overcome. The problems mentioned are not overcome by using an enactment engine. However the enactment engines as well as a server-based tool overcome the problem of long-running workflows being able to be executed remotely from the user's computer.
5. A common workflow scripting language would allow a workflow created by one editor to be opened and edited in another. Users can then use their preferred workflow editor knowing that they can share their workflow descriptions with their peers.
6. An easy install process would also be highly advantageous.

4. Conclusions

In conclusion, there are two ongoing problems that re-occurred throughout the research done. The first was the missing definition of an e-Science workflow script language and the second was the total absence of work towards user tests with workflow interfaces within the e-Science community.

REFERENCES

[1] ALTINTAS, I., BERKLEY, C., JAEGER, E., JONES, M., LUDAESCHER, B. and MOCK, S.: *Kepler: Towards a Grid-Enabled System for Scientific Workflows.* Workflow in Grid Systems Workshop in GGF10 - The Tenth Global Grid Forum, Berlin, Germany, 2004.

[2] ALTINTAS, I., BERKLEY, C., JAEGER, E., JONES, M., LUDÄSCHER, B. and MOCK, S.: *Kepler: An Extensible System for Design and Execution of Scientific*

Workflows. 16th International Conference on Scientific and Statistical Database Management (SSDBM), Santorini Island, Greece, 2004.

[3] BHATTACHARYYA, S. S., BROOKS, C., CHEONG, E., DAVIS, J., GOEL, M., KIENHUIS, B., LEE, E. A., LIU, J., LIU, X., MULIADI, L., NEUENDORFFER, S., REEKIE, J., SMYTH, N. TSAY, J., VOGEL, B., WILLIAMS, W., XIONG, Y ZHAO, Y. and ZHENG, H.: *Volume 1: Introduction To Ptolemy II.* Brooks, C., Lee, E.A., Liu, X. Neuendorffer, S., Zhao, Y. and Zheng, H. eds. Ptolemy II: Heterogenous Concurrent Modeling and Design In Java, 2004.

[4] HERNANDEZ, F., BANGALORE, P., GRAY, J. and REILLY, K.: *A Graphical Modelling Environment For The Generation Of Workflows For The Globus Toolkit.* Workshop on Component Models and Systems for Grid Applications, Held in conjunction with ICS 2004: 18th Annual ACM International Conference on Supercomputing, Saint-Malo, France, 2004, Springer Verlag.

[5] LOVAS, R., DÓZSA, G., KACSUK, P., PODHORSZKI, N. and DRÓTOS, D.: *Workflow Support for Complex Grid Applications: Integrated and Portal Solutions.* 2nd European Across Grids Conference, Nicosia, Cyprus, 2004.

[6] MAJITHIA, S., SHIELDS, M., TAYLOR, I. and WANG, I.: *Triana: A Graphical Web Service Composition and Execution Toolkit.* IEEE International Conference on Web Services 2004.

[7] OINN, T., ADDIS, M., FERRIS, J., MARVIN, D., GREENWOOD, M., GOBLE, C. WIPAT, A., LI, P and CARVER, T.: *Delivering Web Service Coordination Capability to Users.* 2004, http://decweb.ethz.ch/WWW2004/docs/2002p2438.pdf

[8] OINN, T., ADDIS, M., FERRIS, J. MARVIN, D., SENGER, M. GREENWOOD, M., CARVER, T GLOVER, K., POCOCK, M. R., WIPAT, A. and LI, P *Taverna: a tool for the composition and enactment of bioinformatics workflows.* Bioinformatics, 20 (17), 30453054.

[9] SHIELDS, M.: *Triana User Guide*, Cardiff, 2004, http://www.trianacode.org/docs/index.html

[10] TAYLOR, I. SHIELDS, M. and WANG, I.: *Resource Management of Triana P2P Services.* Weglarz, J., Nabrzyski, J., Schopf, J. and Stroinski, M. eds. Grid Resource Management, Kluwer, 2003.

[11] TAYLOR, I., SHIELDS, M., WANG, I. and PHILP, R.: *Grid Enabling Applications Using Triana. in Workshop on Grid Applications and Programming Tools*, Seattle, 2003, GGF Applications and Test beds Research Group (APPS-RG). GGF User Program Development Tools Research Group (UPDT-RG).

[12] THURSTON, C.: *Go with the workflow.* Scientific Computing World, September/October 2004 (78).

[13] WORKFLOW MANAGEMENT COALITION: *The Workflow Management Coalition Specification: Terminology & Glossary.* Winchester, 1999, http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v1013.pdf

[14] WHITE, R., JONES, A., PITTAS, N., GRAY, A., XU, X., SUTTON, T., BROMLEY, O., CAITHNESS, N., FIDDIAN, N., CULHAM, A., BISBY, F., BHAGWAT, S.,

BREWER, P YESSON, C. and WILLIAMS, P *Building a Biodiversity Problem-Solving Environment.* All Hands Meeting (AHM 2004), Nottingham, 2004.

[15] WROE, C., LORD, P MILES, S., PAPAY, J. MOREAU, L. and GOBLE, C.: *Recycling Services and Workflows through Discovery and Reuse.* www.mygrid.org, Manchester and Southampton, 2004, http://www.ecs.soton.ac.uk/ lavm/papers/ahm04-wroe.pdf.

[16] WWW.BERKELEY.EDU: Ptolemy II, 2004, http://ptolemy.eecs.berkeley.edu/ptolemyII/.

[17] WWW.GRIDLAB.ORG: Gridlab - a grid application toolkit and testbed. 2005.

[18] WWW.KEPLER-PROJECT.ORG: Kepler Project, 2004.

[19] WWW.LPDS.SZTAKI.HU: How to install P-GRADE PORTAL, SZTAKI, Budapest, 2004, http://www.lpds.sztaki.hu/pgportal/manual/install/ PORTAL_installation_an_Introduction.html.

[20] MYGRID. *Taverna User Guide*, 2004.

[21] TAVERNA.SOURCEFORGE.NET: Welcome to Taverna, 2004

[22] WWW.TRIANACODE.ORG: The Triana Project, Cardiff University, Cardiff, Wales, UK, 2003.

[23] WWW.LPDS.SZTAKI.HU: MTA-SZTAKI Computer and Automation Research Institute, Hungarian Academy of Sciences, Budapest, Hungary

Production Systems and Information Engineering
Volume 4 (2006), pp. 33–51

A MODEL FOR VISUAL FEATURE EXTRACTION BASED ON THE MAMMALIAN VISUAL CORTEX

BARNA RESKÓ
Computer and Automation Research Institute, Hungary
Cognitive Systems Research Group
resko@sztaki.hu

ZOLTÁN PETRES
Computer and Automation Research Institute, Hungary, Hungary
Cognitive Systems Research Group
petres@tmit.bme.hu

PÉTER BARANYI
Computer and Automation Research Institute, Hungary
Cognitive Systems Research Group
baranyi@sztaki.hu

[Received November 2005 and Accepted May 2006]

Abstract. The present paper proposes a model for intelligent image contour detection. The model is strongly based on the architecture and functionality of the mammalian visual cortex. A pixel-to-feature transformation is performed on the input image as the afferent visual information. The result of the transformation is a three-dimensional array of data representing abstract image features (contour objects), instead of another array of pixels. The contour feature recognition is performed by a vast and complex network of simple units of computation that work together in a parallel way. The use of a large number of such simple units allows a clear structure that can be implemented on a special hardware to allow fast, constant time feature recognition.

Keywords: Visual Feature Array, negative filtering, contour detection

1. Introduction

The main goal of this paper is to present a neurobiological and cognitive psychological analogy based cognitive framework. The framework is based on the biological architecture and cognitive functionalities of the mammalian visual cortex, which is able to perform image contouring in an intelligent way.

Besides the possibilities of practical applications of the framework, it also aims to extend the limits of classical computation.

In order to show why cognitive models can give the necessary boost, consider the example where a test person has to determine whether there is a cat or something else in the shown image, and press a button according to the decision. Such a task is impossible for a computer to perform today, yet a human can do it reliably in half a second or less. This result becomes more shocking if we know that the "processing time" of the basic processing unit of the brain (a typical neuron) is in the range of milliseconds, while the basic processing unit (a logic gate) of a modern silicon-based computer is 5 million times faster. The answer for how the "slow" brain can solve this task lies in its special architecture and particular information representation and processing. It is thus our belief that in order to step beyond the borders of today's computer systems' architectures the basic way of information representation and processing has to be changed. For new ideas we turn to existing cognitive systems in biological architectures to study them, because they already bear the solutions that we are seeking for. A cognitive system is implemented in a biological neural network, where simple units of computation are connected in a very complex structure. Our research goal is to turn the cognitive information processing system into engineering models which can later be organized into a cognitive psychology inspired model running on a biology related computational architecture.

Our work has received inspiration from research about biological visual systems, [1, 2]. This is not to say that the model presented in this paper are necessarily identical with biological visual systems. The ultimate criterion of our work is performance from a technical point of view.

A cognitive process is an abstract concept which can be considered as an information processing function. A cognitive system is composed of many cognitive processes each responsible for a different task. By the complex structure of mutual interaction of the cognitive processes the cognitive system becomes very sophisticated with new limits of computation. A cognitive process only describes a functionality, but it does not say anything about the way of implementation, thus it can be implemented in many ways. One existing implementation of cognitive processes is the cerebral cortex of mammalian animals, where a very complex biological computational architecture provides the computational power for cognitive processes.

Such an architecture is built up by numerous, simple computational elements that can perform only primitive functions like addition, subtraction in a rather short time. These computational elements are connected to each other in a very complex network, like the neurons in the brain. The neural architecture

can be much more efficient in certain tasks than the complex, classical algorithms, by virtue of the decomposition of the problem into thousands of simple independent operations which can be done simultaneously. The elaboration of such simple operations require simple hardware units that can be implemented in a chip with a clear and simple architecture. The resulting architecture is able to perform the computation in a fully parallel way, thus tremendously reducing the computational time. It seems thus to be promising to base the cognitive models on parallel architectures to achieve an efficient operation.

This paper introduces a model strongly based on the cognitive functions of the visual cortex for extracting image features of contour line segments. The model is based on the analogy of the mammalian visual system. Each phase from the retina to the visual cortex is represented in the model by imitating the biological structures and cognitive functions in order to perform similar image transformations and operations. In classical image processing algorithms, such as edge detection using a Sobel filter, both the input and the output are pixels arranged in a matrix. These algorithms thus represent a pixel-to-pixel transformation between two matrices.

The notion of an image feature, or simply a *feature*, is defined as a visual object, which can range from a single pixel or edge element through an oriented line segment until a more complex corner or even a triangle. This suggests the introduction of a hierarchical organization of features along the abstraction dimension. So far, many work has dealt with the hierarchical organization of features according to scale factors [3, 4, 5, 6]. The abstraction hierarchy first introduced by Granlund [7] employs symmetry properties implemented by Gabor functions.

Accordingly, the more complex a feature is, the higher level of abstraction it is classified. A one-pixel-size feature can be considered as a feature of the lowest level abstraction. Similarly to the neural networks in the cerebral cortex, the proposed model implements a pixel-to-feature transformation, which should more precisely be referred to as a low-level-feature to high-level-feature transformation. The result of the transformation is thus a higher level feature abstraction of the input image. The abstract features can also be re-transformed into the lower level features they are composed of. In the case of a feature composed of pixels, this re-transformation will result in a pixel level representation of the features of higher level abstraction. The re-transformation of features into lower level features excludes noise from the result, thus it can be used as a filtering technique, described later in this paper.

The rest of the paper is organized as follows. Section 2 gives an introduction to the visual pathway, how the brain processes an image. Section 3 describes the proposed architecture of the model for high speed image processing. Section 4

is devoted to the model evaluation and experimental results. The fundamental ideas of the hardware realization of our model is discussed in Section 5. Finally, Section 6 concludes the paper.

2. The Visual Pathway from the Retina to the Primary Visual Cortex

The main goal of this paper is to present a cognitive model based on the visual pathway with a special respect on the primary visual cortex. The purpose of this section is to give an overview of the biological and cognitive aspects of early visual information processing, on which the model is based.

Visual processing begins in the retina. The photoreceptors that include 120 million rods and more than 5 million cones are located in the outer plexiform layer of the retina. The rods are sensitive to light intensity and are responsible for phototransduction [8], while cones are sensitive to the wavelength of the light [9]. These photoreceptors modulate the activity of the bipolar cells, which in turn connect with more than one million ganglion cells in each eye. The axons of the ganglion cells leave the eye at the optic disc and form the optic nerve, which carries information from the retina to the brain.

The bipolar cells and the ganglion cells are organized in such a way that each cell responds to light falling on a small circular patch of the retina, which defines the cell's *receptive field.* Both bipolar cells and ganglion cells have two basic types of receptive fields: on-center/off-surround and off-center/on-surround. The center and its surround are always antagonistic and tend to cancel each other's activity [10, 11]. On the other hand, the on/off or off/on arrangement of the receptive field makes ganglion cells more responsive to differences in the level of illumination between the center and surround of its receptive field. Uniform illumination of the visual field is less effective in activating a ganglion cell than is a well placed spot or line or edge passing through the center of the cell's receptive field.

The main target of the axons of the ganglion cells are the lateral geniculate nucleus (LGN) of the thalamus, and the superior colliculus. The LGN is the main conduit to the primary visual cortex where conscious visual perception occurs. The superior colliculus is involved in guiding eye movements and other automatic visuo-motor responses. The primary visual cortex (which is also referred to as V1, the striate cortex, or area 17) populates approximately 2 billion neurons in a two-dimensional sheet about 2–3 mm thick. Visual information processing totals up to a vast portion of cortical activity and is composed of more than a dozen separate areas. In macaque monkeys, the visual cortex constitutes about 50% of the surface area of the entire cerebral

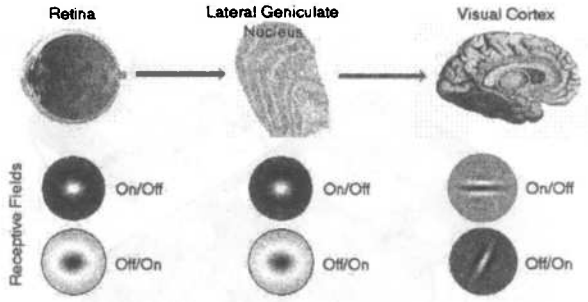


Figure 1. The visual pathway from the retina through the lateral geniculate nucleus to the visual cortex. The shape of the corresponding classical receptive fields varies from circular in the retina and LGN to elongated in the cortex. Orientation selectivity occurs only in cortical neurons.

cortex, while in humans this fraction is about 20%. The primary visual cortex topographically maps the visual field, with neighboring neurons responding to neighboring parts of the visual field.

Neurons in the primary visual cortex can be classified in two major classes according to their response characteristics: simple-cells and complex cells [2]. Simple cells tend to receive afferent projections mostly from the LGN, while complex cells receive projections mostly from other cortical cells [12]. Both of these cells exhibit a property known as orientation selectivity, meaning that they do not respond simply to light or dark in the visual field, but more typically to bars or edges of light with a particular orientation [13].

The visual cortex has a columnar organization on the cellular level. In 1977, Hubel and Wiesel suggested that iso-orientation domains are packed in essentially linear parallel stripes, which Hubel [1] subsequently referred to as the "ice-cube" model. The model of Hubel, and later V1 models [14] suggest that cells in the visual cortex are organized in a 3D structure, where a location on the visual field and an input stimulus preference (*e.g.* orientation preference) can be assigned to each cell, as shown in Figure 2.

While simple cells respond to an oriented edge at a particular position of the visual field, complex cells exhibit more robust functionalities. An example of cortical processing in primary visual cortex is *length-tuning* or *end-inhibition.* Hubel and Wiesel first described complex cells in which the response to a stimulus increases with the length of the stimulus up to some optimum value, after which further increases in length decreased the response [15].

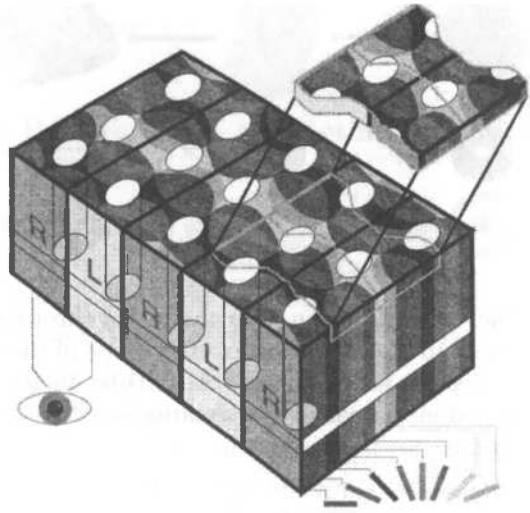


Figure 2. Schematic map of the visual cortex at work. This amazingly orderly "mosaic" of the working brain is formed by three groups of neurons performing different tasks: 1) Black lines mark the borders between columns of neurons that receive signals from the left and right eye and are responsible for the binocular perception of depth. 2) White ovals represent groups of neurons responsible for color perception (blobs). 3) The 'pinwheels' are formed by neurons involved in the perception of shape, with each color marking neurons responsible for a particular orientation of the visual field. (Reprinted with permission from [16])

3. Cognitive model of the visual pathway

A scene projected to the retina becomes a two-dimensional image, which has to be transferred to the brain for further processing. Such an image is composed of image features like regions of a certain color and texture, their boundaries as segments of different orientation and length. The image features make part of more abstract features like simple shapes, curves, circles.

The work of Hubel and Wiesel states the existence of simple and complex neurons in the visual cortex [2]. Tao goes further, and introduces the complexity of neurons as a quantitative descriptor [12]. The larger synaptic distance a neuron is from the input, the more complex it is. This suggests that neurons connected in a complex network can be hierarchically classified into different levels corresponding to the synaptical distance from the input. The neurons

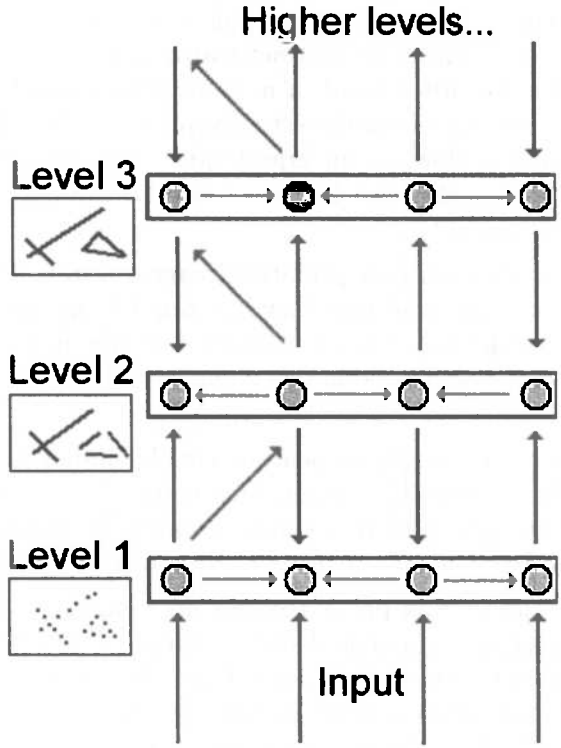


Figure 3. The neural hierarchy

in the n^{th} level may receive input from levels l_i where $l_i \geq n-1$. This means that a neuron in a certain level can receive input from one level bellow or from the same or higher levels. The first level consists of neurons that directly receive an input signal. In this paper we refer to such an organization as a neural hierarchy. Since each level is representing the abstraction of the level bellow, it can be supposed that the higher is the level of a neuron, the more abstract feature it can represent (Figure 3).

The main goal of the authors is to propose a cognitive model, which is able to *understand* (i.e. represent at an abstract level) the basic primitives (features) of an image, analogically to the cerebral cortex. In neurobiology a feature is *understood* when it causes the intensive firing of a set of neurons. In the proposed model a feature is represented by the activation of a single neuron instead of a set. A feature is considered to be *understood* by the model when the neuron corresponding to the actual feature has a high output. The neurons

representing features can project their outputs to higher and lower levels in the neural hierarchy. Projecting the output further up allows the neurons in higher levels to understand more abstract features as the composition of lower level features. On the other hand, a neurons that project their outputs to lower levels in the neural hierarchy actually provide a top-down information flow. This information flow, as an *expectation* may influence the neurons in lower layers to represent different features from the case when only bottom-up information flow is present.

This paper concentrates on how primitive image features (line segments) are understood by the model, and how they can provide an expectation for lower levels. The understanding of more abstract features in higher levels of the neural architecture is not treated in this paper, it will be the subject of further research.

In the visual system a variety of neurons can be found from ganglion cells through LGN cells to cortical neurons, each responding to different preferred afferent stimulation. The preferred stimulation can be described by the properties of the receptive field of a neuron, as described in Section 2.

The proposed model in this paper receives an image on its input, which is immediately subjected to an edge detection filter. This filter is based on the receptive field characteristics of the retinal ganglion cells. In the small region of the visual field which is centered around the position of the receptive field of the ganglion cell the afferent connections have a relatively high positive weight, while in the surrounding regions the synapse weights are inhibitory. The receptive field is modeled with a 3×3 matrix M_1 with higher positive input weight values in the middle and small negative values in the surrounding regions. The sum of the values of the filter matrix have to be zero so that no constant component is added to the result. The matrix as a non-directional derivative filter should be symmetric along all the axes. These two constraints explain the choice of the filter matrix:

$$M_1 = \begin{pmatrix} -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} \\ -\frac{1}{8} & 1 & -\frac{1}{8} \\ -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} \end{pmatrix} \tag{3.1}$$

The output pattern of the cells with input weights of M_1 will be an edge detected image of the original image. It is to note that at this level of neural processing the image features understood (or represented by neural activation) are pixels of an edge detected image, edge elements.

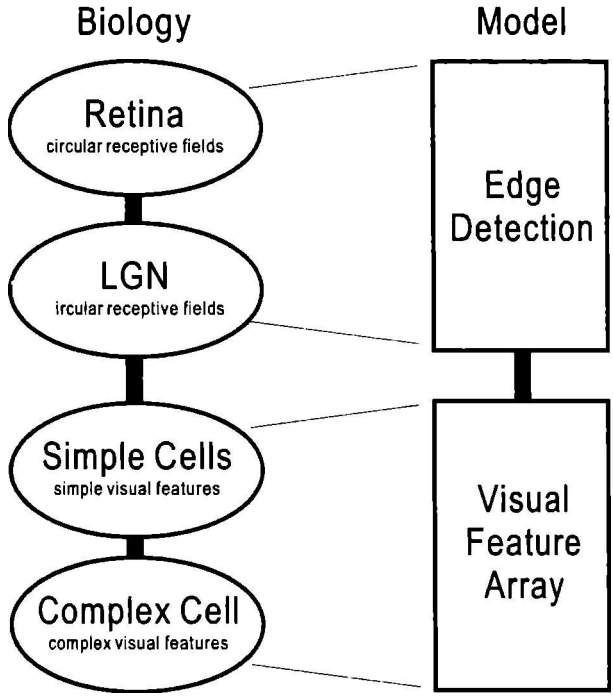


Figure 4. The biological system and the components of the proposed model that cover the biological functionalities.

Going further on the visual pathway we find that the receptive fields of the neurons in LGN are also circular like those in the retina. It rather has an important role in modulating the input to the cortex by attention, but the exact functionality is still a subject of research.

For the above reason we consider the retinal and LGN-neurons as primary edge detectors, and their overall functionality in the aspects of image processing is covered by the M_1 matrix in the model. The input from the cells of such receptive fields project into the visual cortex, where further image processing takes place. The correspondence between the biological functionalities and the model components that cover them are shown in Figure 4.

The image representation in the visual cortex is retinotopic, which means that neighboring regions of the visual field are projected to neighboring regions in the cortex. The neurons of such a region are tuned to respond to a variety of input stimuli described by different receptive fields characteristics, as explained in Section 2. This implies that a vast variety of receptive fields belong to one small region of the visual cortex, and thus to a small region of the visual field.

The variety of receptive fields representing different visual features (*e.g.* line orientations) can be organized along new dimensions.

As a result of the edge detection an edge detected image is available in the matrix I where

$$I \in \mathbb{R}^{n \times m}, \tag{3.2}$$

n and m representing the image dimensions. The elements of the matrix I are bounded, such that

$$I_{i,j} \in [0, 1], \tag{3.3}$$

where $I_{i,j}$ represents the pixel in the i^{th} row and j^{th} column of the matrix I.

Similarly to the visual cortex, several different features can be extracted from the edge detected image I. The extraction of the features begins with the longest line segments, those spanning through the largest angle in the visual field, and thus causing activation in the largest number of ganglion cells, or pixels in the context of a CCD imager. When the first feature is extracted from the edge detected image I, the feature pixels are removed from I, resulting a new matrix that we refer to as $I^{(1)}$ After extracting and removing the k^{th} feature from $I^{(k-1)}$ the matrix $I^{(k)}$ remains. Using this notation the original edge detected image is denoted $I^{(0)}$ This step is necessary to ensure that only one of many possible similar features is extracted from the edge detected image $I^{(0)}$ The k^{th} feature is removed from $I^{(k-1)}$ and added to a two-dimensional matrix F_k, such that

$$\forall i, j, k \quad (F_k)_{i,j} \in \{0; 1\}, \tag{3.4}$$

and the value $(F_k)_{i,j}$ indicates if any pixel of the detected feature k is present in the edge detected image at the position $I_{i,j}^{(k-1)}$

It is important to note that the features to be extracted are ordered by the number of pixels they contain in order to ensure that

$$\mathcal{F}_k \supseteq \mathcal{F}_l, k < l, \tag{3.5}$$

where $\mathcal{F}_k$ is the set of pixels contained by the k^{th} feature. Since there are several image features to be extracted from the image, there will be a matrix F for each of these features. We define the three-dimensional array with the F matrices overlapped along a third dimension as follows:

$$\mathcal{V} \in \mathbb{R}^{n \times m \times r} \tag{3.6}$$

For the tensor $\mathcal{V}$ we introduce the notion of *Visual Feature Array* or *VFA*, where r represents the total number of visual features extracted from the image. By construction, the element $\mathcal{V}_{i,j,k}$ of the VFA represents if an edge pixel $I_{i,j}^{(k-1)}$ belongs to the k^{th} visual feature.

In the VFA each element corresponds to the response of a cortical neuron tuned to a certain feature in a certain location. The representation shown in Figure 2 shows that the neurons tuned to different visual features in the visual cortex are organized in a rather sophisticated system. In the VFA the same features are organized along a third dimension, orthogonal to the other two dimensions. Such a system of visual features yields a 3-dimensional neural array model of the primary visual cortex.

Let's take a closer look on the third dimension of the VFA.

In the visual cortex there are neurons tuned to a whole variety of visual features. The present model includes the orientation selective cortical cells with end-inhibition characteristics. There are other visual features in the brain, such as sensitivity to spatial frequency, eye preference or binocular depth cues, but these features are not included in our model yet. Each feature in the VFA can thus be described by an orientation angle and an optimal length. The possible orientations are equally distributed with a specified angular resolution. The angles represented in the VFA are defined with the angle α and angular resolution θ, such that

$$\alpha \in [0 \dots \pi], \alpha = k \cdot \theta, k \in \mathbb{N}, \tag{3.7}$$

and thus the matrix elements $(F_{\alpha=\pi/5})_{i,j}$ will be values of 1 where an edge line segment with an orientation close to $\pi/5$ is found in the edge detected image at $I_{i,j}$.

The end-inhibition property of the neurons is also formalized in the model. An optimal length l of a neuron is a length to which it gives a maximal response. The different lengths are distributed between the shortest length and the longest length, and their number is h. Since the line lengths are measured in pixels, the shortest possible line segment is 3 pixel long. The maximal length can be chosen taking the requirements of the input image and the available computational capacity into consideration. Normally this value is between 20 and 30 pixels.

Given an angular resolution of θ and the number of different length values h, the number of possible visual features r can be assessed as follows:

$$r = \frac{\pi}{\theta} \quad h. \tag{3.8}$$

A visual feature k is thus characterized by two values, an orientation α and length l. The matrix elements $(F_k)_{i,j}$ will thus have a value of 1 if the edge pixel on the edge detected image $I_{i,j}$ belongs a feature with the characteristics of k.

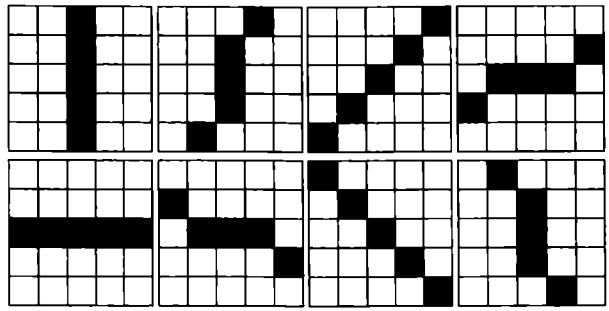


Figure 5. The matrices in the model that represent the receptive fields of cortical orientation tuned end-inhibited cells of 5-pixel-length.

In the visual cortex there are receptive field characteristics that actually define the visual feature the particular neuron is responsive to, as described in section 1 and shown on the bottom of Figure 1. In order to extract the desired features from an edge detected image, for each feature k a mask matrix R_k obtained from a corresponding receptive field has to be defined. In the proposed model the visual features are extracted by a convolution of the edge detected image and a matrix R_k. In the present case the receptive fields are modeled by binary matrices instead of matrices with real values. These matrices contain the sought feature as it may appear on the binary edge detected image. We have chosen to use binary matrices to detect visual features because it is possible to well approximate the sought features, and binary operations are easier to implement in a hardware. A series of mask matrices for all the possible five-pixel-long lines are shown in Figure 5.

Once the VFA is constructed from the edge detected image I, it can be subjected to further transformations in order to extract more abstract features from it. As it was described above, one layer in the VFA contains the pixels that belong to a well-defined feature (i.e. a line of a certain length and orientation). A grouping transformation can be defined on the VFA, which unifies the layers and thus groups the features of the VFA according to different feature properties.

Two basic grouping transformations are defined:

$$\mathcal{G}_o \quad \mathcal{V} \rightarrow \mathcal{V}^{(o)}, \tag{3.9}$$

and

$$\mathcal{G}_l \quad \mathcal{V} \rightarrow \mathcal{V}^{(l)} \tag{3.10}$$

The result of $\mathcal{G}_o(\mathcal{V})$ is $\mathcal{V}^{(o)}$, which contains iso-orientation layers, where all the features of the same orientation are present in one layer. This step was

inspired by the iso-orientation columns found in the primary visual cortex by Hubel and Wiesel, as described in the section 2. On the other hand such a grouping transformation is necessary to find the line crossings and vertices in the VFA.

The result of $\mathcal{G}_l(\mathcal{V})$ is $\mathcal{V}^{(l)}$, which contains iso-length layers, where all the features of the same length are present in one layer. This transformation can be useful in segmenting short and long line segments from each other. Short line segments of arbitrary orientation are usually the components of textures of natural objects (trees or bushes). Longer, parallel and orthogonal line segments usually make part of artificial (man made) objects or scenes, such as an urban scene.

The nodes of $\mathcal{V}$, $\mathcal{V}^{(o)}$ or $\mathcal{V}^{(l)}$ can send their outputs to higher or lower levels of the neural hierarchy. Sending the inputs further up allows further transformations and the recognition of more complex features or objects. Sending the output back in the neural hierarchy allows feedback and reinforcement in lower neural structures.

For instance, line crossing and vertex detection can easily be done by sending the output of the $\mathcal{V}^{(o)}$ neurons further *up* in the neural hierarchy. A layer of neurons organized in a two dimensional matrix $C \in \mathbb{R}^{n \times m}$ receives input from $\mathcal{V}^{(o)}$ and provides an output according to the function f as

$$C_{i,j} = f(\mathcal{V}^{(o)}_{i,j,1}, \mathcal{V}^{(o)}_{i,j,2}, \ldots, \mathcal{V}^{(o)}_{i,j,o}), \tag{3.11}$$

where o is the number of line orientations. A neuron in $C_{i,j}$ will have a high output if it receives more than one active inputs, meaning that there is more than one differently oriented line at the same image location. Figure 6 shows an example, where the red circles are neurons in C indicating line crossings, while the blue circle indicates no line crossing.

It is to note that the use of the original VFA $\mathcal{V}$ is not appropriate in finding the vertices, because two colinear line segments may overlap each other. If so, their overlap will be considered as a vertex, which is not desirable. If the $\mathcal{V}^{(o)}$ is used, only one neuron from one position and one orientation will send input to C, and thus two overlapping collinear line segments will not activate C.

One can consider a third type of grouping transformation on the VFA, which simply groups all the layers into one final layer containing all the extracted features. This transformation equals sending the output of the VFA neurons *down* in the neural hierarchy, and can be used to reconstruct an image by reactivating the pixels that belong to the detected visual features. This reconstruction will include only the features that were extracted from the original image. This implies that the noise (pixels not considered as the part of any feature) will not be present in the reconstructed edge detected image. The

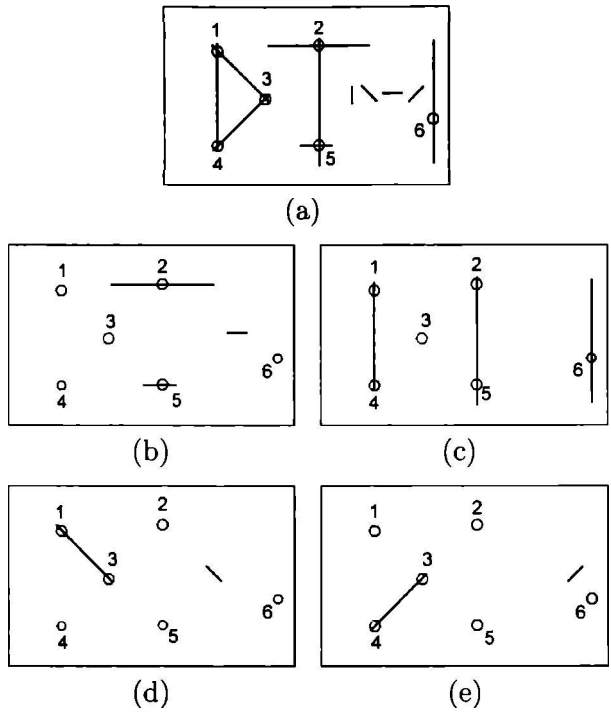


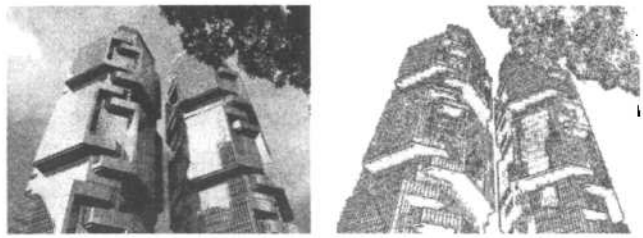
Figure 6. Detection of line segment crossing and vertices. Image (a) shows the edge detected image, images (b)-(e) show the results of four grouping transformations of the VFA ($\mathcal{V}_1^{(o)}, \mathcal{V}_2^{(o)}, \ldots, \mathcal{V}_4^{(o)}$). Red circles indicate line crossings, the blue circle shows an example of no crossing. Red circles contain active neurons in the VFA in more than one group, while the blue circle contains active neurons only on (c).

comparison or merging of the reconstructed and the original edge detected image actually adds information to the original image.

We introduce the notion of *negative filtering* as the process of understanding image primitives and reconstructing the image from them. The notion arose from the fact that on contrary to a filtering process, the above defined process adds useful information to the image, instead of subtracting it.

4. Model evaluation, results

The proposed model has two important advantages compared to classical solutions. By virtue of the simple but numerous computational units (neurons)



(a) (b)

Figure 7. Original test image (a) and the result of the primary edge detection (b)

that work parallel on the solution, the model can perform the proper activation of the VFA and the negative filtering in constant time. This, however requires a parallel hardware implementation of the model.

In this paper only a computer based simulation of the model is presented, which allowed to evaluate its functionalities. The evaluation of the performance was however not possible due to the lack of a hardware implementation. In the rest of this section the different sections of the information flow within the model will be presented.

The input test image used to evaluate the model is shown in Figure 7a. This image is subjected to a primary edge detection as discussed in section 3. The result is a binary image of edge elements, with white dots representing high-contrast points on the original image. This edge-detected image is shown in Figure 7b.

The edge-detected image within the model corresponds to the image that is projected to the visual cortex. In the model, this image is used as the input to the neurons in the VFA. In the present implementation 5 different line lengths were used with the possible orientations to calculate the values of the VFA. These lengths were 3, 5, 9, 17, and 33 pixels.

The VFA layers after the grouping transformations with 3, 9 and 33 pixel-long segments are shown in Figure 8. Using the grouping transformation $\mathcal{G}_l$ that yields the VFA $\mathcal{V}^{(l)}$, having 5 layers each of them containing the line segments of all the possible orientations of a certain length. Three out of these five layers are shown on Figure 8.

The union of the five layers of $\mathcal{V}^{(l)}$ yields the top-down reconstruction of the edge detected image from the detected line segments. The reconstruction will exclude the edge elements detected as noise, which was not recognized as a visual feature (a line segment of certain length and orientation). The final,

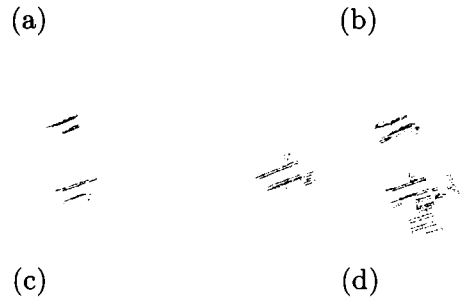


Figure 8. The reconstruction of the edge-detected image from line segments of 3 pixels (a), 9 pixels (b), 33 pixels (c) and the combination of all sub-VFAs (d)

fully reconstructed, negative filtered image composed from the five layers of $\mathcal{V}^{(l)}$ is shown in Figure 8d.

5. Hardware realization

One of the most advantageous properties of the proposed architecture is its native parallelism. A software implementation running on the fastest *von Neumann*-based processor cannot provide fast, $O(1)$ time responses even if they are extended with Single Instruction Multiple Data (SIMD) instruction sets like Intel's Streaming SIMD Extensions (SSE). Parallel processing with multiple simple computational elements, on the other hand, can provide tremendous speedups. The Field-Programmable Gate Array (FPGA) is such a microelectronic device the programming logic of which can be set up according to the users' needs, and some models even allow to be reconfigured during operation time. Thus, the proposed architecture can be implemented in an FPGA, and then can be used as a coprocessor or accelerator card in a PC environment to solve dedicated tasks. Moreover, it can be a stand-alone image processing device that solves the task without the execution of any conventional algorithm.

The proposed architecture requires about 10 000 computational elements to perform the edge detection on a 100×100 pixel image. Then for each direction of each length another 10 000 computational elements are necessary, that is in

total $124 \times 10\,000$. The state of the art FPGA has about 6 000 000 logic cells that is sufficient for about 100–200 000 computational elements. This number copes with what is necessary, while the processed image is still relatively small. However, the famous Moore's Law also applies to FPGAs saying that in about every two years the number of transistors on a silicon chip doubles, thus the number of logical cells is expected to double, too. In addition, some of the modern FPGAs also have the capability of being reprogrammed in runtime. Applying this feature allows the use of only one chip for the processing that can be done by reprogramming the architecture for each task, sacrificing extra processing time. In conclusion, a primary visual cortex based image contour detector chip can be realized in near future by some compromises.

A simple (low resolution) version of the model is being implemented in an FPGA. A serious bottleneck in this solution is the small number of parallel input/output data that can transmitted to and from the FPGA. Apart from this problem, the implementation will give ground to test and evaluate the model operating on a dedicated hardware.

6. Conclusion

A model for intelligent contour detection was presented in this paper. The basic structure and functionality of the model is based on the mammalian primary visual cortex, which can perform edge contour extraction on an edge detected image. The extracted contour pixels are clustered into a hierarchical classification of visual features. The features are organized into a three-dimensional orthogonal array (the VFA) according to their properties. The extracted features are used in two ways: further abstraction or top-down image reconstruction. This latest adds an augmented information space to the original edge detected image, which we refer to as negative filtering.

It is important to note that the goal of this model was not to achieve a qualitative advance in image contour detection, but to make the first step towards a biology and cognitive science inspired vision system. The new approach is expected to lead to a cognitive system overpassing the performance of classical computational methods. Meanwhile, the quality of the contour detection achieved by the proposed model is comparable to classical edge detection algorithms.

The VFA containing different features can be submitted to grouping transformations, that merge layers of the VFA according to certain rules, such as similar line length or orientation. The grouping transformations are necessary for further transformations, such as line crossing and vertex detection.

The model and especially the VFA has been designed to operate in a fully parallel manner. In the present system binary array values were used for the sake of easy hardware implementation. An FPGA or other parallel implementation of the model yields a constant time contour detection and visual feature extraction.

REFERENCES

[1] HUBEL, D.: *Eye, Brain and Vision.* W.H. Freeman & Company, 1995, ISBN 0716760096.

[2] HUBEL, D. H. and WIESEL, T. N.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiology*, **160**, (1962), 106–154.

[3] GRANLUND, G. H. and KNUTSSON, H.: Signal processing for computer vision. *Kluwe Academic Publishers.*

[4] KOENDERINK, J. J. and VAN DOORN, A. J.: The structure of images. *Biol. Cybernet.*, **50**, (1984), 363–370.

[5] LIFSHITZ, L. M.: *Image segmentation via multiresolution extrema following.* Tech. rep., University of North Carolina, 1987.

[6] WITKIN, A.: Scale-space filtering. In *Proc. 8th Internat. Joint Conf. Artificial Intelligence*, 1983, pp. 1019–1022.

[7] GRANLUND, G. H.: In search of a general picture processing operator. *Comput. Graphics Image Process.*, **8**(2), (1978), 155–178.

[8] POLANS, A., BAEHR, W., and PALCZEWSKI, K.: Turned on by Ca^{2+}! the physiology and pathology of Ca^{2+}-binding proteins in the retina. *Trends in Neurosciences*, **19**(12), (1996), 547–554.

[9] DARTNALL, H. J. A., BOWMAKER, J. K., and MOLLON, J. D.: Human visual pigments: Microspectrophotometric results from the eyes of seven persons. *Proc. of Royal Society of London, B.*, **220**, (1983), 115–130.

[10] BARLOW, H. B.: Summation and inhibition of the frog's retina. *J. Physiology*, **119**, (1953), 69–88.

[11] KUFFLER, S. W Discharge patterns and functional organization of mammalian retina. *J. Neurophysiology*, **16**, (1953), 37–68.

[12] TAO, L. SHELLEY, M. MCLAUGHLIN, D., and SHAPLEY, R.: An egalitarian network model for the emergence of simple and complex cells in visual cortex. *PNAS*, **101**(1), (2004), 366–371.

[13] HUBEL, D. H. and WIESEL, T. N.: Receptive fields and functional architecture of monkey striate cortex. *J. Physiology*, **195**, (1968), 215–243.

[14] GRINVALD, A., MALONEK, D., SHMUEL, A., GLASER, D., VANZETTA, I., SHTOYERMAN, E., SHOHAM, D., and ARIELI, A.: *Imaging of Neuronal Activity.* Cold Spring Harbor Laboratory, 1999.

[15] HUBEL, D. H. and WIESEL, T. N.: Receptive field and functional architecture in two nonstriate visual areas (18–19) of the cat. *J. Neurophysiology*, **28**, (1965), 229–289.

[16] GRINVALD, A., MALONEK, D., SHMUEL, A., GLASER, D., VANZETTA, I. SHTOYERMAN, E., SHOHAM, D., and ARIELI, A.: *Imaging of Neuronal Activity*, chap. Intrinsic signal imaging in the neocortex, pp. 1–17. Cold Spring Harbor Laboratory, 1999.

Production Systems and Information Engineering
Volume 4 (2006), pp. 53–63

TRADE-OFF PROPERTIES OF TENSOR PRODUCT MODEL TRANSFORMATION: A CASE STUDY OF THE TORA SYSTEM

ZOLTÁN PETRES
Computer and Automation Research Institute,
Hungarian Academy of Sciences, Hungary
petres@tmit.bme.hu

PÉTER BARANYI
Computer and Automation Research Institute,
Hungarian Academy of Sciences, Hungary
baranyi@sztaki.hu

[Received November 2005 and Accepted March 2006]

Abstract. The Tensor Product (TP) based models have been applied widely in approximation theory, and approximation techniques. Recently, a controller design framework working on dynamic systems has also been established based on TP model transformation combined with Linear Matrix Inequalities (LMI) within Parallel Distributed Compensation (PDC) framework. The effectiveness of the control design framework strongly depends on two main properties of the TP model used. One of them is the approximation accuracy, and the other one is computational complexity. Therefore, the primary aim of this paper is to investigate the relation of the two contradictory goals, namely, the trade-off between the dynamic TP model's accuracy and complexity. The study is conducted through the example of Translational Oscillations with a Rotational Actuator (TORA) system.

Keywords: Tensor Product model transformation, approximation accuracy, computational complexity, TORA System

1. Introduction

The demand for the decomposition of multivariate functions to univariate ones goes back to the very end of the 19th century. In 1900, in his memorable lecture at the Second International Congress of Mathematicians in Paris, D. Hilbert, the famous German mathematician, listed 23 conjectures, hypotheses

concerning unsolved problems which he considered would be the most important ones to solve by the mathematicians of the 20th century [1, 2]. In the 13th, he addressed the problem of multivariate continuous function decomposition to finite superposition of continuous functions of fewer variables. The motivation is straightforward: one dimensional functions are much easier to calculate with, handle and visualize. Hilbert presumed that this problem cannot be solved in general, *i.e.* there exist multivariate continuous function that cannot be decomposed to univariate continuous functions. This was disproved by Kolmogorov in 1957 [3] in his general representation theorem, when he provided a constructive proof.

TP based approximation has reached modeling approaches of non-linear dynamic systems, and furthermore, there are now controller design frameworks based on TP model [4, 5]. Generally, TP model, in broad sense, is an approximation technique where the approximating functions are in tensor product form, whereas a TP model form is a particular approximating function in a TP model. In this paper we consider TP model in a narrower sense, when a TP model is applied to dynamic system control.

A large variety of LMI based control design techniques have been developed in the last decade [6, 7]. Powerful commercialized softwares have also been developed [8] for solving LMIs and related control problems. Recently, a number of LMI based controller designs have been carried out for TP models (also termed as polytopic or TS models in fuzzy theory) under PDC [9]. Further, a TP model transformation has been developed to transfer non-linear dynamic models to TP model whereupon PDC design frameworks can readily be executed [4, 5]. One can find a case study of TP model transformation in the control design of a prototypical aeroelastic wing section [10] that exhibits various control phenomena such as limit cycle oscillation and chaotic vibration.

A crucial point of these control design frameworks is the modeling accuracy. If TP model does not appropriately describe the real system the resulting control may not ensure the required control performance. On the other hand, by increasing the modeling accuracy the model's complexity also drastically increases and makes difficult any further calculation. Therefore, an optimal trade-off has to be chosen between the modeling accuracy and computational complexity for efficient controller design. This paper is devoted to analyze the approximation capabilities and complexity issues of TP model forms when applied to a case study, the Translational Oscillations with a Rotational Actuator (TORA) system.

The paper is organized as follows: Section 2 introduces the fundamentals of TP modeling. Section 3 discuss the TORA system and the properties of the resulting TP forms. Section 4 derives some conclusions.

2. Preliminaries

2.1. Linear Parameter-Varying state-space model

Consider the following parameter-varying state-space model:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\mathbf{p}(t))\mathbf{x}(t) + \mathbf{B}(\mathbf{p}(t))\mathbf{u}(t), \tag{2.1}$$

$$\mathbf{y}(t) = \mathbf{C}(\mathbf{p}(t))\mathbf{x}(t) + \mathbf{D}(\mathbf{p}(t))\mathbf{u}(t),$$

with input $\mathbf{u}(t)$, output $\mathbf{y}(t)$ and state vector $\mathbf{x}(t)$. The system matrix

$$\mathbf{S}(\mathbf{p}(t)) = \begin{pmatrix} \mathbf{A}(\mathbf{p}(t)) & \mathbf{B}(\mathbf{p}(t)) \\ \mathbf{C}(\mathbf{p}(t)) & \mathbf{D}(\mathbf{p}(t)) \end{pmatrix} \in \mathbb{R}^{O \times I} \tag{2.2}$$

is a parameter-varying object, where $\mathbf{p}(t) \in \Omega$ is time varying N-dimensional parameter vector, and is an element of the closed hypercube $\Omega = [a_1, b_1] \times [a_2, b_2] \times \quad \times [a_N, b_N] \subset \mathbb{R}^N$ $\mathbf{p}(t)$ can also include some elements of $\mathbf{x}(t)$.

2.2. Convex state-space TP model

$\mathbf{S}(\mathbf{p}(t))$ can be approximated for any parameter $\mathbf{p}(t)$ as the convex combination of LTI system matrices $\mathbf{S}_r$, $r = 1, \ldots, R$. Matrices $\mathbf{S}_r$ are also called *vertex systems*. Therefore, one can define weighting functions $w_r(\mathbf{p}(t)) \in [0, 1] \subset \mathbb{R}$ such that matrix $\mathbf{S}(\mathbf{p}(t))$ can be expressed as convex combination of system matrices $\mathbf{S}_r$. The explicit form of the TP model in terms of tensor product becomes:

$$\begin{pmatrix} \dot{\mathbf{x}}(t) \\ \mathbf{y}(t) \end{pmatrix} \approx \mathcal{S} \bigotimes_{n=1}^{N} \mathbf{w}_n(p_n(t)) \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix} \tag{2.3}$$

that is

$$\left\| \mathbf{S}(\mathbf{p}(t)) - \mathcal{S} \bigotimes_{n=1}^{N} \mathbf{w}_n(p_n(t)) \right\| \leq \varepsilon.$$

Here, ε symbolizes the approximation error, row vector $\mathbf{w}_n(p_n) \in \mathbb{R}^{I_n}$ $n = 1, \ldots, N$ contains the one variable weighting functions $w_{n,i_n}(p_n)$. Function $w_{n,j}(p_n(t)) \in [0, 1]$ is the j-th one variable weighting function defined on the n-th dimension of Ω, and $p_n(t)$ is the n-th element of vector $\mathbf{p}(t)$. I_n $(n = 1, \ldots, N)$ is the number of the weighting functions used in the n-th dimension of the parameter vector $\mathbf{p}(t)$. The $(N+2)$-dimensional tensor $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N \times O \times I}$ is constructed from LTI vertex systems $\mathbf{S}_{i_1 i_2 \ldots i_N} \in \mathbb{R}^{O \times I}$ For further details we refer to [10, 4, 11]. The convex combination of the LTI vertex systems is ensured by the conditions:

Definition 1. *The TP model (2.3) is convex if:*

$$\forall n \in [1, N], i, p_n(t) : w_{n,i}(p_n(t)) \in [0, 1]; \tag{2.4}$$

$$\forall n \in [1, N], p_n(t) \quad \sum_{i=1}^{I_n} w_{n,i}(p_n(t)) = 1. \tag{2.5}$$

This simply means that $\mathbf{S}(\mathbf{p}(t))$ is within the convex hull of the LTI vertex systems $\mathbf{S}_{i_1 i_2 \ldots i_N}$ for any $\mathbf{p}(t) \in \Omega$.

$\mathbf{S}(\mathbf{p}(t))$ has a finite element TP model representation in many cases ($\varepsilon = 0$ in (2.3)). However, exact finite element TP model representation does not exist in general ($\varepsilon > 0$ in (2.3)), see Ref. [12]. In this case $\varepsilon \mapsto 0$, when the number of the LTI systems involved in the TP model goes to ∞. However, these models also have a finite element TP model transformation, but it is not exact, there is some approximation error. As a result we have

$$\mathbf{S}(\mathbf{p}(t)) \underset{\gamma}{\approx} \mathcal{S} \overset{N}{\underset{n=1}{\otimes}} \mathbf{w}_n(p_n(t)),$$

where the error γ is bounded as:

$$\gamma = \left(\left\| \mathbf{S}(\mathbf{p}(t)) - \mathcal{S} \overset{N}{\underset{n=1}{\otimes}} \mathbf{w}_n(p_n(t)) \right\|_{L_2} \right)^2 \leq \sum_k \sigma_k^2, \tag{2.6}$$

where σ_k are the discarded singular values.

2.3. TP model transformation

The TP model transformation starts with the given LPV model (2.1) and results in the TP model representation (2.3), where the trade-off between the number of LTI vertex systems and the ε is optimized [4]. The TP model transformation offers options to generate different types of the weighting functions $w(\cdot)$. For instance:

Definition 2. ***SN - Sum Normalization*** *Vector* $\mathbf{w}(p)$*, containing weighting functions* $w_i(p)$ *is SN if the sum of the weighting functions is 1 for all* $p \in \Omega$.

Definition 3. ***NN - Non Negativeness*** *Vector* $\mathbf{w}(p)$*, containing weighting functions* $w_i(p)$ *is NN if the value of the weighting functions is not negative for all* $p \in \Omega$.

Definition 4. ***NO - Normality*** *Vector* $\mathbf{w}(p)$*, containing weighting functions* $w_i(p)$ *is NO if it is SN and NN type, and the maximum values of the weighting functions are one. We say* $w_i(p)$ *is close to NO if it is SN and NN type, and the maximum values of the weighting functions are close to one.*

Definition 5. ***RNO - Relaxed Normality*** *Vector* $\mathbf{w}(p)$*, containing weighting functions* $w_i(p)$ *is RNO if the maximum values of the weighting functions are the same.*

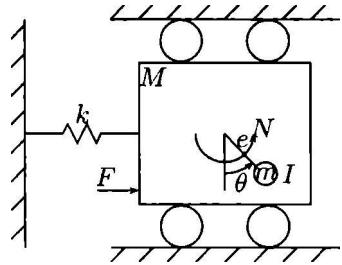


Figure 1. TORA system

Definition 6. ***INO - Inverted Normality*** *Vector* $\mathbf{w}(p)$*, containing weighting functions* $w_i(p)$ *is INO if the minimum values of the weighting functions are zero.*

All the above definitions of the weighting functions determine different types of convex hulls of the given LPV model. The SN and NN types guarantee (2.4), namely, they guarantee the convex hull. The TP model transformation is capable of always resulting SN and NN type weighting functions. This means that one can focus on applying LMI's developed for convex decompositions only, which considerably relaxes the further LMI design. The NO type determines a tight convex hull where as many of the LTI systems as possible are equal to the $\mathbf{S}(\mathbf{p})$ over some $\mathbf{p} \in \Omega$ and the rest of the LTI's are close to $\mathbf{S}(\mathbf{p}(t))$ (in the sense of L_2 norm). The SN, NN and RNO type guarantee that those LTI vertex systems which are not identical to $\mathbf{S}(\mathbf{p})$ are in the same distance from $\mathbf{S}(\mathbf{p}(t))$. INO guarantees that different subsets of the LTI's define $\mathbf{S}(\mathbf{p}(t))$ over different regions of $\mathbf{p} \in \Omega$.

These different types of convex hulls strongly effect the feasibility of the further LMI design. For instance paper [13] shows an example when determining NO is useful in the case of controller design while the observer design is more advantageous in the case of INO type weighting functions.

3. Case study of the TORA system

The Translational Oscillations with a Rotational Actuator (TORA) system[1] was developed as a simplified model of a dual-spin spacecraft [13]. Later, Bernstein and his colleagues at the University of Michigan, Ann Arbor, turned it into a benchmark problem for nonlinear control [14, 15, 16].

The system shown in Figure 1 represents a translational oscillator with an eccentric rotational proof-mass actuator. The oscillator consists of a cart of

[1]Also referred to as the rotational/translational proof-mass actuator (RTAC) system.

mass M connected to a fixed wall by a linear spring of stiffness k. The cart is constrained to have one-dimensional travel. The proof-mass actuator attached to the cart has mass m and moment of inertia I about its center of mass, which is located at distance e from the point about which the proof mass rotates. The motion occurs in a horizontal plane, so that no gravitational forces need to be considered. In Figure 1, N denotes the control torque applied to the proof mass, and F is the disturbance force on the cart.

Let q and $\dot{q}$ denote the translational position and velocity of the cart, and let θ and $\dot{\theta}$ denote the angular position and velocity of the rotational proof mass, where $\theta = 0\,\text{deg}$ is perpendicular to the motion of the cart, and $\theta = 90\,\text{deg}$ is aligned with the positive q direction. The equations of motion are given by

$$\begin{aligned} (M+m)\ddot{q} + kq \qquad & -me(\ddot{\theta}\cos\theta - \dot{\theta}^2\sin\theta) + F \\ (I+me^2)\ddot{\theta} \qquad & -me\ddot{q}\cos\theta + N \end{aligned}$$

With the normalization

$$\begin{aligned} \xi &\triangleq \sqrt{\tfrac{M+m}{I+me^2}}\,q, \qquad \tau \triangleq \sqrt{\tfrac{k}{M+m}}\,t, \\ u &\triangleq \tfrac{M+m}{k(I+me^2)}N, \quad w \triangleq \tfrac{1}{k}\sqrt{\tfrac{M+m}{I+me^2}}\,F, \end{aligned}$$

the equation of motion become

$$\begin{aligned} \ddot{\xi} + \xi \qquad & \varepsilon\left(\dot{\theta}^2\sin\theta - \ddot{\theta}\cos\theta\right) + w \\ \ddot{\theta} \qquad & -\varepsilon\ddot{\xi}\cos\theta + u \end{aligned}$$

where ξ is the normalized cart position, and w and u represent the dimensionless disturbance and control torque, respectively. In the normalized equations, the symbol $(\cdot)$ represents differentiation with respect to the normalized time τ The coupling between the translational and rotational motions is represented by the parameter ε which is defined by

$$\varepsilon \triangleq \frac{me}{\sqrt{(I+me^2)(M+m)}}$$

Letting $\mathbf{x} = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \end{pmatrix}^T = \begin{pmatrix} \xi & \dot{\xi} & \theta & \dot{\theta} \end{pmatrix}^T$, the dimensionless equations of motion in first-order form are given by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u + \mathbf{d}(\mathbf{x})w, \tag{3.1}$$

where

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{-1}{1-\varepsilon^2\cos^2 x_3} & 0 & 0 & \frac{\varepsilon x_4 \sin x_3}{1-\varepsilon^2\cos^2 x_3} \\ 0 & 0 & 0 & 1 \\ \frac{\varepsilon\cos x_3}{1-\varepsilon^2\cos^2 x_3} & 0 & 0 & \frac{-\varepsilon x_4 \sin x_3}{1-\varepsilon^2\cos^2 x_3} \end{pmatrix}$$

Table 1. Parameters of the TORA system

Description	Parameter	Value	Units
Cart mass	M	1.3608	kg
Arm mass	m	0.096	kg
Arm eccentricity	e	0.0592	m
Arm inertia	I	0.0002175	kg m^2
Spring stiffness	k	186.3	N/m
Coupling parameter	ε	0.200	—

$$\mathbf{g}(\mathbf{x}) = \begin{pmatrix} 0 \\ \frac{-\varepsilon \cos x_3}{1-\varepsilon^2 \cos^2 x_3} \\ 0 \\ \frac{1}{1-\varepsilon^2 \cos^2 x_3} \end{pmatrix} \qquad \mathbf{d}(\mathbf{x}) = \begin{pmatrix} 0 \\ \frac{1}{1-\varepsilon^2 \cos^2 x_3} \\ 0 \\ \frac{-\varepsilon \cos x_3}{1-\varepsilon^2 \cos^2 x_3} \end{pmatrix}$$

Note that u, the control input, is the normalized torque N and w, the disturbance, is the normalized force F In the followings consider the case of no disturbance. The parameters of the simulated system are given in Table 1.

3.1. Convex state-space TP model forms of the TORA system

We execute the TP model transformation on the LPV model (3.1) of the TORA. As a first step of the TP model transformation we have to define the transformation space Ω. We define it as $\Omega = [-a, a] \times [-a, a]$ ($x_3(t) \in [-a, a]$ and $x_4(t) \in [-a, a]$), where $a = \frac{45}{180}\pi$ rad (note that these intervals can be arbitrarily defined). The TP model transformation starts with the discretization over a rectangular grid. Let the density of the discretization grid be 101×101 on ($x_3(t) \in [-a, a]$ and $x_4(t) \in [-a, a]$).

3.1.1. Exact finite TP model

The result of the TP model transformation shows that the rank of $\mathbf{S}(p)$ in the dimension of x_3 is 4, whilst in the dimension of x_4 is 2. The singular values in each dimensions are the following: $\sigma_{1,1} = 251.62, \sigma_{1,2}$ 5.7833, $\sigma_{1,3}$ 2.8396, $\sigma_{1,4}$ 0.030969; and $\sigma_{1,2} = 251.63, \sigma_{2,2}$ 5.7833. Therefore the TORA system can be exactly given as the combination of $4 \times 2 = 8$ LTI systems:

$$\mathbf{S}(p) = \sum_{i=1}^{4} \sum_{j=1}^{2} w_{1,i}(x_3) w_{2,j}(x_4) \left(\mathbf{A}_{i,j}\mathbf{x} + \mathbf{B}_{i,j}u\right). \tag{3.2}$$

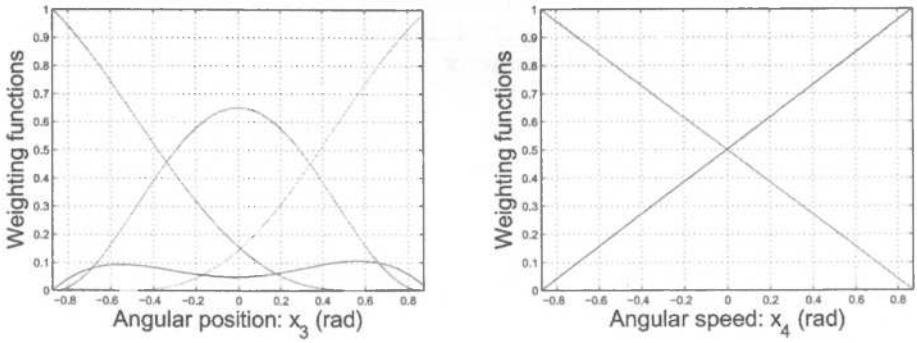


Figure 2. Close to NO type weighting functions of the exact TP model

Let us define the tight convex hull of the LPV model via generating close to NO type weighting functions by the TP model transformation, and depict them in Figure 2.

3.1.2. Approximation Trade-off of the TORA system

Even if the exact finite TP model exists, like in the case of TORA, some other reasons, such as the number of the resulting LMIs for controller design is unmanageable, or the accuracy of the actuator is much worse than the modeling accuracy, etc., can invoke the necessity to reduce the number of LTI systems.

The equation 2.6 gives a bound for the transformation error, but it is only a theoretical maximum and in most cases the resulting model has much better approximation properties. In case of the TORA system only the dimension of x_3 can be reduced, as in the dimension of x_4 if only one LTI system is kept, the Definition 1 cannot be satisfied. Thus, we repeated the TP model transformation and now only the three and two biggest singular values kept in the dimension of x_3. The result TP forms contained $3 \times 2 = 6$ and $2 \times 2 = 4$ LTI systems, respectively. During the transformation the theoretical maximum error is calculated by the equation (2.6), and also after the transformation the approximation error is measured over 10 000 sample points. Table 2 summarizes the results of the trade-off. Figure 3 and 4 show the resulting basis functions of the models.

The trade-off results showed that the size of TP model can be drastically reduced without causing unacceptable approximation error. However, it is worth noticing that the resulting reduced models might behave slightly differently

Table 2. Summary of approximation trade-off of the TORA system

Number of singular values kept	Number of LTIs	Reduction ratio	Theoretical maximal error	Measured maximal error
4	8	0%	0	10^{-12}
3	6	25%	0.0309	0.0007
2	4	50%	2.8699	0.3033

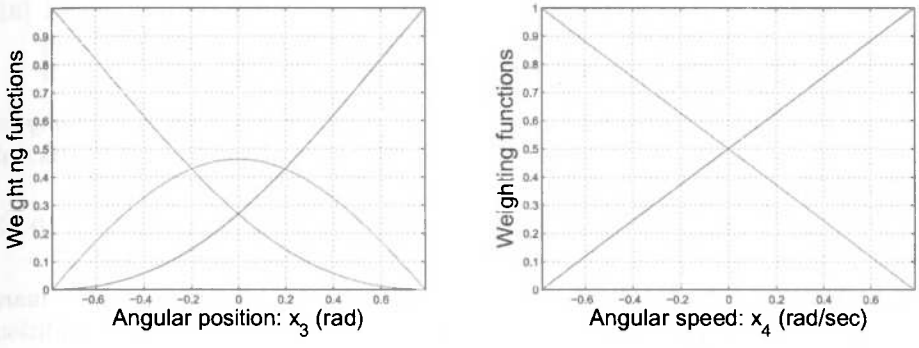


Figure 3. Close to NO type weighting functions of the reduced, 6 LTI TP model

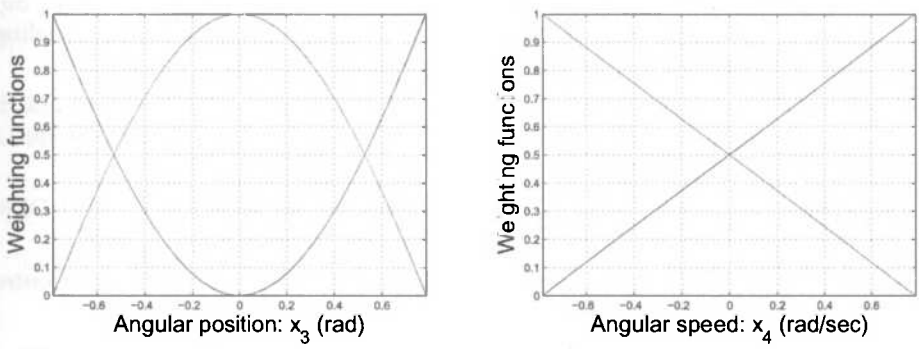


Figure 4. Close to NO type weighting functions of the reduced, 4 LTI TP model

than the exact finite model *e.g.* in controller design. These further checks are necessary to guarantee the needed behavior.

4. Conclusion

This paper shows how the TP model transformation is capable of solving the trade-off problem of the two contradictory goals, the dynamic TP model's accuracy and complexity through the case study of the TORA system. The TP model transformation gives a tool to define the theoretical maximal approximation error during the transformation. However, the case study shows that sometimes it overestimates the real error of the approximation and the model's complexity can be reduced with a large degree.

REFERENCES

[1] HILBERT, D.: Mathematische probleme. In *2nd Internation Congress of Mathematician*, Paris, France, 1900. http://www.mathematik.uni-bielefeld.de/~kersten/hilbert/rede.html.

[2] GRAY, J.: The Hilbert problems 1900–2000. *Newsletter*, **33**, (2000), 10–13. http://www.mathematik.uni-bielefeld.de/~kersten/hilbert/gray.html.

[3] KOLMOGOROV, A. N.: On the representation of continuous functions of many variables by superpositions of continuous functions of one variable and addition. *Dokl. Akad. SSSR*, **114**, (1957), 953–956. (in Russian).

[4] BARANYI, P TP model transformation as a way to LMI based controller design. *IEEE Transaction on Industrial Electronics*, **51**(2), (2004), 387–400.

[5] YAM, Y YANG, C. T. and BARANYI, P *Singular Value-Based Fuzzy Reduction with Relaxed Normalization Condition, Studies in Fuzziness and Soft Computing*, vol. 128. Springer-Verlag, interpretability issues in fuzzy modeling, J. Casillas, O. Cordon, F.Herrera, L.Magdalena (Eds.) edn. 2003.

[6] BOYD, S., GHAOUI, L. E., FERON, E., and BALAKRISHNAN, V Linear matrix inequalities in system and control theory. *Philadelphia PA:SIAM, ISBN 0-89871-334-X.*

[7] SCHERER, C. W and WEILAND, S.: *Linear Matrix Iequalities in Control.* DISC course lecture notes, http://www.cs.ele.tue.nl/SWeiland/lmid.pdf, 2000.

[8] GAHINET, P., NEMIROVSKI, A., LAUB, A. J., and CHILALI, M.: *LMI Control Toolbox.* The MathWorks, Inc., 1995.

[9] TANAKA, K. and WANG, H. O.: *Fuzzy Control Systems Design and Analysis — A Linear Matrix Inequality Approach.* John Wiley and Sons, Inc., 2001.

[10] BARANYI, P Tensor product model based control of 2-D aeroelastic system. *Journal of Guidance, Control, and Dynamics (in Press).*

[11] BARANYI, P., TIKK, D., YAM, Y., and PATTON, R. J.: From differential equations to PDC controller design via numerical transformation. *Computers in Industry, Elsevier Science*, **51**, (2003), 281–297.

[12] TIKK, D., BARANYI, P., PATTON, R. J., and TAR, J.: Approximation Capability of TP model forms. *Australian Journal of Intelligent Information Processing Systems*, **8**(3), (2004), 155–163.

[13] BARANYI, P Output-feedback design of 2-D aeroelastic system. *Journal of Guidance, Control, and Dynamics (in Press)*.

[14] BERNSTEIN, D. S.: Special issue: A nonlinear benchmark problem. *International Journal of Robust and Nonlinear Control*, **8**.

[15] BUPP, R. T., BERNSTEIN, D. S., and COPPOLA, V T.: A benchmark problem for nonlinear control design. *International Journal of Robust and Nonlinear Control*, **8**, (1998), 307–310.

[16] BUPP, R. T., BERNSTEIN, D. S., and COPPOLA, V T Experimental implementation of integrator backstepping and passive nonlinear controllers on the RTAC testbed. *International Journal of Robust and Nonlinear Control*, **8**, (1998), 435–457.

Production Systems and Information Engineering
Volume 4 (2006), pp. 65-79

CONCEPT LATTICE STRUCTURE WITH ATTRIBUTE LATTICES

László Kovács
University of Miskolc, Hungary
Department of Information Technology
kovacs@iit.uni-miskolc.hu

[Received May 2005 and Acceped March 2006]

Abstract. There is an increasing interest on application of concept lattices in the different information systems. The concept lattice may be used for representation of the concept generalisation structure generated from the underlying data set. The paper presents a modified lattice building algorithm where the generated concept nodes may contain not only the attributes of the children nodes but some other generalised attributes, too. The generalisation structure of the attributes is called attribute lattice. Using this kind of lattice building mechanism, the generated lattice and cluster nodes are more natural and readable for humans. The proposed lattice structure can be used in several kinds of information system applications to improve the quality of the query interface.

Keywords: concept lattice, lattice building

1. Standard Concept Lattice

Concept lattices are used in many application areas to represent conceptual hierarchies among the objects in the underlying data. The field of Formal Concept Analysis [1] introduced in the early 80ies has grown to a powerful theory for data analysis, information retrieval and knowledge discovery. There is nowadays an increasing interest in the application of concept lattices for data mining, especially for generating association rules [3]. One of the main characteristics of this application area is the large amount of structured data to be analysed. A technical oriented application field of Formal Concept Analysis is the area of production planing where the concept lattices are used to partition the products into disjoint groups during the optimisation of the production cost [6]. As the cost of building a concept lattice is a super-linear function of

the corresponding context size, the efficient computing of concept lattices is a very important issue, has been investigated over the last decades [5].

This section gives only a brief overview of the basic notations of the theory for Formal Concept Analysis. For a more detailed description, it is referred to [1].

A K context is a triple K(G,M,I) where G and M are sets and I is a relation between G and M. The G is called the set of objects and M is the set of attributes. The cross table T of a context K(G,M,I) is the matrix form description of the relation I:

$$t_{i,j} = \begin{cases} 1 & \text{if } g_i I a_j \\ 0 & \text{otherwise} \end{cases}$$

where $g_i \in G$, $a_j \in M$.

Two Galois connection operators are defined. For every $A \subset G$:

$$f(A) = A' = \{a \in M | \forall g \in A \quad gIa\} \tag{1.1}$$

and for every $B \subset M$

$$g(B) = B' = \{g \in G | \forall a \in B \quad gIa\} \tag{1.2}$$

The Galois closure operator is defined by

$$h = f \circ g$$

and

$$A'' = h(A)$$

is the Galois closure of A. The pair $C(A, B)$ is a closed itemset of the K context if

$A \subseteq G$
$B \subseteq M$
$A' = B$
$B' = A$
$A = h(A)$

hold. In this case A is called the extent and B is the intent of the C closed itemset. It can be shown that for every $A_i \subseteq G$,

$$(\cup_i A_i)' = \cap_i A_i'$$

and similarly for every $B_i \subseteq M$,

$$(\cup_i B_i)' = \cap_i B_i'$$

holds.

Considering the ϕ set of all concepts for the K context, an ordering relation can be introduced for the set of closed itemsets in the following way:

$$C_1 \leq C_2$$

if

$$A_1 \subseteq A_2$$

where C_1 and C_2 are arbitrary closed itemsets. It can be proved that for every (C_1, C_2) pair of closed itemsets, the following rules are valid:

$$C_1 \wedge C_2 \in \phi$$

and

$$C_1 \vee C_2 \in \phi.$$

Based on these features $(\phi, \wedge)$ is a lattice, called closed itemset lattice. According to the Basic Theorem of closed itemset lattices, $(\phi, \wedge)$ is a complete lattice, i.e. the infimum and supremum exist for every set of closed itemsets. The following rules hold for every family $(A_i, B_i), i \in I$ of concepts:

$$\vee_{i \in I}(A_i, B_i) = (\cap_{i \in I} A_i, (\cup_{i \in I} B_i)'')$$

$$\wedge_{i \in I}(A_i, B_i) = ((\cup_{i \in I} A_i)'', \cap_{i \in I} B_i)$$

The structure of a Galois lattice is usually represented with a Hasse diagram. The Hasse diagram is a special directed graph. The nodes of the diagram are the closed itemsets and the edges correspond to the neighbourhood relationship among the itemsets. If C_1, C_2 are itemsets for which

$$C_1 < C_2$$

$$\neg \exists C_3 \in (\phi, \leq) \quad C_1 < C_3 < C_2$$

holds then there is a directed edge between C_1, C_2 in the Hasse diagram. In this case, the C_1 and C_2 concepts are called neighbour concepts.

There are several approaches in the literature to provide an efficiency lattice management. Each of the proposals provides a mechanism to reduce the number of attributes. These methods are usually based on some kind of statistical calculations. The method presented in [11] uses the principal component analysis to eliminate the redundant attributes from the documents. This method is based on the consideration that the occurrences of some attributes may be correlated. According to the principal component analysis, the original m

correlated random variables can be replaced by another set of n un-correlated variables where n is smaller than m. The resulting variables are the linear combination of the original variables. The principal components depend solely on the covariance matrix of the original variables.

Another kind of statistical computation is required if the reduction is based on the relevance values of the attributes. The relevance value of an attribute denotes how important the attribute is in the given object. This relevance value is calculated usually by the 'tf5idf' weighting method. This method defines the relevance value in proportion to the number of occurrences of the attribute in the document f_{ij}, and in inverse proportion to the number of documents in the collection for which the term occurs at least once (n_i):

$$rel_{ij} = f_{ij} \ \ log(\frac{N}{n_i})$$

The attributes having smaller relevance value than a threshold are eliminated from the object descriptor set. This kind of reduction method is used for example in [19].

Although the mentioned algorithms can reduce the number of attributes, providing better efficiency and interpretation, the resulted lattice can not be treated as the optimal one. According to our considerations, this solution may yield in some kind of information lost. This reasoning is based on two elements. First, the information lost is caused by the fact that the parent concepts will contain only some selected attributes of the children and the selected attributes are not always the best to describe the object. Second, during the attribute reduction phase, the meaning of the eliminated attributes will be lost, providing less information in the intersected concept.

To improve the quality and usability of the resulting lattice, a modified lattice and concept description form was developed which is described in the next section in details.

2. Concept Lattice with Attribute Lattice

It is assumed that there exists a lattice-like structure containing the attributes from the objects. This lattice-like structure can be considered as a thesaurus with the generalization relationship among the attributes. Taking the documents as objects and the words as attributes in our example, the attribute lattice shows the specialization and generalization among the different words. In special cases, the lattice may be a single hierarchy. It is also possible to

take several disjoint lattices as they can be merged into a new common lattice. Using this attribute lattice, the usual lattice-building operators are re-defined to generate a more compact and semantically more powerful concept lattice.

The proposed lattice construction algorithm is intended for information systems with a relative narrow problem area. In this case, an attribute lattice can be generated within an acceptable time and effort. It is assumed that the attribute lattice contains only those attributes that are relevant for the problem area in question. In this case, the size of the attribute lattice and the intent part of the concepts will be manageable. According to this assumption, the first phase of the document processing is the attribute filtering when the attributes not present in the attribute lattice are eliminated from the intent parts.

The M' elementset of the attribute lattice is a subset of the M attribute set. This lattice is denoted by the symbol $\Omega(M', \leq)$. The role of the lattice is to represent the generalization - specialization relationship among the attributes. The ordering relation of the attribute lattice is defined in the following way. For any m_1, m_2 attributes in M', m_1 is greater than m_2 ($m_1 \geq m_2$) if m_1 is a generalization of m_2. Based upon the relationship in $\Omega(M', \leq)$ a redefined partial ordering relation is introduced to M as an extension of the $\leq$ relation. This new relation is denoted by $\leq *$ and it is defined in the following way for any $m_1, m_2 \in M$:

$$m_1 \leq *m_2 \Leftrightarrow m_1 \text{ is an ancestor, a generalization of } m_2 \text{ in } \Omega(M', \leq)$$

Taking the words as attributes, for example, the word 'animal' is a generalization of the word 'dog', so 'animal' $\leq *$ 'dog' relation is met.

According to the lattice features, there exists a set of nearest common upper neighbors for any arbitrary pairs of attributes. This set is denoted by $LCA(m_1, m_2)$ for the attribute pair m_1, m_2.

$$LCA(m_1, m_2) = \{m \in M | m \leq *m_1 \wedge m \leq *m_2 \wedge \neg \exists m' \quad m' \leq *m_2 \wedge m' \leq *m_1 \wedge m \leq *m'\} \tag{2.1}$$

The LCA denotes the least common ancestor of two nodes in the lattice. The LCA set contains exactly the leaf elements of the common ancestor lattice for m_1 and m_2. Based on the partial ordering among the attributes, a similar ordering can be defined among the attribute sets. For any $B_1, B_2 \subseteq M$, the $\subseteq *$ ordering relation is given as follows:

$B_1 \subseteq *B_2 \Leftrightarrow \exists f \quad B_1 \to B_2$ function so that $x \leq *f(x)$ for every $x \in B_1$.

Having four sets of words B1(Paris, tennis, cup), B2(capital, sport), B3(capital, sport, car) and B4(sport) the $B_2 \subseteq *B_1$ relation is true as the $f \quad capital \to Paris, sport \to tennis$ function is a good injection. On the other hand, $B_3 \subseteq *B1$ relation is false, as the word 'car' can not be mapped to any word in B_1.

It is easy to see that the normal subset relation is a special case of the $\subseteq *$ relation, i.e.:

$$B_1 \subseteq B_2 \Rightarrow B_1 \subseteq *B_2$$

In this case the $f \quad x \to x$ mapping can be used to show the correctness of the $\subseteq *$ relation.

Based on this kind of subset relation, a new intersection operation can be defined. The definition of the new operator is:

$$B = B1 \cap *B2 = \cup \{LCA(m_1, m_2) | m_1 \in B_1, m_2 \in B_2\} \tag{2.2}$$

The intersection operator results in a set containing the nearest common generalizations of the attributes in the operand sets. If the parent node for every normal attribute of the intent sets is the null attribute (which is equivalent to the case when no attribute lattice is defined), the new $\cap *$ intersection operator will yield in the same result as the standard $\cap$ intersection operator. This is due to the fact that in this case

$$LCA(m_1, m_2) = \begin{cases} m & if m_1 = m_2 = m \\ \oslash & otherwise. \end{cases}$$

Using this kind of subset and intersection operators instead of the usual subset and intersection operators during the concept set and concept lattice building phases, the resulting lattice will be more compact, more readable and manageable than the standard concept lattice. This effect will be achieved by involving attributes into the concept description that would not be present if the standard lattice building method was used.

3. Algorithms for the LCA operation

The key operation in the proposed lattice management is the determination of the LCA set for any arbitrary pair of nodes. This operation is performed several times during the execution of the $\cap *$ intersection operations. As the

intersection is a frequent operation the efficiency of the LCA generation is a key element in the efficiency of the whole lattice management.

The computation of the LCA set can be performed basically on two different ways. In the first family of proposals, the common ancestor nodes are located by traversing the paths connecting the two operand-nodes. To reduce the number of candidate paths, the shortest path is determined first. The shortest path is usually calculated by using matrix multiplication. The second group of approaches for determining the LCA elements is based on the labeling concept. In the labeling approach, every vertex is assigned a description string. This label is used not only for identifying the nodes but to represent the ordering relationship among the nodes. In this case, the parents of an arbitrary node can be determined from the labels without the edge descriptions. Beside the problem of LCA generation, the labeling methods are used also to determine the distance between two nodes. This kind of labeling is called a distance labeling [14].

If the lattice is degenerated to a linear structure, the LCA contains only one node and this LCA node can be determined with a linear processing cost. Harel and Tarjan [15] showed first that the tree-LCA can be generated in a

$$O(n)$$

linear preprocessing time. In the last decades, some new proposals were published having the same $O(n)$ execution cost but using a much more simpler algorithm. One of most recent ones among these proposals is the paper of Bender, Colton and Pemmasani[15]. They present an extremely simple, optimal tree-LCA algorithm. It is shown that the tree LCA problem is equivalent to the RMP, the range minimum problem. In the paper, an $O(n)$ RMP algorithm is presented first and then it is converted to a tree-LCA algorithm with $O(n)$ efficiency. An intensive investigated topic in this area is the identifying the nearest common ancestors in dynamic trees. In [16], Alstup presents a pointer machine algorithm which performs n link and m LCA operations in time

$$O(n + m \ \ loglogn).$$

The main problem of the algorithms based on path traversing is that the relationship among the nodes of the lattice must be stored explicitly. These relationships are usually described by matrixes or by pointers. In this case, the tree is stored by representing explicitly all vertices and all edges. To save the storage space and to improve the execution efficiency, labeling methods

are implemented.

In [17] a simple algorithm is presented that labels the nodes of a rooted tree such that from the labels of two nodes alone one can compute in constant time the label of their nearest common ancestor. The labels assigned to the nodes are of size

$$O(logn)$$

and the labeling algorithm runs in

$$O(n)$$

time. A similar result for this problem can be found among others in [18].

In the case of a lattice or DAG (directed acyclic graph), the LCA problem requires much more computation. In a lattice structure, two nodes may have several LCA nodes. Although the DAG is a widely used structure, the DAG-LCA generation is not so widely investigated as the tree-LCA problem. Based on the work of Bender, Colton and Pemmasani[15], the main results can be summarized as follows. For testing the existence of common ancestors, an ancestor existence matrix is built. Two nodes x and y in lattice G have a common ancestor if and only if (x', y) is in the transitive closure set of the G'' lattice. The G'' lattice is generated by merging the sinks of G' with the sources of G. The G' is the inverse lattice of G, i.e. it contains the same number of nodes and edges but every edge has the inverse direction. The ancestor existence matrix can be computed in

$$O(n \;\; w)$$

time, where w is about 2.376 [15] and $O(nw)$ is equal to the efficiency value of the fastest matrix multiplication algorithm. The transitive closure of a lattice can be generated within the $O(nw)$ efficiency class, too. The computation of the LCA set is based on the consideration that the shortest path in the G'' DAG from node x' to node y goes through the LCA of the corresponding nodes. The generation of LCA for a pair of nodes can be calculated in

$$O(\frac{n \;\; w}{2} - 0.5)$$

time.

There exist some proposals for finding the LCA nodes in graph by labeling method, too. A k-step labeling method is presented in the paper [16] of Talamo and Vocca. A k-step labeling consists of $f_1, .., f_k$ functions where every f_i is

a partial function computable in one step and a composition between f_i and f_{i-1} can be defined. The k-step labeling is a valid labeling if and only if

$$y \in adj(x) \Leftrightarrow (f_k \circ f_{k-1} \circ \quad \circ f_1(x,y)) = y \vee (f_k \circ f_{k-1} \circ \quad \circ f_1(y,x)) = x$$

is met. The paper presents a method for generating the labels where a vertex x has a

$$O(\delta(x) \cdot log^2 n)$$

bit long label and the labels can be computed in

$$O(\delta \cdot n^2)$$

time, where δ denotes the degree of the vertex. The degree of a vertex is equal to the number of adjacent nodes. A modified version of this adjacent labeling can be found among others in [16]. An $L(d,1)$ labeling is a function f that assigns to each vertex a non-negative integer such that if two vertices x and y are adjacent then $|f(x) - f(y)| > d$, and if x and y are not adjacent but there is a two-edge path between them then $|f(x) - f(y)| > 1$.

Considering our requirements, it can be seen, neither of the proposals meets all of the requirements. The main problems in application of the presented methods are the followings:

there is no detailed study on DAG-LCA problem
the path traversing method for DAG [15] retrieves only one element from the LCA
the labeling methods can be used only to determine the adjacent nodes and not all the descendant nodes

Based on these restrictions, it seems reasonable to develop a special DAG-LCA generation algorithm for the lattice structure.

4. A modified DAG-LCA algorithm

The computation of the LCA set can be performed basically on two different ways. In the first family of proposals, the common ancestor nodes are located by traversing the paths connecting the two operand-nodes. To reduce the number of candidate paths, the shortest path is determined first. The shortest path is usually calculated by using matrix multiplication. The second group of approaches for determining the LCA elements is based on the labeling concept. In the labeling approach, every vertex is assigned a description string.

In the case of our search algorithm for finding the LCA nodes, a merging of the path- oriented methods with the labeling methods is implemented. The main idea is to assign a description set to every node in the lattice where this description set has a similar role as the attribute set has in the normal concept lattices. As it is known, there is a strong correlation between the position in the lattice and the content of the intent part. For any pairs of concepts, the concepts are in relation if and only if one of the intent parts is a subset of the other intent part:

$$C_1 \leq C_2 \Leftrightarrow A_2 \subseteq A_1$$

Based on this rule it can be seen that

$$A(LCA(m_1, m_2)) \subseteq A(m_1) \cap A(m_2)$$

also holds where $A(m)$ denotes the intent part of m. In this sense, the search for the LCA nodes may be restricted to the nodes where the intent part is a subset of the intersection of the corresponding A_i and A_j sets. This reduction may increase the efficiency of finding the LCA elements. As a node in the attribute lattice usually does not contain an intent part description, it is not possible to apply this kind of reduction element in the usual lattice building. To include this optimization feature an appropriate intent part should be added to every node of the attribute lattice.

Let B denote the set of binary lists having the same length and containing 0 and 1 elements. If there exists a

$$a \quad M \rightarrow B$$

function which meets the following requirements:

$$a(m_1) = a(m_2) \Leftrightarrow m_1 = m_2$$

$$a(m_1) \subseteq a(m_2) \Leftrightarrow m_1 \geq m_2$$

then

$$a(LCA(m_1, m_2)) \subseteq a(m_1) \cap a(m_2) \tag{4.1}$$

holds also. In these expressions the $\subseteq$ symbol denotes the sub-list operator and the $\cap$ operator is the list intersection. The list-intersection is defined for any l_1, l_2 lists as follows

$$(l_1 \cap l_2)_j = l_{1j} \wedge l_{2j}$$

where the length of the result list is equal to the minimum of the operands lengths. Thus, for example the intersection of 101100 and 111000 is equal to 101000.

This statement can be easily verified as according to (2.1)

$$LCA(m_1, m_2) \geq m_1 \; and \; LCA(m_1, m_2) \geq m_2$$

thus

$$a(LCA(m_1, m_2)) \subseteq a(m_1) \; and \; a(LCA(m_1, m_2)) \subseteq a(m_2)$$

and so

$$a(LCA(m_1, m_2)) \subseteq a(m_1) \cap a(m_2)$$

holds.

To provide an appropriate $a()$ function, the following algorithm is used to calculate the $a(m)$ values. First, the nodes in the lattice are sorted by the depth value. The nodes with low depth value are processed first. Thus before processing of node m, every ancestor of m has been processed already. The root node of the lattice is assigned to an empty list. This root element is the only node with a zero depth value. If all the nodes with depth value less than K are already processed, then the $a()$ values for nodes at depth level $K+1$ are calculated according to the following algorithm.

1. For every m at level $K+1$:
   $a(m) = \cup_{m' \in Parent(m)} a(m')$
2. Nodes having the same $a(m)$ value are extended with tail tags to ensure the uniqueness of the $a(m)$ values.
3. Testing every node at the processed levels. If node m' is not an ancestor of m and $a(m') \subseteq a(m)$ then $a(m')$ is extended with tail tag.
4. The descendants of m' are adjusted to the new m' value.

Lemma. The $a()$ function generated by the given algorithm meets the (4.1) conditions.

Proof. According to the step 2 in the algorithm, every node will have a unique value. In the adjustment phase every processed node will be modified with an unique tag, so the uniqueness of the $a()$ values is ensured in this phase too. According to this considerations, the

$$a(m_1) = a(m_2) \Leftrightarrow m_1 = m_2 \tag{4.2}$$

condition holds.

If the m is a child of m' then $a(m') \subseteq a(m)$. This comes from the fact that the $a(m)$ is generated as the union of all its parents. If $m' \geq m$ then exists a list of parent-child relationship from m to m' Using the transitive property of the relations, it follows that

$$a(m_1) \subseteq a(m_2) \Leftarrow m_1 \geq m_2 \tag{4.3}$$

is met. On the other hand if m' is not greater or equal to m then m' is not an ancestor of m, then the $a()$ value of m' is modified by adding new tags to the list value. After this modification, the $a(m')$ will not be a subset of $a(m)$. Thus

$$\neg a(m_1) \subseteq a(m_2) \Leftarrow \neg m_1 \geq m_2 \tag{4.4}$$

According to the (4.2), (4.3), (4.4) formulas, the (4.1) property is met.

Considering the proposed labeling algorithm, the generated labels are usually not optimal from the viewpoint of the label length. In the tests, the labels were generated for the normal concept nodes having a natural attribute string. Depending on the number of nodes and on the depth of the lattice, the generated labels can be several times longer than the original attribute labels. In the test runs, the proposed labeling algorithm provided always the same lattice relationship among the nodes as the original attribute strings. The length optimality of the generated labels is a topic for further investigations.

After generating the labels, the next step is the identification of the LCA set. In the basic path oriented methods the LCA algorithm consists of the following steps:

1. Generating the A_x ancestor set for x. The ancestors are selected by traversing along the parent-child edges.
2. Generating the A_y ancestor set for y.
3. Calculating the A_{xy} intersection of the two ancestor sets.
4. Selection of vertices in A_{xy} having no descendants in A_{xy}

The cost for the LCA algorithm can be given by

$$O(A_x \quad f \cdot e + A_y \quad f \cdot e + A_{xy} \cdot w)$$

where

f average degree of the vertices, i.e. the average number of parents

e cost for selection of an edge related to a given node. The cost may vary depending on the storage method.

A_x size of the corresponding ancestor set

On the other hand, in the proposed algorithm the generation of the LCA for (x, y) is performed in the following steps.

1. Processing the parents of x in a recursive way
2. If the current element is an ancestor of y, insert the current element into list L and stop the ancestor lookup

3. Selecting elements of L having no descendants in L

In step 2, the ancestor relationship is tested by comparison of the label values. According to (4.3), if

$$a(m_1) \subseteq a(m_2)$$

then m_1 is an ancestor of m_2. If the traversing reaches a y-ancestor, the lookup can be stopped as the ancestors of this node can not be LCA nodes.

The main benefit of this algorithm is the reduced number of nodes to be processed. The cost can be given by

$$O(A'_x \quad f \quad (e+h) + \eta A'^2_{xy})$$

where

A'_x the number of vertices being the ancestor of x but not being an ancestor of y.
η the cost for comparing two labels
A'_{xy} the number of selected border nodes in A_{xy}.

Comparing the two cost expressions, we can see that the combined method is more efficient than the basic method if

1. A'_x is smaller than A_x and A_y
2. η and e have the same magnitude
3. A'_{xy} is smaller than A_{xy}

Based on these considerations, this bottom up traversing is advantageous if the LCA elements are located near to the x and y nodes. On the other hand, if the LCA elements are near to the root of the lattice, a top-down approach provides a better solution. In this case, the algorithm is the following:

1. Selecting the root of the lattice
2. Testing the children of the current node
3. If the label is a subset of the intersection label
4. Selection of vertices in A_{xy} having no descendants in A_{xy}

This algorithm determines the parents for the intersection of x and y. The label of the intersection node is equal to the intersection of the corresponding labels. The cost value can be given by

$$O(A_{xy} \quad f \quad (e+h))$$

where f denotes the average number of children vertices.

An efficient implementation can involve all of the mentioned algorithms. The LCA generation program should include a decision module that is responsible for selecting an appropriate algorithm. As the number of 1 digits in the label is correlated with the level of the node, an approximation of the LCA levels can be given based on the value of the intersection label. The heuristic rule can be summarized as follows: If the number of 1 digits is low in the intersection label, then a top-down traversing method is used, otherwise a bottom-up traversing is applied.

REFERENCES

[1] S. ABITEBOUL AND H. KAPLAN AND T. MILO: Compact labeling schemes for ancestor queries, *Technical report*, INRIA, 2001

[2] S. ALSTUP AND T. RAUHE: Improved labeling scheme for ancestor queries, *Technical report*, University of Copenhagen, 2001

[3] S. ALSTUP AND C. GAVIOLLE AND H. KAPLAN AND T. RAUHE: Identifying Nearest Common Ancestors in Distributed Enviroment, *Technical report*, University of Copenhagen, 2001

[4] M. BENDER AND M. COLTON AND G. PEMMASANI: Least Common Ancestors in Trees and Directed Acyclic Graphs, *Symposium on Discrete Algorithms*, 2001, pp. 845-854

[5] G. CHANG AND W KE AND D. KUO AND D. LIU AND R. YEH: On L(d,1)-labelings of graphs, *Discrete Mathematics*, Volume 220, Issues 1-3, 6 June 2000, pp. 57-66

[6] B. GANTER AND R. WILLE: *Formal Concept Analysis, Mathematical Foundations*, Springer Verlag, 1999

[7] R. GODIN AND R. MISSAOUI AND H. ALAOUI: Incremental concept formation algorithms based on Galois lattices, *Computational Intelligence*, 11(2), 1995, 246-267

[8] K. HU AND Y LU AND C SHI: Incremental Discovering Association Rules: A Concept Lattice Approach, *Proceedings of PAKDD99, Beijing*, 1999, 109-113

[9] M. KATZ AND N KATZ AND D. PELEG: Distance labeling schemes for well-separated graph classes, *STACS 2000, Lecture Notes In Computer Science*, Springer Verlag, 2000

[10] L. KOVACS: Efficiency Analysis of Building Concept Lattice, *Proceedings of 2nd ISHR on Computational Intelligence*, Budapest, 2001

[11] L. KOVACS: A Fast Algorithm for Building Concept Set, *Proceedings of Micro-CAD2002*, Miskolc, Hungary 2002

[12] C. LINDIG: Fast Concept Analysis, *Proceedings of the 8th ICCS*, Darmstadt, 2000

[13] L. NOURINE AND O. RAYNAUD: A Fast Algorithm for Building Lattices, *Information Processing Letters*, 71, 1999, 197-210

[14] S. RADELECZKI AND T. TÓTH: Fogalomhálók alkalmazása a csoporttechnológiában, *OTKA kutatási jelentés*, Miskolc, Hungary, 2001.

[15] J. SILVA AND J. MEXIA AND A. COELHO AND G. LOPEZ: Document Clustering and Cluster Topic Extraction in Multilingual Corpora, *Proc. of the 2001 IEEE Int. Conference on Data Mining*, IEEE Computer Society, pp. 513-520

[16] G. STUMME AND R. TAOUIL AND Y BASTIDE AND N. PASQUIER AND L. LAKHAL: Fast Computation of Concept Lattices Using Data Mining Techniques, *Proc. of 7th International Workshop on Knowlegde Representation meets Databases (KRDB 2000)*, Berlin, 2000

[17] M. TALAMO AND P VOCCA: Representing graphs implicitly using almost optimal space, *Discrete Applied Mathematics*, Elsevier Publ., 2001, pp. 193-210

[18] D. TIKK AND J. YANG AND S. BANG Text categorization using fuzzy relational thesauri, *Technical report*, Chonbuk National University, Chonju, Korea, 2001

[19] M. ZAKI AND M. OGIHARA: Theoretical Foundations of Association Rules, *Proceedings of 3 rd SIGMOD'98 Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'98)*, Seattle, Washington, USA, June 1998.

[20] U. ZWICK: All Pairs Shortest Path in weighted directed graphs- exact and almost exact algorithms, *IEEE Symposium on Foundation of Computer Science*, 1998, pp. 310-319

Production Systems and Information Engineering
Volume 4 (2006), pp. 81–93

A MATLAB GRAPHICAL TOOL TO SUPPORT KNOWLEDGE ENGINEERING

GÁBOR VÁMOS
Budapest University of Technology and Economics, Hungary
Department of Control Engineering and Information Technology
vamos@iit.bme.hu

[Received May 2005 and Accepted May 2006]

Abstract. Bayesian networks give an efficient probabilistic model representation scheme to encode both expert knowledge and information extracted from database. Although the probabilistic model representation and the storage space problem can be solved using Bayesian networks, the model evaluation (inference) operation become complicated. The practical solution of this problem is an optimum between the accuracy of model and the complexity of the model evaluation task. This trade-off is more easily found if the knowledge engineers could receive an immediate feedback during the developing phase of the network about the computational resources required to evaluate the probabilistic model. This paper presents a Bayesian network modeling software tool focusing on its graphical user interfaces and data structures.

Keywords: Bayesian Networks, Knowledge Engineering

1. Introduction

Probabilistic models [8, 6, 7] of real world phenomena are increasingly used in commercial applications, especially in decision support systems and in different type of fault or fraud detection tasks. Such applications have two essential, usually independently considered components: the knowledge base (abstract description of real world phenomena, i.e. a model) and the model evaluation engine.

In the case of an uncertain problem domain, the model is often given as a joint probability distribution function (PDF) over a finite set of variables. For discrete variables, the joint probability function may be stored in a table. This may result simple inference engines at the price of huge required storing capacity which grows exponentially with the number of variables and their

possible values. In fact, the required storage capacity impedes the direct use of joint probability tables in real applications.

Bayesian networks offer an efficient representation structure using conditional probability tables obtained from the factorization of the joint probability function. This factorization is implied by a set of independence relations which is represented by a directed acyclic graph (DAG). This graphical structure helps to integrate two vital knowledge sources when creating the probabilistic model: expert knowledge and information extracted from databases. However, the evaluation of these models is more complex. From a practical point of view, the main problem is to find an optimum between the model accuracy and the complexity of its evaluation. This trade-off should be more easily found if the knowledge engineers could receive an immediate feedback about the computational resources required to evaluate his or her actual probabilistic model.

Our research has a twofold objective. First, we wish to revisit some existing algorithms used to evaluate Bayesian networks, and second, to develop a user friendly environment to design such networks providing information about the computational complexity of the model during its creation. This paper focuses on our second objective and presents a prototype software tool designed using MATLAB.

The remaining part of the paper is organized as follows. The next section (2) gives an overview on the mathematical background of Bayesian networks and on the related probability inference problem. Section 3 summarizes the model creation methodology, the steps of model creation are described in Section 4. Section deals with the implemented Bayesian modeling software tool. The first part of the section is devoted to the main role of the user interface. This is followed by a short overview about the main concepts of system architecture. Section 6 ends with the description of the implemented functions. Section 7 specifies the data structures used to maintain the DAG property during the editing process. Conclusions and some remarks about the future works close the paper.

2. Bayesian networks

The objective of the modeling is to find an abstract representation of the information coded in the database and that of the human knowledge accumulated as experience.

In probabilistic models all variables are aleatory variables and each record in the database is a realization of these variables. Then the model is given by the joint probability distribution function (PDF) over all variables as

$$P \quad V_1 \times V_2 \times \quad \times V_n \to [0,1] \tag{2.1}$$

Bayesian networks [7] are widely used to represent efficiently a joint PDF, so it become one of the most popular uncertainty knowledge representations and reasoning technique in AI. The structure is able to code and to integrate expert knowledge and formerly measured values of problem domain variables (stored in databases).

Definition 1 (Bayesian networks). *A Bayesian network over a set of variables* $\mathbf{U} = \{V_1, \ldots, V_n\}$ *consists of a graphical and a quantifying component:*

1. *Graphical component. Directed acyclic graph:* $\mathcal{G}$. *Each node in the graph represents a variable in* $\mathbf{U}$. *The set of parents of a variable* V *is denoted by* Π_V.
2. *Quantifying component. Each variable* V *in* $\mathbf{U}$ *(i.e. each node in* $\mathcal{G}$*) is quantified with a conditional probability distribution (CPT) denoted by* $P(V|\Pi_V)$.

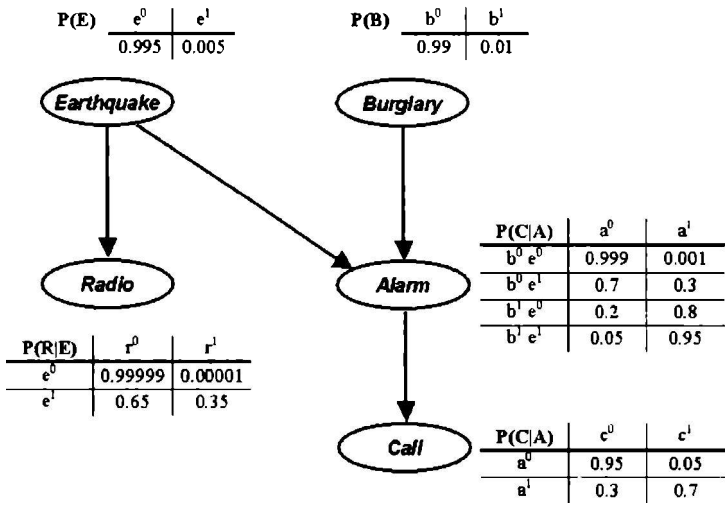


Figure 1. The classical [7] Alarm Bayesian network with CPTs

These random variables can be either discrete or continuous depending on the nature of the problem domain. We will consider discrete variables in the sequel. The Bayesian network corresponds to a joint PDF over $\mathbf{U}$:

$$P(\mathbf{U}) = \prod_{i=1}^{n} P_i(V_i|\Pi_{V_i}). \tag{2.2}$$

The query of a variable V_q in a Bayesian network based on a set of evidences $\mathbf{E} \subset \mathbf{U}$ (variables with known values or distributions) corresponds to an inference procedure. The inference in a Bayesian network consists of fixing the values of a subset of the variables and to marginalize (2.2) w.r.t. V_q:

$$P_{V_q}(V_q) = \sum_{\mathbf{U}\setminus\{V_q,\mathbf{E}\}} P(\mathbf{U}). \tag{2.3}$$

This results a probability distribution over V_q which combines the knowledge encoded in the network (i.e. in $P(\mathbf{U})$) and in the collected evidence. The direct marginalization according to (2.3) is a computationally expensive operation because it needs numerous evaluations of the joint PDF coded by the network. Therefore the marginalization takes place usually in a different secondary structure called tree of clusters using the so called Probability Propagation in Tree of Clusters (PPTC) algorithm [4]. Roughly speaking, cluster trees are undirected, acyclic graphs whose nodes are clusters containing sets of the nodes from the original network. The rules of transformations leading from the original Bayesian network to a cluster tree ensure that both entities correspond to the same joint PDF

There are several exact and approximate inference algorithms, whose computational complexity is NP-hard [1, 2]. The most popular exact inference algorithm is the so called clique-tree propagation algorithm [4]. Our research places the emphasis on this approach [3].

3. A probabilistic modeling methodology using Bayesian networks

The applied model creation methodology is meshing to the degree of observability of problem domain [5]. We pressume the full observability over many human experts and large datebases. According to our approach the model creation process consists of several overlapping steps, some of them being repeated in an iterative way. This section summarizes the main working phases of this process.

Since Bayesian networks yield an effective way to represent factorizable joint PDFs, our methodology is especially customized to this approach. The following human collaborators intervene during the model creation process [9]:

- Human Expert: he or she holds possession of practical and/or theoretical knowledge about the problem domain.
- Statistician: skilled in the classical statistical methods of data analysis.
- Knowledge Engineer: proficient in the creation of knowledge based probabilistic model.

The model creation process consists of the following steps.

- Surveying knowledge sources: discovery of the appropriate knowledge sources. The probabilistic model is generally a theoretical (mathematical) abstraction, which depicts efficiently the behavior of the circumscribed real world domain. In our approach, the model integrates the knowledge hiding in measured data patterns (stored in databases) and the knowledge of Human Expert (collected using questionnaires and interviews). This phase is undertaken by the Knowledge Engineer in cooperation with the Human Expert.
- Raw data preprocessing and determining the relevant probabilistic variables. This phase processes the raw data gathered about the problem domain. In the possession of reports made during the preceding phase with the Human Expert, the Knowledge Engineer determines a potential collection of probabilistic variables (including future evidence and query variables) in line with the Statistician, who analyzes the quality and relevance of raw databases w.r.t. the set of probabilistic variables.
- Stipulating the structure of probabilistic network. The Statistician and the Knowledge Engineer divide the probabilistic network into partitions. Some of them will be trained using data from database, the rest will be appraised by the Human Experts.
- Putting the pieces together. The Knowledge Engineer assembles the trained and appraised parts of model using again the Human Expert's knowledge. This results the first raw but complete probabilistic model.
- Finalizing the qualitative and quantitative composition. The Knowledge Engineer verifies the Bayesian network using test data.

Recall that at each phase one may need to step back and reiterate previous phases if the actual results are not satisfying.

4. Model creation steps

The probabilistic model representation and the storage space problem of a joint PDF can be solved using Bayesian networks, but the model evaluation (inference) operation become complicated.

Recall that the first objective aims to obtain quantitative information to create the graphical component and the second one helps to determine qualitatively the conditional PDFs associated to the nodes of the Bayesian network. This modeling procedure results an iteration of steps as illustrated in Figure 2.

The transformation between the cluster tree and the Bayesian network (already mentioned in Section 2) is not a one-to-one relation, since several cluster tree may correspond to the same Bayesian network. A cluster tree is said to be

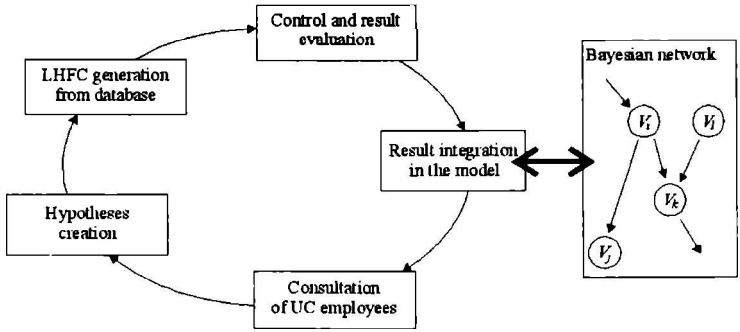


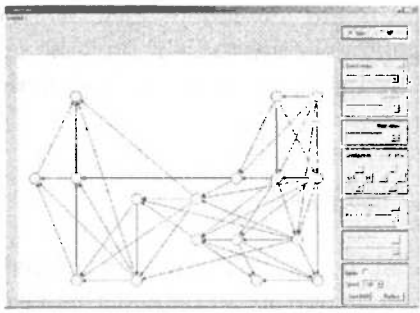
Figure 2. Iterative modeling with Bayesian network

optimal if it is of minimal width. Roughly speaking, the width of the cluster tree is directly related to the computational cost of a query evaluation. The search for the optimal cluster tree is a complex task impeding the possibility to give the Knowledge Expert an exact complexity measure of the designed network at each step of the above methodology.

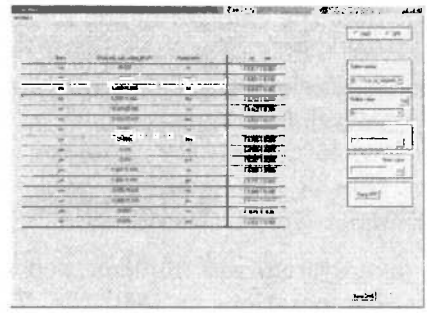
5. The modeling software tool

It is clear from the methodology described in the previous section that the Knowledge Engineer has the occasion to trade-off between the accuracy of the probabilistic model and the complexity of the inference task during the creation of each direct connection between a pair of nodes. There are a lot of commercial Bayesian software tools integrated with ergonomic graphical interfaces to support this graph manipulation process. From the well distributed software packages we emphasize Netica, GeNIe, Hugin. [10, 11, 12]. Some of them have high level abstraction interfaces, which can be accessed for development using up-to-date programming languages (C++, Java etc.).

However, these environments are too closed for the optimization at low level of PPTC, so we have constructed a new framework using MATLAB, including a simple graphical but a more complex developer interface. In this environment several optimization approaches (testing different triangulation and complexity information feedback heuristics) become realizable and examinable. In this section our framework is presented that supports the Knowledge Engineer to find the compromise between accuracy and computational complexity.



(a) Surface to edit the DAG component



(b) Surface to edit the CPT component

Figure 3. Graphical user interfaces

5.1. GUI functionality and structure

The GUI has two distinct windows for supporting the modifications of the DAG (directed acyclic graph) structure and the numerical data of CPT (conditional probability distribution). Figure 3 shows screenshots of both windows.

There are two frames in each window, the right ones (larger) visualize the DAG or the CPT, the left ones (smaller) displays the modification commands.

In the DAG manipulation view (Figure 3) some operations are just modifying the vista of the graph (scroll the graph or a node), the others are changing the structure of the DAG (add/remove edges). The main commands for the DAG view are the following:

- Scrolling figure: the displayed graph is movable in the window.
- Adding or removing a node.
- Manipulating a node. After the selection of a node (the current node is marked using different color) the most important operations become executable:
  - modification of the position of the current node in the picture,
  - changing the name of the current node,
  - assigning new parent to the current node (add a new edge), removing a parent from the current node (delete an edge).

It is important to pay more attention to operation where a new edge is added to the graph. We present this problem in detail in Section 6.

The role of the left frame in CPT manipulation view (Figure 3) is the same as in DAG view, but the bigger right pane becomes two sided, because the tasks of displaying and manipulating probability values are executed in the same pane.

Recall that the changes in the DAG structure are not independent from the context of CPTs. For example if one edge is deleted, the CPT of the child node becomes simpler. This software tool is capable to handle such situations using some default strategies.

6. Document/View architecture

The pilot system was implemented in Matlab 6.1, which is well optimized to carry out operations on matrices. The fundamental ideas behind this visualization interface follows the well known and the widely used Document/View architecture. There are distinctly constructed matrices for the calculation-storing process and the visualization proposition. These differences are demonstrated with the two variants of data representation form of Bayesian Networks.

Tables 1 and 2 enumerate the main components of data structure in which Bayesian networks are stored and used during the evaluation (inference) process.

Table 1. BNet data structure for storing Bayesian networks.

Name	Type	Description of use
adjMatrix	sparse	adjacency matrice
nodeSize	int. vector	each nodes values
nodes	structure	representing the nodes
edgeConstraints	sparse	training strategy
timeConstraints	struct	training strategy

Table 2. Node of BNet data structure for storing Bayesian networks

Name	Type	Description of use
varName	string	current node name
varStateNames	string vector	name of the values
selfVar	double	identification the nodes
selfVarSize	integer	values of current node
CPD	matrice	conditional probability table

The three types of data are the following:

- Basic data: These components correspond to the mathematical definition of Bayesian networks.

- Sub data: It makes executable and balance able such procedures as training the network using database
- Redundant data: Some pieces of information in the Basic data are replicated in other structures in order to make efficient some operations during the inference algorithms. For example, the number of values of nodes (i.e. variables) are stored implicitly and explicitly in each node structure and this data also appears in a collection of BNet nodeSize (which is in turn useful in the triangulation process).

Tables 3 and 4 contain the graph visualization structures.

Table 3. "drawingnode" matrix for the visualization

Name	Type	Description of use
index	integer	numerical identification
name	strings	verbal identification
posx	double	horizontal location
posy	double	vertical location
markerhandler	double	accessing the graphical object
texthandler	double	accessing the graphical object

Table 4. "drawingedge" matrix for the visualization

Name	Type	Description of use
parIndex	integer	start point identification
chIndex	integer	end point identification
parposx	double	horizontal location of parent node
parposy	double	vertical location of parent node
chposx	double	horizontal location of child node
chposy	double	vertical location of child node
markerhandler	double	to access the graphical object
arrowhandler	double	to access the graphical object

The main data types in these tables are the following:

- Basic data: the most important data from the Bnet basic data, like the node and adjacency information.
- Localization data: horizontal and vertical positions of graphical components.
- Data to the graphical objects: these handlers ensure the modifications of graphical components accessing to a complex data structure.

The basic data of matrix drawingedge and drawingnode render a permeable way between the document and the view structure. Besides to the identical content, the adjacency matrix is mapped to the column of parIndex and chIndex.

7. Incremental graph expansion

Referring to Subsection 5.1, the problem of graph structure modification will be demonstrated focusing on the graph extension. There are two ways to modify the structure of a connected graph: adding or removing the directed connection between nodes. The edge removing operation is simpler: The Knowledge Engineer selects the current node, the GUI loads the name of the parents of the current node into the Remove parent menu. The user selects one parent node (the corresponding edge become marked) from this menu, and then using the Ok button the operation is executed. After then the corresponding field of adjacency matrix become zero and the CPT of current node will be changed.

In contrast of the edge removing operation, the edge adding function more complex, because during the operation one needs to guarantee that the extended graph is still acyclic. The adding edge procedure is not too complex for the first sight, the operation is decomposed to the following steps. First a start node has to be assigned. Then the GUI loads to the Add child menu the names of potential children to the current node. The user selects one node from this list. After the approvement of actual operation, the adjacency matrix and the CPT of child node will be modified.

The key momentum is the selection of potential children to the current node. In this step the GUI selects the nodes which may hurt the DAG property if considered as children. To support this operation, a special structure is maintained to represent the reachability of each node from the other nodes. This information is stored in the so called `reachable_from` matrix. This matrix has two columns, the first contains the identifier of a node, the second contains the list of nodes, from which the identified node is reachable along directed edges. We define the size of this matrix as the total number of all elements in the list of the second column.

The algorithm operates according to an elimination scheme. Every node without children node is processed in the `while` sequence, which is executed n times. The currently processed node is selected first. Then the reachability list of the selected node is actualized with the parent nodes using the union operation. Going further, one has to check the lists of the other nodes. If the

selected node is contained in some list of other node, this list must be actualized with the list of selected node (taking the union of the two list). Then the current node is eliminated.

Figure 4 shows a chain graph with 6 nodes. The `reachable_from` matrices for this graph reads

1.
2. 1
3. 1, 2
4. 1, 2, 3
5. 1, 2, 3, 4
6. 1, 2, 3, 4, 5

This example represents the worst case size of the `reachable_from` matrix. For n nodes, it is not greater then: $\frac{n \cdot (n-1)}{2}$ The proof of this statement uses the following three lemmas. Recall that the $\mathcal{G}$ graph is ordered, if there is a bijection α such that: $\alpha \quad \mathbf{V} \rightarrow (1, 2, \ldots \ n)$

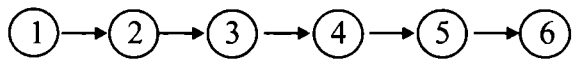


Figure 4. Chain DAG with 6 nodes

Lemma 1. *There is an ordering of nodes for every $\mathcal{G}(\mathbf{V}, \mathbf{E})$ DAG, such that every edge directs from a lower numbered node to a higher numbered node ($i < j$ for every pair of $(i, j) \in \mathbf{E}$).*

Proof. The proof consists of giving the algorithm resulting the ordering.

Input: $\mathcal{G}(\mathbf{V}, \mathbf{E})$ DAG.

Output: node ordering.

Temporal variables:

- $\mathcal{G}'$: graph is the actual states of elimination sequence
- $\mathbf{R}$: contains the set of not eliminated nodes whose have not descendant node, but not parent node in $\mathcal{G}'$
- V: the actually processed (eliminated node)
- T: temporal node
- $\mathbf{P}$: set of nodes, which appears as the first element of $(V_i, V_j) \in E$ edge pairs
- $\mathbf{C}$: set of nodes, which appears as the second element of $(V_i, V_j) \in E$ edge pairs

Initialize: creation of $\mathbf{C}$ and $\mathbf{P}$, $\mathbf{R} = \mathbf{C} \backslash \mathbf{P}$, $\mathcal{G}' = \mathcal{G}$, $n = |\mathbf{V}|$, $i = 1$

```
while i ≠ n
    V ∈ R
    α(V) = i //numbering the
             //selected node
    i = i + 1
    P = P\V //eliminating the
             //selected node
    C = C\V //eliminating the
             //selected node
    R = C\P //G' = G'(V\V ,E\(V,.))
endwhile
```

Lemma 2. *A given graph $\mathcal{G}$ is acyclic if there is an ordering α such that for every edge (V_i, V_j) we have $\alpha(V_i) < \alpha(V_j)$.*

Proof. The lemma is shown by indirection. Consider a graph $\mathcal{G}$, which have an ordering and a hypothesized cycle. Let $g \in V$ be the node with the lowest ordering of the assumed cycle. But, by the construction of the ordering, it is impossible to direct an edge into g from any other node of cycle, hence g cannot be a node of the hypothetical cycle. Therefore the Lemma follows.

Lemma 3. *Let $\mathcal{G}(\mathbf{V}, \mathbf{E})$ be a DAG. The $\mathcal{G}$ contains a maximal number of edges, if $\mathbf{E}$ contains every branches (V_i, V_j) such that $\alpha(V_i) < \alpha(V_j)$, i 1 . n, $j = 1 \ldots n$ for a given ordering α of $\mathbf{V}$*

Proof. From Lemmas 1 and 2, it follows that such a graph exists. It follows also from the construction that there is an edge between any two nodes. Hence any new edge one can put in the graph is such that the ordering of its starting node is grater that the ordering of its ending node. But such an edge creates a cycle with the already existing edge between the same nodes, hence the graph is maximal.

Proposition 1. *The maximal size of* `reachable_from` *matrices is $n(n-1)/2$.*

Proof. By construction, the size of the maximal DAG as defined in Lemma 3 is $n(n-1)/2$. Since the set of edges of all DAGs is a subset of the edges of the maximal DAG (choosing the right ordering) the proposition follows.

This proposition gives in fact the worst case for the number of operations needed to check weather the introduction of a new node hurts the DAG property.

8. Conclusion

In this paper Bayesian network based modeling techniques, and a network editor tool and its GUI have been presented. The Matlab environment was suitable for implementing incremental model generation methods. The open environment makes it possible to extend and improve the application; this way several optimization approaches (testing different triangulation and complexity information feedback heuristics) become realizable and examinable.

REFERENCES

[1] G.F. Cooper, *The computational complexity of probabilistic inference using bayesian belief networks*, Artificial Intelligence (1990), no. 42, 393–405.

[2] P. Dagum and M. Luby, *Approximating probabilistic inference in bayesian belief networks is np-hard*, Artificial Intelligence **1** (1993), no. 60, 141–153.

[3] C. Huang and A. Darwiche, *Inference in belief networks: a procedural guide*, Intl. J. Approximate Reasoning **3** (1996), no. 15, 225–263.

[4] S.L. Lauritzen and D.J. Spiegelhalter, *Local computations with probabilities on graphical structures and their applications to expert systems*, Proc. of the Royal Statistical Society (1988), no. 50, 154–227.

[5] K. Murphy, *A Brief Introduction to Graphical Models and Bayesian Networks*, Department of Computer Science at U. C. Berkeley, 2001

[6] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*, Morgan Kaufmann, San Mateo, Calif., 1988.

[7] P. Norvig and S.J. Russel, *Mesterses intelligencia modern megkelben (artificial intelligence. a modern approach.)*, Panem-Prentice Hall, Budapest, 2000.

[8] D.E. Heckerman and M. Henrion E.J. Horowitz H.P. Lehmann G.F. Cooper M.A. Shwe, B. Middletown, *Probabilistic diagnosis using a reformulation of the internist-1/qmr knowledge base*, Meth. Inform. Med. (1991), no. 30, 241–255.

[9] G. Vámos and Á. Nagy and B. Kiss, *Bayesian network based modelling for flaw detection in metallic fusion welds using x-ray images*, Proc. of 4th Workshop on European Scientific and Industrial Collaboration Promoting Advanced Technologies in Manufacturing, Wesic, Miskolc (2003), no. 4, 181–186.

[10] Netica Application, http://www.norsys.com/netica.html

[11] GeNIe Development Environment, http://genie.sis.pitt.edu/downloads.html

[12] Hugin Expert Systems, http://www.hugin.com

Production Systems and Information Engineering
Volume 4 (2006), pp. 95-113

OPTIMISING THE SUPPLIER SYSTEM OF NETWORK-LIKE OPERATING ASSEMBLY PLANTS

MÓNIKA NAGY
University of Miskolc, Hungary
Department of Materials Handling and Logistics
altmoni@uni-miskolc.hu

ÁGOTA BÁNYAI
University of Miskolc, Hungary
Department of Materials Handling and Logistics
altagota@uni-miskolc.hu

JÓZSEF CSELÉNYI
University of Miskolc, Hungary
Department of Materials Handling and Logistics
cselenyi@snowwhite.alt.uni-miskolc.hu

[Received: October 2005 and accepted June 2006]

Abstract. This paper is analysing how the different factors influence the suppliers of each component parts and the optimum number of yearly transports, using a simplified cost function as objective function, based on a general model used for designing and running network-like operating supplier-assembly logistics systems, which has been detailed in former publications. The authors introduce the data structure required for the calculation in the following chapter, then they analyse the influence of each characteristics of the objective function parameters and their relative ratios of one to another in relation to several suppliers and customers, using the method of total discount with limited number of transports as well as using a heuristic algorithm (former analyses referred to one customer i.e. one production company). In the end part, evaluation and comparison of results of the two methods (limited total discount and heuristic algorithm) related to the introduced model and drafting further tasks required for optimum operation of the supplier system will take place, such as taking the capacity limits of the suppliers and combined supplies of several component parts from some suppliers into consideration.

Keywords: optimisation, logistics

1. Introduction

The aim of this paper is to solve a partial optimisation problem not known so far, during which the optimum supplier is to be selected for each component and for

each production company and to determine the optimum number of transports of components into the production companies in a network-like operating supplier-assembly logistics system. There are several approaches to the matter of this paper, i.e. the optimisation of supplier logistics tasks, e.g. purchasing and production model for one product – Hill (1998). Several optimisation methods are known: linear programming – Pan (1989), game theory – Tallury (2002), neural networks – Siying (1997). Novelty of this paper appears in its network-like nature, in its methods of optimising as well as in its approach to the objective functions, especially to the cost functions. The specified system is considered as network-like operating, because analyses are carried out assuming several suppliers, several production companies and several customers. Optimisation is made with several parameters of objective functions and with several limitations. Among the objective functions, the cost function is of great importance, in whose context, it means new ways that realistic costs are used; our philosophy is based on displaying specific costs and natural characteristics when assuming these costs. [6, 7, 8, 9]

2. Objective functions and conditions of optimising the supplier system

In the operation of assembly networks, the optimisation of the supplier logistics system is a highly important task, where different objective functions and conditions must and can be taken into consideration.

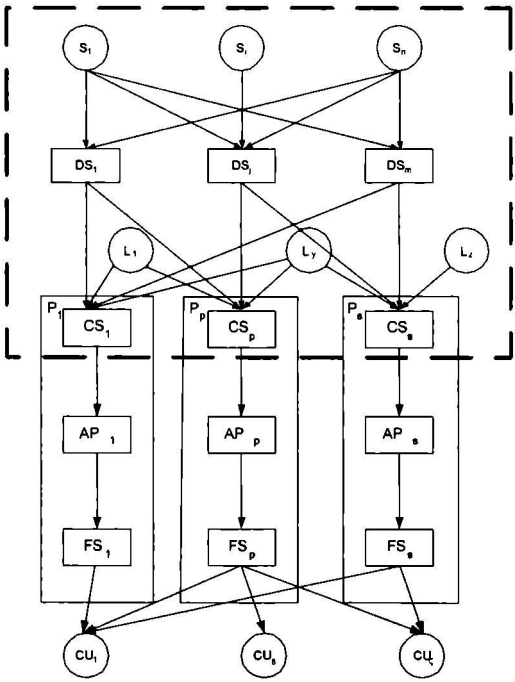


Figure 1. Network-like operating supplier-assembly logistics system

In the first step of the present optimisation, a cost function is chosen as objective function. Besides the method of total discount, the drafted multi-parameter optimisation task requires a heuristic method, which is to be solved in several steps, with a feedback after each step. First, the present network-like logistics system, its objective functions, the influence of ordered quantity on the specific costs, the optimum selection of appointed suppliers and a simplified objective function and its parameters for optimising the number of yearly transports are blocked in.

This system consists of the following units: production companies (P_p), in which component stores (CS_p), assembly plants (AP_p), finished goods stores (FS_p) (finished goods are transported from here to the customers (CU_δ)) are indicated. Transports of component parts can take place into the above mentioned component stores: indirectly from the group of preferential suppliers (S_i), i.e. through distribution stores (DS_j), or by direct transports in case of bypassing the distribution stores. The preferential suppliers provide the so called brand featured component parts, which guarantee the quality connecting to a brand name for the customers. Suppliers within the right proximity to the assembly plants make up the group of local suppliers (L_γ), from where only direct transports can take place into the production companies. [3]

The following cost function is the objective function when analysing the preferential suppliers:

$$C_{gpi}^{S} = C_{gpi}^{SP} + C_{gpi}^{ST_1} + C_{gpi}^{DS} + C_{gpi}^{ST_2} + C_{gp}^{CS} \rightarrow \min \tag{2.1}$$

Where C_{gpi}^{S} is the total cost of component g manufactured by preferential supplier i, which has been ordered by assembly plant p within the present partial system, which consists of the following cost components: C_{gpi}^{SP} is the total purchasing cost of component g in case of purchasing from the preferential supplier, $C_{gpi}^{ST_1}$ is the total transport cost of component g from the preferential supplier to the distribution store, C_{gpi}^{DS} is the total storage cost of component g in the distribution store, $C_{gpi}^{ST_2}$ is the total transport cost of component g from the distribution store to the assembly plant and C_{gpi}^{CS} is the total storage cost of component g in the component store of assembly plant p.

The following objective function refers to the local suppliers:

$$C_{gpy}^{L} = C_{gpy}^{LP} + C_{gpy}^{LT} + C_{gp}^{CS} \rightarrow \min. \tag{2.2}$$

Superior (index) letter L refers to the local suppliers. In this case, supplies take place without distribution stores. In the former studies [1, 2], function connections

of each cost component were unfolded generally. At present, a typical version is being introduced, using those cost function components, which are required for selecting the suppliers and scheduling the transports.

2.1. Simplified cost function

In the first step, a simplified formula of the cost function of the preferential suppliers is used as objective function, which of course, applies to direct transports only.

$$C^{S}_{gpi} = C^{SP}_{gpi} + C^{ST}_{gpi} + C^{CS}_{gpi} \rightarrow \min , \tag{2.3}$$

where C^{S}_{gpi} is the total yearly cost of component g in the present partial system, which consists of the following cost components: C^{SP}_{gpi} is the yearly total purchasing cost, C^{ST}_{gpi} is the yearly total transport cost, C^{CS}_{gpi} is the yearly total storage cost at the production company's component store. (Similar to 2.1.) Formulas for calculating total costs and the cost function components of each total costs are described hereinafter.

2.2. Cost function components

2.2.1. Purchasing cost

$$C^{SP}_{gpi} = s^{SP}_{gpi}(q^{S}_{gpi}) * Q^{S}_{gpi} \tag{2.4}$$

Specific purchasing cost: $s^{SP}_{gpi} = s^{SP}_{gpi}(q^{S}_{gpi})$, where q^{S}_{gpi} means the quantity of component g shipped at the same time by supplier i to production company p. Yearly total purchasing cost can be calculated according to the above mentioned, where Q^{S}_{gpi} means the total quantity ordered yearly. Formula (2.4) assumes that every supply of any component part consists of the same quantity during the year and the component part consumption at the production company is uniform during the examined period of one year.

2.2.2. Transport cost

The following formula is to calculate the total yearly transport cost, assuming that component parts purchased at the same time will also be transported at the same time:

$$C^{ST}_{gpi} = n^{S}_{gpi}\left(Entier\frac{q^{S}_{gpi}}{c_{\phi g}} + \Phi\right) * l^{ST}_{ip} * s^{ST*}_{Gpi}, \tag{2.5}$$

where n^{S}_{gpi} means the number of shipments in one year, i.e. how many shipments of component part g by supplier i to production company p takes place during the examined period. Parameter $c_{\varphi g}$ defines the capacity of the vehicle, i.e. what quantity of component g can be shipped by vehicle φ at the same time. Parameter s^{ST}_{ip} indicates the transport distance between the suppliers and the production company. Parameter s^{ST}_{gpi} is the specific transport cost referring to the average shipped quantity and parameter s^{ST*}_{gpi} is the specific transport cost referring to one shipment. If $\frac{q^{S}_{gpi}}{c_{\varphi g}} = Integer$, then $\Phi = 0$, else $\Phi = 1$, i.e. if the quantity to be shipped is integral multiple of the vehicle capacity, then obviously, the number of required transport vehicles is the quotient result of the above formula. Otherwise, an additional transport vehicle must be put in the shipping progress, even with empty tonnage.

2.2.3. Storage cost

The total storage cost can be calculated as follows, where $\vartheta = 1$ year means the examined period:

$$C^{CS}_{gp} = n^{S}_{gpi} * q^{S}_{gpi} * \frac{1}{2}\frac{\vartheta}{n^{S}_{gpi}} * s^{CS*}_{gp} \tag{2.6}$$

Parameter s^{CS*}_{gp} is the specific storage cost, which means the storage cost of one piece of component g at production company p. The cost calculation according to formula (4) assumes that shipped components are consumed uniformly.

2.3. Specific costs as functions of q^{S}_{gpi}

Specific purchasing cost: $s^{SP}_{gpi} = s^{SP}_{gpi}(q^{S}_{gpi})$, where q^{S}_{gpi} means the quantity of component g transported at the same time from supplier i to production company p.

The figure above shows the step function of specific purchasing cost of component g ordered by production company p and supplied by supplier i. A similar specific cost function can be made for each component part from each supplier. This nature of the function arises from the assumption that higher ordered quantity is accompanied with lower specific purchasing costs. The total yearly purchasing cost can be calculated as follows, where Q^{S}_{gpi} means the yearly ordered quantity:

$$s_{gpi}^{SP}(q_{gpi}^{S}) = \frac{C_{gpi}^{SP}}{Q_{gpi}^{S}} \tag{2.7}$$

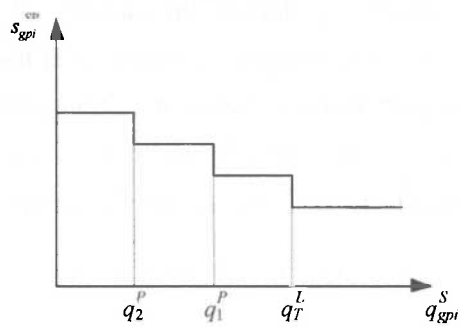


Figure 2. Change of specific purchasing cost as a function of ordered quantity

Specific transport cost: $s_{gpi}^{ST} = s_{gpi}^{ST}(q_{gpi}^{S})$. The specific transport cost can be calculated as follows (2.8). Parameter s_{gpi}^{ST} is the specific transport cost referring to the average quantity and parameter s_{gpi}^{ST*} is the specific transport cost referring to one piece.

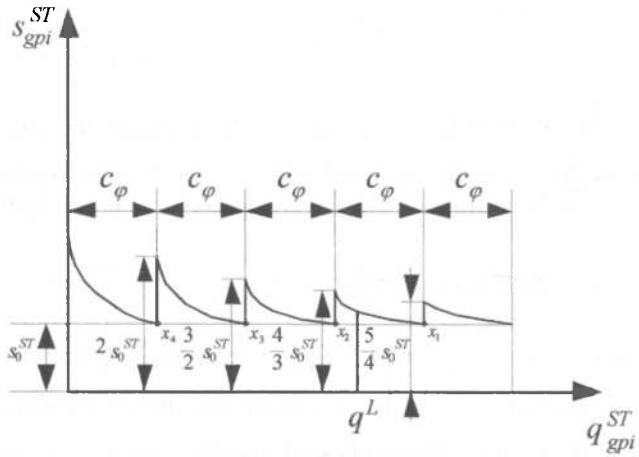


Figure 3. Change of specific transport cost as a function of ordered quantity

Specific transport costs decrease hyperbolically as well as the maximum values of each range also decrease hyperbolically (ranges refer to the transport capacities).

$$s_{gpi}^{ST} = \frac{C_{gpi}^{ST}}{Q_{gpi}^{S}} = \frac{1}{q_{gpi}^{S}} \left(Entier \frac{q_{gpi}^{S}}{c_{\phi}} + \Phi \right) * l_{ip}^{ST} * s_{gpi}^{ST*} \tag{2.8}$$

Specific storage cost: $s_{gp}^{CS} = s_{gp}^{CS}(n_{gpi}^{S})$

$$s_{gp}^{CS} = \frac{C_{gp}^{CS}}{Q_{gpi}^{S}} = \frac{\vartheta}{2n_{gpi}^{S}} f_{gp}^{CS*} \tag{2.9}$$

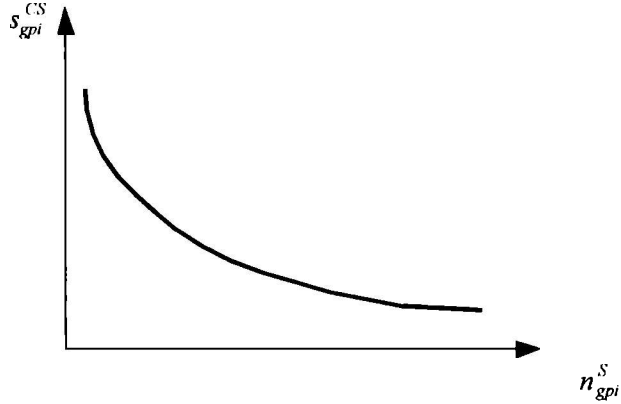


Figure 4. Specific storage cost as a function of yearly transport number

On the one part, the specific storage cost changes as a function of number of transports (hyperbolically); on the other part, it changes as a function of the transported quantity linearly.

A short evaluation of specific cost functions: the specific purchasing cost is constant above the q_T^L limit quantity; the specific transport costs are the lowest at the X points; the specific cost of storage makes up a hyperbolic function as a function of yearly transport number.

The main principle of the heuristic algorithm arises from examining the specific costs: quantity transported at one time may be decreased until transport and purchasing costs do not increase.

The specified simplified cost function can be calculated by two methods. One is the method of total discount, limited regarding the number of transports, which means that the cost function is calculated for numbers of transports discussed in the further parts of this paper (this time only for them) regarding each supplier, component part and production company. The other method is to go through a

heuristic algorithm, whose principle is detailed later. The goal is to compare the results of these two methods within the confines of this paper.

3. Methods of optimisation

3.1. Limited total discount (I.)

On the grounds of the assumed base data, using formula (2.3), the optimum cost is calculated for different numbers of transports regarding component g, supplier i and production company p and the optimum supplier is determined. Because of the large extent of the example, calculations are made only with n_{gpi} = 1, 2, 3, 12 and 48 yearly transports. (Therefore is limited the total discount.) In the first step, calculations are made regarding one production company, then regarding several companies (three companies within the confines of this paper).

3.2. Heuristic algorithm (II.)

When compiling the algorithm, specific purchasing, transport and storage costs have been taken as bases. The essential part of the algorithm examines the given specific cost functions (purchasing and transport costs), determines where the breakpoints are, it calculates step-by-step the costs at these breakpoints and then it calculates the total cost. All these calculations are made while the total costs decrease. At the resulted optimum point, it is also resulted that which component parts should be supplied by which suppliers how many times a year, i.e. the optimum supplier is selected for each component and for each production company and also the yearly transport number is determined regarding a given component and a given production company by this.

3.3. Principle of heuristic algorithm for optimising transports for each component part

3.3.1. Optimising the transports for each component part

1. Determine q^L value (limit quantity) for that supplier of the given component part, where s_{gpi}^{SP} is minimum at q^L, i.e. the specific purchasing cost. It is called limit quantity, because in case of ordering higher quantity than this, the specific purchasing cost of the given component does not decrease further from the given supplier for the given production company, therefore it is not economical to transport higher quantity than this at one time from the point of view of the total cost.

2. Take the point, where $q_T > q^L$ and $q_T \equiv q_{x_1}$ i.e. q_T point at x_1 point after q^L

At x_1 point, there is one of the breakpoints of the specific transport cost, which means that aiming at the maximum utilisation of transport capacity, shipments can be started with x_1 quantity.

3. At q_{X_1} point, determine $C^S_{gpi1} = C^{SP}_{gpi1} + C^{SP}_{gpi1} + C^{CS}_{gp1}$, thus total cost for the given component is resulted at x_1 point.

4. Examine the point, where $q_T < q^L$ and $q_T = q_{x_2}$, i.e. q_T point at x_2 point before q^L, because there is another breakpoint of the specific transport cost.

5. At q_{T_2} point, determine a total cost, i.e. $C^S_{gpi2} = C^{SP}_{gpi2} + C^{ST}_{gpi2} + C^{CS}_{gp2}$.

6. If $C^S_{gpi1} < C^S_{gpi2}$, i.e. total cost at x_2 point is higher than at x_1 point, then it should be examined, if there is a $q_{x_2} > q_1^P > q^L$ range of $s^{SP}_{gpi} = s^{SP}_{gpi}(q^S_{gpi})$ function, i.e. find a breakpoint in the given range of the specific transport cost function, since there is a purchasing cost decrease at this breakpoint. If yes, then take $q_T \equiv q^*_{x_2}$ and calculate the total cost at this point ($C^{S^*}_{gpi2} = C^{SP^*}_{gpi2} + C^{ST^*}_{gpi2} + C^{CS^*}_{gp2}$).

If the resulted total cost is less than total cost at x_1 point ($C^{S^*}_{gpi2} < C^S_{gpi1}$), then $q_T = q^*_{x_2}$ If it is higher ($C^{S^*}_{gpi2} > C^S_{gpi1}$), then using formula $q_T \le \frac{q^*_{x_2} + q_{x_1}}{2} = q^{**}_{x_2}$, find the next point, an ordered quantity, which is between the limit quantity and $q^*_{x_2}$ Determine $C^{S^{**}}_{gpi2} = C^{SP^{**}}_{gpi2} + C^{ST^{**}}_{gpi2} + C^{CS^{**}}_{gp2}$, i.e. the total cost at this point, too. If $C^{S^{**}}_{gpi2} < C^{S^*}_{gpi2}$ and $C^{S^{**}}_{gpi2} < C^S_{gpi1}$, then $q_T = q^{**}_{x_2}$ If $C^{S^{**}}_{gpi2} < C^{S^*}_{gpi2}$ and $C^{S^{**}}_{gpi2} > C^S_{gpi1}$, $\Delta q_{x_2} = q_{x_1} - q^{**}_{x_2} < \Delta q_0$, then $q_T = q_{x_1}$ If $\Delta q_{x_2} > \Delta q_2$, then $q_T = \frac{q^{**}_{x_2} + q_{x_1}}{2} = q^{***}_{x_2}$

Value of q_{x_1} is to be converged step-by-step, until it is closer than Δq_0, i.e. $\Delta q^*_{x_2} < \Delta q_0$.

7. If $C^S_{gpi1} > C^S_{gpi2}$, i.e. the total cost is less at x_2 point than at x_1 point, then sort the common set of $\{(q_{V_2}; q_{V_3}; ...)$ *és* $(q_{x_3}; q_{x_4}; ...)\}$ in decreasing order.

Calculate the cost functions ($C^S_{gpi3}; C^S_{gpi4}; \ldots$), until $q_{T\min} \geq 0{,}1 * q^S_{gpi}$ or $n^S_{gpi} \leq 52$. respectively.

Determine the minimum values of the above calculated costs: $C^S_{gpi0} = Min\{C^S_{gpi3}; C^S_{gpi4}; \ldots\}$ It results $C^S_{gpi0} \rightarrow q^S_{gpiT0}$, i.e. the optimum quantity to order, which also results the optimum number of transports.

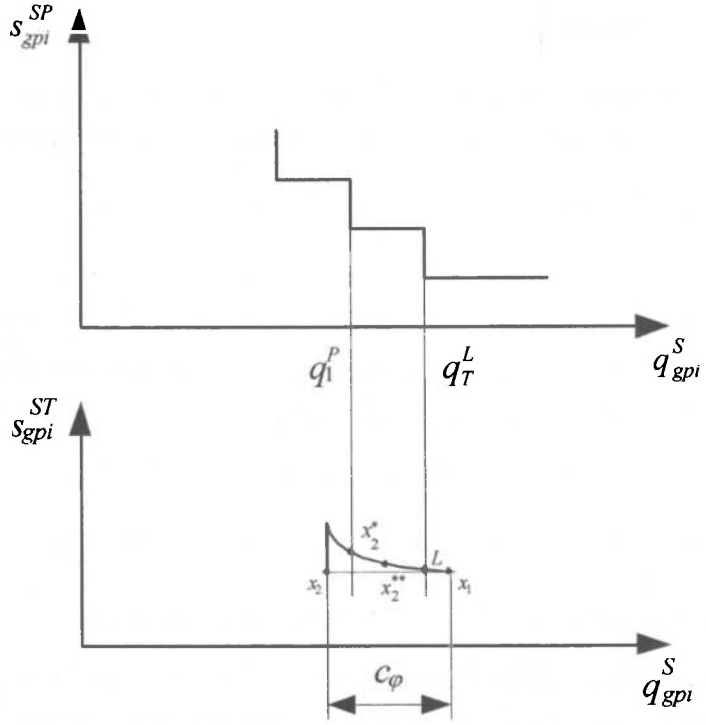


Figure 5. Specific purchasing and transport costs

8. It must be examined for the optimum of q^B_{gpiT0}, if there is any other supplier (v), where $C^{SP*}_{gpi0} + C^{ST*}_{gpi0} < C^{SP}_{gpi0} + C^{ST}_{gpi0}$ i.e. the sum of purchasing and transport cost is more favourable. If $C^{SP*}_{gpi0} + C^{ST*}_{gpi0} < C^{SP}_{gpi0} + C^{ST}_{gpi0}$, then supplier i will be selected. If several i suppliers can be found, then that supplier will be selected, where $\{C^{SP}_{gpi0} + C^{ST}_{gpi0}\} \rightarrow \min$.

The above optimisation must be carried out for each component part and each supplier.

4. Data structure required for examining optimisation methods

To analyse the present system, several base data must be assumed so that the mentioned two methods can be used on this system. Base data in this example can be divided into constant and variable data.

4.1. Constant data of the examinations

g=4, examinations cover four types of component parts,

p=3, costs are calculated for three production companies (in the first example, examinations covered one production company)

φ_i=1, calculations concern one transport vehicle,

$C_{1g} = [700,700,800,1000]$ vehicle capacity matrix, which is defined as follows: transport vehicle no. 1 can carry the given quantity of the component part no. 1, i.e. the vehicle can carry 700 pcs of component part no. 1 at the same time. Similar definitions apply to the other components, too.

$$Q^{B}_{gpi} = \begin{bmatrix} 120 & 100 & 150 \\ 90 & 130 & 120 \\ 150 & 80 & 200 \\ 30 & 50 & 100 \end{bmatrix}$$ [kpcs/year] yearly ordered quantity, e.g. production company no. 2 orders yearly 100,000 pcs. of component part no. 1,

s^{SP}_{gpi} specific purchasing cost can be read out of the graphs (see e.g. Figure 2.) as a function of the purchased and at the same time transported quantity.

$s^{ST^*}_{gpi} = 0{,}8k_0 \ [EUR/travel\_km]$ specific transport cost for each supplier,

$$s^{CS^*}_{gp} = \begin{bmatrix} 0.004 & 0.004 & 0.004 \\ 0.005 & 0.005 & 0.005 \\ 0.005 & 0.005 & 0.005 \\ 0.005 & 0.005 & 0.005 \end{bmatrix} k_0 \ [EURO/(pieces * day)]$$ specific storage cost.

4.2. Variable data of the examinations

During the examinations

i=7, optimum supplier is selected out of 7 suppliers,

in the examinations, all cases of yearly transport numbers n_{11i} = 1, 2, 3, 12 and 48 are to be considered,

distance matrix, its elements mean the transport distances from each supplier in km (transposed matrix is shown here):

$$L_{ip}^{ST} = \begin{bmatrix} 40 & 50 & 70 & 75 & 80 & 100 & 125 \\ 50 & 60 & 80 & 60 & 85 & 60 & 110 \\ 120 & 90 & 70 & 95 & 100 & 90 & 70 \end{bmatrix} km$$

5. Analysing optimum selection of suppliers

Among the examination results, characteristics of suppliers summarised in Figure 6 to 8 are analysed as follows.

Sum of purchasing and transport costs are shown in case of each suppliers: $C_{gpi}^{SP} + C_{gpi}^{ST} = C_{gpi}^{'}(i,n)$, five values of yearly transport numbers can be seen as well as four types of component parts (regarding one production company).

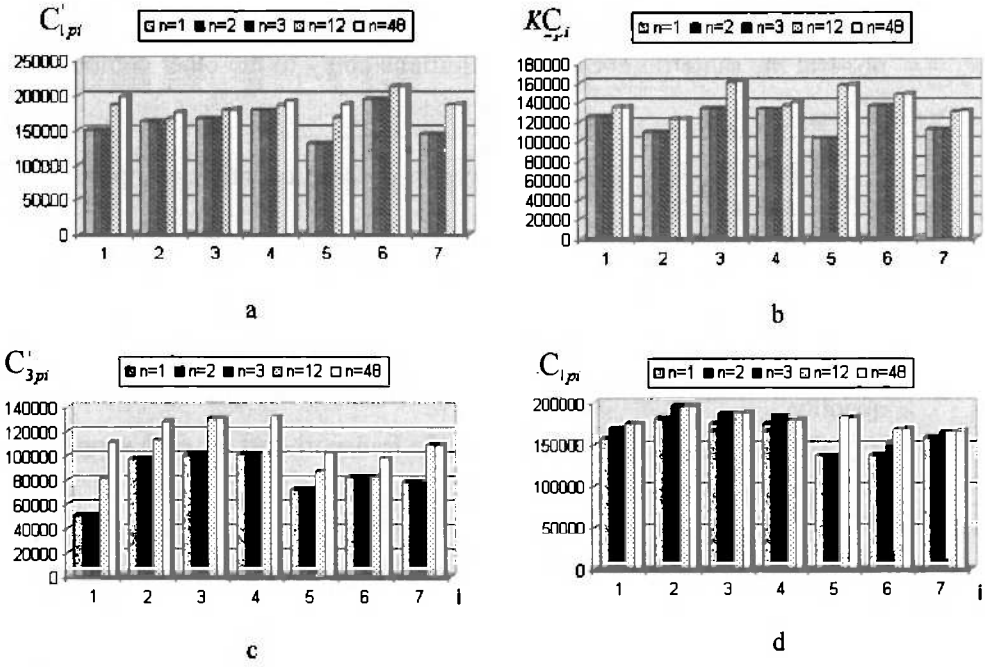


Figure 6. Sum of purchasing and transport costs as a function of yearly transport number

The following can be set out of Figure 6:

the sum of the two costs:

- at components nos. 1, 2 and 4, at n=1, 2 and 3 yearly transports and at supplier no. 5, there are the minimum values; at component no 3, at n=1, 2, 3 and 12 yearly transports and at supplier no. 1, there are the minimum values,

- at components nos. 1 and 2, at n=12 and 48 yearly transports and at supplier no. 2, there are the minimum values; at component no 3, n=48 yearly transports and at supplier no. 6, there is the minimum value; at component no 4, n=12 and 48 yearly transports and at supplier no. 7, there are the minimum values,
- in case of component no 1, at n=1, 2, 3, 12 and 48 yearly transports and at supplier no. 6, there are the maximum values, while at component no. 4, and at supplier no. 2, there are the maximum values,
- at n=1, 2 and 3 yearly transports, in case of component no. 2 and at supplier no. 6, there are the maximum values, while in cases of n=12 and 48 yearly transports, at supplier no. 3 show the highest costs; highest values of component no. 3 in cases of n=1, 2, 3 and 12 yearly transports and at supplier no 3, there are the maximum values; while in case of n=48 yearly transports and at supplier no. 4, there is the highest cost.

In Figure 7, the total yearly cost is shown for different yearly transport numbers and for different component parts in case of the optimum supplier $C_{gpi}^{SP} + C_{gpi}^{ST} + C_{gp}^{OS} = C_{gpi}^{S}(n)$. Values in the graph are accompanied with a k_0 factor, which makes the resulted values relative.

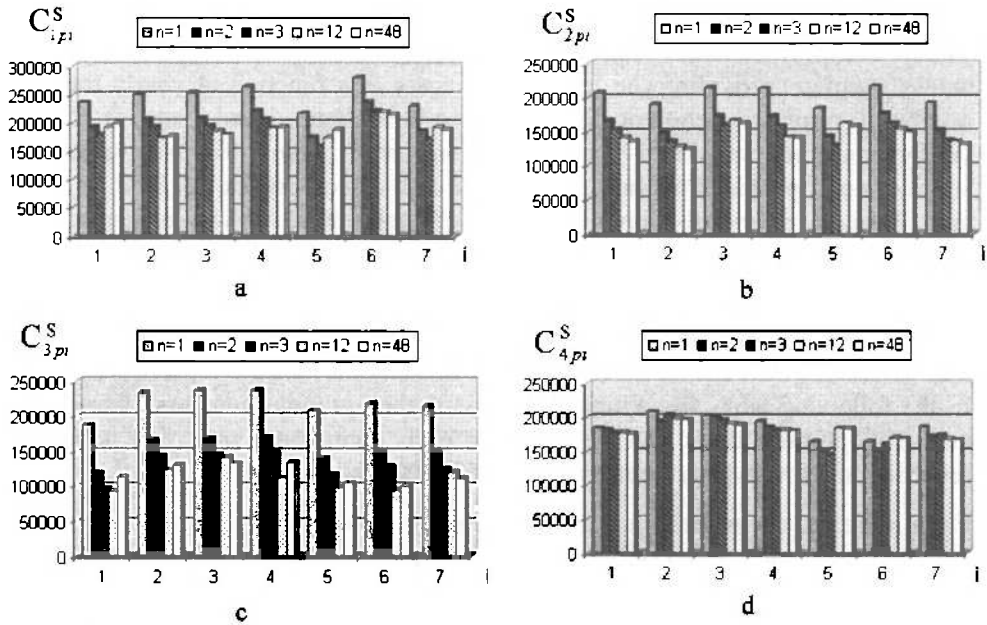


Figure 7. Change of total cost in cases of different yearly transport numbers

Differences between the sums of each total cost regarding each component part and considering each yearly transport numbers and each suppliers are the follows: at component no. 1, difference between the highest and lowest costs is: 105584 k_0; at component no. 2, it is: 90878.1 k_0; at component no. 3, it is: 145604.8 k_0; while at component no. 4, it is: 62010 k_0. Analysing Figure 7, it can be set out that if the optimum supplier is selected, then at n=2, 3, 12 and 48 yearly transport numbers, the total costs hardly change.

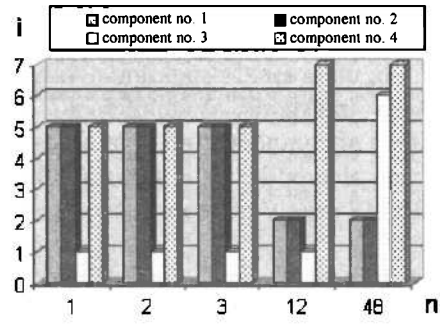


Figure 8. Selecting optimum supplier as a function of yearly transport number regarding each component part

In Figure 8, the optimum suppliers $i_0 = i_0(n)$ are delineated in cases of different yearly transport numbers regarding four types of component parts. This figure shows suppliers providing the lowest total costs as a function of yearly transport number. It turns up unequivocally, that at components no. 1 and 2, at n=1, 2 and 3 yearly transports, supplier i=5 is the best and at n=12 and 48 yearly transports, supplier no. 2 provides the best solution. In case of component no. 3, at n=1, 2, 3 and 12 yearly transports, supplier i=1 is the optimum and at n=48 yearly transports, supplier no. 6 is the optimum supplier. Regarding component no. 4, at n=1, 2 and 3 yearly transports, supplier no. 5 is the optimum, while in cases of n=12 and 48 yearly transports, it is practical to order from supplier no. 7. All these are consonant with the results of Figures 3 to 8.

In the following table, the optimum suppliers, the optimum numbers of transports and the optimum costs can be seen for each component part. The total cost regarding the four types of components is 523402.2 k_0.

Examining the optimum solutions for each components and the results of supply combinations made based on these solutions (e.g.: in case of n=1 transport a year, components nos. 1, 2 and 4 are supplied by supplier no. 5 at the same time), it can be set out that the cost function is reduced only by 0.008 to 1.382 percent. It can be established from all these, that transport cost is not as dominant as purchasing and

storage costs in case of the present example. Although the total cost is not reduced considerably based on the above results, however taking other points of view into account (only two suppliers are to be in communication with, contingent price reductions because of purchasing three types of component parts, fixed and constant number of yearly supplies for each components, etc.) it is worth to purchase the appointed three types of components from supplier no. 5 as well as component no. 3 from supplier no. 1 three times a year.

Table 1. Optimum suppliers, the optimum numbers of transports and the optimum costs

Component no. (g)	Optimum supplier no. (i)	No. of optimum yearly transports (n)	Optimum cost ($C^S_{gpi} * k_0$)
1	5	3	160336
2	2	48	124470.9
3	1	12	92550.3
4	5	3	146045

In cases of different yearly transport numbers, the following table shows, which component parts to be supplied by which suppliers.

Table 2. Connection of the component parts and the suppliers

No. of transports	Component no. → Optimum supplier
1	$1,2,4 \rightarrow 5;\ 3 \rightarrow 1$
2	$1,2,4 \rightarrow 5;\ 3 \rightarrow 1$
3	$1,2,4 \rightarrow 5;\ 3 \rightarrow 1$
12	$1,2 \rightarrow 2;\ 3 \rightarrow 1;\ 4 \rightarrow 7$
48	$1,2 \rightarrow 2;\ 3 \rightarrow 6;\ 4 \rightarrow 7$

It has been displayed that in case 1 production company, 4 types of component parts and 7 suppliers, how to select the optimum supplier using limited total discount, and which is the most favourable among the given yearly transport numbers regarding the costs.

6. Comparing the methods for optimising supplier logistics systems

In the following, results of the introduced two methods are given for three production companies. In the following table, optimum suppliers, yearly transport numbers and costs as results of total discount and heuristic algorithm can be seen for each component parts. In case of four components and regarding different numbers of yearly transports, the total cost is 523402.2 k_0 by total discount and it is 495341 k_0 by heuristic algorithm. It can be seen that result of heuristic algorithm is better regarding the total cost. It is because the above mentioned cause, i.e. by total discount, optimum case is determined for the given yearly transport numbers only, because of the large extent of the example.

Table 3. Results of total discount and heuristic algorithm

Component no. into Production co. no. (g-p)	Optimum supplier no. (i)		Optimum no. of yearly transports (n)		Optimum cost ($C_{gpi}^{S} * k_0$ [*EUR*])	
	I.	**II.**	**I.**	**II.**	**I.**	**II.**
1 into 1	5	5	3	4	160336	152908
2 into 1	2	5	48	3	124470.9	130131
3 into 1	1	1	12	9	92550.3	66257
4 into 1	5	5	3	3	146045	146045
Total					**523402.2**	**495341**
1 into 2	5	5	3	3	134125.82	134125.82
2 into 2	2	5	48	5	180687.96	173145
3 into 2	6	1	12	5	51267.64	42600
4 into 2	6	5	3	5	242656.64	237525
Total					**608738.06**	**587395.82**
1 into 3	5	5	3	7	203780	205503.17
2 into 3	7	5	48	4	169033.25	167135
3 into 3	1	1	12	12	99400.64	99400.64
4 into 3	6	5	3	10	515380.78	467125
Total					**987594.67**	**939163.81**
Grand total					**2119734.93**	**2021900.63**

If each component is compared in case of production company no. 1, then practically, very little difference turns up regarding components nos. 1 and 2. At component no. 4, the two methods give the same results, considerable difference is resulted at component no. 3 only, which is 26293.3 k_0, but the algorithm is better. This can be because of the before mentioned, i.e. total discount has been done for the given numbers of transports, because of the large extent of the example. It can be seen well that also in this case, both methods give supplier no. 1 as optimum, difference only occurs between the yearly transport numbers, which results in the already mentioned difference. Regarding production company no. 2, there is also a component part (component no. 1), for which both methods give the same results. Considerable differences occur for the optimum suppliers and optimum transport numbers of methods I and II. However, if the results of both methods are approached from the cost side, then it can be noticed that the difference is not so much examining components nos. 2 and 4. For component no. 3, the difference is 16.9 percent in favour of the heuristic algorithm. For production company no. 3, in case of component no. 3, both methods give the same results. In case of component no. 1, it results the same supplier, but with different transport numbers. However, this is the only case, where total discount gives better result regarding the costs. In relation to the other three production companies, total discount gives worse results by 4.615 percent than heuristic algorithm.

Data in the above table are shown is the following graph.

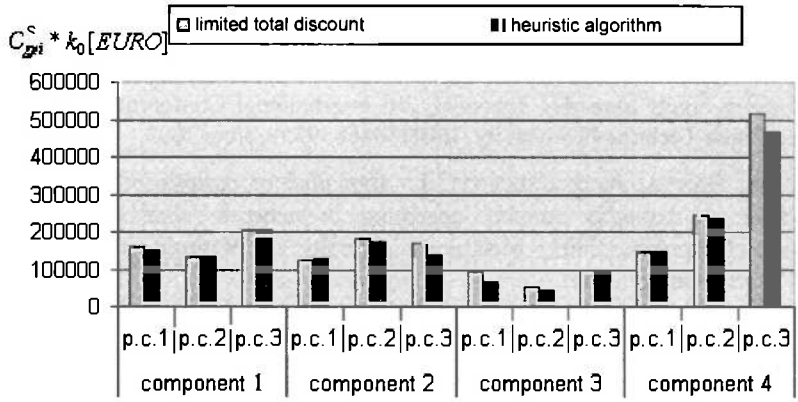


Figure 9. Comparing results of total discount and heuristic algorithm

7. Summary

In this paper, a network-like operating logistics system, objective functions as well as an optimisation algorithm for the simplified objective function is being presented. This paper shows the optimum numbers of suppliers and yearly transports for each component parts in cases of three production companies, seven

suppliers and four types of component parts; it displays the results of the two methods used for cost optimisation and their comparison. Efficiency of heuristic method is proven unequivocally, besides that it results lower costs and it needs considerably less calculation mainly in tasks of great extent. (It must be noticed, that only limited total discount has been made, because of the large extent of the example, therefore heuristic method resulted better solutions.) The heuristic algorithm results better solution unequivocally, if there is no need to decide on fixed numbers of transports and it requires less calculations. If the number of transports are fixed, then the two methods are of the same accuracy, but in this case, calculation method is much simpler and it demands less time. Henceforth, a correction method will be introduced, which takes the capacity limits of the suppliers and the possibility of transporting several types of component parts at the same time into account. One of the authors' intentions is to examine it referring to four types of comonent parts, that using contractions in case of each component, to what extent lesser costs can be achieved compared to the optimum solutions resulted herein.

Acknowledgements

The research work described briefly in this paper has been supported by OTKA project No.: F037525, and No.: T038382.

REFERENCES

[1] NAGY, M.; BÁNYAI Á. & CSELÉNYI, J.: *Analysis of purchasing logistics system of electronic products assembly networks,* 3rd International Conference on AED 2003, Prague, Czech Technical University, ISBN8086059359, June 2003.

[2] NAGY, M.; BÁNYAI Á. & CSELÉNYI, J.: *Algorithm of optimisation of selection of purchasers of assembly systems operating in networks and supply schedules,* Miskolcer Gespräche 2003, Miskolc, University of Miskolc, ISBN9636615950, pp.:29-34, September 2003.

[3] NAGY, M.; BÁNYAI Á. & CSELÉNYI, J.: *Optimization of schedule of transport and selection of purchasers of assembly systems operating in networks*, microCAD 2004 International Scientific Conference, University of Miskolc, ISBN9636616205, pp.: 91-96, March 2004.

[4] NAGY, M.; BÁNYAI Á. & CSELÉNYI, J.: *Data structure of network-like operating assembly plants for the optimisation of the supplier system* (in Hungarian), OGÉT 2004, XII. Nemzetközi Gépész Találkozó, Erdélyi Magyar Műszaki Tudományos Társaság – EMT, Csíksomlyó, 2004. április 22-25. ISBN9738609798, pp.:216-220.

[5] NAGY, M.; BÁNYAI Á. & CSELÉNYI, J.: *Sensibility analysis of optimization of purchasing logistic system as a function of yearly transport number*, Modelling and

Optimisation of Logistic Systems – Theory and Practice, Edited by T. Bányai and J. Cselényi, University of Miskolc, 2004.

[6] APPLE, J. M. *Plant Layout and Material Handling.* John Wiley &Sons, ISBN 0471 07171 4, New York, 1977.

[7] CSELÉNYI, J. & TÓTH, T. *Some questions of logistics in the case of holonic production system.* Journal of Intelligent Manufacturing. 9. 113-118, ISSN 0956 5515, 1998.

[8] SIMCHI-L D.; CHEN X. & BRAMEL J. *The logic of Logistics. Theory, algorithms and applications for logistics and supply chain management.* Springer Series in Operations Research. Editors: Peter Glynn, Stephen Robinson. ISBN 0-387-22199-9, 2005.

[9] M. CAMARINHA-MATOS L. *Virtual enterprises and collaborative networks.* IFIP 18th World Computer Congress, Kluwer Academic Publishers. ISBN 1-4020-8138-3, 2004.

Production Systems and Information Engineering
Volume 4 (2006), pp. 115-124

USING REGRESSION TREES IN PREDICTIVE MODELLING

TAMÁS FEHÉR
University of Miskolc, Hungary
Department of Information Engineering
feher@ait.iit.uni-miskolc.hu

[Received November 2005 and accepted May 2006]

Abstract. Tasks where the value of an unknown parameter has to be estimated are typical in data mining projects. The solution of this kind of problems requires the creating of a predictive model. There are several methods for the solution of this type of tasks, e. g. the decision trees and the regression. In the paper an algorithm is demonstrated which combines the properties of the two techniques. Tests on different datasets show the efficiency of the method compared to the results which were provided by the traditional regression and decision tree algorithms.

Keywords: data mining, predicitve modelling, decision tree, regression

1. Introduction

Databases contain a lot of hidden information that can be used by business decisions. During the classification and prediction the value of an unknown parameter is predicted. While in the classification tasks the value of a categorical variable has to be defined, in the prediction tasks the value of a continuous variable is estimated. The model which gives estimation for the value of the target variable by the help of the predictor parameters is named predictive model. The predictive models build up by the help of a training dataset. Their use-value reveals itself when they are able to give estimation from samples which they have never seen. Several techniques are used in the predictive modelling e. g. neural networks, regression, decision trees [8][9]. This paper concentrates on the last two methods.

The traditional decision trees serve basically for the solution of classifications tasks. In each leaf there is a class label that shows, which class the questionable object falls in. However, there are decision tree methods, which make possible the estimating of a continual variable, in such a way that they put a regression model in the leaves. These complex models are called *regression trees*.

The first regression tree method is the CART algorithm [1], which puts a constant function to the leaves. During the years, further algorithms were developed, which put linear functions into the leaves for the more efficient estimation of the continuous value (M5[2], TSIR[3], RETIS[4], SMOTI[5], HTL[6], SECRET[7]).

The difference between the algorithms is in the induction of the tree. The most important factor is how the algorithm finds the split points. The most of the algorithms use a variance-based approach for the measure of the quality of cutting. In the original algorithm of CART the best split in a node is the one that minimizes the excepted impurity I_{exp}, given by the formula:

$$I_{exp} = p_l I_l + p_r I_r, \tag{1}$$

where p_l and p_r denote the probabilities of transition into the left and the right side of a split and I_l and I_r are the corresponding impurities, resting on variance of the children nodes. The variance in a node can be computed by the next formula:

$$\sigma^2(E) = \frac{1}{W(E)} \sum_{e_i \in E} w_i (y_i - \mu(E))^2 \tag{2}$$

where E denotes the set of elements in a node, w_i is the weight of the i^{th} element and $W(E) = \sum_{e_i \in E} w_i$ The y_i is the value of the i^{th} element and $\mu(E)$ denotes mean class value of E.

2. Regression models in the leaves

In [4] it is recognized that in the case of using linear regression in the leaves of the tree, it is not sure, that the goodness of the split has to be judged by the variance. Figure 1 shows, although the variance would be minimal by the cut 'a', nevertheless it seems to be reasonable to bring in a new measure of goodness and to accomplish the split in the point 'b'

According to the above mentioned aspects, the measure of the cut's goodness is the fitting error of the regression model on the left and right side. In the next formula:

$$I(E) = \frac{1}{N_l + N_r} \left[\frac{1}{N_l} \sum_{e_i \in E_l} (y_i - g_l(\vec{x}_i))^2 + \frac{1}{N_r} \sum_{e_j \in E_r} (y_j - g_r(\vec{x}_j))^2 \right] \tag{3}$$

I(E) denotes the measure of a cut's goodness. N_l and N_r are the number of elements in the left and the right side of the cut, E_l and E_r denote the set of elements in the left and the right side. Function g_l and g_r represent the regression plane through elements in the left child and in the right child. The actual values of the elements are denoted by y_i and y_j.

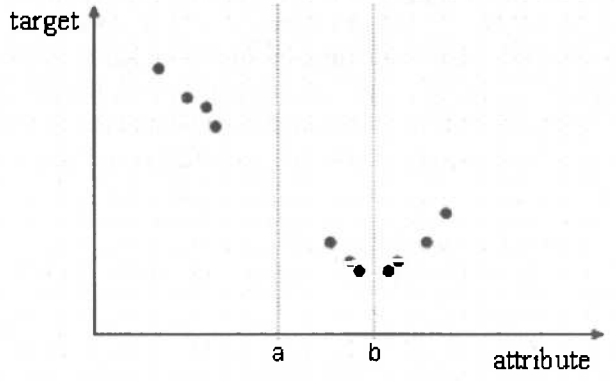


Figure 1. The appropriate split is in the point 'b'

The disadvantage of this method is the huge computing time. In the original paper the author used this algorithm on datasets which contained only few observations. Considering that the measure of cut's goodness is the fitting error of the regression models on the two sets, theoretically all of the splits would have to be executed along all of the dimensions in order to decide, what is the best split on a given node. For example on a dataset, which contains 10^4 observations and which has 10 attributes, about 10^5 splits would have to be performed. This would mean building of $2x10^5$ models (only for one level). In the case of a tree with five levels computing of about 10^6 regression models would be necessary. It is very high number, especially if it is considered that a real-life database can be substantially bigger, having regard of number of elements and dimensionality.

The algorithm presented in this paper extends the idea demonstrated in [4]. In order to building-up the tree finishes in feasible time, in the new algorithm is not accomplished all of the cuts. If the cutting is put in at only every 10^{th}, 100^{th} or 1000^{th} value, and thereby the number of splits along a dimension is reduced to order of magnitude of 10. At price of this compromise may be made not the best, but a good model. This is supported by the running results (see Chapter 4).

3. Regression Tree Algorithm

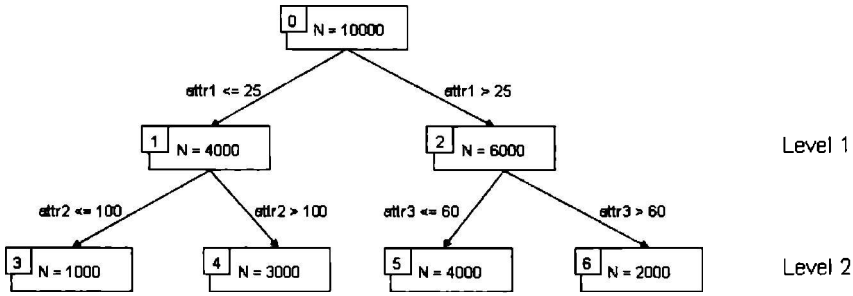
The core of the algorithm is a process, which carries out the given splits along every dimension (the number of splits is determined as an input parameter) and then chooses the split along which cutting the dataset the two regression hyperplanes can be fit with the minimal error (the average error as the measure of the split's goodness is the average of mean square errors weighted by the number

of elements on the left and the right side). So, it makes the regressions to each split. Choosing the best split, two tables has been generated storing the elements of the two child-nodes. The descriptor information of the nodes is written into a special table describing the tree. These pieces of information are: the path from the root to the node, the number of elements in the node, the parameters of the regression and if the node is a leaf then it is signed with a binary flag (see Figure 2). The node is a leaf, if

(i) the tree reached the given depth,
(ii) the leaf does not contain more elements than twice the minimal number of elements.

This process is accomplished iteratively on each level. The number of iterations is equal to the number of nodes on the former level. The input of the procedure is a node from the former level in each iteration step.

The output is a table (see Figure 2), and it contains all information, which is necessary to building-up of the tree.



ID	Leaf	Rule	N	Intercept	Param1	Param2	
1	0	attr1 <= 25	4000	123.456	0.789	2.345	
2	0	attr1 > 25	6000	234.567	1.234	5.432	
3	1	attr1 <= 25 & attr2 <= 100	1000	87.654	0	0.987	
4	1	attr1 <= 25 & attr2 > 60	3000	34.567	0.456	0.023	
5	1	attr1 > 25 & attr3 <= 100	4000	54.321	2.345	0	
6	1	attr1 > 25 & attr3 > 60	2000	12.345	0.456	0.789	

Figure 2. Example illustrating the algorithm

The algorithm achieves a stepwise linear regression in any cases. In the stepwise method, variables are added one by one to the model, and the F statistic for a

variable to be added must be significant at a given level (SLENTRY). After a variable is added, however, the stepwise method looks at all the variables already included in the model and deletes any variable that does not produce an F statistic significant at a determined level (SLSTAY). Only after this check is made and the necessary deletions accomplished can another variable be added to the model. The stepwise process ends, when none of the variables outside the model has an F statistic significant at the given SLENTRY level and every variable in the model is significant at the SLSTAY level, or when the variable to be added to the model is the one just deleted from it.

The algorithm is implemented in SAS Base [10]. The benefit of this environment is the speed (among others): the running time of a stepwise regression on a dataset containing more ten thousands of observations, is a split seconds.

3.1. The assignation of cutting points

The assignation of cutting points works by the undermentioned algorithm.

Input: minimal number of elements in a leaf (MIN), the number of intervals through cuttings, i. e. number of cuts + 1 (N), dimension (DIM)

Step 1: sorting the values along dimension DIM
Step 2: detaching first elements from the list in size of MIN
Step 3: detaching last elements from the list in size of MIN
Step 4: dividing the remainder range to N-2 parts

The algorithm handles exceptions, as follows:

Exception 1: If there are less available values than N, then these values turn split bounds, except when thereby it is not possible to make groups with MIN size in the forepart and in the end of the dimension. In this case the algorithm moves forward to the interior of dimension and tries to assign the next value as lower/upper bound. This property guarantees for example the recognition a binary attribute.

Exception 2: If the number of elements is less than 2*MIN, it does not begin to split dimensions, because in this case it is not possible to create two nodes with MIN size.

Exception 3: If lots of elements belong to a given value along a dimension, exceptions can occur, too. For example there is a database, where half of the customers has 0 dollar on the account and only another half of the customers possesses an account with positive balance. Supposing that after cutting of the first MIN pieces customer, the split point is in the middle of value 0. Since simultaneously only one dimension is in the focus, it cannot be distinguished

between customers with balance 0. The algorithm recognizes this situation and shifts the split points.

Ergo the algorithm stands for working well-balanced, namely it makes all of a size cuts (excluding exceptions, and except the front and the end of dimensions).

4. Experimental Results

There are results from running the algorithm on three different datasets. In case of each dataset the target variable is estimated with regression at first, then with a decision tree algorithm implemented by SAS Enterprise Miner and with the regression tree algorithm at last. The average relative error of the predicted values is chosen for the measure of predictive power of the different methods.

10 percent of the data was separated for testing in case of each dataset. Further 10 percent was separated for the validation in case of SAS decision tree.

The first dataset was artificially generated. In this dataset the number of dimensions is 3 and the number of records is 100. The continuous target variable can be estimated by two predictor variable. One of them is a binary and the other is an interval variable. The dataset can be shown on Figure 3.

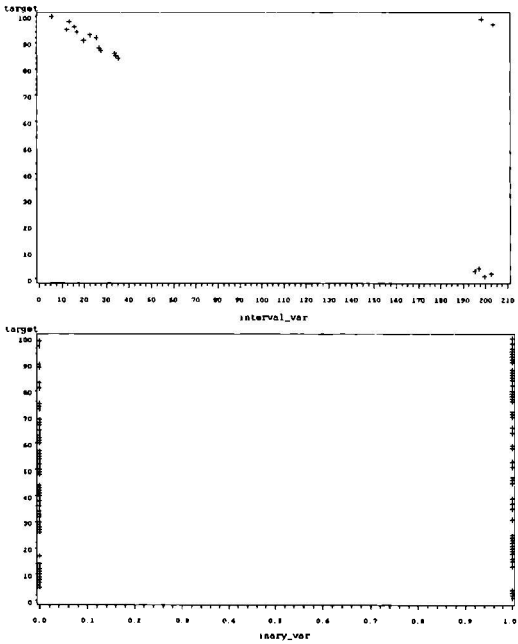


Figure 3. Different views of the dataset1

Investigating the figure it can be seen immediately: in order to the prediction is the most exact, the dataset has to be cut along the binary variable, because the fitting error of the regression models is the smallest in this manner.

Table 1 contains the results of the regression, the decision tree and the regression tree.

Table 1. Model errors on dataset1

Model	Average relative error (%)
Regression	120.92
Decision Tree	104.43
Regression Tree	5.18

As the results show, the behaviour of target variable in dataset1 cannot be modelled acceptable neither with regression nor with decision tree. However the regression tree gives an adequate model. Nevertheless it is not allowed to come to profound conclusions, because this dataset was generated to indicate that there are situations, where the regression tree algorithm performs much better than the traditional methods.

In the followings a real-life database is investigated. The data issue from a medical database. It is not too large, it contains 470 observations and the number of dimensions is 10. The target variable is a continuous quantity named PWV (pulse wave velocity) which means the velocity of the compression wave occurring by the heart-beat [11]. From this parameter can be reasoned the state of the blood-vessels. This value is estimated from other data, such as age, systole and diastole blood pressure, etc.

Table2. Model errors on dataset2

Model	Average relative error (%)
Regression	2.84
Decision Tree	4.59
Regression Tree	0.52

The target variable can be predicted very exactly from the predictor variables. All of the models give little error but it can be seen, that the predictive power of the regression tree is the best.

The third dataset issues from a moneyed corporation environment. It contains 20018 observations and 10 dimensions. The target variable is the income which can be estimated very poorly from the available predictor variables, but the best estimation is given by the regression tree.

Table 3. Model errors on dataset3

Model	Average relative error (%)
Regression	83.10
Decision Tree	53.08
Regression Tree	46.76

4.1. Fine tuning of the models

All of these models can be tuned by different parameters. Since the number of possible model variants is huge, finding the absolute best solution is a difficult task. More models were built to each dataset. The parameters of stepwise regression were constants in the case of every running (slstay = 0.08, slentry = 0.08, see Chapter 3). The SAS Decision Tree can be tuned by several parameters (e. g. splitting criterion: F test / variance reduction, minimum number of observations in a leaf, maximum depth of tree). The regression tree algorithm can be tuned by three parameters: maximum depth of tree, minimum number of observations in a leaf and the number of splits along a dimension (per levels).

The tables contain only the test results of the model variants, which gave the best average relative error on the test data. The documentation of fine tuning's process is not in the paper because of lack of space, but speaking in a general way it can be stated, that the regression tree performed better than conventional regression in any case, and also better than decision tree almost in all cases (the decision tree was able to be better in only such cases, where it was well-tuned and the regression tree was roughly-tuned).

5. Conclusions

Extending the idea presented in [4], a new algorithm has been developed, which combines the properties of regression and decision trees. The result is a decision tree whose leaves contain regression models. By finding the split points the fitting error of the regression model on the left and right side is used as the measure of the cut's goodness. In order to the induction of the tree finishes in an acceptable time, in the algorithm is not carried out all of the cuts. This compromise redound an good predicitve model.

According to the results of the tests the model is much better than the traditional regression (it results from the structure of the model), and it is better than the decision tree used in some commercial software products, as well.

5.1. The speed of the algorithm

The method accomplishes actually the series of splitting and stepwise regression steps. Running the algorithm on the third database (which is the largest) the running time was about 20 minutes on an average computer (PIII, 850 MHz, 384 MB RAM) in the case of a relative big tree (5 levels). Although there are no results referring to this, according to the experiences from the tests, the running time depends minimally on the number of the observations. The most important factor is the number of the splits. The relation between the number of the splits and the running time is linear.

5.2. Further Developments

The algorithm misses the pruning for the present. But with appropriate setting up of the number of the elements in a node the overfitting can be minimalized. Even so completing this algorithm with a pruning method some performance increase could be reached.

From the point of view of the running time it is an important question, how many splits are executed during the build-up of the tree. It would require further researches answering of the question: what kind of algorithm would be optimal to determinate the cut points.

Acknowledgements

The author would like to thank Professor *László Cser*, *Bulcsú Fajszi* and *Márton Zimmer* for the support and the valuable pieces of advice.

REFERENCES

[1] BREIMAN, L., FRIEDMAN, J., OLSHEN, R., STONE, J.: *Classification and regression tree*, Wadsworth & Brooks, 1984.

[2] QUINLAN, J. R.: *Learning with continuous classes*, in Proceedings AI'92, Adams & Sterling (Eds.), World Scientific, pp. 343-348, 1992.

[3] LUBINSKY, D.: *Tree Structured Interpretable Regression*, in Learning from Data, Fisher D. & Lenz H.J. (Eds.), Lecture Notes in Statistics, 112, Springer, pp. 387-398, 1994.

[4] KARALIC, A.: *Linear Regression in Regression Tree Leaves*. In Proceedings of ISSEK'92 (International School for Synthesis of Expert Knowledge) Workshop, Bled, Slovenia, 1992.

[5] MALERBA, D., APPICE, A., CECI, M., MONOPOLI, M.: *Trading-off local versus global effects of regression nodes in model trees.* In H.-S. Hacid, Z.W. Ras, D.A. Zighed & Y. Kodratoff (Eds.), Foundations of Intelligent Systems, 13th International

Symposium, ISMIS'2002, Lecture Notes in Artificial Intelligence, 2366, 393-402, Springer, Berlin, Germany, 2002.

[6] TORGO, L.: *Functional Models for Regression Tree Leaves.* Proc. 14th International Conference on Machine Learning, 1997.

[7] DOBRA, A., GEHRKE, J.: *SECRET - A Scalable Linear Regression Tree Algorithm.* In Proc. of ACM SIGKDD, pp. 481-487, 2002.

[8] HAN, J., KAMBER, M.: *Data Mining – Conceptions and Techniques.* Panem, Budapest, 2004. (in Hungarian)

[9] FAJSZI, B., CSER, L.: *Business Knowledge in the Data.* Budapest, 2004. (in Hungarian)

[10] SAS INSTITUTE INC.: http://www.sas.com/

[11] HAST, J.: *Self-mixing interferometry and its applications in noninvasive pulse detection.* http://herkules.oulu.fi/isbn951426973X/html/x957.html

Production Systems and Information Engineering
Volume 4 (2006), pp. 125-145

COMPARISON OF DIRECT AND INDIRECT DISTRIBUTION OF A NETWORK-LIKE OPERATING LOGISTICS INTEGRATED ASSEMBLY SYSTEM

BÉLA OLÁH
University of Miskolc, Hungary
Department of Materials Handling and Logistics
altbela@uni-miskolc.hu

TAMÁS BÁNYAI
University of Miskolc, Hungary
Department of Materials Handling and Logistics
alttamas@uni-miskolc.hu

JÓZSEF CSELÉNYI
University of Miskolc, Hungary
Department of Materials Handling and Logistics
cselenyi@snowwhite.alt.uni-miskolc.hu

[Received October 2005 and accepted June 2006]

Abstract. This paper introduces the limitations and objective functions of planning of a network-like operating logistics integrated assembly systems. The optimal assignment of assembly plants to the final product requirements of the end users is discussed in detail, and related cost functions are worked out. Solution methods of optimisation are described in the next chapters. The sensitivity analysis of the assignment algorithms concerning to products and assembly plants is completed by a simple example and comparison of different variations is showed. Finally the system of one distribution warehouse model and the description of its algorithm is showed. Novelty of this paper appears in its network-like nature, in its methods of optimising as well as in its approach to the objective functions, especially to the cost functions.

Keywords: assignment, logistics, optimisation, assembly system

1. Introduction

The network-like operating logistics integrated assembly system means when the production planning is planned integrated by the purchasing and distribution logistics system, accordingly we search aggregate optimum of: not merely the production but also the logistics resources and factors. The network-like means that the same product can be assembled by several assembly plants in different points, and the components needful to assembling can be purchased from several different

sited suppliers. Additionally the network-like means that the procurement of components and the distribution of final products may be direct and indirect, in other words by the help of distribution warehouse. In case of the network-like operating systems the logistics integrated production planning details how search the optimal result having regard to capacity-limits and conditions, fulfill to the requirements of the end users according to described objective functions.

Mathematical modelling and optimisation of a network-like operating assembly system as an integrated logistics system there is no even early attempt in the international scientific literature. A mere scattering of publications for the logistics integrated production scheduling can be found in the international literature [9, 10, 11, 12]. Therefrom can be determined that the used objective functions and conditions are in accord with the used by us. At the same time in the system drawn by us the objective functions and the conditions can be demonstrated in an other form, it follows from this, that this defined principle can be used for the optimisation. All these require that we lean on the considerable results made in the Department of Materials Handling and Logistics at the University of Miskolc for solution of this logistics model [1, 4, 13, 14].

The globalisation and the decreasing of the production depth leaded to sweeping changes of the market of firms. These changes can be recognised by the increasing of cooperative industrial processes based on horizontal and vertical networking depending on the depth of competitiveness [7]. In the model (Fig. 1) the amount of the final products ordered by an end user in a given time interval is given.

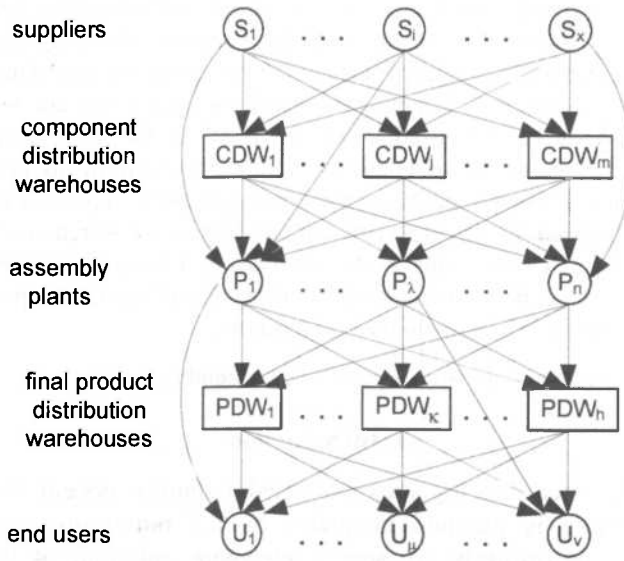


Figure 1. Network-like operating assembly system

The optimal operation of this complex and large cooperative logistics system requires absolutely modern theoretical establishment of planning and control methods [1]. The task to be completed is the logistics integrated assignment task, which includes the distribution and storage of final products and the storage of components. Different objective functions and conditions must and can be taken into consideration during the solution of these tasks. In the first case the cost function is chosen as objective function, whose components are detailed in [3]. The optimisation is completed by a hierarchical jointed feedback heuristic method due to the high number of cost function parameters to be optimised. The modules of a multistage optimisation and the principles [4], solution methods and heuristic algorithm of the assignment are elaborated in [5].

This system consists of the following units: assembly plants (P_λ), component distribution warehouses (CDW_j) and final product distribution warehouses (PDW_κ). Transports of component parts can take place into the above mentioned assembly plants: indirectly, i.e. through component distribution warehouses, or by direct transports from the suppliers (S_i) in case of bypassing the distribution warehouse. Transports of final products occur to the end users (U_μ) in like manner, i.e. direct delivery from the assembly plants or indirectly through final product distribution warehouses.

The authors worked out the assignment of final products requirements of separate end users to the assembly plants relating to time-intervals ϑ with the following simplyfied coditions in paper [3]:

we have taken no notice of the preparation cost separate from the assembly cost, in this planning level not come to optimise the assembly lot size;

the specific assembly cost is conditioned by only final product type;

only the direct distribution model is analysed by the help of shuttle tours.

2. Aims of the paper

This paper shows the followings:

assignment algorithms of the before-mentioned simplyfied network-like operating system are stated;

exact data-model is elaborated, by the help of it three optimal assignment variations are worked out;

the optimum sensitivity analysis are accomplished, the specific assembly cost is conditioned by the assembly plants per final products;

that model and algorithms are worked out which is suited to sort out the better from the direct (without distribution warehouse) and indirect distribution.

In any case only the distribution with shuttle tours is analysed, but it can be under further investigation, what cost reduction can be achieved with the solution of distribution tasks by the help of round tours.

2.1. The total cost function of the model

$$C = C^P + C^T + C^S + C^A + C^{RT} + C^\Psi + C^W + C^D \rightarrow min.\ [€] \tag{2.1}$$

which can be obtained as a sum of the following costs: purchase costs of components (C^P), transportation costs of components (C^T), storage costs of components (C^S), assembly costs (C^A), changeover costs of assembly lines (C^{RT}), costs of standby of lines (C^Ψ), warehousing charges of final products (C^W), and distribution costs of final products (C^D).

We simplify the total cost function (2.1) for the determination of the annual amount of the final products of the individual user and then only the distributional and assembly costs should be considered. Because this paper considers the optimisation of the assembly and the delivery, the warehousing costs of components and final products cannot be taken into consideration and the considered costs are also global and simplified. The above-mentioned cost-components have not to be taken into account by optimisation, because these components are not known by this step of assignment, but we will take these into account in after modules and effects of these components will appear from the principle of feedback.

2.2. The objective function of the assignment in case of the product k

$$C_k^I = C_k^A + C_k^D \rightarrow min.\ [€] \tag{2.2}$$

where C_k^A is the assembly cost, C_k^D is the distribution cost. The matrix Q gives the annual quantity ordered from product k by the user μ. The searched matrix Y shows that the user μ gets the final product k from the assembly plant λ or not. $y_{k\mu\lambda}$ can take values 0 or 1 with the following condition: $\sum_{\lambda=1}^{n} y_{k\mu\lambda} = 1$ (case a), or what part of the final product k will be transported into the end user μ from the assembly plant λ (case b). Conditions are: $0 \le y_{k\mu\lambda} \le 1$ and $\sum_{\lambda=1}^{n} y_{k\mu\lambda} = 1$

2.3. The considered and simplified objective functions in case of the product k

2.3.1. Distribution cost

$$C_k^D = \sum_{\lambda=1}^{n} \sum_{\mu=1}^{v} c_k^D Q_{k\mu} y_{k\mu\lambda} s_{\mu\lambda}\ [€] \tag{2.3}$$

c_k^D is the specific delivery cost of final product k, $s_{\mu\lambda}$ is the length of the delivery route between user μ and assembly plant λ.

2.3.2. Assembly cost

$$C_k^A = \sum_{\lambda=1}^{n}\sum_{\mu=1}^{v} Q_{k\mu} y_{k\mu\lambda} c_{k\lambda}^A \quad [€] \tag{2.4}$$

where $c_{k\lambda}^A$ is the specific assembly cost in case of the product k in the plant λ. The assembly cost by the plants has to be calculated as the weighted average cost of the capacity of assembly lines because we do not know yet onto which line will be assembled it. As well the lower $L^L = \left[\ell_{k\lambda}^L\right]$ and the upper limit $L^U = \left[\ell_{k\lambda}^U\right]$ of annual produced amount of every assembly plant have to be defined. An exclusion matrix has to be defined which gives which plant which product does not able to assemble. Conditions are

$$\ell_{k\lambda}^L \le \sum_{\mu=1}^{v} Q_{k\mu} y_{k\mu\lambda} \le \ell_{k\lambda}^U . \tag{2.5}$$

2.3.3. Objective function

The objective function (3.1) becomes the following formula by the considered and simplified objective functions:

$$C_k^I = \sum_{\lambda=1}^{n}\sum_{\mu=1}^{v} c_k^D Q_{k\mu} y_{k\mu\lambda} s_{\mu\lambda} + \sum_{\lambda=1}^{n}\sum_{\mu=1}^{v} Q_{k\mu} y_{k\mu\lambda} c_{k\lambda}^A \rightarrow min. \ [€] \tag{2.6}$$

It can be seen that each element is a function of $y_{k\mu\lambda}$ The following formula arises if the $Q_{k\mu} y_{k\mu\lambda}$ is put before the brackets:

$$C_k^I = \sum_{\lambda=1}^{n}\sum_{\mu=1}^{v} Q_{k\mu} y_{k\mu\lambda} (c_k^D s_{\mu\lambda} + c_{k\lambda}^A) \rightarrow min. \ [€] \tag{2.7}$$

which is a multivariable linear programming (LP) problem [2] with $n \times v$ pieces (decision) variables ($y_{k\mu\lambda}$), and with $n+v+n \times v$ pieces (limiting) conditions:

$$\sum_{\lambda=1}^{n} y_{k\mu\lambda} = 1, \sum_{\mu=1}^{v} Q_{k\mu} y_{k\mu\lambda} \le \ell_{k\lambda}^{U} \;\&\; 0 \le y_{k\mu\lambda} \le 1 \tag{2.8}$$

The total cost function contains $n \times v \times p$ pieces variables ($y_{k\mu\lambda}$), additionally the number of conditions is $p \times (n+v+n \times v)$, so this problem requires for optimisation 2^{npv} steps (p is the number of final product types, n is the number of assembly plants and v is the number of end users). It results from this that in the event of few plant and user the size of this problem grows exponentially. Some solutions of the (extensive) LP problem with several thousand variables and conditions are the following in [8]: Revised Simplex Algorithm, Product Form of the Inverse, Using Column Generation, Dantzig-Wolfe Decomposition Algorithm, Karmarkar's Method, etc. Because of the large size, the authors worked out two heuristic algorithms (Algorithm A and B) [6] for the solution of the problem.

3. Algorithms of assignment based on simplified cost functions

3.1. Algorithm A

1) Choose a final product and check which end users placed an order for this product. Choose the end user with the largest ordered quantity.
2) Find the plant, whose distribution cost is most favourable having regard to capacitance limits of plants. Take the next user in decreasing order of ordered quantity and choose assembly plant to it.
3) Find the plant, which can assemble this product at the least cost, and look at it has any capacitance, if yes then check the possibility of change for the other users in decreasing order of ordered quantity. Take the next plant in ascending order of assembly cost.
4) Take the following product and repeat step 1.

3.2. Algorithm B

1) Choose a final product and check which users placed an order for this product.
2) Constitute the all possible relations of assembly plant-end user, and we choose them, where distribution costs are most favourable having regard to capacitance limits of assembly plants.
3) Find the plant which can assemble this product with the least cost, and look at it has any capacitance, if yes then check the possibility of change for the other end users in ascending order of distribution cost. Take the next assembly plant in ascending order of assembly cost.
4) Take the following product and repeat step 1.

In step 2 for both algorithms have respect to those relations, where the capacitance

limit of plant enables the ordered quantity of the user to be assembled. In step 3 partial ordered quantities are also changed.

4. Determination of datamodel needful to sensitivity analysis

A program has been implemented using Delphi programming language, which solves the assignment problem using Hungarian method, algorithm A and B showed in chapter 3. The program dynamically handles the number of plans, users and products. The order matrix, capacity matrix, route matrix, distribution cost and assembly cost matrix can be fed into the computer by automatic and manual. The program makes it possible to save, load and print parameters and results.

The basic data are the followings: $n=3$, $v=6$, $p=8$. Values of the matrix Q can change between 1000 and 6000 pieces, the average of these values is about 2000. The data structure is defined by relative variables for the farther easier changes. Accordingly the order matrix is given in next form: $q_{k\mu} = q_0 a_{k\mu}$, where q_0 is the basic ordered quantity, which is independent of products and users, $a_{k\mu}$ is the relative ordered quantity of user μ from product k. The modification of value q_0 can generate quantity change. The modification of $a_{k\mu}$ can create structure change. In like manner the capacity matrix can be written down in next form: $\ell^U_{k\lambda} = \ell^U_0 a^U_{k\lambda}$

$$Q = 2\begin{bmatrix} 0.5 & 2.5 & 0 & 0 & 1.5 & 1 \\ 0 & 1 & 2 & 3 & 2 & 0 \\ 3 & 0 & 1 & 0.5 & 0 & 0 \\ 1.5 & 0 & 0 & 1 & 2.5 & 0 \\ 2 & 1.5 & 1 & 0 & 3 & 0 \\ 0 & 0 & 2 & 0 & 0 & 3 \\ 0 & 0.5 & 0 & 0 & 1 & 2 \\ 1 & 0 & 3 & 0 & 0.5 & 0 \end{bmatrix} \left[\frac{1000\text{pieces}}{\text{cycle}}\right], L^U = 4\begin{bmatrix} 1.5 & 1 & 0.5 \\ 0 & 2 & 2 \\ 0.5 & 0.5 & 1.5 \\ 1 & 1.5 & 0 \\ 1.5 & 0 & 2.5 \\ 0.5 & 1 & 1 \\ 0 & 1 & 1 \\ 1.5 & 0 & 1 \end{bmatrix} \left[\frac{1000\text{pieces}}{\text{cycle}}\right]$$

The values of the route matrix S can change between 20 and 250 km, the average value is about 100 km.

Ratio of the specific distribution and assembly cost: the values of $\delta = c^D_0 / c^A_0$ can be 0.2, ..., 2, let δ be 1 now. The formula respecting the calculation of the specific distribution cost: $c^D_k = c^D_0 a^D_k$, where $c^D_0 = c_0 \delta$ is the distribution basic cost, which is independent of products, $a^D_{k\beta} = a^D_{00} a^D_{k0} a^D_{0\beta}$ is the parameter of proportionality, its

value for average product and vehicle $a^D_{00} = 1$, $a^D_{k0} = 0.8 - 1.2$, $a^D_{0\beta} = 0.7 - 1.3$. The formula of the calculation of the specific assembly cost: $c^A_{k\lambda} = c^A_0 a^A_{k\lambda}$, where $c^A_0 = c_0$ is the assembly basic cost, which is independent of products, $a^A_{k\lambda} = a^A_{00} a^A_{k0} a^A_{0\lambda}$ is the parameter of proportionality, where $a^A_{00} = 1$, $a^A_{k0} = 0.7 - 1.4$, $a^A_{0\lambda} = 0.8 - 1.2$.

We defined a data-structure, which is suitable sensitivity analysis and comparison of the different optimisation methods too. The sensitivity analysis covers only the products, but its data-model is useable for sensitivity analysis of the specific costs. During investigation the matrix Q, L^U and S are fixed. We suppose values of c_0, δ, $a^A_{0\lambda}$ and $a^D_{0\beta}$ to be constant. Let the value of the last two parameters be 1, so the assembly parameter of proportionality is independent of assembly plants and the delivery parameter of proportionality is independent of delivery vehicles, so the vehicle is given. During the sensitivity analysis regarding product costs by both algorithm A and algorithm B only the value of a^A_{k0} and a^D_{k0} change between the above-defined limits (the parameters of proportionality depend on only the products). In the following we complete the comparison of the two methods for different products by a simple example.

$$S = 100 \begin{bmatrix} 0.2 & 0.8 & 1.5 \\ 2.5 & 0.6 & 1.2 \\ 1.8 & 2 & 1 \\ 0.6 & 0.5 & 1.5 \\ 2 & 1 & 2.5 \\ 2.2 & 1.2 & 0.2 \end{bmatrix} [\text{km}]\ C^D = c_0 \delta \begin{bmatrix} 0.6 \\ 0.7 \\ 0.8 \\ 0.9 \\ 1 \\ 1 \\ 1.1 \\ 1.2 \end{bmatrix} \left[\frac{\text{/ piece}}{100\text{km}}\right], C^A = c_0 \begin{bmatrix} 0.7 \\ 1 \\ 1.2 \\ 0.8 \\ 1.3 \\ 0.9 \\ 1.4 \\ 1.1 \end{bmatrix} [\text{ / piece}]$$

5. Sensitivity analysis

5.1. Sensitivity analysis of algorithms concerning to products

By the help of the parameters the values of the specific costs and the orders presented in the objective function can be simply changed, and so parameter sensitivity analysis can be done. The three-dimensional matrix Y is converted in

the interest of the briefer representation, that in the plane the matrix $y_{\mu k}$ can be seen and the values λ are represented smaller numbers. The indexes H, A and B of the matrix Y refer to the methods. This example is solved using the Hungarian method in case of $\delta=1$ we get the following matrix Y in %:

$$Y = \begin{bmatrix} 100^{0^0} & 0^{0^0} & 33^{17^{50}} & 100^{0^0} & 100^{0^0} & 0^{0^0} & 0^{0^0} & 100^{0^0} \\ 20^{80^0} & 0^{0^{100}} & 0^{0^0} & 0^{0^0} & 0^{0^{100}} & 0^{0^0} & 0^{100^0} & 0^{0^0} \\ 0^{0^0} & 0^{0^{100}} & 0^{0^{100}} & 0^{0^0} & 0^{0^{100}} & 50^{50^0} & 0^{0^0} & 33^{67^0} \\ 0^{0^0} & 0^{67^{33}} & 0^{100^0} & 50^{50^0} & 0^{0^0} & 0^{0^0} & 0^{0^0} & 0^{0^0} \\ 100^{0^0} & 0^{100^0} & 0^{0^0} & 0^{100^0} & 33^{0^{67}} & 0^{0^0} & 0^{100^0} & 100^{0^0} \\ 0^{0^{100}} & 0^{0^0} & 0^{0^0} & 0^{0^0} & 0^{0^0} & 0^{67^{33}} & 0^{0^{100}} & 0^{0^0} \end{bmatrix} [\%]$$

This example is solved using the algorithm A, the results are the followings:

$$Y = \begin{bmatrix} 100^{0^0} & 0^{0^0} & 0^{0^{100}} & 100^{0^0} & 100^{0^0} & 0^{0^0} & 0^{0^0} & 0^{0^{100}} \\ 100^{0^0} & 0^{100^0} & 0^{0^0} & 0^{0^0} & 0^{0^{100}} & 0^{0^0} & 0^{100^0} & 0^{0^0} \\ 0^{0^0} & 0^{0^{100}} & 100^{0^0} & 0^{0^0} & 100^{0^0} & 0^{100^0} & 0^{0^0} & 100^{0^0} \\ 0^{0^0} & 0^{100^0} & 0^{100^0} & 50^{50^0} & 0^{0^0} & 0^{0^0} & 0^{0^0} & 0^{0^0} \\ 0^{100^0} & 0^{0^{100}} & 0^{0^0} & 0^{100^0} & 0^{0^{100}} & 0^{0^0} & 0^{100^0} & 0^{0^{100}} \\ 0^{0^{100}} & 0^{0^0} & 0^{0^0} & 0^{0^0} & 0^{0^0} & 33^{0^{67}} & 0^{0^{100}} & 0^{0^0} \end{bmatrix} [\%]$$

Finally it is solved using the heuristic algorithm B, the result is the matrix Y_B:

$$Y = \begin{bmatrix} 100^{0^0} & 0^{0^0} & 0^{0^{100}} & 100^{0^0} & 100^{0^0} & 0^{0^0} & 0^{0^0} & 0^{0^{100}} \\ 100^{0^0} & 0^{100^0} & 0^{0^0} & 0^{0^0} & 0^{0^{100}} & 0^{0^0} & 0^{100^0} & 0^{0^0} \\ 0^{0^0} & 0^{0^{100}} & 100^{0^0} & 0^{0^0} & 100^{0^0} & 0^{0^{100}} & 0^{0^0} & 100^{0^0} \\ 0^{0^0} & 0^{100^0} & 0^{100^0} & 0^{100^0} & 0^{0^0} & 0^{0^0} & 0^{0^0} & 0^{0^0} \\ 0^{100^0} & 0^{0^{100}} & 0^{0^0} & 20^{80^0} & 0^{0^{100}} & 0^{0^0} & 0^{100^0} & 0^{0^{100}} \\ 0^{0^{100}} & 0^{0^0} & 0^{0^0} & 0^{0^0} & 0^{0^0} & 33^{67^0} & 0^{0^{100}} & 0^{0^0} \end{bmatrix} [\%]$$

It is worth analysing, what the results of the three methods (Y_H; Y_A; Y_B) after the assignment of assembly plants to the final product requirements of the end users. Individual elements of matrix Y are labelled different tokens under the followings:

square - the suitable elements of matrix Y_H; Y_A and Y_B are same;
hexagon - the suitable elements of matrix Y_A and Y_B are same;
circle - the suitable elements of matrix Y_H and Y_A are same.

From the ended tokens can be traced, that from the 6x8=48 elements of the matrix:

there is full coincidence (square) by 32 elements, so (32/48)x100=66,67% consist. If these elements are projected for end users (μ^*) and final products (k^*) the number of same elements can be comprised in vectors:

$$\mu^*=[6;6;4;6;3;7]\ \sum_{k=1}^{6}\mu_k^* = 32\ ;\ k^*=[4;3;4;4;4;4;6;3]\ \sum_{\mu=1}^{8}k_\mu^* = 32 \qquad (4.1)$$

13 elements of the algorithm A and B (hexagon), thus (13/48)x100=27,08% is coincided. The results can be also detailed in vectors like previous:

$$\mu^{**}=[2;2;3;1;4;1]\ \sum_{k=1}^{6}\mu_k^{**} = 13\ ;\ k^{**}=[2;3;2;0;2;1;0;3]\ \sum_{\mu=1}^{8}k_\mu^{**} = 13 \qquad (4.2)$$

up to two elements of algorithm A and Hungarian method (circle) (both of them in the event of product 4) so (2/48)x100=4,17% add.

the 3 methods gave several results by as far as 1 element (final product 6 of end user 3).

Testing results are summarised in Table 1. If the Hungarian method is compared to the heuristic methods it can be traced that the Hungarian method is over 11 per cent ($167.910c_0/189.110c_0$=0,8879) better than the algorithm A. It can be seen that the Hungarian method is also over 11 % ($167.910c_0/189.920c_0$=0,8841) better than the algorithm B. The algorithm A approaches to the optimal solution only 0,4 per cent ($189.110c_0/189.920c_0$=0,9957) better than the algorithm B, which arises therefrom, that the value $a_{0\lambda}^{A}$ is supposed constant. Testing results show that the Hungarian method guarantees the optimum much better as opposed to the heuristic methods.

Table 1. Testing results of assignment algorithms per product in costs (c_0)

Pro-duct	Hungarian method			Algorithm A			Algorithm B		
	assembly	delivery	summa	assembly	delivery	summa	assembly	delivery	summa
1.	7700	6900	14600	7700	9660	17360	7700	9660	17360
2.	16000	10780	26780	16000	12740	28740	16000	12740	28740
3.	10800	6560	17360	10800	10480	21280	10800	10480	21280
4.	8000	6030	14030	8000	6030	14030	8000	6840	14840
5.	19500	20400	39900	19500	23000	42500	19500	23000	42500
6.	9000	10800	19800	9000	13200	22200	9000	13200	22200
7.	9800	3740	13540	9800	3740	13540	9800	3740	13540
8.	9900	12000	21900	9900	19560	29460	9900	19560	29460
Total	90700	77210	167910	90700	98410	189110	90700	99220	189920

This table proves the algorithm A and B to give the same results except for one case (final product 4), at the same time all algorithms give the same result just in case of product 7, additionally the result of only the algorithm A analogise with the optimum in case of product 4.

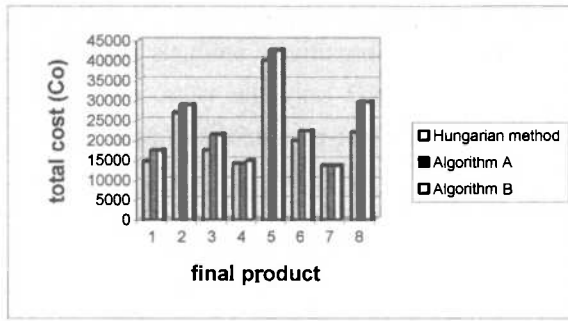


Figure 2. Results of the applied methods for products

The results of total cost are represented in Fig. 3 in the event of different values δ (0.2; 0.5; 0.8; 1; 1.2; 1.5; 1.8; 2) by both two algorithms and Hungarian method.

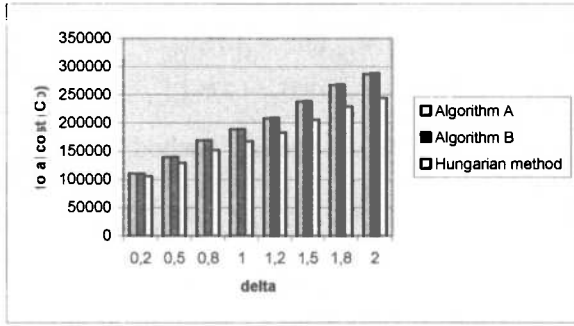


Figure 3. Results of the methods in case of different values δ

In Fig. 3 can be experienced that the given total cost results by the methods linear increase with increase of the value δ. At the same time, if the value δ grows from 0,2 to 2, so it decuples, the total cost will increase 2,6-fold or 2,3-fold. Between two heuristic algorithms there is no great difference in case of different δ, because in case of the specific assembly cost independent of assembly plants did not befall changes in second step of the algorithms, so in fact the final product requirements of end users assigned to assembly plants by only the distribution cost. In all probability, if our investigations are amplified for specific cost dependent on assembly plants, then the algorithm A generate better results, than the algorithm B. It must be observed, there is no represented matrices Y in case of different values δ, but the accordant matrices are all in harmony.

5.2. Sensitivity analysis of algorithms concerning to assembly plants

The basic data are analogised with previous example except that the value of a_{0i}^{A} is not constant, so the assembly parameter of proportionality is conditioned by assembly plants too. During the sensitivity analysis regarding assembly costs by both algorithm A and algorithm B only the value of a_{0i}^{A}, a_{k0}^{A} and a_{k0}^{D} change between the above-defined limits (the parameters of proportionality depend on only products and assembly plants).

$$C^{A} = c_{0} \begin{bmatrix} 0.7 & 0.525 & 0.875 \\ 1 & 0.75 & 1.25 \\ 1.2 & 0.9 & 1.5 \\ 0.8 & 0.6 & 1 \\ 1.3 & 0.975 & 1.625 \\ 0.9 & 0.675 & 1.125 \\ 1.4 & 1.05 & 1.75 \\ 1.1 & 0.825 & 1.375 \end{bmatrix} \left[\text{ / piece} \right]$$

This example is solved using the Hungarian method in case of $\delta=1$ we get the following matrix Y in %:

$$Y_H = \begin{bmatrix} 100^{0^0} & 0^{0^0} & 33^{17^{50}} & 100^{0^0} & 100^{0^0} & 0^{0^0} & 0^{0^0} & 100^{0^0} \\ 20^{80^0} & 0^{0^{100}} & 0^{0^0} & 0^{0^0} & 0^{0^{100}} & 0^{0^0} & 0^{100^0} & 0^{0^0} \\ 0^{0^0} & 0^{0^{100}} & 0^{0^{100}} & 0^{0^0} & 0^{0^{100}} & 50^{50^0} & 0^{0^0} & 33^{0^{67}} \\ 0^{0^0} & 0^{67^{33}} & 0^{100^0} & 50^{50^0} & 0^{0^0} & 0^{0^0} & 0^{0^0} & 0^{0^0} \\ 100^{0^0} & 0^{100^0} & 0^{0^0} & 0^{100^0} & 33^{0^{67}} & 0^{0^0} & 0^{100^0} & 100^{0^0} \\ 0^{0^{100}} & 0^{0^0} & 0^{0^0} & 0^{0^0} & 0^{0^0} & 0^{33^{67}} & 0^{0^{100}} & 0^{0^0} \end{bmatrix} [\%]$$

This example is solved using the heuristic algorithm A, the results are the followings:

$$Y_A = \begin{bmatrix} 100^{0^0} & 0^{0^0} & 0^{33^{67}} & 100^{0^0} & 100^{0^0} & 0^{0^0} & 0^{0^0} & 100^{0^0} \\ 100^{0^0} & 0^{0^{100}} & 0^{0^0} & 0^{0^0} & 0^{0^{100}} & 0^{0^0} & 0^{100^0} & 0^{0^0} \\ 0^{0^0} & 0^{0^{100}} & 0^{0^{100}} & 0^{0^0} & 0^{0^{100}} & 50^{50^0} & 0^{0^0} & 33^{0^{67}} \\ 0^{0^0} & 0^{100^0} & 100^{0^0} & 50^{50^0} & 0^{0^0} & 0^{0^0} & 0^{0^0} & 0^{0^0} \\ 0^{100^0} & 0^{50^{50}} & 0^{0^0} & 0^{100^0} & 17^{0^{83}} & 0^{0^0} & 0^{100^0} & 100^{0^0} \\ 0^{0^{100}} & 0^{0^0} & 0^{0^0} & 0^{0^0} & 0^{0^0} & 0^{33^{67}} & 0^{0^{100}} & 0^{0^0} \end{bmatrix} [\%]$$

Finally this example is solved using the heuristic algorithm B, the result is matrix Y_B:

$$Y_B = \begin{bmatrix} 100^{0^0} & 0^{0^0} & 0^{0^{100}} & 100^{0^0} & 100^{0^0} & 0^{0^0} & 0^{0^0} & 100^{0^0} \\ 100^{0^0} & 0^{0^{100}} & 0^{0^0} & 0^{0^0} & 0^{0^{100}} & 0^{0^0} & 0^{100^0} & 0^{0^0} \\ 0^{0^0} & 0^{0^{100}} & 100^{0^0} & 0^{0^0} & 100^{0^0} & 0^{0^{100}} & 0^{0^0} & 33^{0^{67}} \\ 0^{0^0} & 0^{67^{33}} & 0^{100^0} & 50^{50^0} & 0^{0^0} & 0^{0^0} & 0^{0^0} & 0^{0^0} \\ 0^{100^0} & 0^{100^0} & 0^{0^0} & 0^{100^0} & 0^{0^{100}} & 0^{0^0} & 0^{100^0} & 100^{0^0} \\ 0^{0^{100}} & 0^{0^0} & 0^{0^0} & 0^{0^0} & 0^{0^0} & 33^{67^0} & 0^{0^{100}} & 0^{0^0} \end{bmatrix} [\%]$$

It is worth analysing too, what the results of three methods (Y_H; Y_A; Y_B) after the assignment of assembly plants to the final product requirements of the end users. It can be traced, that from the 6x8=48 elements of the matrix:

– there are full coincidence by 37 elements, so (37/48)x100=77,08% consist.

2 elements of the algorithm A and B, thus (2/48)x100=4,17% are coincided.
up to 4 elements of algorithm A and Hungarian method, so (4/48)x100=8,33% add.
3 elements of algorithm B and Hungarian method, so (3/48)x100=6,25% are coincided.
the three methods gave several results by as far as two elements.

Testing results are summarised in Table 2. If the Hungarian method is compared to the heuristic methods it can be traced that the Hungarian method is over 3 per cent ($147.085c_0/152.070c_0=0,9672$) better than the algorithm A. It can be seen that the Algorithm B is 8,3 per cent ($159.290c_0/147.085c_0=1,0830$) worse than the Hungarian method. The algorithm A approaches to the optimal solution more than 4,5 per cent ($152.070c_0/159.290c_0=0,9547$) better than the algorithm B.

Table 2. Testing results of assignment algorithms per product in costs (c_0)

Pro-	Hungarian method			Algorithm A			Algorithm B		
duct	assembly	delivery	summa	assembly	delivery	summa	assembly	delivery	summa
1.	7350	6900	14250	7525	9660	17185	7525	9660	17185
2.	11200	10780	21980	11200	11480	22680	11200	10780	21980
3.	7825	6560	14385	7500	8160	15660	8175	10480	18655
4.	6350	6030	12380	6350	6030	12380	6350	6030	12380
5.	15675	20400	36075	15250	20900	36150	15675	23000	38675
6.	7400	10800	18200	7400	10800	18200	7400	13200	20600
7.	5075	3740	8815	5075	3740	8815	5075	3740	8815
8.	9000	12000	21000	9000	12000	21000	9000	12000	21000
Total	69875	77210	147085	69300	82770	152070	70400	88890	159290

On the analogy of the former example the Hungarian method guarantees the optimum but this is very time-consuming, on the contrary, the algorithm A and B give only approximation.

The Table 2 proves the algorithm A and B to give the same results in the moiety of cases (final product 1, 4, 7, 8), at the same time all algorithms give the same result in three cases (product 4, 7, 8), additionally the result of only the algorithm A analogise with the optimum in case of product 6 and the result of only the algorithm B analogise with the hungarian method in case of product 2.

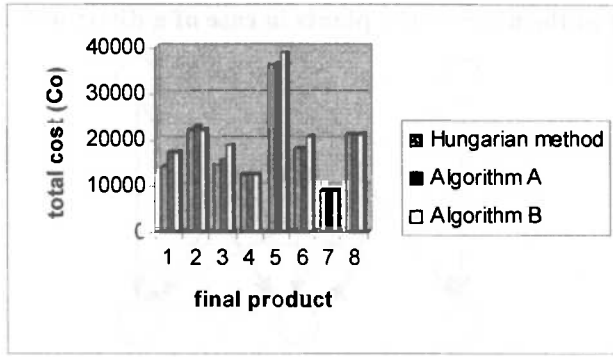


Figure 4. Results of the applied methods for products

The results of total cost are represented in Fig. 5 in the event of different values δ (0.2; 1; 2) by both two algorithms and Hungarian method too.

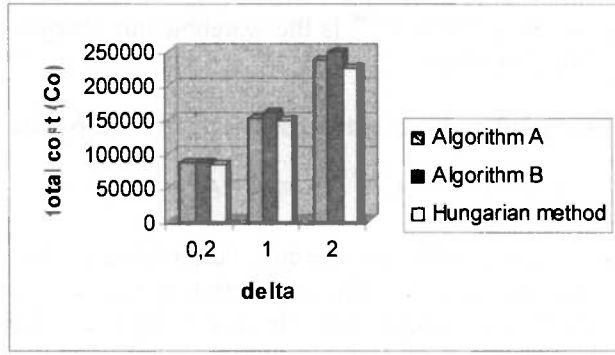


Figure 5. Results of the methods in case of different values δ

In Fig. 5 can be experienced that the given total cost results by the methods nonlinear increase with increase of the value δ. At the same time, if the value δ grows from 0,2 to 2, so it decuples, the total cost will increase $234.840c_0/86.247c_0=2,7$ or $248.180c_0/87.715c_0=2,8$-fold. Between two heuristic algorithms there is a great difference in case of different δ, because in case of the specific assembly cost dependents on assembly plants have already befallen changes in second step of the algorithms, so in fact the final product requirements of end users assigned to assembly plants by not only the distribution cost.

In the next step we analyse that in case of one-distribution warehouse model how modify the simplified cost function (purchase and delivery cost) in accordance with direct delivery. Subsequently we analyse that the indirect delivery when (for what conditions) become necessary and profitable.

6. Assignment of the users to the plants in case of a distribution warehouse

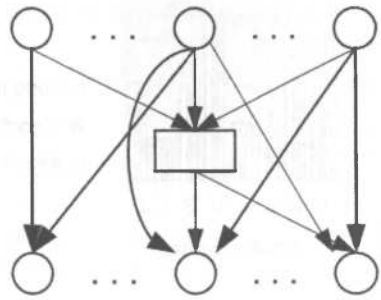


Figure 6. One distribution warehouse model

In case of distribution warehouse (indirect delivery) the costs will be as follows:

$$C = C^{PW} + C^{WW} + C^{DW} \rightarrow min.\ [€] \tag{6.1}$$

where C^{PW} is the purchase costs, C^{WW} is the warehousing charges and C^{DW} is the distribution cost of final products.

6.1. Objective function of assignment in case of product k by indirect delivery

$$C_k^2 = C_k^{PW} + C_k^{DW} \rightarrow min.\ [€] \tag{6.2}$$

Do not have to take account of the storage cost, because the distribution warehouse built that in the purchase cost. By the distribution warehouse the cost function come the following (that is analogous formula arise to the direct delivery):

$$C_k^2 = C_k^{PW} + C_k^{DW} \quad \sum_{\mu=1}^{v} Q_{ku} y_{ku}^W (c_k^D s_u^W + c_k^{PW}) \ [€] \tag{6.3}$$

6.2. Delivery from distribution warehouse and the specific purchase cost

The specific purchase cost from the distribution warehouse in case of product k:

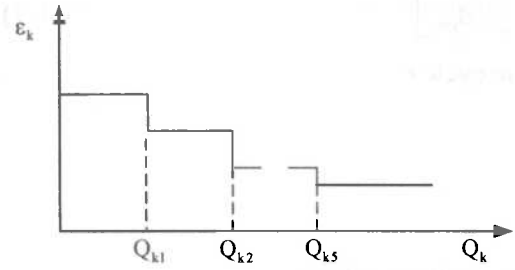
be conditioned by the weighted value of maximal assembly capacity of assembly plants;

$$\overline{c_k^A} = \sum_{\lambda=1}^{n} \frac{\ell_{k\lambda}}{\ell_{k0}} c_{k\lambda}^A \ [€], \text{ where } \ell_{k0} = \sum_{\lambda=1}^{n} \ell_{k\lambda} \tag{6.4}$$

be conditioned by the ordered amount and the ordering incoming time.

$$c_k^{PW} = \varepsilon_k \{Q_k\} \alpha_k \{t^{OW} - t^{OP}\} \overline{c_k^A} \ [€] \tag{6.5}$$

The function $\varepsilon_k\{Q_k\}$:



$$\varepsilon_{k1} = 1,3 \quad 0 < Q_k < 1000$$
$$\varepsilon_{k2} = 1,2 \quad 1000 < Q_k \leq 2000$$
$$\varepsilon_{k3} = 1,1 \quad 2000 < Q_k \leq 3000$$
$$\varepsilon_{k4} = 1,0 \quad 3000 < Q_k < 4000$$
$$\varepsilon_{k5} = 0,9 \quad 4000 < Q_k < 5000$$
$$\varepsilon_{k6} = 0,8 \quad 5000 < Q_k$$

Figure 7. Specific purchase cost

t_k^{OP} is the ordering time of product *k*: $t_k^{OP} = t_0 \Delta_k$, that after ordering incoming into distribution warehouse must be fulfilled the demand by this time and t_k^{OW} is the ordering time of the final product *k* by distribution warehouse. If

$t_k^{OP} > t_k^{OW}$, then $\alpha_k = 1$, there is no overcharge because of in retard order;

$t_k^{OP} < t_k^{OW}$, then the Figure 8. determines the value of α_k.

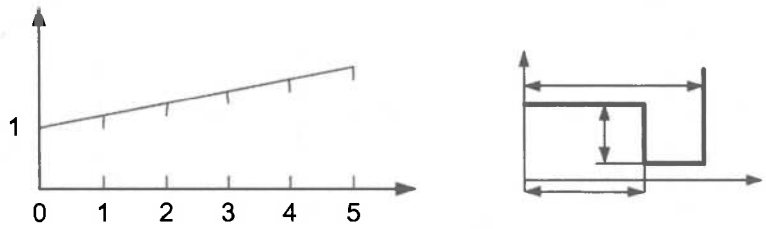


Figure 8. Determination of the value of α_k and the case of $t_k^{OP} < t_k^{OW}$

Remarks:

by $t_k^{OP} > t_k^{OW}$ $\alpha_k = 1$, because the distribution warehouse can get the product optimistically from assembly plants;

by $t_k^{OP} < t_k^{OW}$, the amount Q_k have to storage in the distribution warehouse for term of $t_k^{OW} - t_k^{OP}$

Ordering amount from product *k* by the user μ in the cycle *r*:

$$\overline{Q}_{kr} = \left[Q_{kr1} \cdots Q_{kr\mu} \cdots Q_{krv} \right] \tag{6.6}$$

Ordering time of ordering amount from product *k* by the user μ in the cycle *r* is given in former cycles of the term of ordering appearance (t_k^{LP} arises from this):

$$\Delta_{kr} = \left[\Delta_{kr1} \ldots \Delta_{kr\mu} \ldots \Delta_{krv}\right] \tag{6.7}$$

If the end user μ from the product k in the cycle r

preorder a cycle before: $\Delta_{kr\mu} = 1$;

preorder f cycles before: $\Delta_{kr\mu} = f$.

6.3. Exploration of possibility of direct, indirect delivery

We analyse, that the ordering amount of the end user μ exceed the cycle capacity of the assembly plant λ or not. If the end user μ in the cycle r by the plant λ $Q_{kr\mu} \le \ell_{kr\lambda}$, then $\Theta_{kr\mu\lambda} = 1$ and $Q_{kr\mu} > \ell_{kr\lambda}$, then $\Theta_{kr\mu\lambda} = 0$, where $\ell_{kr\lambda}$ is the free cycle capacity of cycle r in the assembly plant λ. The matrix Θ_{kr} means, which assembly plants can fulfill the product k in the cycle r to the end user μ. The matrix $C_k^l = C_k^A + C_k^D$ which comprises the (direct) assembly and delivery cost is given.

$$\Theta_r^k = \begin{array}{c} \\ 1 \\ \mu \\ v \end{array} \begin{array}{c} \begin{array}{ccc} 1 & \lambda & n \end{array} \\ \begin{bmatrix} & & \\ & \Theta_{r\mu\lambda}^k & \\ & & \end{bmatrix} \end{array} \qquad C_1^k = \begin{array}{c} \\ 1 \\ \mu \\ v \end{array} \begin{array}{c} \begin{array}{ccc} 1 & \lambda & n \end{array} \\ \begin{bmatrix} & & \\ & C_{1\mu\lambda}^k & \\ & & \end{bmatrix} \end{array} \tag{6.8}$$

In every cycle the capacity of every assembly plant from the product k is known.

$$\bar{L}_k = \left[\ell_{k1} \ldots \ell_{k\lambda} \ldots \ell_{kn}\right] \tag{6.9}$$

where $\ell_{k\lambda}$ is the maximal assembly capacity per cycles in case of final product k.

6.3.1. Determination of marginal cases of direct delivery to the end users from the assembly plants

The direct delivery consists in case of that end users (μ) from the product k in the cycle r, where the followings are fulfilled:

in case of end user μ it can be seen the assembly plant in ordering time of the cycle r, where the delivery amount is less, than the free assembly capacity (in case of more solution we have to choose where the total cost C_k^l is minimal);

out of assembly plants which fulfill the former condition, those continue to exist the direct delivery, where the total cost of direct delivery is less than in

case of indirect delivery from the distribution warehouse (if there are more that assembly plants, then we choose that whose total cost is minimal);
if by the analysed order μ there is no direct delivery, because by the ordering amount in the cycle r

- do not have a sufficiency of free assembly capacity of every plant and/or
- delivery from the distribution warehouse is soluble less total cost $C^1_{kr\mu} > C^2_{kr\mu}$ where $C^1_{kr\mu}$ and $C^2_{kr\mu}$ is the total cost of the direct delivery and the delivery from the distribution warehouse in case of final product k in the cycle r by the demand of the end user μ.

6.3.2. Further principles to the algorithm for optimisation of the direct or indirect delivery possibility

the schedule have to be accomplished by cycle time t_0;
in accordance with schedule by the order of the user the sequence per produce:

- we begin that order, where the ordering time t^{OP} is the maximum;
- if in case of more end users there are equal ordering time, then we choose that $C^1_{k\mu\lambda}$ is the minimal having regard to the matrix (6.8/a) and (6.8/b);
- the chosen cost form part of μ and λ is less than the delivery from the distribution warehouse $C^1_{k\mu\lambda} < C^2_{k\mu}$, then the delivery to the end user μ will be from the assembly plant λ in the cycle r. In this case the row μ of the matrix Θ_{kr} is cancelled and by the vector $\bar{L}_{kr}$ the value $\ell_{kr\lambda}$ will be decreased with the measure of capacity decrease, we fix the assembly amount in the matrix G_{kr}, where $G_{kr\mu\lambda}$ is the assembly amount from the final product k in the cycle r in the assembly plant λ for the end user μ;
- if in the above case the: $C^1_{k\mu\lambda} \geq C^2_{k\mu}$, then the delivery comes from the distribution warehouse, the delivery amount of product k can be ordered, in this case we cancel the row μ of the matrix Θ_{kr}, but the vector $\bar{L}_{kr}$ do not change, the product get into the matrix $\pi_{k\mu r}$, which shows the delivery amount from the product k in case of the end user μ in the cycle r;
- we continue this algorithm until all order of cycle $r=1$ have graded;
- we have to determine $\bar{L}_{kr0}$ which is the free assembly capacity in the cycle r, which arises therefrom that $y_{kr\mu\lambda}$ may 1 or 0, that is the demand of one end user will be fulfilled by only one assembly plant.

6.3.3. Additional algorithm for the case b)

- we analyse the assembly capacity vector $\bar{L}_{kr0}$;

we search the minimal element of the matrix (6.8/b): $C^{l}_{k\mu\lambda}$ and we analyse the matrix Y^{a}_{kr}, that the end user μ_0 get where the product k, if

- from the assembly plant λ_0, then jump to the next step and search the next minimal $C^{l}_{k\mu_0\lambda_0}$;
- it is fulfilled elsewhere, then we have to analyse, that pass to the free capacity's debit delivery some of the order from the assembly plant λ_0 the modified cost $C^{l*}_{k\mu\lambda} < C^{l}_{k\mu\lambda}$, or $C^{l*}_{k\mu\lambda} < C^{2}_{k\mu}$;
- in case of the modification we get the lower cost, then the revised matrix Y^{b}_{kr} fix the result (we choose the actual elements of the matrix) and revise the vector $\overline{L}_{kr0}$;
- continue the analysis of the next minimal value $C^{l}_{k\mu_0\lambda_0}$ of matrix (6.8/b) until the every element of the matrix has analysed, the produced modified assignment matrix Y^{b}_{kr} is better than Y^{a}_{kr}.

7. Conclusions and future works

The scientific paper proves that in the network-like operating assembly systems the Hungarian method hard to use by the large-sized problems and the two heuristic methods worked out for optimal assignment of assembly plants to the final product requirements of the end users by simplified cost function to give nearly equivalent result. If the ratio of the specific assembly and delivery basic cost is changed - provided that the specific assembly costs are constant and do not depend on the assembly plants - then the total cost is on the linear increase with the increase of the specific delivery basic cost, but tenfold increase of the delivery cost results in only about 2,5-fold increase of the total cost. If the specific assembly cost depends on assembly plants then the total cost has already increase nonlinear with increase of the specific delivery cost, and tenfold increase of the delivery cost results in only about $224.295c_0/84.745c_0$=2,6-fold increase of the total cost.

In the near future we would like to amplify the described model with more distribution warehouses and to analyse the change of the optimum in comparison to the solution of one-distribution warehouse model by the fulfilment of the final product requirements of the end users.

REFERENCES

[1] CSELÉNYI, J. TÓTH, T.: *Mathematical model for optimisation of a product assembly system integrated by logistics and operating in a network like way*, WESIC 2001 Workshop on European Scientific and Industrial Collaboration, Published by Drubbel Institute for Mechatronics, Twente, pp. 81-92, 2001.

[2] EVANS, J. R.: *Applied Production and Operations Management*, West Publishing Company, Cincinnati. 1993.

[3] OLÁH, B., BÁNYAI, T., CSELÉNYI, J.: *Logistical tasks of co-operative assembly plants*, 3rd International Conference on AED, Prague, pp. 110, 2003.

[4] OLÁH, B., BÁNYAI, T. CSELÉNYI, J.: *Algorithm of optimal assignment of assembly plants and end users within the framework of products in co-operative assembly system*, Miskolcer Gespräche, Miskolc, pp. 145-150, 2003.

[5] OLÁH, B., BÁNYAI, T., CSELÉNYI, J.: *Sensitivity analysis of optimal assignment of assembly plants and end users within the framework of products in a cooperative assembly system*, microCAD 2004, University Press, Miskolc, pp. 97-102, 2004.

[6] OLÁH, B., BÁNYAI, T., CSELÉNYI, J.: *Optimal assignment of assembly plants to the final product requirements of the end users in a cooperative assembly system*, 15th International DAAAM Symposium, Vienna, pp. 321-322, 2004.

[7] TÓTH, T.: *Design and Planning Principles, Models and Methods in Computer Integrated Manufacturing*, University Press, Miskolc, 1998. (in Hungarian)

[8] WINSTON, W.: *Operation Research: Applications and Algorithms*, Aula Press, Budapest, pp. 765-793, 2003.

[9] SMILOWITZ, K. R. DAGANZO, C. F.: *Cost Modeling and Design Techniques for Integrated Package Distribution Systems*, Industrial Engineering and Management Science Working Paper, Northwestern University, pp. 04-06, 2004.

[10] JÜNEMANN, R.: *Planung- und Betriebsführung-Systeme für die Logistik*, Verlag TÜV Rheinland, 1990.

[11] SCHÖNENBURG, E., HEINZMANN, F., FEDDERSEN, S.: *Genetische Algorithmen und Evolutionsstrategien (Eine Einführung in Theorie und Praxis der simulierten Evolution)*, Addison-Wesley, 1994.

[12] SCHÖNSLEBEN, P.: *Integrates Logistik Management, (Planung und Steuerung von umfassenden Geschäftprozessen.)*, 22-5 Springer Verlag Berlin, Heidelberg, 1998.

[13] CSELÉNYI, J.: *Mathematisches Modell und Algorithmus des Montageprozesses für ein elektronisches Produkt*, Magdeburger Schifften zur Logistik, Logistik an der Universität Miskolc, Aktuelle Logistikforschung, Magdeburg,, pp. 35-44, 2002.

[14] CSELÉNYI, J., TÓTH, T.: *Interrelations between Logistics and Production Control*, Proceeding of MANSA'94 9th National Conference of the South African Institute of Industrial Engineers, Manufacturing in South Africa, pp. 365-370, 1994.

Production Systems and Information Engineering
Volume 4 (2006), pp. 147-180

VIRTUAL ENTERPRISE (VE) TOOLS FOR SOLVING COOPERATION AND INTEGRATION ISSUES IN MANUFACTURING INDUSTRY

TIBOR TÓTH
University of Miskolc, Hungary
Department of Information Engineering
Production Information Engineering Research Team (PIERT)
of the Hungarian Academy of Sciences;
toth@ait.iit.uni-miskolc.hu

FERENC ERDÉLYI
University of Miskolc, Hungary
Department of Information Engineering
Production Information Engineering Research Team (PIERT)
of the Hungarian Academy of Sciences;
erdelyi@ait.iit.uni-miskolc.hu

[Received May 2006 and accepted June 2006]

Abstract. Virtual Enterprise (VE) frameworks (computer networks, as well as application systems for production engineering and management) enlarge the application possibilities of information and communication technology. An important novelty of VE is the fact that is offers various common tools for managing business and planning processes, as well as production, supply and customer relation processes including their operations, goals, monitoring and control. Integration and cooperation are the key issues for the continuous improvement and business process reengineering activities of agile manufacturing enterprises.

The authors propose a seven level cooperation model suitable for supporting discrete production engineering activities and processes in manufacturing enterprises. From the year 2000 up to the present two remarkable research and development projects supported by the Hungarian government have been organized by two consortiums with members from universities, an academic research institute, and large and small companies for solving integration and cooperation tasks with computer applications. In the course of the research new models, methods, optimization techniques and software prototypes have been developed. The paper also outlines some promising experiences with application.

Keywords: Virtual Enterprise (VE), integration paradigms, enterprise modelling, Computer Integrated Manufacturing (CIM), aggregate production planning.

1. Introduction

Globalisation is an economic and social tendency of great importance, which has an effect on the world as a whole. As a consequence of its emergence, geographic, national, regional and other boundaries have been disappearing. Globalisation is an objective process, the sources of which can be found in the objective tendencies of sciences, technologies, the world economy and world politics. At present the judgement on globalisation is a contradictory matter. On the one hand, it provides opportunities for faster development of relatively undeveloped countries. On the other hand, dangers of the global extension of crisis symptoms can also be perceived.

The economic organisations of the advanced world have been forced by globalisation to improve their competitiveness and innovation capabilities in a short time. One of the most typical phenomena of this tendency is the wide-spread application of new information and communication technologies in technical and business processes. One of the phenomena of large long-term effects at the turn of the twenty-first century, as reported in expert studies, is the implementation of an integrated, world-wide computer network, the Internet, which has brought the vision of an "information society" close to reality.

The fast progress of the Internet has resulted in a qualitative change in the environmental circumstances of economic processes:

- The technical possibility of accessing information has been increased to a great extent;
- Information is a fundamental value, which does not decrease proportionally when shared;
- Management, business and technical decision-making processes have also drastically accelerated.

These circumstances have a strong effect on business and technical processes, which can be observed in both the operation of economic organisations and their documentation. In developed countries some kind of local information systems have already been built up by the overwhelming majority of economic organisations and are connected to the Internet. Nevertheless, these systems are not integrated to the proper extent, i.e. the facilities for applications related to the organisation as a whole, and those concerning the cooperation with external partners, are missing.

Recognition of the information technology based integration of production processes was first declared by *J. Harrington* [8] more than thirty years ago. The major car factories have been forced to use electronic documentation to a great extent (e.g. for engineering models, NC/PLC and robot programs, operation sheets,

measuring instructions), which leads to the paradigm of Computer Integrated Manufacturing (CIM). Putting this paradigm into practice has yielded considerable results in the automotive industry, but, at the same time, typical failures have also taken place in other fields.

The paradigm of CIM has been transformed to a significant extent during the past 25 years (see Fig.1) [24]. Computerised integration of physical manufacturing systems has been followed by the integration of engineering design and planning systems. In the third step the integration of enterprise management processes has made it possible to manage resources in a completely integrated way. At last, the concept of Virtual Enterprise (VE) can be considered as a result of a full integration of enterprise business environment, clients and suppliers.

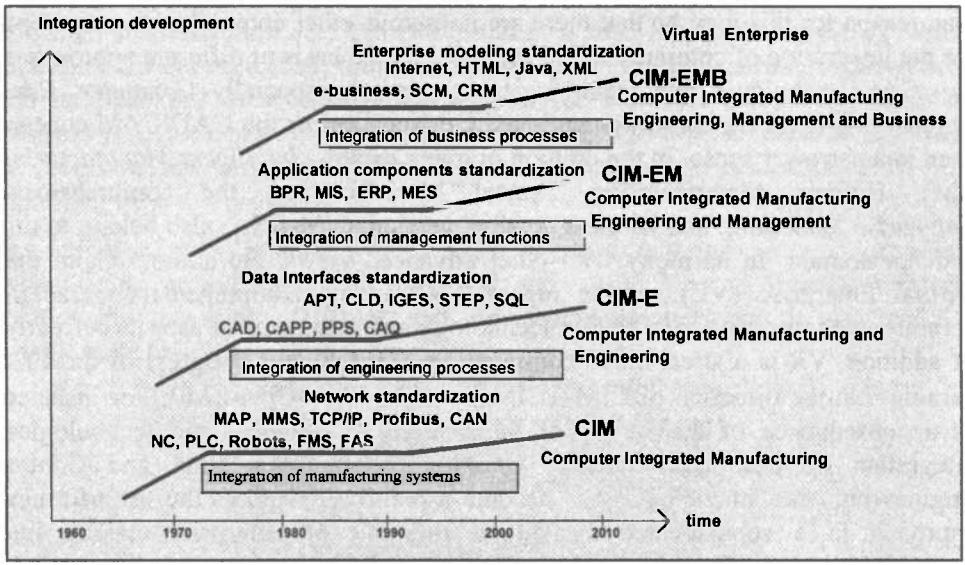


Figure 1. Evolution of the paradigm of CIM

As regards the integration of production processes, the most important factor was the introduction of network standards (MAP, MMS, Ethernet, TCP/IP, Profibus, etc.). Integration of engineering design applications has made data and model interfaces the focus of research (APT, CLD, IGES, STEP, etc.). The accomplished and accepted standards have been widely used. Integration of business processes has been established by the spreading of the large systems with integrated databases (*Enterprise Resources Planning* = ERP; e.g. Oracle, SAP). Integration has been extended by Internet and web technology to the complete business environment of enterprise. It has been proved that enterprise management functions can be integrated with the functions of cooperation in terms of the market, suppliers, clients and partners. The scope of these complex and far-reaching issues

is summarized in some excellent monographs [14], [16], [1]. The book of *Francois B. Vernadat* is especially outstanding [24]. CIM has proved to be an extremely effective paradigm in the course of its fairly long evolution and has kept its importance up to now.

2. Virtual Enterprise

The concept of "Virtual Enterprise" came to be about ten years ago and has spread in the professional literature very rapidly [9]. In the opinion of sceptics it is only a vogue word or, if you like, just a slogan, and the content of the concept, from the point of view of engineering and especially from the aspect of science, is very poor.

One reason for this may be that there are numerous other comprehensive concepts for the integration of computer-aided activities on the basis of different approaches, e.g. CAxx techniques and technologies in general, especially *Computer Aided Engineering* (CAPE), the aforementioned CIM-concept or the CAD/CAM concept used in a narrower sense. In the opinion of many experts, *Intelligent Manufacturing* (IM), Holonic Manufacturing, Fractal Manufacturing, the comprehensive *Enterprise Modelling*, and its most detailed version, CIM-OSA, also belong to this concept domain. In harmony with other advanced views, the authors claim that Virtual Enterprise (VE) can be regarded as a new comprehensive paradigm combining engineering information technology and management aspects currently. In addition, VE is a streamlined continuation of the known progress of the CIM paradigm in the direction of CIM→CIM-E→CIM-EM→CIM-EMB. For instance, as a consequence of the results of the last years, technical and technological integration of Computer Science, Communication Engineering and Control Engineering (the "magic" 3C) has become a reality. It requires the use of a new approach to a sophisticated system of relations of enterprises taking into consideration organisational, marketing and technological interrelationships. VE, in this concern, is a new and comprehensive technical-business paradigm, i.e. a combination of principles, models and methods, which supports more effective and successful management of economic organisations, both for producing and supplying enterprises [6]. As a paradigm, VE promises successes for the companies and institutions functioning and competing in a globalised business environment.

Now, let us attempt to define the VE-paradigm. We may follow two approaches differing from each other to a certain extent:

(1) VE is an occasionally established cooperating system of autonomous organisations (enterprises, affiliated firms) based upon electronic information processing and organisational integration, which makes it possible for the participating organisations to be able to effectively utilise extra resources, and

not only those which are physically available at a particular organisation without significant expansion [6]. In this sense, VE is mainly a paradigm of *integration between enterprises*.

(2) VE is a continuous cooperating system of autonomous functional organisations (settlements, departments, factory units) based upon electronic information processing and organisational integration, which enables participating organisations to operate shared resources in an effective way, without any considerable extension of the resources available physically at the individual organisations [9]. In this formulation VE is mainly a comprehensive paradigm of *integration within the enterprise*.

It is easy to see that VE is a special form of firm operation based on electronic information system and services (the Internet is also included if needed), which facilitates the organisational units of the firm in question, its partner organisations and its customers in accessing data and service resources effectively, initiating business processes, and carrying them out in a safe way with no need to search for or access resources physically. In Figure 2 we attempt to demonstrate the VE-paradigm from this aspect.

In *Jan Hopland's* opinion, "It is clear we are entering an age in which organisations would spring up overnight and would have to form and reform relationships overnight. 'Virtual' had the technology metaphor. It was real and wasn't quite real." [9]

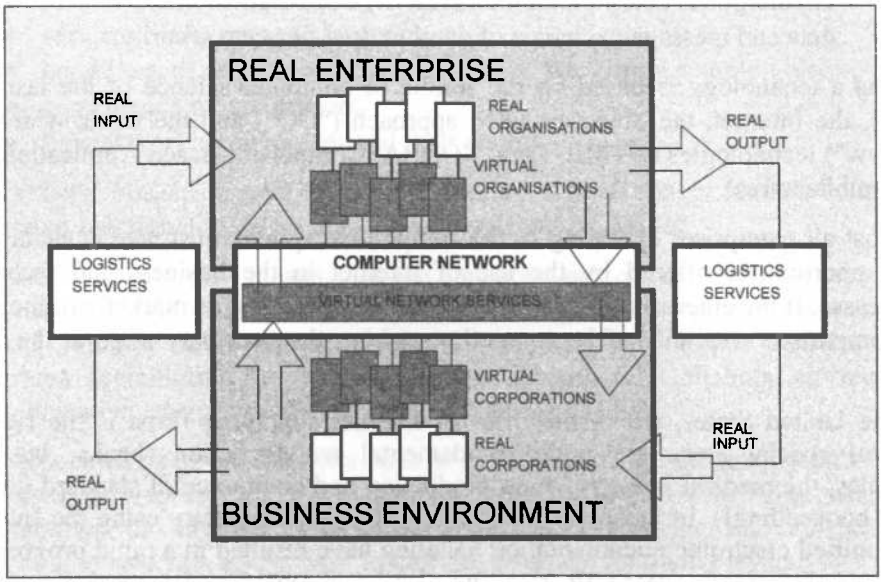


Figure 2. A computer network based functional model of VE

The tasks of establishing a VE can be drafted as follows:

- Integration of business and technology process, including all connections with internal and external suppliers, partners and customers;
- Deeper knowing, modelling and harmonising the processes, revising the constraints and objectives in order to create the conditions for optimization;
- Creating and implementing tools suitable for utilisation by human resources in order to promote real time data exchange, knowledge sharing and cooperation.

The more detailed requirements for VE require that it should support:

- coming into existence of dynamic relationships, groupings and co-operation,
- existence of organizational and geographic dividing,
- different types of communication,
- all kinds of teamwork and undertaking of different roles,
- exchange of information in different forms as much as possible,
- changing the organizational frames and the different life-duration of groups,
- shared use of resources,
- integration of existing working tools and working methods,
- unambiguous determination of authority and responsibility,
- data and message exchange of one-hundred-per-cent security.

VE as a technology is based on the results of computer science of the last five years, the Internet, the object-oriented approach ("OO") and the world-wide-web ("www") technologies (HTML, Java, CORBA, Component Based Applications, 3-tier architectures).

Almost all enterprises operating in the competitive sphere must take advantage of the opportunities offered by the use of Internet in the business and technical processes. If the enterprise does not do so, the risk of losing its market will increase in comparison with that of its competitors taking the necessary steps at the right time.

In the United States, where the offer of Internet supplying firms is the richest, several special areas are under fundamental reconstruction (banks, business supplies, the medical industry, mass production and commerce of standard quality (e.g. bookselling)). In these areas Intelligent Agent Technology using the Internet and unified electronic documentation handling have resulted in a rapid progress. A similar revolutionary reconstruction can also be expected in the areas of supplier and logistic systems as well as in the field of recycling technologies.

3. Integration and Cooperation as the Key to VE

The general architecture of VE is demonstrated in Fig. 2. As can be seen, the physical subsystems (organizations) of the real enterprise are integrated with virtual organizations through a total virtual transformation. Connection of the physical and virtual subsystems is maintained by means of data acquisition sensors, programmable automatons and data input actions of intelligent human agents. Clients of the virtual organizations use virtual services of the suppliers on a network promoted to independent agents. Virtual organizations and agents are present not only within the firm but in the market (business) environment as well Certain differences can only be found in responsibility, authority and the sphere of action of the partners.

The main components of VE are as follows:

- virtual organizations (subsystems),
- virtual services (functions),
- virtual resources and documents,
- virtual working processes,
- a computer network.

A VE is a designed system in which the following entities and attributes are designed:

- classes of the partners participating potentially in the virtual system,
- properties and responsibility of the current classes,
- services of the current classes,
- conditions of coming into being in case of the current sample-objects,
- competence of the partners,
- typical working processes and their results.

The well-established success of the VE paradigm is to be sought in the principle of integrated functionality. What is the meaning of this principle?

Integrated functionality means that the functions are carried out by specialized (optimised) agents in an effective system. The agents, however, are capable of solving complex tasks characterized by temporary or durable collaboration, integrated (combined) by means of interrelationships. Collaboration requires communication, co-operation and co-ordination capabilities.

As a consequence of the characteristics of the VE-paradigm the principles, models and methods represented by it require different tools in the organizational collaboration of different types. It is expedient to arrange the collaboration according to layers. A seven-layer model can mean the following layers:

(1) Collaboration of competitors in the case of contradictory posed interest (open system);
(2) Collaboration of business partners with partially agreeing interest (e-business);
(3) Customer-vendor collaboration with temporary interest (e-commerce);
(4) Collaboration of suppliers with close interest (supplier chain);
(5) Collaboration of affiliated companies with the same interest (virtual factory);
(6) Collaboration of functional organizations under the same conditions (CIM);
(7) Collaboration of operators working on the base of the same plans (CAM).

VE supposes that the same or similar technology can be used for collaboration of the different layers.

The VE-paradigm presumes that the intelligent partners of the collaboration layers mentioned above are capable of following the same samples and principles within the same framework, in their own interest.

VE supposes that collaboration is a behaviour strategy that can be regarded as an optimum strategy for partners giving a positive reply to the initiative. This assumption is well-supported by the results of modern game-theory, established by *John von Neumann*. The latest results of artificial intelligence research also support the fact that optimization of the decision-making series for longer terms has always suggested the conceptions open for collaboration as successful, in contradiction to aggressive strategies.

4. Production Planning in Integrated and Cooperative Environment

The production planning and scheduling (PPS) system is one of the most important functional subsystems of modern digital enterprises. The functions and the services of the PPS systems have significantly changed in the VE environment. In recent years the development of control and product information systems has yielded special application structures. These structures are functionally layered and consist of four horizontal levels with many components in every layer. These layers are the following (see Fig. 3.):

1. Enterprise Resource Planning, ERP
2. Computer Aided Engineering, CAE (CAD/CAP)
3. Manufacturing Execution Systems, MES
4. Manufacturing Automation, MA.

The main horizontal layers are connected by special communication bridges. The production planning and scheduling systems take a prominent role in this situation, forming a functional and integrated „bridge" over all four layers connecting the control and decision functions, as well as the production process controlling and execution systems.

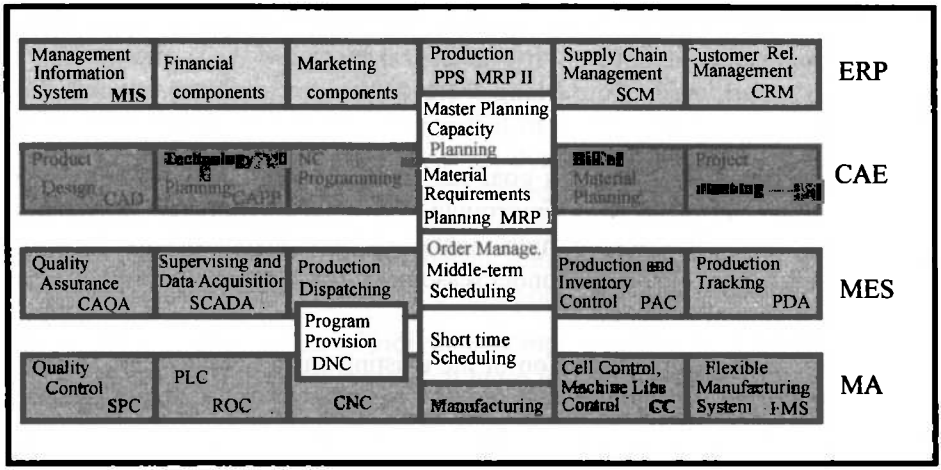


Figure 3. Multi level computer application system for manufacturing

The most widely accepted form of production planning and scheduling systems is the Hierarchal Production Planning (HPP) structure. The greatest advantage of this approach is the realization of the hierarchal modelling which is the only opportunity in case of a large-sized scheduling task (even using the current tools of information technology) to allocate an effective solving system and easy-to-survey Human Machine Interfaces (HMI) to the scheduling tasks.

Production planning and scheduling is one of the most important technical activities for enterprises in the machine manufacturing industries. Production planning is carried out at several hierarchy levels in general. The task of aggregate production planning is to generate quantitative and scheduling data on production in the medium and long run (usually for three months and one year, respectively). The input data of aggregate planning are: the orders based on market demands, specification of the products and technology processes, the internal and external (supply chain) capacities available, as well as stocks [21]. The output is the *aggregate production plan* and the *master schedule*.

In order to summarize the requirements for production planning we have to start from the goals of the business policy of the firm in question. They can be as follows:

- *Improved customer service.* Nowadays this business goal is top priority. Keeping the market and attracting new customers is a precondition of realization of every other business goal. This goal can only be obtained by means of guaranteed quality of products, meeting deadlines and product specifications, as well as offering advantageous prices.

- *Increasing revenue.* This business goal, at the level of production planning and control, requires continuous improvement and control of the macro-parameters of the so-called production triangle (readiness for delivery, stocks level, capacity utilization).
- *Lowering working capital.* This goal can also be achieved by simultaneously and in a synchronized way improving the macro-parameters (production indices) of the production triangle mentioned previously. Meeting deadlines and minimizing the stock level under reasonable constraints have a direct effect on working capital demand.
- *Managing fixed assets.* Utilization of the existing capacities invested in earlier is a fundamental condition to realize effective production capable of ensuring the profit expected. In case of well-proven products, successful accomplishment of the accepted external orders depends on, in most cases, the capacities available.
- *Reducing operation costs.* Under the conditions of the prices agreed and fixed in contracts the net profit can mainly be influenced by decreasing the operation costs and lead times, as well as by optimal utilization of the resources (machines, workers, materials).

It is easy to see that the concept of competitive enterprise can only be defined in a complex manner. The primary business goals can only be influenced through improving the secondary manager (or performance) indices. Effectiveness of production planning and control can only be ascertained after the results obtained in money, i.e. with a delay. Factual influence of the previously made decisions related to scheduling of the production activities (i.e. concerning their quantitative and time-based distribution) can only be ensured by means of a smoothly operating activity-based controlling system.

5. VE and Production of Individual Machines

Nowadays there are numerous examples of the implementation of VE application systems in different branches of industry. The production of individual machines to customer order is one of the most typical application fields. The production of individual machines, machine systems, technological equipment and establishments requires more flexibility and organization and it is necessary to cooperate with the suppliers, partners and customers to a higher degree than usual, both inside and outside the enterprise.

The tasks of aggregate production planning are very different in mass production and in one-of-a-kind production. In large series and mass production the most important viewpoint is to harmonize prediction of the market demand and

utilization of the capacities available. In the case of production of complex products, production planning has to be subordinated to the interest of successful realization of the external orders obtained. Here the demand for flexibility of production is significantly greater than that of mass production and the deadlines are stricter. Production planning has to be dynamic and incremental. This means that aggregate production planning is controlled not by the start of planning periods but by the order-events appearing in changing dates. The new orders necessitate rearranging the work quantities previously allocated to production but this is also limited by the conditions determined by the work in progress.

In the case of the production of individual complex machines and machine systems the project-like approach becomes even more important [15]. Project-like planning became a typical aspect of production planning in a make-to-order machine manufacturer firm. A complex machine of great value, that has been made to order and is to be assembled of numerous parts, can be considered as the project product. Such a complex product is usually made as a special version of another similar machine made and sold in a previous period, i.e. the new machine to be made to order can be considered as a further developed and more-or-less modified version of a similar one previously made and sold in a successful way. The activities and processes of a project are based on experience of previous similar production activities and processes on the one hand, as well as on the unchanged and standardized engineering documentation and specific data of the new project including the new technical documents attached, on the other hand.

The task of production planning is the decomposition of the projects in question into production activities, determining the resource demand (specification, machines, workers, material, energy) for all of these activities. A fundamental feature of the resource model is the available capacity of the given *resource class* depending on the *production calendar*. The resource model used by aggregate production planning is an abstract one and is connected with the high-level activities of production process. Every aggregate activity requires at least one resource suitable for carrying out it.

The typical activities in the practice of an enterprise producing individual machines are the following:

1. Engineering design
2. Electrical design
3. Part manufacturing
4. Component purchasing
5. Mechanical assembly (mounting)
6. Electrical assembly (wiring, mounting)
7. Putting into operation, testing

8. Product delivery.

According to the demands of the production planning we can more or less define activities as we did above. The task of production is the realization of the activities A_i, belonging to the project-set $P=\{P_1,P_2,\cdots,P_j,\cdots,P_J\}$ under determined deadline, capacity and precedence constraints. To the activity type set $A=\{A_1,A_2,\cdots,A_p,\cdots,A_P\}$ a resource type set $R=\{R_1,R_2,\cdots,R_k,\cdots,R_K\}$ is allocated where $K \geq P$ The effective projecting models make it possible to utilize several resources by a given activity, too.

Available capacity is defined as the capability of resource class R_k for doing a certain job, available in the course of the given work-week (the unit of measurement is working hours/week). In the aggregate planning models it is expedient to model the time by means of a series of discrete time intervals δt In most cases the discrete time unit is one work-week. On the discrete time scale let t be the serial number of time interval, i.e.: $t=(1,2,\cdots,T)$. At the planning time horizon the so-called *relative time* is $\tau = t \cdot \delta t$ At the end of the time horizon used in modelling we have $\tau_s = T \cdot \delta t$ Considering this time horizon there is an internal resource capacity for every time unit according to the calendar: $c_k(t)$, $k=1,2,\cdots,K$ The production scheduling model treats the available capacity, after it has been fixed, as a strict constraint.
The production of individual machines can be accommodated to the changing demands of orders only by applying extremely flexible capacities. If there are few orders obtained then the utilization of capacities might be very unfavourable. On the other hand not only the internal capacities should be taken into account but the external capacities based on suppliers as well, in order to fill the external orders obtained. The external capacity $s_k(t)$ $k=1,2,\cdots,K$ similarly to the internal one, is more expensive in general. The external capacity generally is also constrained. Further the planning of the rate of the internal and external capacities depends on the expectations and circumstances of the market as well.

6. Production Planning Scenarios in Individual Machine Manufacturing

In the production of individual machines, production planning can be classified into three different scenario types. They are as follows:

1. ***Project work for tender***
   This is the basic version of project planning. It consists of the analysis of demand (or interest) of the potential purchaser (or customer), a feasibility study of the project and determination of the main data of the project. The

deadline of the project previously accepted has to be determined on the basis of such a model, in which the activities and their work demand are only known at an estimated level.

2. ***Detailed project work***

   This is the detailed, main version of project planning. It consists of all the known phases of product design, technology process planning and production planning on the basis of the customer's order. The project must be included in the actual projects running in the same period. Scheduling of the project is to be carried out by taking into consideration the actual business goals and by fixing the constraints and the objective function.

3. ***Redesign, replanning and rescheduling of projects***

   This is a correcting and modifying version of project planning. It is used when certain modification is needed because of an unexpected factor that has arisen in the course of parallel project execution. The factor can be a change in the business processes, in the production policy or in the engineering specifications. Other reasons are unexpected business events or unexpected events in the technology process, changes in the constraints and objective functions or uncertainty factors.

For project-like production planning another key issue is what we consider to be the optimal production plan. As is known [16], in order to qualify as achieving the production goals three natural state variables (macro-parameters) are needed and they are also sufficient at the same time. These complex state variables can be labeled the "Production Triangle" [22]. They are as follows:

1. The average utilization of resources;
2. The readiness for delivery, i.e. the reciprocal value of the average lead time of the external orders;
3. The average stock level fixed in production.

These complex state variables, of course, are not independent of each other. Any of them can be improved to the detriment of the other two.

In the production planning of individual machine systems the alternative objective functions of a project scheduler suitable for optimisation appear as the special descriptions of the Production Triangle (see: Fig. 4.). The objective functions are:

1. The weighted sum of the external capacities utilized;
2. The weighted sum of due-date tardiness of the projects;
3. The number of projects released at the same time.

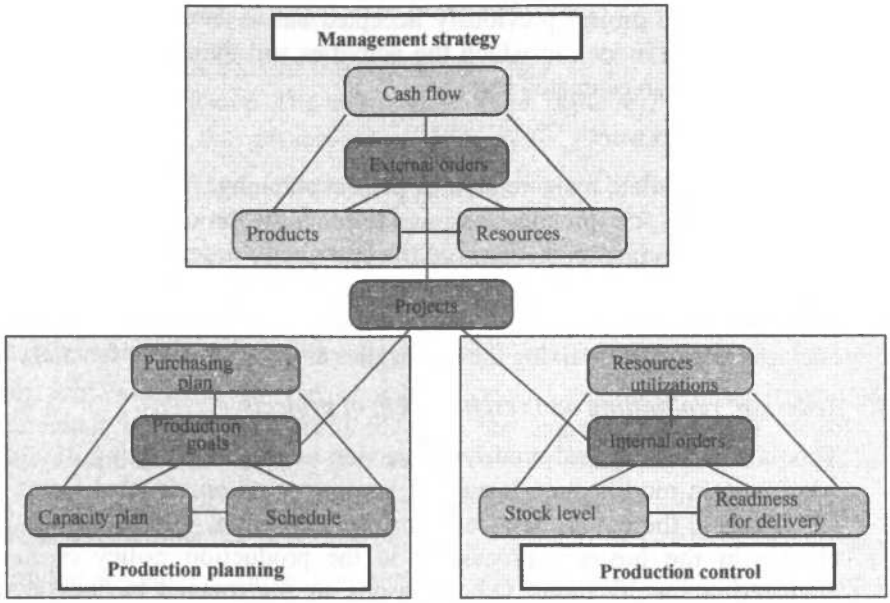


Figure 4. The integrated goals and tools of production management

In project-based production the successful realization of external orders is an essential and primary business goal that has to be supported by the utilization of external capacities as well. However, the maximal utilization of internal capacities is also expected. The most important characteristic of readiness for delivery is to meet the due dates (terms of delivery) fixed in the contract. Any deviation from the term of delivery either may not be allowed (hard constraint) or may be an objective to be minimized.

The task of the scheduler of project activities is to determine those production activities (both in quantity and in time) that meet all the constraints and minimize the objective function in the domain allowed.

The first objective function of project work gives a good solution, typically, in the case of overloaded resources. If the jobs required by the actual order-book of the firm cannot load the resources in the planning period then the value of external capacity demand is equal to zero and there can be numerous scheduling solutions suitable for meeting the constraints. In many cases it is difficult to decide if improving the stocks level or improving the readiness for delivery should be the objective targeted at such a time. The conflict between the short-term and long-term goals makes the situation even more complicated.

The philosophy of the schedulers used at present is, in general, that the constraints are the important ones; they have to be met by all means. There can also be several

production plan solutions (schedules) meeting all the constraints. It is possible to select the most suitable of them on the basis of heuristic considerations. Of course, an exact optimum is out of question here. The larger the number of permissible solutions, the more robust the optimum is, and the less sensitive it is to the changing circumstances.

7. The Role of Rates in Production Scheduling Models

Production processes are typically cumulative ones. This explains the important integrating role of the rate-based state variables in the planning and controlling of production processes.

These kinds of state variables are state characteristics concerning a time unit.

Some typical examples:

- *Material removal rate , Cutting rate* (cm^3/min)
- *Operation rate* (pieces/min)
- *Production rate* (working hours/time unit)
- *Activity rate* (working hours/time unit)
- *Demand rate* (number of products/time unit)
- *Capacity rate* (specific source work volume/time unit).

In order to control the production processes the rates must be controlled in time. The production scheduling plans specify the work volume engaged capacities and their dependence on time with which the production processes can be realised successfully or optimally in some sense, of course meeting the described requirements. The characteristics of rates have a great influence on the scheduling models and the methods needed to solve them. Figure 5 represents the four basic types of production rates.

The four basic types appear in the production scheduling model in the following manner:

1. The typical shop floor level scheduling model of part manufacturing processes. The combinatorial optimization task is NP hard. The solution can be achieved by heuristic considerations, constraint programming or a searching AI procedure.
2. The scheduling model in large series and mass production or in continuous production (for instance in the chemical industry). The extent of the series (production mass) varies. The optimization process can be carried out by heuristics or by the method of hybrid dynamic activity control.

3. The flexible model of low level Production Activity Control. The time period of operations can be controlled in a limited way (*Process Management*). There is only heuristics (OPT).
4. The scheduling model of high level project-like activity. It can be modelled by methods of large size from the field of the mixed integer linear programming. The solver has to meet high requirements.

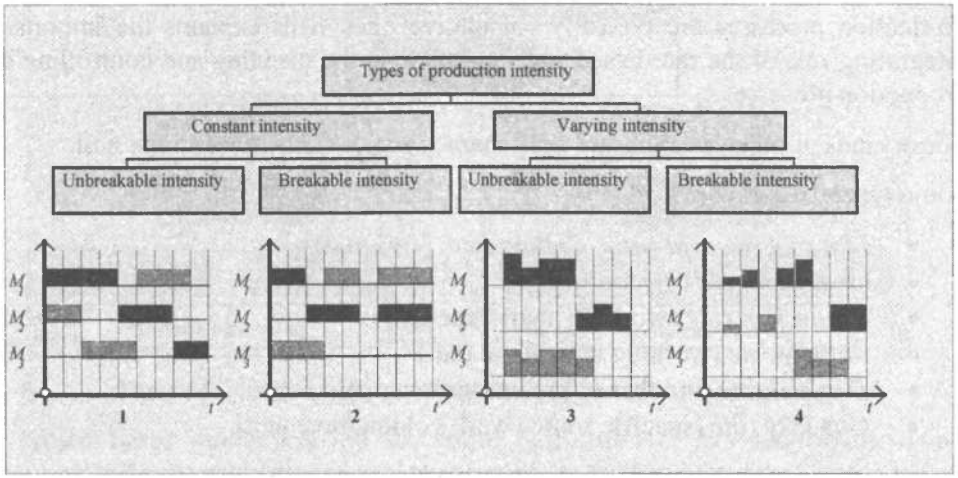


Figure 5. The effect of production rate upon the scheduling model

In the Department of Information Engineering at the University of Miskolc, scientific investigations have been carried out for a long time related to the role of production rate type state variables at the different hierarchy levels of production management, from the well-known material removal rate (MRR) to the rates of the main production activities.

Based our experience in the control of production processes, it is clear that the *process rate* (process intensity) is of great importance. If we consider production control as a closed control loop then the basic signal of control is the production rate. The rate of production processes can be measured in the measuring unit [working hours used/time unit] in the most general manner. At the level of operations the production rate depends on the *technological rate* that can be measured in measuring units [number of products/time unit], [removed material volume/time unit]. In cutting technology processes where the finishing processes are of great importance, the measure of rate is [machined surface/time unit] and in case of chemical technology processes [processed mass (volume)/time unit] [18].

The technological rate for cutting, as a state variable in time, can be defined in an indirect way:

$$\int_0^t Q(\tau)d\tau = V(t) \quad \text{i.e.} \quad Q(t) = dV(t)/dt,$$

where $Q(t)$ is the cutting rate changing in time and $V(t)$ is the material volume removed until the time t. In case of technology process planning it is expedient to use the cutting rate related to one revolution of the main spindle.

Then $Q(t) = A(t) \cdot v_e(t)$, where $A(t)$ is the momentary effective cross section of cutting and v_e is the feeding speed. This equation can also be used in case of multiple-edged tools (see Fig.6.).

The cutting rate defined in this way is a suitable tool for optimization of cutting operations. In planning and production control the average rate $\overline{Q}$ is advantageously used for a given operation or operation element that makes it possible to estimate the primary time of cutting (the machining time) t_m; $t_m = V/\overline{Q}$. Here V is the material volume removed in the given operation.

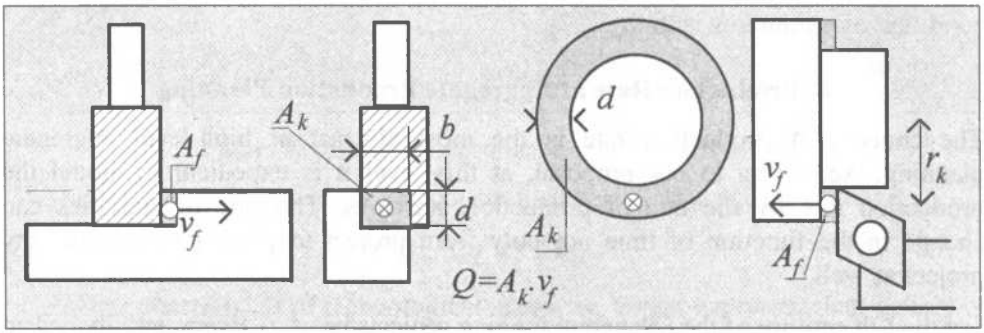


Figure 6. The technological rate for cutting

The term of cutting rate is suitable for formulating the new model of optimal technological data. The cost of cutting operations can namely be expressed as a function of cutting rate. Some factors of this function decreases by increasing the rate (for example time proportional costs) while other factors increases as well (tool cost, loss because of rejects, etc.). The average optimal rate – in the sense of minimum cost – can be determined and this makes it possible to increase the number of alternatives in production control decisions (especially in case of NC machines).

The rate of technological operations is the reciprocal value of the operation time, its measuring unit is [1/min]:

$$q_0 = \frac{1}{t_0} = \frac{1}{t_m + t_a}, \text{ if } t_m >> t_a \quad \text{then} \quad q_0 = \frac{\overline{Q}}{V}.$$

Here q_o is the rate of operation, and t_0 is the operation time, which can be approximated by the machining time in case of short auxiliary time t_a.

In general, part manufacturing demands a consecutive series of operations, and therefore the average rate of part manufacturing, referring to work pieces or series, is an aggregate production characteristic:

$$q_p = \frac{n_p}{t_f} \text{ [pieces/min], where } t_f = \sum t_{prep} + \sum t_o + \sum t_w$$

Here n_p is the lot size, t_{prep} is the preparation time and t_w is the time of the work piece spent in waiting. Summing has to be extended to all the operations of the series executed so far. The average rate of part manufacturing referring to work piece series plays a great role at the shop-floor level and in medium-term scheduling where the equilibrium of demand rate and production rate is the condition of production stability [17].

8. Production Rate at Aggregate Production Planning

The concept of production rate is the most abstract at high-level aggregate planning. According to our proposal, at this level it is expedient to model the production rate by the rate of production activities. The rate of activities can change in the function of time not only from project to project but within any project as well.

Let the *i*-th activity of the *j*-th actual running projects be A_i, Every activity has an earliest starting date and a latest completion due date (deadline). The former is determined by the precedence of the activities and the latter depends on the project deadline. Let us denote these two dates with e_i and d_i, respectively. Both dates will be determined in the course of aggregate planning. Any project means a defined product to be manufactured, the technology process planning of which gives the engagement $r_{i,k}$ [working hours] demanded by the project activity to the resource used by the activity, in a cumulative way. At preliminary planning for a bid this, of course, can only be based on engineering estimations, however after having carried out detailed process planning it can be calculated from the technology process plans. For a given activity one or more resource engagements can also be allocated but this fact will have importance in the planning phase of the capacity-constrained production scheduling only.

We can give an implicit definition for *activity rate* in the case of aggregate planning:

$$\sum_{t=e_i}^{d_i} q_{i,k}(t) \cdot \delta t = r_{i,k}$$

Hence, the activity rate $q_{i\,k}(t)$, changing in time discretely, is the activity concerning the time unit (for example a week) demanded by the i-th project, which loads the k-th resource. The "stepped" function $q_{i\,k}(t)$ is called the profile of activity (Fig. 7.). For the profile numerous constraints can be defined, which must be taken into consideration in the course of production planning and scheduling.

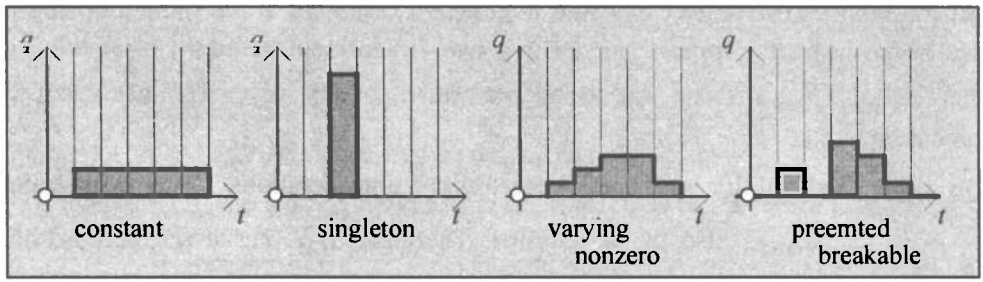


Figure 7. Discrete time profiles of production activities

The rate of a project activity can be constrained from below and from above:

$$q_{mi} \le q_i(t) \le q_{Mi}.$$

The upper constraint is of technological character, which expresses that the rate of the given activity cannot exceed the maximum value even if there were free capacity available for this purpose. (It is not possible to design or to assemble a machine with optionally great rate even in the case when working capacity is available.) The lower constraint can express the fact that if we have already started with a certain activity then a minimum expenditure is needed for it in every time interval. If $q_{mi} = 0$ then the activity in question can be interrupted, otherwise it cannot be done. A correct modelling of the rate constraints is of fundamental importance for the scheduling of projects because the model of the scheduler is obviously sensitive to the right boundaries.

From the viewpoint of a feasible scheduling plan the maximal rate allowed q_M is a key issue. The maximum value of the rate can be constrained as follows. Let the "time window" of the i-th activity be $\Delta t_i = d_i - e_i$. The activity rate has a

minimum (and maximum at the same time) during which the activity can be executed keeping this rate every week:

$$1 \le \left\lceil \frac{r_i}{q_{Mm}} \right\rceil \le \Delta t_i \quad \text{from which we obtain} \quad q_{Mm} \ge \frac{r_i}{\Delta t_i}.$$

This means a lower constraint for the maximum of the activity rate, i.e. if the last inequality is not satisfied the project deadline cannot be met even with constant working without any interruption. Another constraint for q_M can be originated from technological features of the competent resource of the activity. For each activity a minimum time interval for completion of the activity can be determined according to experience. Hence the maximum of the activity rate cannot exceed this number even there were greater parallel capacity available. This limit can depend both on the project type and the utilized resources at the same time and it can be given for the project planner in a two-dimensional table (p_{ik}). Obviously, the relation $p_{ik} \cdot \delta t \le \Delta t_i$ has to be performed, otherwise the project deadline cannot be met.

On the basis of the aforementioned considerations, the constraint $\frac{r_i}{p_{ik} . \delta t} \ge q_{MM} \ge q_M$ is also to be satisfied. Therefore q_M has to be kept within bounds:

$$\frac{r_i}{\Delta t_i} \le q_{Mm} \le q_M \le q_{MM} \le \frac{r_i}{p_{ik} . \delta t}.$$

We showed the limits for the maximum value of the rate related to project activities. (see Fig.8.) We can define the relative value of the rate by $x_i(t) = \frac{q_i(t)}{r_i}$

Here x_i means the actual fraction of the rate and can be expressed in percent. The relative value of the maximum rate allowed can also be defined in a similar way, as follows: $a_{iM} = \frac{q_{iM}}{r_i} \cdot 100\,\%$.

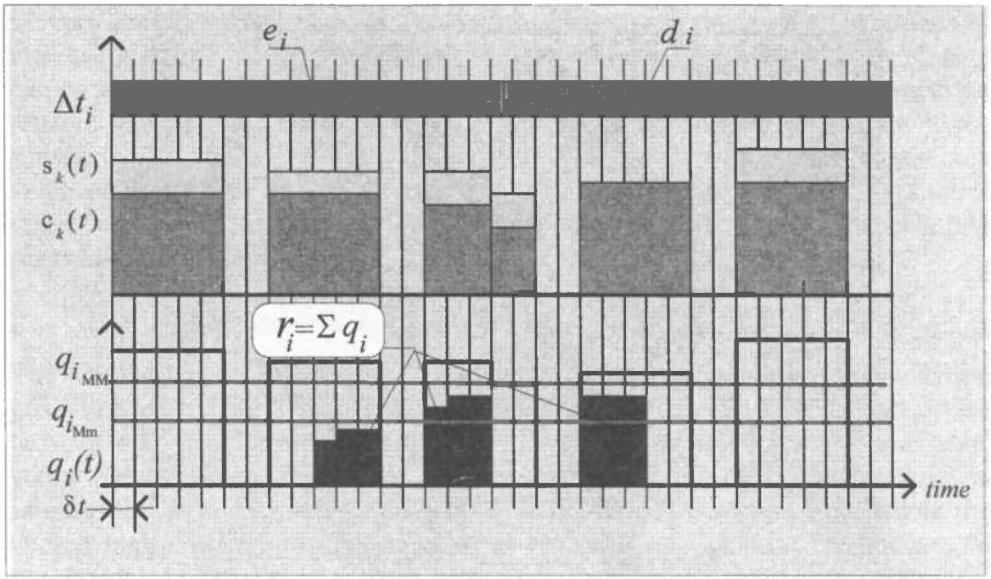


Figure 8. Constraints for activity rate and capacity

Fine modelling of the loading profile of project activities is a problem treatable in a more complex way.

The profile can be modelled with a graph or a conventional *Gantt*-diagram in a rough way only, because these graphic tools only concentrate on the time conditions and partial deadlines. As regards the resource demand of the project, only a constant or periodically constant rate can be modelled.

Demonstration of the loading profiles with a set of time-functions is better but there exists the danger of it being not easy to survey (see Fig. 9).

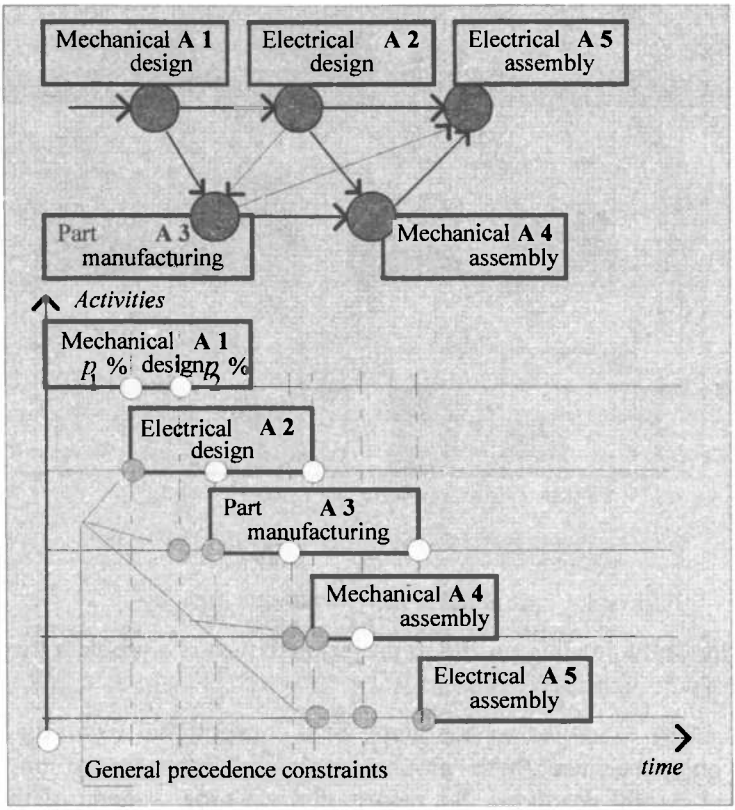


Figure 9. Modelling of precedence constraints for different activities

There are precedence constraints between the activities. On the one hand, they originate from the technology process itself; on the other hand, they can be deduced from business goals and considerations between the projects. Precedence, for instance, can be described by a directed acyclic graph $D=(N,A)$. The simple or special precedence $A_i \rightarrow A_j$ means that activity A_j can only start if A_i has ended. It can also be interpreted as a more general precedence $A_i \overset{p}{\rightarrow} A_j$, which means the activity A_j can only start if A_i was completed to p %. In aggregate production planning the latter is typical. For modelling precedence a binary variable can be allocated to every activity fraction, $x_i(t) \Rightarrow z_i(t)$, which shows whether the rate is allowed in the given time interval. The function $z_i(t)$ for activity A_i is a "stepped" function, which separates the interval $\Delta t_i = d_i - e_i$ into two sections. One of the sections is allowed for the activity, the other is not.

Further complicated constraints can be specified, for instance, for the gradually increasing profile of the rate. In case of two activities depending on each other it can be required that the relative rate of the dependent activity cannot exceed the relative rate of the other one. This might mean a cumulative restriction, if the summed value of a rate within a limited period cannot be greater than the summed value of the rate of any previously started activity. In such cases the activities "feed" each other, for instance, the delivery of the part drawings and scheduling plans in folders is a precondition of the beginning of part manufacturing.

9. Rate Based Optimization Model for Production Planning

The first basic task of aggregate production planning is to choose those production goals that are to be achieved in the planning period. Planning is carried out on the basis of market predictions, the orders of customers and the capacities available, taking into consideration the specifications and quantitative data of the products to be manufactured. The second problem of production planning is to schedule the chosen high-level production activities in time and in a quantitative manner. In project-like production planning this can be done by choosing those specific production loads (i.e. discrete production rates) that appear on the resources in the chosen planning horizon.

These tasks can also be solved in several ways and the task of computerized production planning applications is to support this solving process. In general, aggregate production planning models yield constrained discrete optimum problems, and the solution process of these problems is supported by the results of Operations Research.

Considering the fact that there are effective computer solvers suitable for solving linear programming problems, it is worth investigating those models of the aggregate production planning that can be solved by these solvers. The problem has been investigated by a research consortium consisting of five Hungarian partners for the last two years: the Computer and Automation Research Institute of the Hungarian Academy of Sciences (CARI-HAS), Budapest University of Technology and Economics, the University of Miskolc and two firms from the competitive sphere. Several models of the joint research work show promising results [5],[12].

The next model was developed for supporting the aggregate planning activities of a factory manufacturing individual machine systems. The model is elaborated by the researchers of CARI-HAS collaborating with research workers of Budapest University of Technology and Economics and the University of Miskolc within the framework of the research project Digital Factory.

The relative production rate as a state variable is defined by $x_i(t) = q_i(i)/r_i$ and means the loading fraction of the activity in the t-th time interval. It is obvious that

$$\sum_{t=e_i}^{d_i} x_i(t) = 1 .$$

The relative production rate of project level can have a value between the limits 0 and 1. For the sake of simplifying the model let us assume that every activity can be interrupted and therefore any value of $x_i(t)$ can also be equal to zero.

Let the goal of business policy be the maximal utilization of internal resources. In this case the rate of utilization of external capacities is to be minimized so that the objective function of the project scheduler is to minimize the utilization of external capacities. Hence, the objective function is:

$$\sum_k [w_k \sum_t y_k(t)] \Rightarrow \min ,$$

where $y_k(t) = \max[0, (\sum_i q_{i,k}(t)) - c_k(t)]$ is the rate of external capacity used in the t-th time interval and w_i is the weighting factor expressing the properties of the resource in question.

The task of the project-based production scheduler is to determine those relative production rate fractions $x_i(t)$ and external demands $y_k(t)$ which meet all the constraints related to times, capacities and sequences, as well as to minimize the objective functions.

The constraints are as follows:

$x_i(t) = 0$ if $t \le t \le e_i$ and $d_i \le t \le T$ (The activity has to be completed in the given time window.)

$\sum_{t=e_i}^{d_i} x_i(t) = 1$ (Every activity has to be carried out entirely);

$\sum_{i,k} r_{i,k} . x_{i,k}(t) \le c_k(t) + y_k(t)$ $1 \le t \le T$ (All the demands are covered by the internal and external capacities);

$y_k(t) \le b_k(t)$ (The external capacity is also limited);

$x_i(t) \le a_{iM} . z_i(t)$ (The rate of activity cannot be greater than that allowed and it can only be different from zero in that interval where it is allowed by the precedence control condition).

If there is no solution of the planning task with the given data then it is the task (or decision) of the production engineer to intervene interactively in the computer-aided planning process. In order to solve the problem it can be expedient to slacken certain constraint(s) or to define a new production planning task by changing the demands of the project.

The aforementioned strategy of project work results in a solution most typically in resource-overloaded cases. If the task is not resource-overloaded then the value of the objective function is obviously zero and there can be numerous solutions for meeting the constraints. At that time the task of the project scheduler is to suggest those solutions from the possible and allowed solutions considering which profile of rate changing is the most suitable for meeting the requirements of the production goal.

If it is not important or not possible to take the external capacities into consideration, then the objective function can be an expedient function of the deviations from deadlines. This function, for instance, manages the exceeding of deadlines stricter than completion before due date, because the latter only increases the stock level.

It is clear that project activities in several aspects differ from the activities of part manufacturing at the shop-floor level. In general the operations as activities cannot be interrupted, and in scheduling processes it is not common to define operation by a changing rate. The precedence constraints are stricter and the scheduling plan can be well represented by a Gantt-diagram.

The nowadays commonly-used applications for aggregated production planning separately manage the tasks of Material Resource Planning and Capacity Planning in order to cope with the difficulties of hard calculation. The result is an aggregated Master Schedule which guarantees meeting the constraints even in the case of large-sized problems, however it gives little information about the alternatives and possible solutions.

10. Estimating Procedures at Similarity Based Production Planning

Important tools of the aggregate production planning are those estimating procedures that estimate the probable structure of activities and the utilization of resources on the basis of the similarity of the products. Under such circumstances the modular structure of these products, the principles of Group Technology (GT), and similarity-based estimations can have an important role.

Machine manufacturers meet the task of aggregate production planning in the period of tender, when obtaining the order is an outstanding business goal. If the production plans of the product meeting the requirements of customer are not available then inserting the project into the running tasks requires careful aggregate planning that includes planning alternatives of the "What would happen if..." type as well. Here the most important factors are a well-established delivery deadline and a reliable estimation of the probable capacity overloading.

In the course of the realization of a project two different hierarchies have to be taken into consideration:

- the structural hierarchy of the product (complex machine) constituting the base of the project in question,
- the technological hierarchy realized in the manufacturing process.

Structural hierarchy reflects the physical reality of the product, as well as subordination of the main machine units adequate to the major functions. We assume that a product can be dissected into four hierarchy levels at the very most:

(1) the complex (complete) machine

(2) a machine unit

(3) an assembly unit

(4) a part group.

We hold it natural that only those projects that belong to the same structural hierarchy level can be compared to each other.

In the hierarchy of the production process we allow two levels, namely

1. the level of aggregate activities of a complete project,

2. the level of operations of the production activities.

The first step of similarity-based production planning is to allocate the project to be planned to a product hierarchy level. After this, at the given hierarchy level, we select the similarity projects from the projects previously completed. This is an algorithm consisting of several steps. We make a list including the operations executed in the projects, the utilization of capacities, and the times for planning, manufacturing and assembly. The operation set obtained in this way can also be supplemented with several specific operations if needed for realization of the new project, and if they have not appeared in any similar project so far. So we thus obtain a possible set of operations. We allocate the operation times occurring already in the completed projects to these operation sets in a primary table. The similarity based selection, after all, will be executed by means of a secondary table that qualifies the similar projects on the basis of the occurrence of operations and

the operation times within defined tolerances. On the basis of activities of the projects selected in this way we can get a fairly good estimation for the production time requirements of the project activities planned.

11. Application Experiences in Industrial Environment

As we have already touched on the fact in Section 9, in 2001 a three-year long research and development project started in Hungary. The project, entitled Digital Factory, was led by CARI-HAS, and several departments of universities and industrial factories participated in it. One of its clusters aimed at the elaboration of a large-sized project scheduling system, which can be applied in an industrial environment [15].

During the development and application process of a technical system several well-proved methodological results were used (*Enterprise Modelling and Integration*). These results are based on the experiences of several great paradigms, for example CIM, Concurrent Engineering, Virtual Enterprise, TQM, BPR, IMS, etc.

One of the best summaries (but not the only one) of the development methodology is the reference model CIM OSA [24]. In the course of our work this framework system was considered as a reference model. This framework is suitable for giving a clear survey and nodes of the determination and solution of the partial tasks in the course of research and development process (see Fig.10.).

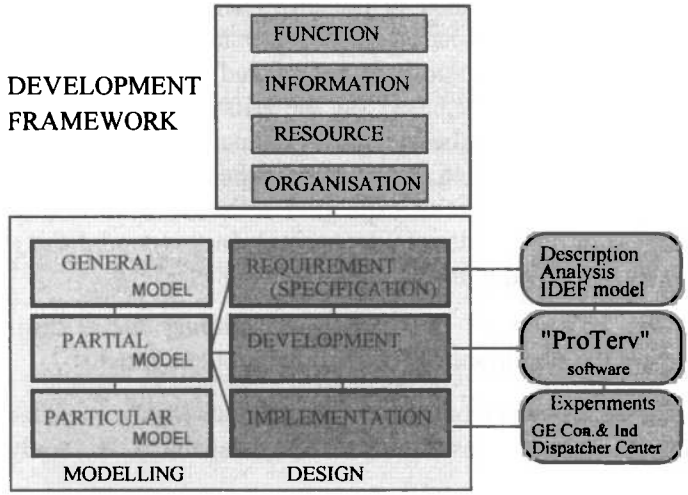


Figure 10. Application of the development framework CIM OSA

The *general* requirements of the integrated production planning and scheduling system were based on the results of the basic research. The main requirement is modelling the activities of the projects competing for the finite resources, as well as

the scheduling of the resource loading capacities of these projects considering the constraints and the objective functions for optimization.

The *partial* requirements were determined by the production tasks of a factory making individual machines. The *particular* solution aimed to satisfy the special requirements of the machine factory *GE Consumer & Industry.*

The requirement analysis was followed by the system- and program planning. The next step was the implementation of the project planner and capacity scheduler software application ProTerv. We had to find a functional and mathematical model suitable for the functional requirements and at the same time we had to elaborate a solver algorithm. In the course of the information planning it was necessary to develop an application data model and an integrated input/output model; both of them in several iterative steps. The organization tasks determined the Human Machine Interface (HMI) of the production planner and scheduler systems and the different ways of operation and utilization. The most important software components were the efficient solvers (CPLEX and ILOG Solver) and the program developing tool (Windows.NET).In order to formulate the functional requirements of ProTerv we elaborated a detailed text-centred descriptive model, an SADT-type hierarchal graphical model and a mathematical model for presenting the operations research task.

When planning and scheduling a project in the production of individual machines the requirements are to be allocated to the characteristics of the production. In production planning the project-based model provides the most advantageous conditions. Decisions of the medium run aggregated production plans (projects) and their scheduling in time are based on the existing and expected customer orders, the running projects and the available production and supplier capacities. The primary goals are the following: realization of the obtained orders meeting the deadlines, utilization of the production capacities at the highest level, minimization of the quantities of supplier's orders (outsourcing). In this model the set of high-level activities to be scheduled typically consists of 4-8 elements.

For improving the solution of the project scheduling tasks the production management stresses the importance of the following requirements:

- Increasing the efficiency of the management decisions;
- Enlarging the set of possible decisions, capability of analysis of the alternatives;
- Reducing the production costs, increasing effectiveness and profit;
- Meeting deadlines in a safer way;
- Decreasing the risk of erroneous decisions.

At the project planning process the roles and tasks of the material plans, capacity plans and production scheduling plans can be easily distinguished. The requirements of the project scheduling method of new approach are given below.

The scheduling has to cope with the problems arising from the partiality of scheduling, the uncertainty of data and events, as well as the periodic validity of plans (see Fig.11.).

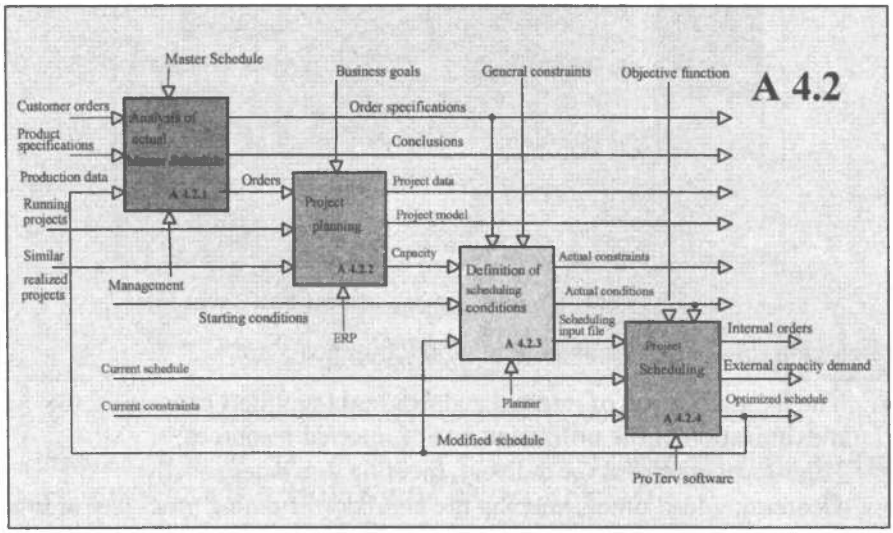


Figure 11. A SADT-type graphic functional model element of ProTerv

As far as possible the production planning, the engagement of capacities and high-level scheduling are to be managed together, although these tasks are traditionally separated in ERP systems.

It has to take into consideration equally the economical and engineering aspects of production, the duality of constraints and manager goals, the demand of the profiles of scheduling plans.

The quick and clear definition of project scheduling conditions has to be supported by an interactive graphic interface (Fig. 12).

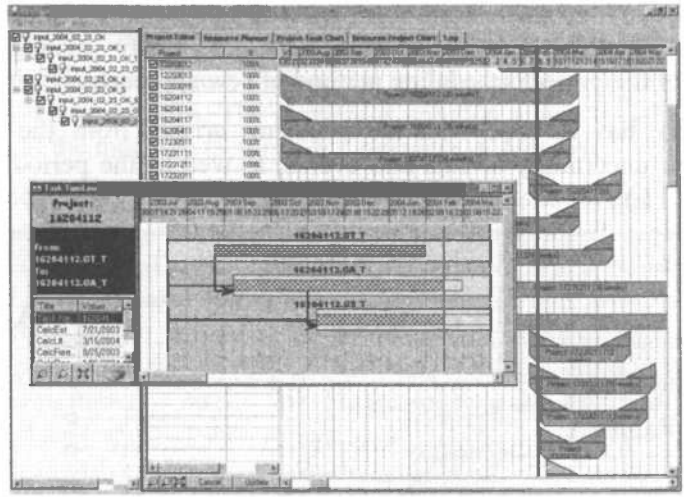


Figure 12. Interactive interface of a project-based scheduler

Requirements concerning business and production goals are:

- The harmonization of internal and external (supplier) capacities, the maximization of the utilization rate of internal resources;
- High-level readiness for delivery, meeting due dates strictly.
- Decreasing lead times, keeping the number of running processes at low level.

Software requirements for the scheduler:

- Modern information platform (Windows XP and .NET are used by the prototype);
- Effective algorithm for the task and solver (CPLEX and ILOG solver);
- Modular structure, maintenance services;
- Possibility of improvement and reutilization.

The most important application requirements:

- Pure model, functional correctness, examinations of consistence;
- Reliability, tested menus;
- Graphical human-machine interface, easy to manage;
- Possibility of integration (ERP, MES; Text-type I/O files).

The improved application software ProTerv is based on many components. The model generating process, the examination of consistence and database treatment ensure the controlled task description for the scheduler component. The main

source of input data is the database ERP and the interactive human-machine interface (Fig. 13).

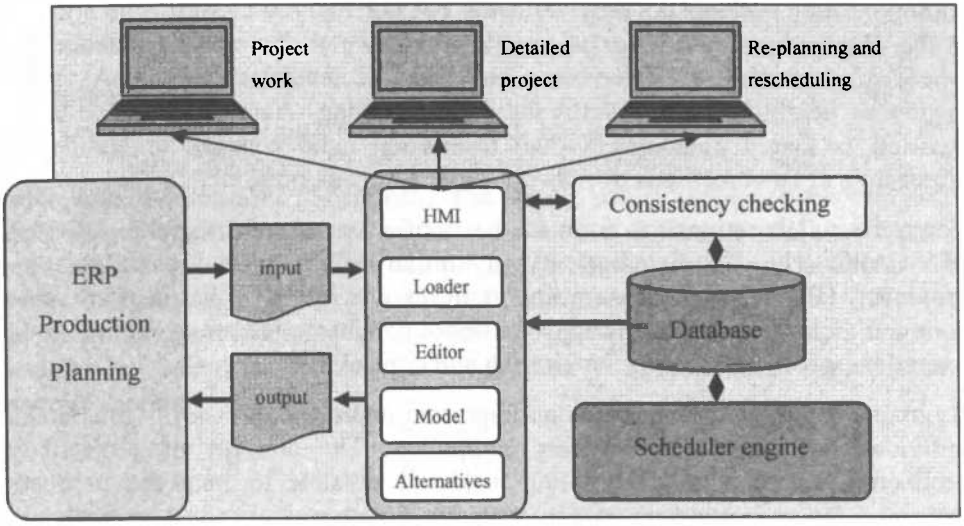


Figure 13. Structure of a project-based production scheduler

The scheduler provides the scheduling plans for project activities as output, representing them in tables and in a graphical way (Fig. 14).

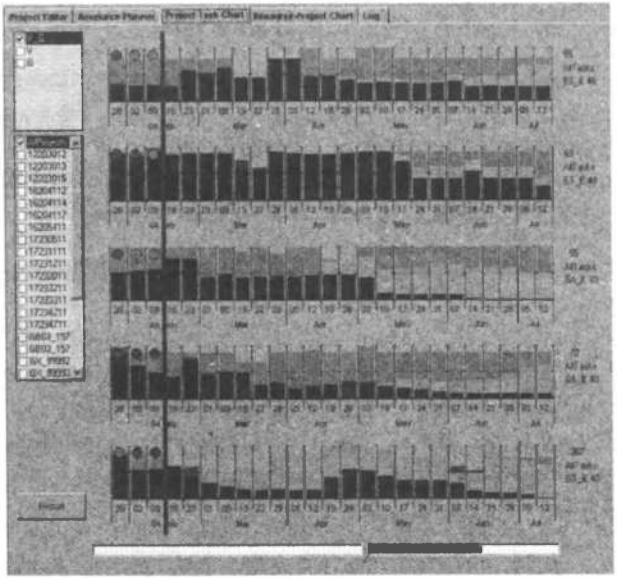


Figure 14. Graphic form for scheduling plans of ProTerv

12. Conclusions

VE-frameworks enlarge application possibilities of information technology and communication systems not only vertically but horizontally as well. The principles of the VE paradigm have revealed new possibilities at the layers (for instance CIM) where a great deal of experiences has been accumulated so far. As typical examples, flexibility and re-configurability of joining, shared knowledge-base in addition to shared databases, virtual teamwork, rapid changes in competence, application of BPR methods by network-organisational tools, etc.

The same collaborations can form the basis of a reassessed integration of CAPP-PPS-CAPC. The VE paradigm has a similar effect on the paradigms used previously (JIT, TQM, CIM). An important novelty of VE is that it offers several common tools for production-supply-business-commerce processes including their operations, goals, scheduling, monitoring and control.

Aggregate production planning is an important and difficult task of firms making individual machines and complex equipment. The concept of project-based production planning and scheduling makes it possible to treat the production activities and the engagement of capacities together. A production scheduling model can be based on the rate of project activities. The model of project activities is significantly different from the scheduling model of shop-floor control. The high-level activities of the projects can be interrupted and can be planned at a changing rate. The activities can also use several different resources. The profile of rates in time can be influenced by additional constraints.

We have applied the rate-based model of aggregate production planning in R&D work carried out within the framework of a consortium. The model proved to be successful for different products and production profiles as well. The experimental computerized applications are being tested at present. The advantages offered by the rate-based aggregate production scheduler are as follows:

- The number of occasions when the deadline is over-run decreases;
- The lead times of projects decreases;
- The set of external orders can also change advantageously;
- Utilization of capacities increases and will be more balanced;
- The use of overtime and external capacities decreases;
- The bottle-necks can be recognized and can be treated in a better way than earlier;
- The number of jobs in process decreases.

In addition, an important benefit can be the increase of the co-ordination of engineering functions and the improvement of the integration of the chief engineer department and shop-floor levels. Alternative solutions of modelling of the

production processes increase efficiency of management decisions. On the basis of experience, a reengineering process of greater scale can be realized for improving and controlling the working process of the production planning organization.

Acknowledgements

The research work presented in this paper was carried out within the framework based on the collaboration of academic and industrial partners. The topic of the project is *Digital Factories, Production Networks* (Project No. 2/040/2001, project leader: *László Monostori, CARI-HAS*). The project was supported by the Hungarian Government within the framework of the National Research and Development Program. The research was carried out by the Production Information Engineering Research Team (University of Miskolc) in collaboration with the Office for Research Groups Attached to Universities and Other Institutions, Hungarian Academy of Sciences. The authors would like to express their thanks for the financial support.

REFERENCES

[1] ASKIN, G. A., STANDRIDGE, C. R. (1993): *Modeling and Analysis of Manufacturing Systems*. John Wiley and Sons Inc., New York.

[2] BUZACOTT, J. E., SHANTHIKUMAR, J. G. (1993): *Stochastic Models of Manufacturing Systems*. Prentice Hall Inc., New Jersey.

[3] DETZKY, I. FRIDRIK, L., TÓTH, T. (1989): *On a New Approach to Computerized Optimization of Cutting Conditions.* Proc. of the 2nd World Basque Congress. Bilbao, V.1. pp. 129-141.

[4] DUFFIE, N., FALU, I. (2002): *Control Theoretic Analysis of a Closed Loop PPC System*. Annals of the CIRP V. 51/1 pp. 379-382.

[5] ERDÉLYI, F., TÓTH, T., SOMLÓ, J., KOVÁCS, A., KÁDÁR, B., MÁRKUS, A., VÁNCZA, J. (2002): *Production management: taking up the challenge of integration.* 3rd Conference on Mechanical Engineering. Budapest, pp. 705-709.

[6] GARANSON, H.T. (1999): *The Agile Virtual Enterprise*. Quorum Books. Westport, USA.

[7] GOLDRATT, E. M. (1994): *Theory of Constraints*. North River Press. New York.

[8] HARRINGTON, JOSEPH, JR. (1973): *Computer Integrated Manufacturing*. Reprint, New York: Robert E. Krieger Publishing Co., 1979.

[9] HOPLAND, JAN, AND SAVAGE, CHARLES M (1989): *Charting New Directions.* Digital Enterprise 3, No. 1. pp. 8-12.

[10] HUNT, V. D. (1989): *Computer Integrated Manufacturing Handbook.* Chapman and Hall Ltd, New York.

[11] KIS, T. (2003): *A Branch and Cut Approach for scheduling projects with variable intensity activities.* 6th Workshop on Models and Algorithms for Planning and Scheduling Problems. Aussois, France, pp. 160-172.

[12] KIS T., ERDŐS G., MÁRKUS A., VÁNCZA J. (2004): *A Project- Oriented Decision Support System for Production Planning in Make-to-order Manufacturing.* ERCIM News. 2004. July. No.58, pp. 66-67.

[13] KIS T. (2004): *Project Scheduling.: a Review of Recent Books.* Operations Research Letters. 33. pp. 105-110.

[14] KUSIAK, A., DORF, R. C. (1994): *Handbook of Design, Manufacturing and Automation.* John Wiley & Sons Inc. New York.

[15] MÁRKUS A., VÁNCZA J., KIS T. (2003): *Project Scheduling Approach to Production Planning.* Annals of the CIRP. V. 52/1/2003, pp. 359-361

[16] MONKS, J. G. (1987): *Operations Management: Theory and Problems.* McGraw Hill Book Company, New York.

[17] PERKINS, J. R., KUMAR, P. R. (1989): *Stable, Distributed Real-Time Scheduling of Flexible Manufacturing Systems.* IEEE Trans.on Aut. Cont. V. 34, N.2, pp. 139-148.

[18] RAVIGNANI, G. L., TIPNIS, V. A., FRIEDMAN, M. Y. (1977): *Cutting Rate Tool Life Function (R-T-F). General Theory and Application.* Annals of the CIRP, V. 25/1. pp. 295-301.

[19] STARBEK, M., GRUM, J. (2000): *Operation lead time control.* Robotics and Computer Integrated Manufacturing. N.16. pp. 443-450.

[20] TÓTH, T., ERDÉLYI, F. (1997): *The Role of Optimization and Robustness in Planning and Control of Discrete Manufacturing Processes.* Proc. of the 2nd World Congress on Intelligent Manufacturing Processes and Systems. Springer Verlag, Budapest, pp. 205-210.

[21] TÓTH, T. (1999): *New Principles and Methods in the Computerized Integration of Process Planning and Production Control.* Publ. Univ. of Miskolc. Series C, Mechanical Engineering. Vol.49. pp. 173-187.

[22] TÓTH, T. (1998): *Planning Principles, Models and Methods in Computer Integrated Manufacturing.* University Press, Miskolc, Hungary (in Hungarian).

[23] VÁNCZA J., KIS T. KOVÁCS A. (2004): *Aggregation the key to integration Production Planning and Scheduling.* Annals of the CIRP. V. 53/1/2004, pp. 377-380.

[24] VERNADAT, F.B. (1996): *Enterprise Modeling and Integration.* Chapman & Hall Ltd, London.