Production Systems and Information Engineering
Volume 3 (2006), pp. 3-20

ALGORITHMS FOR E-MARKETPLACES INTEGRATED WITH LOGISTICS

LÍVIA KACSUKNÉ BRUCKNER
International Business School
Department of Methodology and Information Systems
lkacsuk@ibs-b.hu

[Received October 2005 and accepted February 2006]

Abstract. The E-Marketplace Model Integrated with Logistics (EMMIL) is a family of models that bring together three entity types – seller, buyer and third party logistics service provider - in the same transaction optimising the total cost and/or maximising the revenue. This article is focusing on the allocation algorithms of the buyer oriented EMMIL. The line-haul model is analysed in detail, the combinatorial model is solved for a major class of possible scenarios.

Keywords: B2B e-marketplace, logistics

1. Introduction

Ongoing researches in the fields of supply chain analysis and e-business have an emerging interrelated area of investigation, seeking for new methods in supply chain optimisation [1]. This paper is aimed at contributing to this interdisciplinary research field.

Business-to-Business (B2B) e-marketplaces facilitating trade between businesses are supply chain management (SCM) tools of high importance. Today a substantial part of supply chains are managed across the Internet still they contain a surprisingly high amount of inefficiencies [2].

Logistics services may be handled internally or outsourced to third party logistic (3PL) providers. The standard B2B marketplace models today do not facilitate the integration of logistical solutions into the negotiation between buyers and sellers thus total cost optimisation is practically impossible. E-marketplaces selling goods either do not offer logistical solutions at all or offer a single solution or 2-3 possibilities in different time-cost ranges. The so called integrated marketplaces offer both goods and logistical services, but the goods must be selected first and then logistics providers' market can be reached with a click.

To address part of the SCM inefficiency problems a new model of B2B marketplaces is introduced in [3]. This model integrates three sides of the business into the same transaction - sellers, buyers and third party logistics service providers (3PL) - creating the possibility of a higher level optimisation compared to the

traditional e-marketplaces. The new model is called EMMIL, meaning e-market place model integrated with logistics. This paper gives insight to the theory behind the EMMIL model.

2. Related Research in Auction Theory

Auctions are widely used market mechanisms both in traditional and electronic commerce. In the last two decades researchers of auction theory and practitioners have made huge efforts in order to facilitate, support, optimise and standardise electronic negotiations. The aim is to find incentive and computationally manageable auction mechanisms for different business scenarios.

An auction is a resource allocation process, its main components according to [4] are the following: resources, market structure, preference structure, bid structure, matching supply with demand and information feedback.

Resources might be classified as single or multiple items with single or multiple units each having single or multi-attribute specifications. In combinatorial auctions resources are traded in bundles in which case the value of the goods is not equal to the sum of the individual values.

The market structure determines both the number of buyers and sellers participating in a transaction and the mechanism of negotiation. We can distinguish between seller-oriented, buyer-oriented and intermediary marketplaces. Buyer-oriented marketplaces run reverse auctions for an entity with strong buying potential in order to minimise procurement cost.

The bid structure allows the bidders to show their preference structures. We have designed a new bid structure that is much closer to the real world situations than the bid structures discussed in the literature of auction theory.

Matching supply with demand – winner determination – is the crucial point in all auctions. In simple cases is rather straightforward, but in case of general combinatorial auctions is NP-complete so authors in the literature tend to address particular circumstances and find solutions for restricted areas. Linear and integer programming are natural tools for optimisation problems but as the number of participants increases complexity causes difficulties in producing the results within a business feasible duration. Applying metaheuristics, particularly branch and bound algorithms seem to be a very promising direction for tackling these problems [5],[6] that is why we are using a combination of integer programming and branch and bound techniques.

3. General Scheme of EMMILs

In this we give an overview of the general architecture of the EMMIL family of three-sided e-marketplace models that integrate logistics service providers to goods' markets.

Third party logistics providers (3PLs) usually offer one or more of the following services: transportation, warehousing, packaging, unit-load grouping and bulk-breaking. By integrating logistics to goods' marketplaces we mean that the logistics providers are placing their offers step by step during the negotiation of buyers and sellers then each time a combined proposal showing the total cost is created by the marketplace. With this approach the business decisions can be based on the total cost which opportunity has never been offered by any marketplaces.

The general structure of EMMIL can be seen on Figure 1. This framework serves as a base for designing different type of models. In the centre there is the

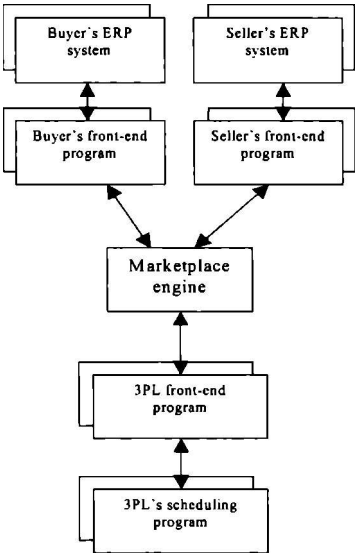


Figure 1. General structure of EMMIL

marketplace engine to which the entities from all the three sides – buyer, seller and 3PL – are joining with the help of front-end processors. It is assumed that all participants have advanced resource planning and scheduling programs that can provide the front-end processors with the relevant data within a short time.

The ratio of the number of buyers and sellers determines if the model is buyer-oriented, seller-oriented or intermediary (exchange) type. All three kinds of marketplaces need a different composite auction mechanism. Here we deal with procurement auctions only since we focus on the buyer-oriented EMMIL models (EMMIL/BM). We use a composite reverse auction with discrete rounds of open bidding that alternate between sellers and 3PLs. The general bidding process is independent from the bidding structure and can be outlined by the flowchart in Figure 2.

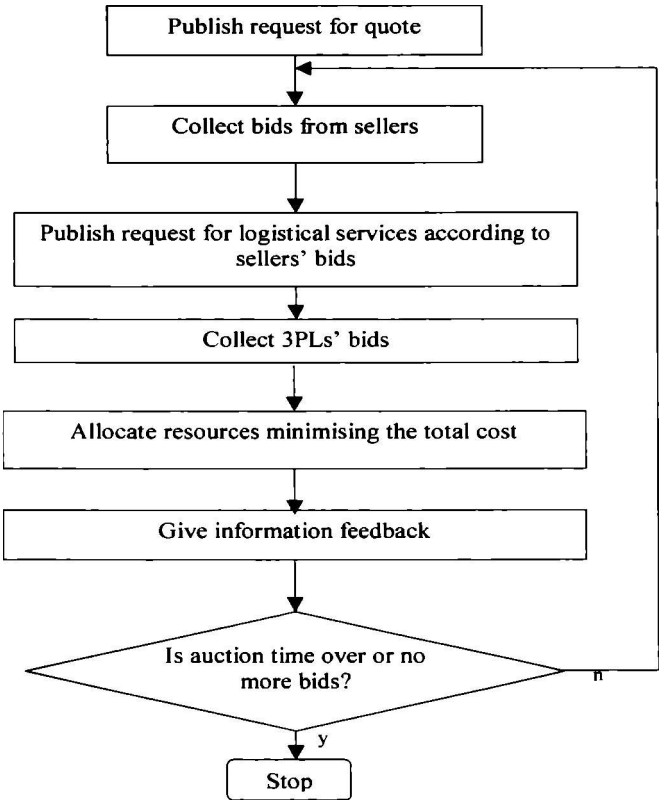


Figure 2. Allocation algorithm for buyer-oriented EMMILs

It can be seen that the auction terminates if no new bids are placed from either sides or a pre-set auction time is over. The auction fails if the set of optimal combination of bids remains empty. In the following chapters we go into the details of buyer oriented EMMIL models and the composite procurement auctions.

4. Formulation of EMMIL/BM

Participants of the marketplace are: a single buyer looking for a set of non-digital goods in specified quantities, many sellers (suppliers) who wish to sell certain quantities of some of the required goods and many third party logistics providers that undertake transportation jobs and warehousing if needed. A seller may act as 3PL as well in case of having logistical capabilities, e.g. vehicles for the transportation.

The products are assumed to be homogenous from the transportation point of view which means they belong to the same transportation class and are packed

uniformly. We assume that bulks will not be broken thus required quantities will be integer numbers. We shell focus on road transport.

4.1. Cost Considerations

In order to create an incentive bid structure we are investigating the cost structure of sellers and 3PLs since they have independent private values that is basically the sum of their costs and their target profit margin. Cost structure of EMMIL models are discussed in detail in [7], here only a short summary is given.

Cost accounting [8] classifies costs as variable or fixed according to how they react to the changes in activity. Variable costs - like cost of material built into a product - change linearly with the volume, fixed costs like building depreciation or insurance - do not change for a longer period of time. There are also semi-fixed or step-fixed costs that can be described by a step function. In practice total production and service costs are always semi-variable, they contain both fixed and variable components. We can say that economies of scale can be achieved if the volume of an activity increases up to the limit that can be handled by the same fixed cost value. What we are looking for is an appropriate price structure for the seller side and the 3PL side that reflects the nature of their costs and can be handled by the marketplace engine.

In case of goods the different cost ranges are usually realised by quantity discounts according to pre-set volume ranges. There are several papers in auction literature dealing with quantity discount models, e.g. [9], [10], [11]. In traditional commerce we can also find discount rates based on overall spending on purchases. According to these principles we shall introduce a complex bid structure for sellers in formula (3) allowing any of these two discount types or the combination of them.

Costs of basic logistical activities include handling, transportation and warehousing. Cost calculations should be based on processes but here we are going to simplify the model and use an overall fix and variable cost structure related to a particular tour of a truck regardless of the actual process that takes place between loading the goods at the sellers' plant and unloading them at the buyer.

The fix cost of a tour should cover the organisational overhead and the direct fixed costs. Direct fixed costs of transportation are related partly to the operation of terminals, communication and information systems applied for vehicle tracking, partly to the type of the vehicles, their purchase costs, insurance and general maintenance. Rights-of-way can also form fixed costs if they are pre-paid for a longer period Variable costs relate directly to the movement of a particular load from point A to point B. They include labour (driver's wages), fuel and depreciation of the vehicle as well as maintenance associated with the usage. They depend on the type of the vehicle, the distance, the road conditions, and in case of heavy cargo some of them like cost of fuel can be influenced by the weight of the load as well. The measurement can be [EUR / km] or [EUR/ ton km]. We have to

mention that this latter measurement can be misleading if the truck is not full since the major part of fuel and similar costs do occur even if the truck is empty. We may say that these are semi-fixed costs for a line-haul operation where the fix part refers to the cost of moving the empty truck from A point to B and the variable part depends on the weight of the freight - very often non-linearly.

In addition to these variable costs there can be several others that are not proportional to the distance or weight such as casual highway tolls, detention (delay) costs or accommodation costs of the driver in case of long distances. Carriers very often have to cover the back-haul costs as well, i.e. moving back the vehicle from B point to A, adding joint costs to the front-haul transportation. These can only be avoided with back-haul freight or optimised vehicle routing. In case of a consolidation or split-delivery every stop-off means extra costs because of the detention.

Carriers usually apply rates and tariffs for pricing reflecting more or less the fixed and variable costs. They usually set an initial sum that depends on the geographical area and overall requirements of the transportation. Second part of the rate is a unit price related to the distance and the weight or volume of the load, measured e.g. in [EUR / pallet km]. In EMMIL marketplaces carriers should be able to bid for certain routes (line-haul or consolidation tour) with uncertain quantities. A useful cost structure can be given if we create a sum of the indirect and direct fixed cost and the semi-fixed costs of moving the vehicle from starting point (a seller) to the end point (the buyer). This will be the fixed cost of transportation in a bid. The variable cost will be proportional to the volume transported where extra costs of moving the load and handling can be accounted for.

Warehousing cost is composed of handling and storing costs. Handling cost is proportional to the volume so it be aggregated with the previously discussed handling costs. The problem is that storing is proportional with the volume and the time. Still it can be incorporated into the variable cost structure since the time of necessary storing can be calculated before placing thus it can be taken as a fix factor. These considerations lead to the bid structure given in formula (4).

4.2. Bid Structure

First we introduce some basic notations then formulate the buyer's request for quotation (1).

N Number of items (products)

i Item identifier

M Number of suppliers

k Supplier identifier

L Number of 3PLs

l 3PL identifier

Φ type of unit-load

φ class of product

$$R=\{W,\varphi,\Phi,[Q_i, P^h_i], E, U]\}\ i=1,2,\ldots,N \quad (1)$$

W Target location (warehouse of the buyer)

Q_i, required quantity of product i [unit]

P^h_i upper limit of price of product i [EUR/unit]

E earliest arrival time

U latest arrival time

According to the principles set in section 4.1. we shall introduce a complex bid structure for sellers (2) allowing any of these two discount types or the combination of them:

$$B_s^k=\{ W^k,[Q^{kv}_i,P^{kv}_i], [Q^{kh}_i,],[\Xi^k_g, \Delta^k_g], E^k, U^k\}, \quad (2)$$

where

$i \in \{1,...,N\}$, $v= 1,...,V^k_i$ $g=1,...,G^k$

B_s^k bid of seller k

W^k warehouse location of seller k

V^k_i Number of quantity discount categories used by seller k for item i

G^k Number of categories for overall discount at seller k

Q^{kv}_i, minimum quantity of item i in discount category v at seller k [unit]

Q^{kh}_i, maximum quantity of item i at seller k [unit]

P^{kv}_i unit price of item i in volume discount category v at seller k [EUR/unit]

E^k earliest shipping time at seller k

U^k latest shipping time at seller k ($>=E^k$)

Ξ^k_g lower limit of overall discount category g at seller k [EUR]

Δ^k_g discount factor of overall discount category g at seller k (>0, <1)

If $V^k_i=1$ then no volume discount is given for product i. If no overall discount is given then $G^k=1$ and $\Xi^k_1=\Delta^k_1=0$ is set. If a seller does not have product i to sell then sets $V^k_i=1$ and $Q^{kh}_i = Q^{kv}_i = P^{kv}_i = 0$.

To make further considerations easier we introduce some preliminary concepts. Let A be the set of identifiers of sellers $A=\{1,2,...,M\}$. Denote $\boldsymbol{P(A)}$ the power set of A.

Now we are going to formulate the bid structure of 3PLs as (3) shows.

$$B_L^l=\{[\Gamma_j^l, F_j^l, V_j^l, {}^mT_j^l, {}^hT_j^l]\}, \quad j=1,2,...,G^l, \tag{3}$$

where

$G^l=\left| B_L^l \right|$ number of routes specified in bid of 3PL l

$\Gamma_j^l \in P(A)$ route as a combination of sellers' identifiers

F_j^l Fix cost of delivering a standard size truck of goods on route Γ_j^l

V_j^l Variable cost of delivering goods on route Γ_j^l (cost/unit load) [EUR/unit]

${}^mT_j^l$ Minimum time period needed for delivering all goods on route Γ_j^l [day]

${}^hT_j^l$ Maximum time period needed for delivering all goods on route Γ_j^l [day]

This is a combinatorial bid, we allow 3PLs to bid for consolidation delivery referring to certain combinations of sellers. This means that they are ready to collect the goods from these sellers with as many trucks as it will be needed and it does not mean that they are going to traverse all the sellers with one truck.

5. The Optimisation Problem

In this section we formulate the objective function (4) of the transactions for the EMMIL/BM marketplaces in the most general way:

$$\min\left(\sum_{k=1}^{M}\left(\sum_{i=1}^{N}P_i^kQ_i^k\right)\left(1-\Delta^k\right)+\sum_{l=1}^{L}\sum_{j\in S^l}x_j^l\left(F_j^l\left\lceil\left(\sum_{k\in\Gamma_j^l}\sum_{i=1}^{N}Q_i^k\right)/Z\right\rceil+V_j^l\sum_{k\in\Gamma_j^l}\sum_{i=1}^{N}Q_i^k\right)\right), \tag{4}$$

where:

Q_i^k Purchased quantity of product i from seller k [unit]

P_i^k Unit-price of product i at seller k as a step function of quantity [EUR/unit]

Δ^k Discount given as a step function of the total purchase cost at seller k

Z Standard truckload size [unit]

$x_j^l \in \{0,1\}$ decision variable, $x_j^l=1 \Leftrightarrow$ offer j of 3PL l is selected as winner.

There are several constraints that have to be formulated. First of all, the buyer wants to buy the required quantity of each product as (5) expresses:

$$\sum_{k=1}^{M}Q_i^k=Q_i, \qquad i=1,2,...,N. \tag{5}$$

We cannot buy any product from a seller in less quantity than the specified minimum level or more than the maximum (6):

$$Q_i^{k1} \le Q_i^k \le Q_i^{kh} \quad \forall i = 1,2,..N, \forall k = 1,2,... \quad M \tag{6}$$

Time constraints in (7) state that the products must be available at the sellers early enough for arriving at the buyer before the latest receiving time but must not arrive too early:

$$(E^k + {}^hT_j^l \le U) \wedge (U^k + {}^mT_j^l \ge E) \forall l \in \{1,2..., L\}, \forall j \in \{1,2,..., G^l\}, \forall k \in \Gamma_j^l. \tag{7}$$

Constraint (8) says that we select a bid $\Gamma^l_j \in D^l$ of a 3PL for delivery only if we purchase some goods from all the suppliers in that combination:

$$x_j^l \le Sign(\sum_{i=1}^{N} Q_i^k), \quad \forall\ k \in \Gamma_j^l. \tag{8}$$

All sellers selected for the transaction – and only those- must be reached by one and only route as (9) formulates:

$$\sum_{l=1}^{L} x_k^l = Sign(\sum_{i=1}^{N} Q_i^k), \quad \forall\ k \in \{1,2..., M\}. \tag{9}$$

We apply certain business policies as well. Constraint (10) expresses that the number of suppliers of goods should be kept within specified limits in order to reduce the additional costs of maintaining relationships with suppliers and avoid exposure [9]:

$$S_{\min} \le \sum_{l=1}^{L} Sign(\sum_{j=1}^{G^l} x_j^l) \le S_{\max}, \tag{10}$$

where

S_{min} minimum number of suppliers allowed,

S_{max} maximum number of suppliers allowed.

We have to determine the x^l_j binary and Q^k_i integer variables to minimise the costs and satisfy the constraints. In the following chapter we discuss the possible solution methods.

6. Winner Determination Algorithms

The objective function formulated in (4) is nonlinear and it is not smooth, not even continuous. In order to solve it easily in the following chapters we show a linearisation algorithm. We start with models without discounts then we show how

to deal with the discounts as well. We also disregard from the time constraints since the inappropriate offers can be easily filtered first.

6.1. Line-haul Models without Discounts

In this scenario we exclude all the discounts so the objective function will be (11), and constraint (8) will be replaced by (12)

$$\min\left(\sum_{k=1}^{M}\left(\sum_{i=1}^{N} P_i^k Q_i^k + \sum_{l=1}^{L} x_k^l\left(F_k^l\left\lceil\left(\sum_{i=1}^{N} Q_i^k\right)/Z\right\rceil + V_k^l\sum_{i=1}^{N} Q_i^k\right)\right)\right), \tag{11}$$

$$x_k^l \le Sign\left(\sum_{i=1}^{N} Q_i^k\right), \quad \forall\ k \in A. \tag{12}$$

We would like to derive an equivalent problem with a piece-wise linear objective function from (11) by determining the aggregated minimal logistical cost functions for line-haul transportation from each seller. In order to do this we introduce $g_k^l(Q)$ in (13) which is the logistical cost function of purchasing from seller k and using logistical services of 3PL l. This function has discontinuity at each multiple of the truck capacity Z and it is linear between the jumps. Taking the minimum in l we get $g_k(Q)$, the best logistical cost function for seller k. (14) This function consists of concave, continuous polygons between nZ and $(n+1)Z$ as is illustrated in Figure 3.

$$g_k^l(Q) = F_k^l\lceil Q/Z\rceil + V_k^l Q, \tag{13}$$

$$g_k(Q) = Min_l\, g_k^l(Q). \tag{14}$$

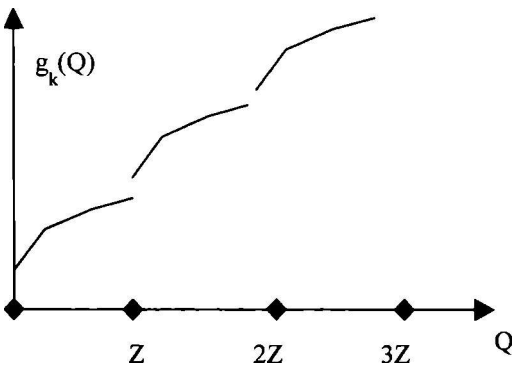


Figure 3. General scheme of the minimal logistical cost function from seller k.

Before introducing the algorithm for constructing $g_k(Q)$ we formulate a few lemmas underpinning the method.

Lemma 6.1.1.

If $j, l \in \{1,...,L\}$ $j \neq l$ and $\exists\ b^l \in B_L^l$, $b^j \in B_L^j$ logistical bids from seller k, $(F_k^l, V_k^l) \subset b^l$ $(F_k^j, V_k^j) \subset b^j$ such that $F_k^l \geq F_k^j \wedge V_k^l \geq V_k^j$, then $g_k^l(Q) \geq g_k^j(Q)$ if $Q \geq 0$.

Remark: This means if both fix and variable costs are smaller in one of the two bids than the same relation holds for the total costs always.

Proof

Matching the definition of costs and the conditions of the lemma we get:

$g_k^l(Q) = F_k^l \lceil Q/Z \rceil + V_k^l Q$ and $g_k^j(Q) = F_k^j \lceil Q/Z \rceil + V_k^j Q \Rightarrow$

$g_k^l(Q) - g_k^j(Q) = (F_k^l - F_k^j) \lceil Q/Z \rceil + (V_k^l - V_k^j) Q \geq 0$. Thus the lemma is proved.

Consequently we can eliminate a logistical bid if we find another one with smaller fix and variable costs. In case of equality we have to use business rules based on previous experiences to find out which 3PL should be kept.

Lemma 6.1.2.

If $Z>0$ integer, $Q \geq 0$ real number, then $Q / \lceil Q/Z \rceil \leq Z$.

Proof

According to divisibility by Z we can express $Q = nZ + r$, $0 \leq r < Z$, $n \geq 0$ integer

If $r=0$, then $Q=nZ$, so $Q/\lceil Q/Z \rceil = nZ/n = Z$

If $r>0$, then $\lceil Q/Z \rceil = n+1$

$Q/(n+1) = n/(n+1)Z + r/(n+1) = Z - (Z-r)/(n+1) < Z$ since $r<Z$.

Lemma 6.1.3.

If the logistical bids of 3PLs j and l are such that $F_k^l \geq F_k^j \wedge V_k^l < V_k^j$

and $(F_k^l - F_k^j) / (V_k^j - V_k^l) \geq Z$, then $g_k^l(Q) \geq g_k^j(Q)$ for $Q \geq 0$.

Proof

Using Lemma 6.1.2. we rearrange the outer sides of the following inequality to get the statement.

$(F_k^l - F_k^j) / (V_k^j - V_k^l) \geq Z \geq Q / \lceil Q/Z \rceil$ hence $F_k^l \lceil Q/Z \rceil + V_k^l Q \geq F_k^j \lceil Q/Z \rceil + V_k^j Q$

From this lemma it follows that in such a case we can eliminate the bid of the 3PL l from the offers referring to seller k since we have a better choice.

In all other cases the cost functions will intersect resulting in alternating minimal offers. The following lemma shows that we do not have to compute the

intersections of all pairs of functions since those that can not be eliminated can be sorted in a way that only adjacent functions do intersect each other.

Lemma 6.1.4.

If $S_k=\{ g_k^l(Q),\ l=1,2,...,L_k \}$ is a set of logistical cost functions related to seller k that cannot be eliminated from seeking the minimal function, i.e. $\neg\exists j,j'\ j\neq j'$, such that $g_k^{j'}(Q) \geq g_k^j(Q),\ Q\geq 0$, then the elements of S_k can be sorted in a way that $F_k^1 \leq F_k^2 \leq ... \leq F_k^{Lk}$ and $V_k^1 \geq V_k^2 \geq ... \geq V_k^{Lk}$

Proof

Let us suppose that the lemma is not true. Then $\exists\, j,j' \in \{1,\ldots, L_k\}, j\neq j',\ F_k^{j'} \geq F_k^j \wedge V_k^{j'} \geq V_k^j$ but then lemma 6.1.1. causes $g_k^{j'}(Q) \geq g_k^j(Q)$, which contradicts the conditions of the lemma. Hence the lemma is proved.

We introduce Q^T for the total quantity required by the buyer:

$$Q^T \quad \sum_{i=1}^{N} Q_i \tag{15}$$

Now we describe the algorithm for the linearisation of the objective function (11).

Algorithm 6.1.

BEGIN

1. Let $S_k=\{ g_k^l(Q),\ l=1,2,...,L \}$
2. $\forall\ l=1,...,L\text{-}1$

   IF $\exists\, j\in\{1,...,L\}, j\neq l, F_k^l \geq F_k^j \wedge V_k^l \geq V_k^j$

   THEN we eliminate the function from the set based on Lemma 6.1.1.

   $S_k := S_k \setminus g_k^l(Q)$

   ENDIF
3. $\forall\ g_k^l(Q) \in S_k$

   IF $\exists\ g_k^j(Q) \in S_k, j\neq l,\ (F_k^l - F_k^j) / (V_k^j - V_k^l) \geq Z$

   THEN we eliminate the function from the set based on Lemma 6.1.3.

   $S_k := S_k \setminus g_k^l(Q)$

   ENDIF
4. Sort the remaining elements of S_k by the fix costs in ascending order. Then based on Lemma 6.1.4 we get

   $F_k^1 \leq F_k^2 \leq .. \leq F_k^{Lk}$ and $V_k^1 \geq V_k^2 \geq .. \geq V_k^{Lk}$

5. Determine the intersection points of the consecutive functions by solving (16). Denote the number of intersections by A_k . The intersection points will form a sequence of increasing real numbers. $0<q_1^k<q_2^k<...<q_{A_k}^k \le Q^T$ We also include the multiple values of Z if they are not in the sequence. Taking the lower cost values on each interval we get the minimal cost function $g_k(Q)$. Denote $l(q_j^k)$ the identifier of the best 3PL at q_j^k

$$F_k^j \lceil Q/Z \rceil + V_k^j Q = F_k^{j+1} \lceil Q/Z \rceil + V_k^{j+1} Q \qquad 0 < Q \le Q^T \tag{16}$$

6. Associate an F_k^l, V_k^l value pair with each q_l^k intersection point denoting the fix and variable cost valid from that point until the next one in the sequence. Calculate $C_k^l = F_k^l \lceil q_l^k / Z \rceil$ for each intersection point.

7. Let $q_0^k = 0$ and $q_{A_k+1}^k = Q^T$ closing elements to the sequence with $C_k^1 = 0$.

END

After this process we can transform (12) to (17) achieving the goal of linearisation. We have Q_i^{kl} , integer, and y_k^l binary variables where y_k^l decides weather the total quantity purchased from seller k lays in the lth. interval of the aggregate minimal cost function when buying Q_i^{kl} quantity of product i from seller k. This problem with the constraints given in (18)-(23) can be solved by any commercial integer programming packages.

$$Min \sum_{k=1}^{M} \sum_{l=1}^{A_k} \left(\sum_{i=1}^{N} P_i^k Q_i^{kl} + C_k^l y_k^l + V_k^l \sum_{i=1}^{N} Q_i^{kl} \right) \tag{17}$$

$$y_k^l \in \{0,1\} \tag{18}$$

$$Q_i^{kl} \ge 0 \quad \forall i \in \{1,...,N\} \tag{19}$$

$$Q_i^{kl} \le Q_i^{kh} \quad \forall i \in \{1,...,N\} \tag{20}$$

$$\sum_{i=1}^{N} Q_i^{kl} - q_l^k y_k^l \ge 0 \quad \forall l \in \{1,...,A_k\}, k \in \{1,...,M\} \tag{21}$$

$$\sum_{i=1}^{N} Q_i^{kl} - q_{l-1}^k y_k^l \le 0 \quad \forall l \in \{1,...,A_k\}, k \in \{1,...,M\} \tag{22}$$

$$\sum_{k=1}^{M} \sum_{l=1}^{A_k} Q_i^{kl} = Q_i \tag{23}$$

Theorem 6.1.

If minimising the objective function (17) with constraints (18)- (23) results in $\{Q_i^{kl'}, Y_k^{l'}\ k\in A,\ i\in\{1,...,N\},\ l'\in\{1,...,A_k\}\}$, then this solution can be transformed into $\{Q_i^k, X_k^l\ k\in A,\ i\in\{1,...,N\},\ l\in\{1,...,L\}\}$, the solution of (11) with constraints (5), (6), (12).

Proof

Based on lemmas 6.1.1. - 6.1.4. and linearisation algorithm 6.1. the two solutions can be converted into each other as it (24) and (25) show:

$$Q_i^k = \sum_{l'=1}^{A_k} Q_i^{kl'} \tag{24}$$

$$X_k^l \begin{cases} 1, & \text{if } \exists\, l'\in\{1,2...,A_k\},\ Y_k^{l'}=1 \ \wedge l(q_{l'}^k) = l \\ 0, & \text{otherwise.} \end{cases} \tag{25}$$

The constraints match as:

(23) $\Rightarrow$ (5)

(19) $\wedge$ (20) $\Rightarrow$ (6) as from excluding quantity discounts $Q_i^{k1}=0\ \forall\ i\in\{1,...,N\},\ k\in A$

(21) $\wedge$ (22) $\Rightarrow$ (12)

Hence the theorem is proved.

The number of variables of the linear optimisation problem is expressed in (26) and it is rather high comparing to the original $M(N+L)$ variables. We give an estimate to A_k, the number of linear pieces of the logistical cost function relevant to seller k in order to keep control on the complexity of the problem.

$$(N+1)\sum_{k=1}^{M} A_k \le M\cdot(N+1)\cdot Max_k(A_k) \tag{26}$$

The number of shipments from a seller cannot be higher then $\lceil Q^T/Z \rceil$ which, taking lemma 6.1.4. into consideration, results in the estimate (27):

$$Max_k(A_k) \le \left\lceil Q^T / Z \right\rceil L. \tag{27}$$

Theorem 6.2. will show that in case of normally distributed parameters L can be replaced by a constant and hence A_k has an upper limit given in (28):

$$Max_k(A_k) \le 6\left\lceil Q^T / Z \right\rceil. \tag{28}$$

Theorem 6.2.

If the variable part of the logistical costs follows a normal distribution then the mean of the maximum number of 3PLs contributing to the minimal cost function in an interval $[Q1, Q2]$ is not more than 6, regardless of the number of bidding 3PLs.

Proof

Let ξ be the random variable of the fix logistical costs and η the random variable of the variable logistical costs where η follows a normal distribution. We sort the 3PLs according to their specified fix costs in ascending order this way determining a random sort on their variable costs. (Figure 4) For any ϑ random variable of normal distribution it is true with confidence of. 0.99 that $|\vartheta^{max}-\vartheta^{min}| < 6\sigma(\vartheta)$ where σ is the standard deviation of ϑ. Furthermore the mean of the distance between any two values of this distribution can be estimated by $\sigma(\vartheta)$, hence the mean of the length of the longest monotone sequence cannot be larger than 6. Along with Lemma 6.1.4. this gives the proof to the theorem.

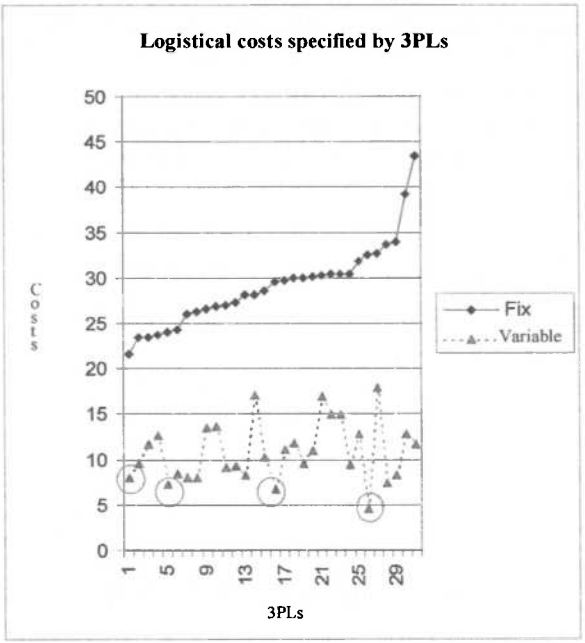


Figure 4. Monotone sequence in a random seguence of variable logistical costs

6.2. Line-haul Models with Quantity Discounts.

This scenario allows individual volume discounts on items but 3PLs can bid for line-hauls only. Here the same solution can be applied as above but new variables and constraints for discount categories should be added. Due to the limited space

we only give some hints instead of the proper formulas. The linear optimisation problem can be formulated similarly to (17) adding decision variables showing if the quantity of product i purchased from seller k falls into discount category j. It is also needed a set of a pre-computable constant showing the cost of the quantity equal to the lower limit of this category j.

6.3 Models Allowing Consolidation Delivery

If we allow combinatorial bids on transportation then the linearisation might result in an unmanageable high number of variables. To overcome this problem we created a set of branch and bound solutions for different scenarios. Here we describe a heuristic algorithm for the case when all suppliers have enough quantities of all products, i.e. we release constraint (6). In the algorithm we investigate the costs consequences of purchasing from different combination of sellers exploiting constraint (10).

Let A be the set of identifiers of sellers $A=\{1,2,...,M\}$. Denote $\boldsymbol{P}(A)$ the power set of A. An $S\in\boldsymbol{P}(A)$ is linearly ordered since the elements are different natural numbers. Denote Max (S) the highest element in S. We introduce $A^{\{m\}} \subseteq \boldsymbol{P}(A)$, $m \in A$, for equal size subsets of $\boldsymbol{P}(A)$. Denote $A^{\{m\}}=\{S \mid (S\in\boldsymbol{P}(A)) \wedge (|S|=m)\}$ We order each $A^{\{m\}}$ lexicographically. We order sets of different sizes naturally, if $S\in A^{\{m\}}$ and $S'\in A^{\{n\}}$ then $S'>S \Leftrightarrow n>m$. We introduce Parent(S) = $S \setminus$ Max(S). It can be seen that if $|S|=1$ then Parent(S)= Ø. Denote Gen(S) the tree generated by S using the rule Gen(S)=$\{S': S' \in \boldsymbol{P}(A), S'>S\}$.

This way $\boldsymbol{P}(A)$ consists of M disjoint trees as it is illustrated in Figure 6. The ordering we introduced goes across the trees level by level.

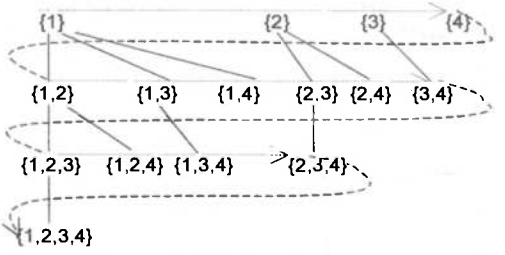


Figure 6. Traversing trees in P({1,2,3,4})

Denote Opt(S) the optimal solution of (4) in $S\in\boldsymbol{P}(A)$ with $C^{opt}{}_{\Sigma}(S)$ total cost of procurement from sellers in S, buying something from all of them:

$\mathrm{Opt}(S)=\{[Q^k],[x_j^l]\}$, $i\in\{1,...,N\}$, $k\in S$, $l\in\{1...,L\}$, $\Gamma^l_j\subseteq S$

Denote $w(s)$ the highest price vector for all item in Opt(S), $w(S)=(w_1(S), w_2(S),...,w_N(S))$ and $\bar{F}^{\pi}_{opt}$, V^k_{opt} the logistical cost parameters in Opt(S) for $k\in S$. Let P_i^k the best unit price of product i at seller k.

Lemma 6.3.1.

If S', $S \in P(A)$ where $S'=S\cup\{k'\}$ and $\forall\ i\in\{1,...,N\}\ P_i^{k'}>= w_i(S)$ and $F^{k'}{}_{opt} >= F^{k}{}_{opt}$ $V^{k'}{}_{opt}>= V^{k}{}_{opt}\ \forall\ k\in S$ then $C^{opt}{}_{\Sigma}(S')> C^{opt}{}_{\Sigma}(S)$.

Proof

The purchase cost of goods cannot be reduced since all prices are higher than in the previous combination. We have to add an extra location where the transportation costs are also higher according to the assumption. The only way of reducing the cost could be replacing the transport of two or more partly loaded trucks from the original sellers by one from the new seller but if we could do this then Opt(S) would not be an optimal solution either. Hence the lemma is proved.

In the **winner determination algorithm** we create a domain set D first by pruning the ordered set $P(A)$ according to the constraints limiting the number of suppliers. Then we traverse the nodes of D in the defined order and create a set of candidate best solutions using a linearised model for the limited set of sellers. During this process we keep pruning subtrees that cannot contain candidate solutions based on Lemma 5.3.1. At the end we compare the candidate solutions at the remaining nodes of D and choose a final solution with minimal total cost.

The speed of the algorithm depends on the size of subtrees that can be pruned. Better result can be achieved if we order the suppliers first in descending order by the total cost that would occur if we purchased everything from one supplier. We can also pre-calculate a lower limit of the total cost and exit the algorithm if a candidate solution reaches it within a given tolerance.

7. Summary and Further Research Plans

EMMIL E-Marketplace Model Integrated with Logistics is a triangular marketplace based on the contribution of buyers, sellers and logistics service providers in the same transaction. It aims at finding partners with minimising the total purchase cost or maximising the revenue depending on the type of the marketplace. This paper focused on the buyer-oriented model (EMMIL/BM) where a buyer carries out procurement from many sellers. A composite reverse auction mechanism was defined with discrete rounds of open bids that alternate between sellers and 3PLs. The suggested bid structure was justified by cost considerations. A few scenarios were examined from algorithmic point of view for line-haul and consolidation deliveries.

The implementation process has been just started. Algorithms for line-haul models were tested so far by a prototype implementation using the Solver program of MS EXCEL [13]. Unfortunately this program can handle only 140 variables, but the tests have underpinned the theoretical results perfectly. A full scale implementation is currently being carried out. The problem is NP-complete but we believe that applying realistic business constraints in the number of contracted sellers will keep

the computational time within acceptable limits even for high number of variables. A further stage of this research is the development of the distributed optimisation algorithms according to principles already set in [12].

REFERENCES

[1] SIMCHI-LEVI, D. WU, SHEN, EDS.: *Handbook of Quantitative Supply Chain Analysis, Modeling in the e-Business Era.* Kluwer Academic Publishers, 2004.

[2] PARKER, B.: *Total Cost of Relationship Should Drive Collaboration Strategies Return on Relationship Revisted.* AMR Research, www.amrresearch.com/content, July 2003.

[3] KACSUKNÉ, L. BRUCKNER, CSELÉNYI, J.: *E-Marketplace Model Integrated with Logistics.* MicroCAD International Scientific Conference Miskolc, 2004.

[4] KALAGNANAM, J., PARKES, D.: *Auctions, bidding, and exchange design* in *Handbook of Quantitative Supply Chain Analysis, Modeling in the e-Business Era.* Simchi-Levi, D. Wu, Shen, Eds. Kluwer Academic Publishers, 2004.

[5] ROTHKOPF, A. P., HARSTAD, M. H., HARSTAD, R.: *Computationally manageable combinatorial auctions.* Management Science, vol. 44, pp. 1131-1147, 1998.

[6] SANDHOLM, T.: *An algorithm for optimal winner determination in combinatorial auctions.* Artificial Intelligence, vol. 135, no. 1, pp. 1-54, 2002.

[7] KACSUKNÉ, L. BRUCKNER., CSELÉNYI, J.: *Cost considerations of the EMMIL E-Marketplaces*, in Bányai T, Cselényi J. (eds.): Logistics Networks – Models, Methods and Applications pp. 235-248. University of Miskolc, Miskolc 2005.

[8] DRURY, C.: *Management and Cost Accounting*, 5th Int. ed. Thomson, 2000.

[9] HOHNER, G., RICH, J., NG, E., REID, G., DAVENPORT, A. J., KALAGNANAM, J. R., LEE, S. H., AN, C.: *Combinatorial and quantity discount procurement auctions benefit Mars, incorporated and its suppliers INTERFACES.* vol. 33, no. 1, pp. 23-35, 2003.

[10] CHEN, R. R. JANAKIRAMAN, G., ROBIN, R., ZHANG, R. Q.: *Efficient auctions for supply chain procurement.* Johnson Graduate School of Management, Cornell University, Ithaca, NY, Tech. Rep., 2002.

[11] ESO, M., GHOSH, S., KALAGNANAM, J., LADANYI L.: *Bid evaluation in procurement auctions with piece-wise linear supply curves.* IBMResearch, Yorktown Heights, NJ, USA, Research Report RC 22219,2001.

[12] KACSUKNÉ, L. BRUCKNER, KISS, T.: *Using Grid-technology to Implement an e-Marketplace Integrated with Logistics.* Dapsys International Conference Budapest, 2004.

[13] KACSUKNÉ, L. BRUCKNER: *Models and Algorithms for E-Marketplaces Integrated with Logistics*, Ph.D. dissertation, University of Miskolc 2005. (in Hungarian)

Production Systems and Information Engineering
Volume 3 (2006), pp. 21-38

A PROPOSED METHOD TO HANDLE CLASSIFICATION UNCERTAINTY USING DECISION TREES

NORBERT TÓTH
Budapest University of Technology and Economics, Hungary
Department of Measurement and Information Systems
ntoth@mit.bme.hu

BÉLA PATAKI
Budapest University of Technology and Economics, Hungary
Department of Measurement and Information Systems
pataki@mit.bme.hu

[Received November 2005 and accepted May 2006]

Abstract. A novel method is proposed in this paper to handle the classification uncertainty using decision tree classifiers. The algorithm presented here extends the decision tree framework to give the ability of measuring the confidence of the classification. Using this algorithm a certain number of the input samples are rejected as "risky points" in order to obtain a smaller misclassification rate on the remaining points. The algorithm is being integrated into a Medical Decision Support System where a confindence number to every classification is required.

Keywords: decision tree, classification uncertainty, CART, misclassification, medical decision support system

1. Introduction and Inspiration

Breast cancer is one of the most common form of cancer among women. Every 12th woman suffers from this disease at least once in her lifetime [1]. Since the cause of breast cancer is unknown, early detection is very important. If detected early, the five-year survival rate exceeds 95% [1].

Currently mammography (X-ray examination of the breast) is the most efficient method for early detection. In a mammographic session usually two images are taken of both breasts. Craniocaudal (CC) is a top view, mediolateral (ML) is roughly a side view image of the breast. Radiologists typically notice suspicious-looking structures in one view and then verify their suspicion by checking the corresponding area of the other view of the same breast. The most important mammographic symptoms of breast cancer can be divided into two main classes: microcalcifications (a group of small white calcium spots) and masses (usually approximately round object brighter than its surrounding tissue).

If a global screening were done, a huge number of mammograms (approximately one million images every year in Hungary) would require diagnostics. The main goal is to create a tool that can ease the work of radiologists by filtering out the true negative cases and draw attention to the suspicious ones. Such Medical Decision Support System for Mammography is being developed in cooperation with radiologists in the Budapest University of Technology [2].

In the system several detection algorithms are working parallel to each other, looking for different kinds of abnormalities (e.g. microcalfications, masses) or different kinds of features to detect the same type of abnormality. Since markings (spots that show the location of an abnormality) created by the detection algorithms cannot be 100% certain, a confidence value was introduced to the system. Each marking is accompanied by this value, showing the diagnoses certainty. The higher this value, the more possible is that the marking is a true abnormality. This value is also used by post-processing algorithms to filter out the most likely false positive markers (the ones with the lowest confidence value). Each algorithm is produces this confidence value, although in different ways.

One of the methods uses decision trees to classify a certain number of features at a location of the image [3]. The result of this classification can be normal tissue or abnormality. If the features are classified as abnormal tissue a marking is generated. To generate the confidence value the original decision tree algorithm was modified to handle classification uncertainty.

This paper discusses a novel extension to the original Classification and Regression Tree (CART) framework (proposed by Breiman et al, 1984) [4] to handle classification uncertainty.

2. Extension of the Decision Trees

2.1. Basis of the Current Work

The origin of decision trees dates back to 1963, when the AID (Automatic Interaction Detection) program [5] was developed at the Institute for Social Research, University of Michigan, by Morgan and Sonquist. They proposed a method for fitting trees to predict a quantitative variable. The AID algorithm created regression trees. A modification to the AID was the THAID algorithm [6] in 1973 by Morgan and Messenger which handled nominal or categorical responses. The THAID program created classification trees. Now several decision tree approaches exist, e.g.: CART, ID3, C4.5 [7], C5, THAID CHAID, TREEDISC, etc.

One the most widespread used decision tree framework is the Classification and Regression Trees (CART, 1984) [4] developed by Breiman et al. Their work is

based on the original ideas of the AID and the THAID algorithms. We used their work as basis for our enhancements to the decision tree methodology.

2.2. Decision Tree Basics

Decision trees can be used to predict values or classify cases. Because in our work (mammographic image analysis) classification is the main issue; therefore from now on we restrict our discussion to classification trees. Classification trees are used to predict membership of cases or objects in the classes of a categorical dependent variable from their measurements on one or more predictor variables.

Decision tree methods use supervised learning to recursively divide the observations into subcategories in such a way that these subcategories differ from each other as much as possible while each subcategory is as homogenous as possible. The outcome of a decision tree building algorithm is a directional graph connecting nodes. Each node of the graph represents a set of observations. There are two kinds of nodes: "terminal" and "non terminal" Non terminal nodes are also referred as "internal" nodes. These nodes incorporate a "splitting rule", which is used to split the observations into subcategories. Terminal nodes are also referred as "leaf" nodes. These nodes represent the dependent variable – in our case – the predicted class of the observations (figure 1).

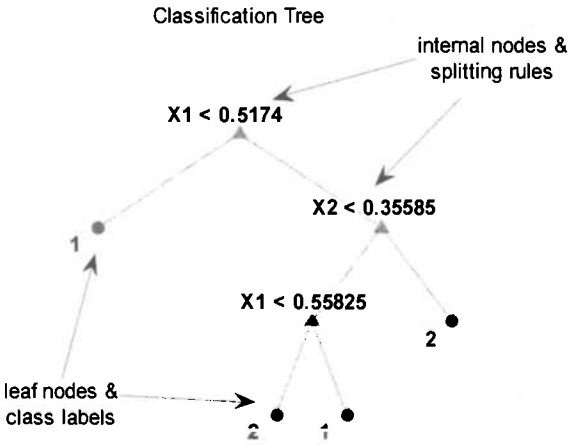


Figure 1. Sample classification tree with 3 splits and 2 classes.

There are various ways to grow a decision tree. For example there are several options to implement the splitting rule or the selection of the right sized tree. For

our work we implemented CART algorithm [4] and used it as a basis for further development. Properties of the tree growth algorithm:

- Splitting rule

A splitting rule deals with observations reaching that specific node. It is used to divide that group of observations into subgroups. We use 2-way binary (smaller / bigger) splits on one variable. At each node a single variable is tested if bigger or smaller than a certain threshold value. At each node impurity is defined to measure the homogeneousness of the node. The best split is the one that creates the purest nodes. Given a node t with estimated class probabilities $p(j|t)$, $j=1..Nc$, where Nc is the number of classes, a measure of node impurity for given t

$$i(t)=f[p(1|t), \ldots, p(Nc|t)] \tag{2.2.1}$$

is defined and an exhaustive search is made to find the split that most reduces tree impurity. This split maximizes the impurity gain

$$\Delta i(t)=i(t)-pLi(tL)-pRi(tR), \tag{2.2.2}$$

where pL and pR is portion of observations falling to left or right child node (tL, tR) according to split. To measure node impurity the Gini diversity index [4] was adopted, which has the form

$$i(t) = \sum_{j \neq i} p(j \mid t) p(i \mid t) \tag{2.2.3}$$

and can be rewritten as

$$i(t) = (\sum_{j} p(j \mid t))^2 - \sum_{j} p^2(j \mid t) = 1 - \sum_{j} p^2(j \mid t). \tag{2.2.4}$$

The Gini index ensure that for any split s the impurity can only decrease: $\Delta i(s,t) \geq 0$.

- Determining the right sized tree

In general, bigger trees having more splits, give better classification rate on the training data. However they tend to overfit, giving worse classification rates on the test data. To determine the right sized tree – that gives the best error rate (R) on the test data and avoids overfitting – there are 2 options. The first is to stop splitting according to a certain criterion. According Breiman's [4] and our experiments as well this is not recommended. The better way to determine the right sized tree is to grow a tree that is much too large and than "prune" it upwards iteratively until we reach the root node. After this test sample error

estimates (R) are used to select best subtree that has minimal error on the test data. We implemented the Minimal Cost Complexity (MCC) pruning algorithm [4]. In MCC pruning a cost-complexity measure is introduced:

$$R\alpha(Tt)=R(Tt)+\alpha|Tt|, \tag{2.2.5}$$

where α is the cost-complexity parameter (real number), Tt is the subbranch starting at node t (if t=1, than $T1=T$ the original tree) and $|Tt|$ is number of terminal nodes on the subbranch Tt. The higher the value of the α parameter the greater the cost of more complicated trees. In this sense the tree complexity is defined by the number of its terminal or leaf nodes. To get a series of pruned subtrees we start from the original tree and we perform a "weakest-link cutting" This is done in the following way:

set

$$R\alpha(\{t\})=R(\{t\})+\alpha \tag{2.2.6}$$

and

$$R\alpha(Tt)=R(Tt)+\alpha|Tt|. \tag{2.2.7}$$

As long as $R\alpha(Tt)< R\alpha(\{t\})$ the branch Tt has a smaller cost-complexity than the single node $\{t\}$. In other words it is "worth" to keep this node expanded. At a critical value of α the two cost-complexities become equal, than keeping only a single node $\{t\}$ instead of an expanded branch Tt is preferable. To find this critical α, the following equation must be solved:

$$\alpha = \frac{R(\{t\}) - R(Tt)}{|\mathrm{T}| - 1}. \tag{2.2.8}$$

This critical α value has to be calculated for all internal nodes of the tree, and than the smallest is the "weakest-link" This means that node is the one that – if we increase α – has to be "closed" to get better cost-complexity value for the entire tree T. Closing means to prune the tree at that location, to replace the branch Tt with the single node t.

Using this method we get a series of smaller and smaller subtrees according to the increasing value of α. To select the best tree we can use test sample or cross-validation error estimates. We used 10-fold cross-validation [4] to estimate the misclassification rate. In this sense the best tree is the smallest one that has minimal cross-validation (or test sample) error (figure 2).

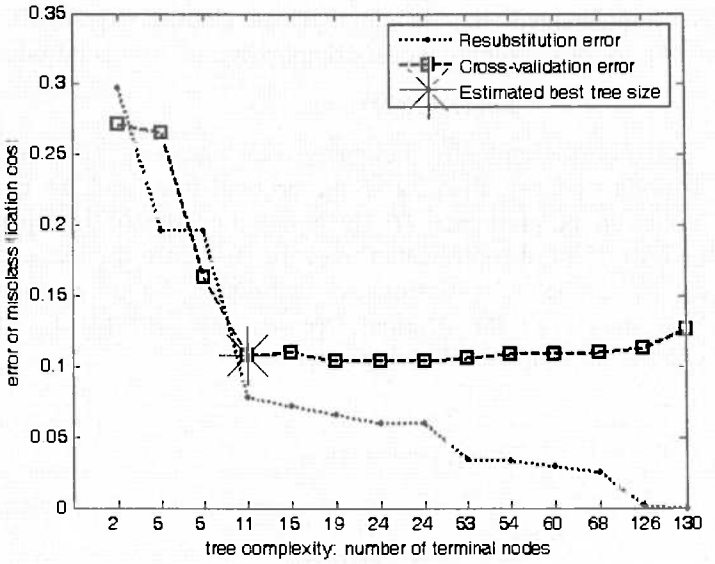


Figure 2. The best tree is the one that has minimal cross-validation error. Resubstitution error means the error on the learning data.

Decision trees produced by the CART algorithm have some favorable properties compared to other methods. They are easily interpretable and can be used to classify data very quickly. Another good property is that the decision boundary can be easily identified. In the next sections we will introduce an algorithm to provide a confidence value to the classification result of the tree. This algorithm makes use of the clear structure of the decision trees and the explicitly defined decision boundary.

2.3. Dealing with Classification Uncertainty

A classification tree divides the input space into a certain number of sections. These sections have their class label according to the leaf that defines the actual section. If the input vector of the predictor variables falls into a section, the corresponding class label is returned. Dealing with the classification uncertainty or classification confidence the main assumption is that the confidence of the classification is proportional to the distance from the closest decision boundary that splits between different class labels.

According to the previous assumption, to get a classification certainty value we need to measure the shortest distance to the closest decision boundary that splits

between different classes, or equally the shortest distance to the closest section with different class label.

The proposed algorithm to measure the shortest distance from the closest decision boundary that splits between different classes is the following:

1) First the actual data is classified using the decision tree: a leaf node is reached, which defines a section in the input space and an output label.

2) To get the distances to the other sections the input data point is projected to all of the decision boundaries. The projection rules are calculated only once (see 2.4 determining projection rules), right after the tree growth process and stored together with the tree structure.

   If the input space contains N variables than the decision boundaries of a section are maximum N-1 dimensional hyper planes (figure 3, 4). A 0 dimensional boundary only exists if all variables in the input space exist in the path from the root node to the leaf node.

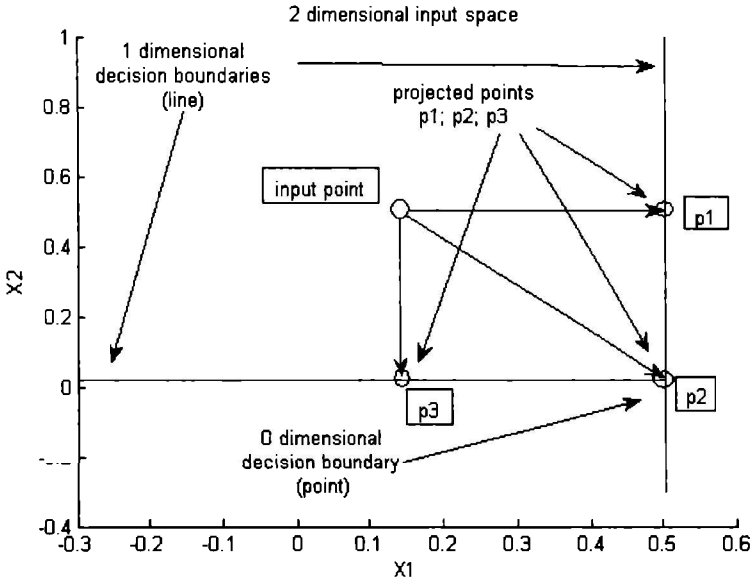


Figure 3. Sample decision boundaries in 2 dimensional input space and the projected points of the input data point.

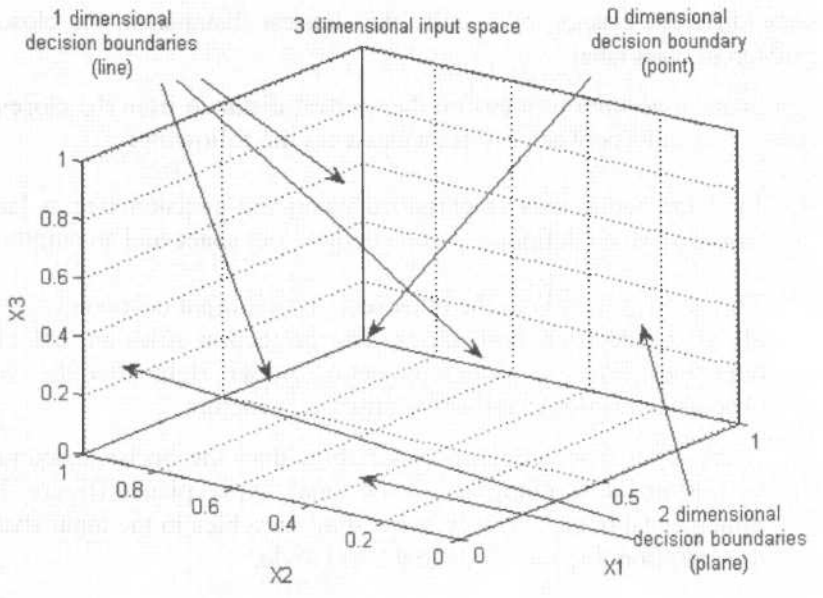


Figure 4. Sample decision boundaries in 3 dimensional input space.

3) The distance between the projected points and the input point is calculated.

4) Take out the projected point that has minimal distance from the input point.

5) Check if that projected point is on a decision plane that splits between different classes (see 2.5 checking projected points).

6) If yes, the output certainty value is the distance between the projected point and the input data. If no, take out the next projected point with minimal distance and repeat steps 5 and 6.

The algorithm described above returns the shortest distance to the closest decision boundary that splits between different classes.

2.4. Determining the Projection Rules

A set of critical points that we call "projection rules" has to be determined for each leaf node. These critical points will be the closest points on the boundary of the actual input space section. We call these "projection rules" because most of these

points are not fully defined, they are maximum N-1 dimensional hyper planes (the input space is N dimensional).

To determine the projection points for a certain leaf the following algorithm is proposed:

Initialize a boundary matrix that will contain the boundary values for the actual input space section marked by the leaf node. This matrix has equal number of rows to the number of input variables in the decision tree. The matrix has 2 columns, because each variable can border the actual segment from above and from below. Initialize the border matrix with infs (abbreviation for infinite) as if no border was present to the actual segment to any direction. Than we go from the leaf node up to the root node taking out the splitting variables and split values. When we take out a split we also check if we came up from a smaller child or from a bigger child. When a split value is taken out we insert it into the boundary matrix into the row identified by the variable and into column 1 if it was a smaller child, and into column 2 if it was a bigger child. If we encounter a split that already in the boundary matrix we skip that because that is not a boundary to the section marked by the leaf node.

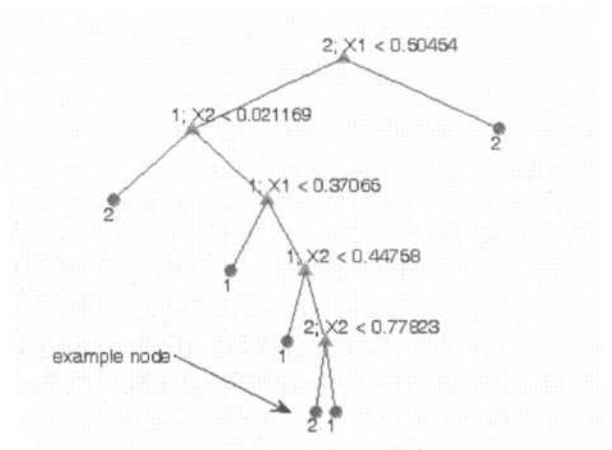


Figure 5. Sample tree with an example node.

The boundary matrix for the example node (figure 5.):

$$BM = \begin{bmatrix} 0.504 & 0.370 \\ 0.778 & 0.447 \end{bmatrix}. \tag{2.4.1}$$

Now the boundary matrix is extended with a column containing infs (referring to infinites). The extended boundary matrix:

$$BMe = \begin{bmatrix} 0.504 & 0.370 & \text{inf} \\ 0.778 & 0.447 & \text{inf} \end{bmatrix}. \tag{2.4.2}$$

To get all the critical points we have to get all the permutations of the elements of the extended boundary matrix, keeping the order of the variables. The points from the matrix *BMe*:

$$\begin{aligned} &P1=(0.504, 0.778) \\ &P2=(0.504, 0.447) \\ &P3=(0.370, 0.778) \\ &P4=(0.370, 0.447) \\ &P5=(\text{inf}, 0.778) \\ &P6=(\text{inf}, 0.447) \\ &P7=(0.504, \text{inf}) \\ &P8=(0.370, \text{inf}). \end{aligned} \tag{2.4.3}$$

The first 4 critical points are fully defined, they are actual points (0 dimensional hyper planes). The rest of them are not fully defined they are (in this case) 1 dimensional hyper planes: lines (figure 6).

These projection rules depend only on the decision tree and independent from the input data. As a result they have to be calculated only once, which saves a considerable amount of time.

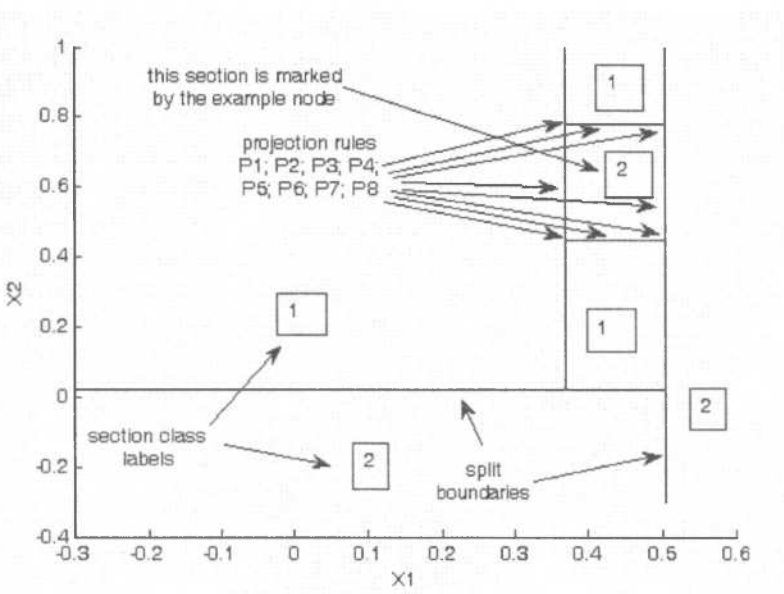


Figure 6. The 8 critical points (projection rules) for the example node.

We use these critical points (projection rules) to project the input data point to section boundaries. The projection technically means to insert the coordinates of the input point into the projection rules replacing the infs. We have to project the input point with the rules from *each* leaf node. These projected points will be the closest boundary points to the input point. However it can not be known if a projected point on a section boundary that splits between different classes. This has to be determined individually for each projected point (see 2.5 checking projected points).

2.5. Checking Projected Points

A projected point is not necessarily on a boundary that splits between different classes. Checking each side of a decision boundary in the worst case requires 2^*N*-1 points to be classified by the tree if we checked all sections around the projected point. However we only have to determine the class of the section on the other side of the boundary in the direction of the projection. We do not have to check the surrounding points because those are not the closest border points to the corresponding area. Figure 7 shows a simplified illustration.

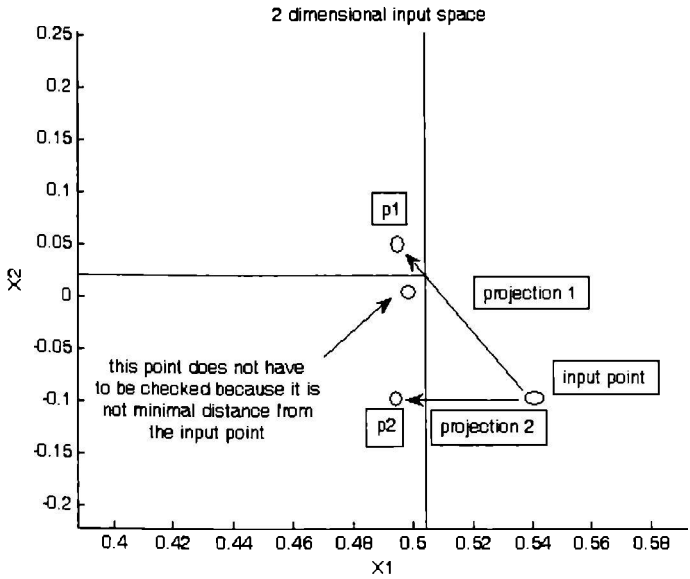


Figure 7. Checking projected points. The point marked does not have to be checked because $p2$ is in the same section and is closer to the input point (using the algorithm given in section 2.3 it is already checked by the time we get to checking $p1$).

Checking technically means to "push" the projected point further in the projection direction to an ε distance into the corresponding section and than use the tree to classify the point (figure 7, $p1$ and $p2$).

3. Test Results and Conclusion

3.1. Test Result on 2 Dimensional Data

The algorithm is first demonstrated on a 2 dimensional dataset. The input data is shown in figure 8. The input points are marked with an 'x' or ' ' according to their class label. After the decision tree is grown (using cross-validation and MCC pruning), the input space is covered with a grid and the distance from the decision boundaries are calculated in the grid's points using the algorithm presented in section 2.3. Figure 9 displays the distance from the decision boundaries.

There were N=1354 data points in the sample dataset, containing roughly equal number of class 1 and class 2 members. 1248 points were correctly and 106 were incorrectly classified by the tree. This gives 7.8% misclassification error rate.

Using the introduced algorithm we calculate the certainty value (the distance from the decision boundaries) for each input point. We determine a certainty threshold

such that if the certainty for a given input point is smaller than this threshold the tree rejects the classification. For this application the threshold is defined to keep 76% of the correctly classified samples (figure 10). In this case 90% percent of the incorrectly classified cases are filtered out (96 points out of 106). 960 cases are classified out of the total 1354, which is around 71% of the total number of input points. 29% percent (394) of the input samples are considered as risky, meaning the calculated classification certainty value was under the threshold. From the classified ones 10 are misclassified, which means 0.1% misclassification rate. Visually this means the method filters out the risky cases inside a "safety lane" around the decision boundaries (figure 11).

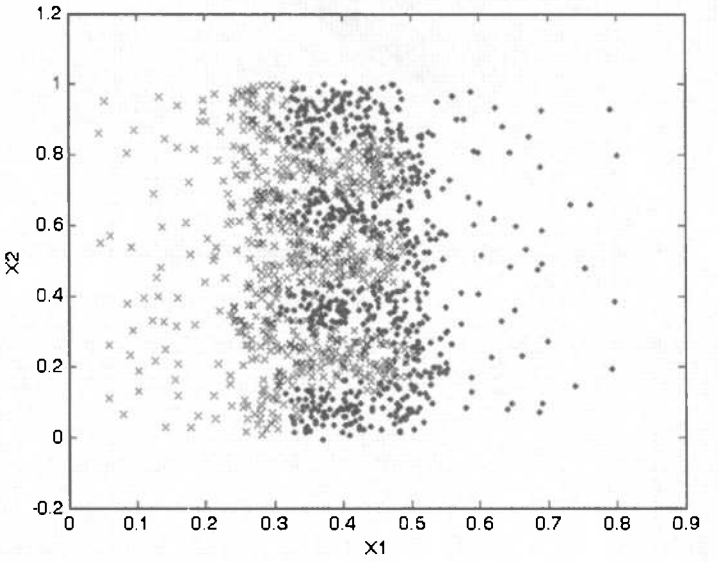


Figure 8. A 2 dimensional sample dataset. The input points are marked with an 'x' or '.' according to their class label.

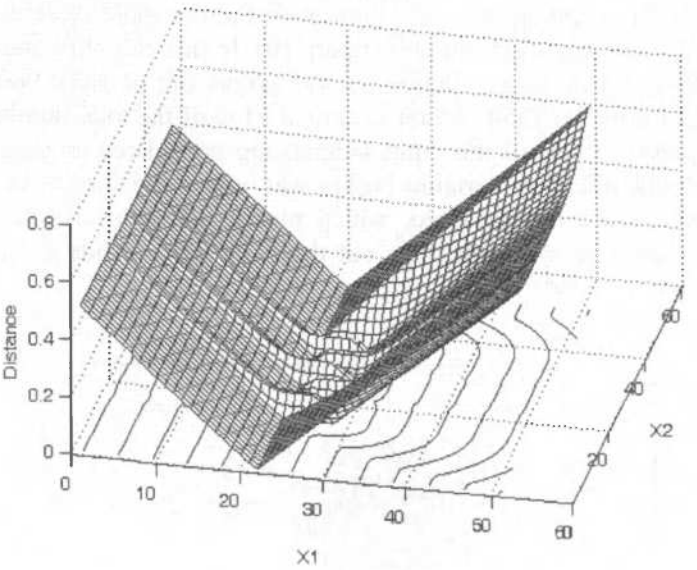


Figure 9. Distance from the decision boundaries.

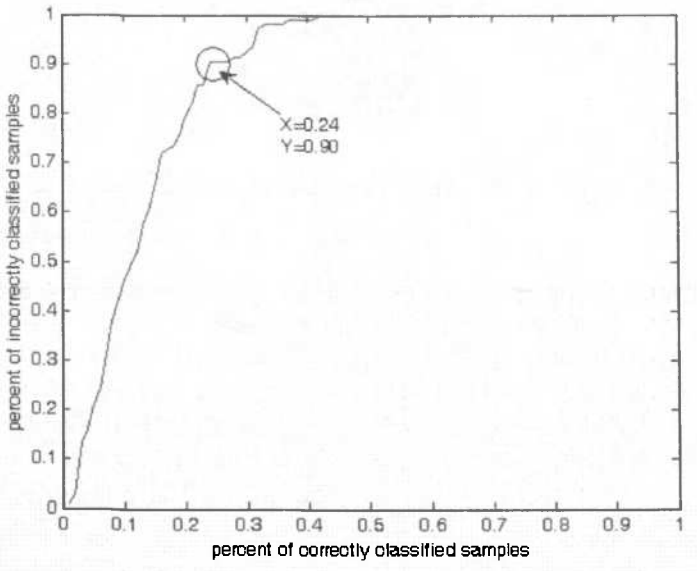


Figure 10. Defining the threshold value to keep 76% of the correctly classified samples.

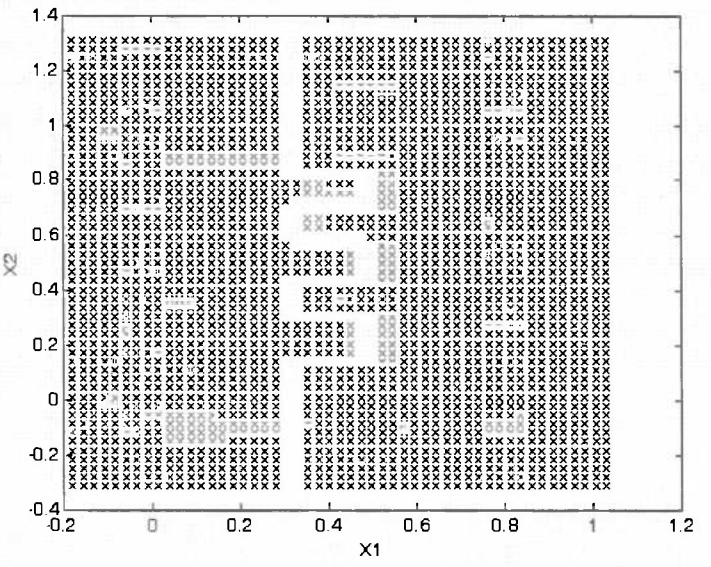


Figure 11. The "safety lane" around the decision boundary.

3.2. Test Results on 9 Dimensional Data

The algorithm is now demonstrated on a 9 dimensional data set. This dataset is the "breast-cancer-wisconsin" dataset downloaded from the UCI Machine Learning Repository [8].

2 experiments were made with different trees. The database contained N=699 data points.

In the first experiment the tree misclassifies 33 points out of the 699, which gives 4.9% misclassification rate. The certainty threshold is determined to keep 90% percent of the correctly classified samples (figure 12). This case the tree rejects 12.7% of the input points but the misclassification rate on the remaining points reduces to 1.5%.

In the second experiment the tree misclassifies 23 points out of the 699, which gives 3.3% misclassification rate. The certainty threshold is determined to keep 93% percent of the correctly classified samples (figure 13). This case the tree rejects 8% of the input points and the misclassification rate on the remaining points reduces to 2%.

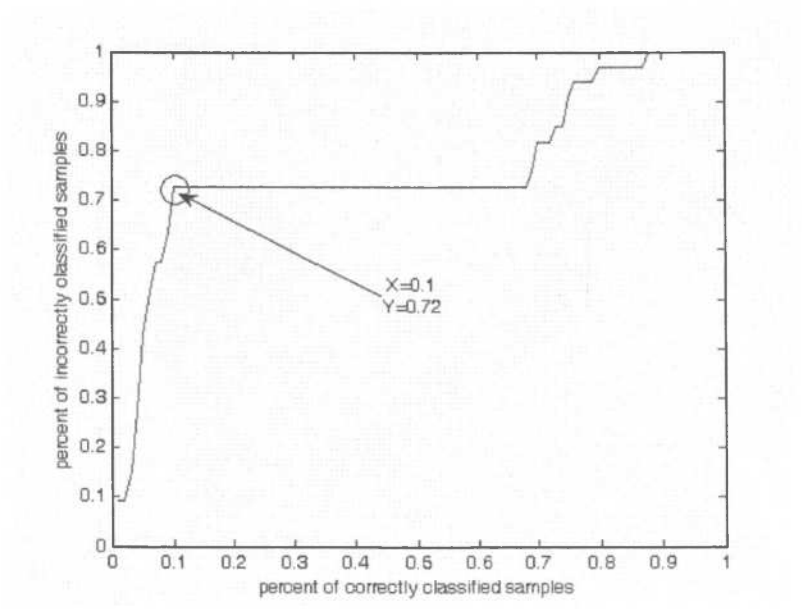


Figure 12. Defining the threshold value to keep 90% of the correctly classified samples. 72% of the misclassified samples are filtered out.

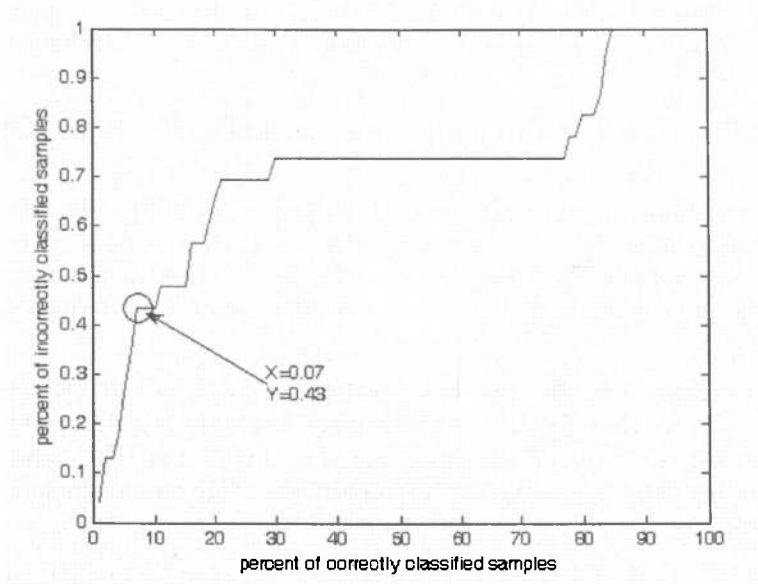


Figure 13. Defining the threshold value to keep 93% of the correctly classified samples. 43% of the misclassified samples are filtered out.

3.3. Conclusion and Future Work

A method was presented in this paper to extend the decision tree framework. The proposed extension gives the possibility to determine classification certainty.

The method proposed was tested on two sample datasets. A certainty threshold was determined to filter out the most "risky" classifications with low certainty values. Results indicate that the proposed algorithm can significantly reduce the misclassification error, in the cost of rejecting a portion of the input points, considering them as "risky" points. The algorithm takes advantage of the clear structure of the decision trees and the explicitly defined decision boundaries. The projection rules have to be determined only once after the tree growth process. Calculating the distance from the relevant decision boundaries involves projecting the input point, than measuring the distance to these projection points (section 2.4) and finally checking these points for class changes (section 2.5). This results in a reasonably fast algorithm and gives accurate distance information.

The method is being integrated into the Medical Decision Support system that is under development in the Budapest University of Technology. Currently results on mammographic data are very preliminary and will be published during the next year.

In the presented examples the key parameter when using the method is the value of the classification certainty threshold. This parameter controls the balance between the rejection rate and the classification certainty. In the demonstrative examples above this threshold was determined manually. Current research focuses on finding a method to automatically determine this threshold, which is optimal in certain means.

Acknowledgements

This work was sponsored by Research and Development Secretariat of the Hungarian Ministry of Education under contract IKTA 102/2001 and by the Hungarian Fund for Scientific Research (OTKA) under contract T046771.

REFERENCES

[1] HIGHNAM, R., BRADY, M.: *Mammographic Image Analysis*. Springer, 1999.

[2] HORVÁTH, G., VALYON, J., STRAUSZ, GY., PATAKI, B., SRAGNER, L., LASZTOVICZA, L., SZÉKELY, N.: *Intelligent Advisory System for Screening Mammography*. Proc. of the IEEE Instrumentation and Measurement Technology Conference, IMTC '2004. Como, Italy, May 18-20. Vol.3. pp. 2071-2077.

[3] SZÉKELY, N., TÓTH, B., PATAKI *A.*: *Hybrid System for Detecting Masses in Mammographic Images.* Proceedings of IMTC/04, 21th IEEE Instrumentation and Measurement Technology Conference, Como, Italy, 18-20 May 2004, pp. 2065-2070.

[4] BREIMAN, L., FRIEDMAN, J., OLSHEN, R., STONE, C.: *Classification And Regression Trees*. Chapman & Hall, 1984.

[5] MORGAN, J.N., SONQUIST, J.A.: *Problems in the Analysis of Survey Data, and a Proposal.* Journal of the American Statistical Association, 2963, 58, 415-35.

[6] MORGAN, J.N., MESSENGER, R.C.: *THAID- a Sequential Analysis Program for the Analysis of Nominal Scale Dependent Variables.* Technical Report, Survey Research Center, Institute for Social Research, Univeristy of Michigan, 1973.

[7] QUINLAN, R.J.: *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993, his website: http://www.rulequest.com

[8] *UCI Machine Learning Repository*, http://www.ics.uci.edu/~mlearn/MLRepository.html

Production Systems and Information Engineering
Volume 3 (2006), pp. 39-56

SURVEY ON VARIOUS INTERPOLATION BASED FUZZY REASONING METHODS

ZSOLT CSABA JOHANYÁK
Kecskemét College, GAMF Faculty, Hungary
Department of Information Technology
johanyak.csaba@gamf.kefo.hu

SZILVESZTER KOVÁCS
University of Miskolc, Hungary
Department of Information Technology
szkovacs@iit.uni-miskolc.hu

[Received November 2005 and accepted April 2006]

Abstract. Approximate fuzzy reasoning methods serves the task of inference in case of fuzzy systems built on sparse rule bases. This paper is a part of a longer survey that aims to provide a qualitative view through the various ideas and characteristics of interpolation based fuzzy reasoning methods. It also aims to define a general condition set for fuzzy rule interpolation methods brought together from an application-oriented point of view. The methods being presented also can be applied in the first level of systems built on hierarchical fuzzy rule bases.

Keywords: interpolative fuzzy reasoning, general conditions on rule interpolation methods, sparse fuzzy rule base

1. Introduction

Approximate reasoning methods play an important role in fuzzy logic inference systems. They are required in the case of so-called sparse rule bases. The sparse attribute denotes that the antecedent universes contain at least one partition that according to [13] can be characterized by the formula (1.1):

$$\mathrm{supp}\left(\bigcup_{k=1}^{n} A_{ik}\right) \neq X_i\,, \tag{1.1}$$

where X_i is the i^{th} input universe, A_{ik} is the k^{th} set of the partition of X_i and supp is the support.

With other words in the sparse case the rules do not cover all the input universes whereupon for some observations no rule exists whose premise would overlap the observation at least partially. Essentially a sparse rule-base takes its origin from one of the three reasons specified below:

1. The rules generated from information obtained from experts or from other sources (e.g. neural network-based learning techniques) do not cover all the possible observation values.
2. Gaps between the fuzzy sets can be arisen during the fine-tuning of the system due to the modification of the shape and position of membership functions (Fig. 1.).

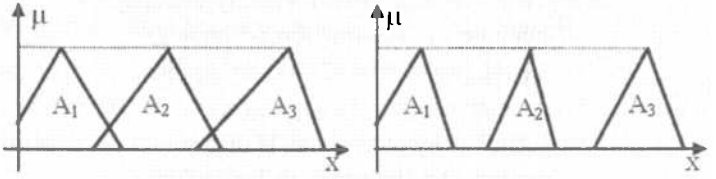


Figure 1. Before and after the fine tuning

3. The number of the state variables is so high that even if all the possible rules can be found out they could not be stored under the given hardware conditions. Taking no notice of the conditions mentioned above the number of the rules grows on. The great number of the rules increases the duration of the inference, too. Thus the performance of the system is decreasing. Making a rule-base sparse artificially [9] or/and transforming it into a hierarchical one (e.g. [26, 27]) could be a possible solution for such cases.

The classical inference methods (e.g. compositional rule of inference) methods are not able to produce an output for the observations covered by none of the rules. That is why the systems based on a sparse rule base should adopt inference techniques, which can perform approximate reasoning taking into the consideration the existing rules. The most applied used methods for this purpose are called interpolative methods.

2. General Conditions on Rule Interpolation Methods

A unified condition system related to the interpolative methods would make the evaluation and comparison of the different techniques based on the same fundamentals possible. However, according to the existing literature (e.g. [1, 7, 20, 21, 22]) can be found only partly consistent conditions and condition groups, which are put together taking different points of view into consideration. Therefore, as a step towards the unification, the conditions considered to be the most relevant ones from the application-oriented aspects are going to be reviewed and based on them, some of the well known methods are going to be compared in the followings.

General conditions on rule interpolation methods:

1. *Avoidance of the abnormal conclusion* [1, 7, 20]. The estimated fuzzy set should be a valid one. This condition can be described by the constraints (2.1) and (2.2) according to [20].

$$\inf\left\{B_{\alpha}^{*}\right\} \leq \sup\left\{B_{\alpha}^{*}\right\}, \qquad \forall \alpha \in [0,1], \tag{2.1}$$

$$\inf\left\{B_{\alpha 1}^{*}\right\} \leq \inf\left\{B_{\alpha 2}^{*}\right\} \leq \sup\left\{B_{\alpha 2}^{*}\right\} \leq \sup\left\{B_{\alpha 1}^{*}\right\}, \qquad \forall \alpha_1 < \alpha_2 \in [0,1], \tag{2.2}$$

where inf and sup are the lower and upper endpoints of the actual α-cut of the fuzzy set.

2. *The continuity of the mapping between the antecedent and consequent fuzzy sets* [1, 7]. This condition indicates that similar observations should lead to similar results.

3. *Preserving the "in between"* [7]. If the antecedent sets of two neighbouring rules surround an observation, the approximated conclusion should be surrounded by the consequent sets of those rules, too.

4. *Compatibility with the rule base* [1, 7]. This means the condition on the validity of the modus ponens, namely if an observation coincides with the antecedent part of a rule, the conclusion produced by the method should correspond to the consequent part of that rule.

5. *The fuzziness of the approximated result.* There are two opposite approaches in the literature related to this topic [22]. According to the first subcondition (5.a), the less uncertain the observation is the less fuzziness should have the approximated consequent [1, 7]. With other words in case of a crisp observation the method should produce a crisp consequence. The second approach (5.b) originates the fuzziness of the estimated consequent from the nature of the fuzzy rule base [20]. Thus, crisp conclusion can be expected only if all the consequents of the rules taken into consideration during the interpolation are singleton shaped, i.e. the knowledge base produces certain information from fuzzy input data.

6. *Approximation capability* (*stability* [e.g. 21]). The estimated rule should approximate with the possible highest degree the relation between the antecedent and consequent universes. If the number of the measurement (knot) points tends to infinite, the result should converge to the approximated function independently from the position of the knot points.

7. *Conserving the piece-wise linearity* [1]. If the fuzzy sets of the rules taken into consideration are piece-wise linear, the approximated sets should conserve this feature.

8. *Applicability in case of multidimensional antecedent universe.*

9. *Applicability without any constraint regarding to the shape of the fuzzy sets.* This condition can be lightened practically to the case of polygons, since piece-wise linear sets are most frequently encountered in the applications.

3. Surveying Some Interpolative Methods

The techniques being reviewed can be divided into two groups relating to their conception. The members of the first group produce the approximated conclusion from the observation directly. The second group contains methods that reach the target in two steps. In the first step they interpolate a new rule that antecedent part at least overlaps the observation. The estimated conclusion is determined in the second step based on the similarity between the observation and the antecedent part of the new rule.

Further on mostly the case of the one-dimensional antecedent universes are presented for the sake of easy understanding of the key ideas of the methods. As several methods need the existence of two or more rules flanking the observation, therefore it is assumed that they exist and are known. The methods are not based on the same principles, hence sometimes they approach the topic of the rule interpolation from different viewpoints.

3.1. The Linear Interpolation Introduced by Kóczy and Hirota and the Derived Methods

The first subset of the methods producing the approximated conclusion from the observation directly contains the technique introduced by Kóczy and Hirota and those ones that have been derived from it aiming its extension and improvement. First the most famous member of this group, the KH interpolation is reviewed.

3.1.1. KH Interpolation

The key idea of the method developed by Kóczy and Hirota [9] is that the approximated conclusion divides the distance between the consequent sets of the used rules in the same proportion as the observation does the distance between the antecedents of those rules (3.1). This is the fundamental equation of the fuzzy rule interpolation [1] (FEFRI). The proportions are set up separately for the lower and upper distances in the case of each α-cut.

The development of the KH method was made possible by the definition of the fuzzy distance [8] and the fact that fuzzy sets can be decomposed into α-cuts, the calculations can be made by the α-cuts and the conclusion sets can be composed from the resulting α-cuts (resolution and extension principle).

$$d_\alpha^i\left(A_1, A^*\right) : d_\alpha^i\left(A^*, A_2\right) = d_\alpha^i\left(B_1, B^*\right) : d_\alpha^i\left(B^*, B_2\right) \tag{3.1}$$

where A_1, A_2 are the antecedent sets of the two flanking rules, A^* is the observation, B_1, B_2 are the consequent sets of those rules, B^* is the approximated conclusion, i can be L or U depending on lower or upper type of the distance. The technique adopted for the determination of the consequent is an extension of the classic Shepard interpolation [16] for case of the fuzzy sets. The method requires

the following preconditions to be fulfilled: the sets have to be convex and normal with bounded support, and at least a partial ordering should exist between the elements of the universes of discourses. The latter one is needed for the definition of the fuzzy distance.

The most important advantage of the KH interpolation is its low computational complexity that ensures the fastness required by real time applications. Its detailed analysis e.g. [11, 12, 17] led to the conclusion that the result can not be interpreted always as a fuzzy set, because by some α-cuts of the estimated consequent the lower value can be higher than the upper one (Fig. 2.). The above listed publications defined application conditions that enabled the avoidance of the abnormal conclusion.

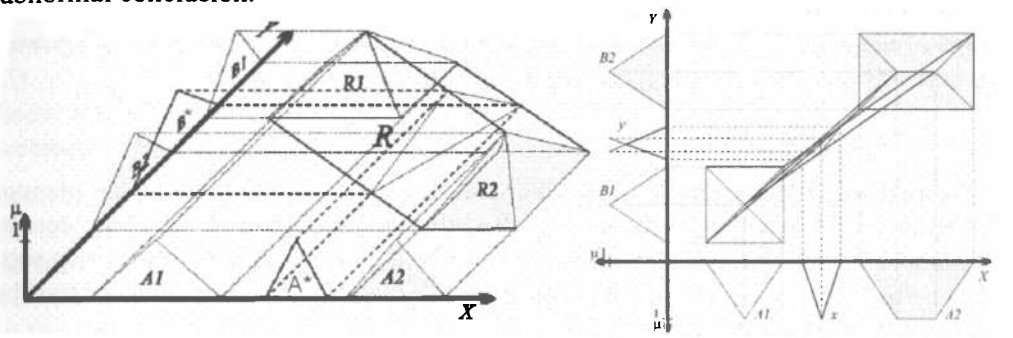


Figure 2. KH interpolation

Theoretically, an infinite number of α-cuts are needed for the exact result if there are no conditions related to the shape of the sets. However, in practice driven by need for efficiency mostly piece-wise linear generally triangle shaped or trapezoidal sets can be found, because these can be easily described by a few characteristic points. Thus supposing the method preserves the linearity completing the calculations for a finite small number of α-cuts could be enough. Although the preceding assumption is not fulfilled, in most of the applications it does not matter because of the negligible amount of the deviation [11, 12, 20].

The KH method was developed for one-dimensional antecedent universes. However, it can be applied in multi-dimensional case using distances calculated in Minkowski sense. It can be simply proven that this technique fulfils *conditions* 3, 4, 5.b and 8. The stabilized (general) KH interpolation [21] also satisfies the condition 6.

The recognition of the shortcomings of the KH interpolation has led to the development of many techniques, which modified or improved the original one or offered a solution for the task of the interpolation using very new approaches. Further on some methods improving the KH technique are reviewed emphasizing those properties which are considered by the authors to be the most important.

3.1.2. Extended KH Interpolation

Several versions of the KH interpolations were developed which allow taking into consideration more than two rules during the determination of the consequence. Their common feature is that the approximation capability of the technique is getting better with the growth of the number of the rules taken into consideration.

In [9] a technique is proposed that takes into consideration the rules weighted with e.g. the reciprocal value of the square of the distance. This approach reflects that the rules situated far away from the observation are not as important as those ones in the neighbourhood of the observation.

The authors of [21] suggest using formulas for the calculation of endpoints of α-cuts of the approximated consequence, which contain the distance on the n^{th} power, where n is number of the antecedent dimensions.

3.1.3. The VKK Method

The method developed by Vass, Kalmár and Kóczy [23] worked out the problem of abnormal conclusion introducing modified distance measures, namely the central distance and width ratio. However, it cannot be applied in case of some crisp sets. Like the KH method it does not conserve the linearity, but the deviance can be proven to be negligible [1].

3.1.4. Interpolation by the Conservation of Fuzziness (GK Method)

The starting point of the method introduced by Gedeon and Kóczy [3] is that in many applications the supports of the antecedent sets are much more larger than the support of the observation. In such cases the significant feature of the observation is its distance from the nearest flanks of the neighbouring antecedent sets [13].

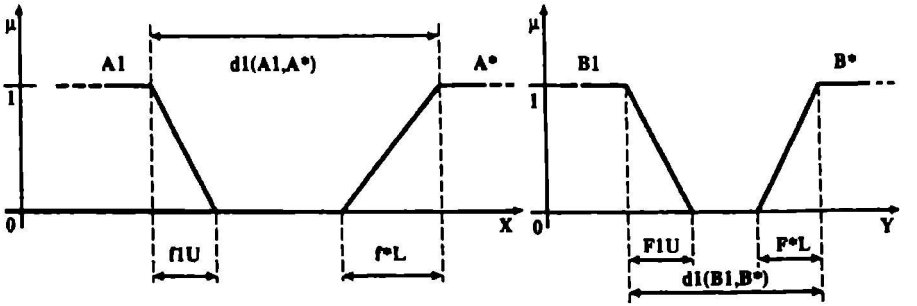


Figure 3. Distance and fuzziness measures [13]

The method was developed for the case of convex normal trapezoidal (incl. triangle shaped and crisp) fuzzy sets. It measures the distance of the sets by the Euclidean

distances among the cores (d1(A1,A*) on Fig. 3.). In multidimensional case the Euclidean sum of the distances measured in each dimension is considered. The technique introduces the term of fuzziness of a set (f1U, f*L, F1U and F*L), which is a quantity calculated separately for the left and right flank of the set as the horizontal distance of the respective endpoint of the support and the respective endpoint of the core.

During the interpolation of the conclusion (B*) the flanks are determined by calculating their fuzziness (F*L and F*U). The applied formulas take into consideration the fuzziness of the observation, the distances to the neighbouring antecedent and consequent sets and the neighbouring fuzziness of those sets. The farther sides of the flanking sets are not taken into consideration according to the principle that the interpolated conclusion should be based on "nearby" information [3]. The core points of the approximated conclusion are determined by simple linear interpolation between the nearest core points of the flanking antecedent and consequent sets.

Although the method is not an α-cut based one and has no direct connection with the FEFRI still it is presented in this group of techniques because the way of determining the estimated conclusion is in full accordance with the FEFRI [13]. The GK interpolation is conservative with respect to the degree of local fuzziness in the rule base [3]. On the basis of the literatures [3, 13] it can be stated that the method fulfils the *conditions* 1, 3, 5.a and 8.

3.1.5. Interpolation by the Conservation of Relative Fuzziness (KHG Method)

Kóczy, Hirota and Gedeon introduced a refined version of the GK method in [13]. This interpolation technique is in fully accordance with the FEFRI. It is also applicable in case of arbitrary shaped convex and normal fuzzy sets and in such crisp cases when the use of its ancestor is not possible [13]. It is extended for the multiple dimensional cases, too.

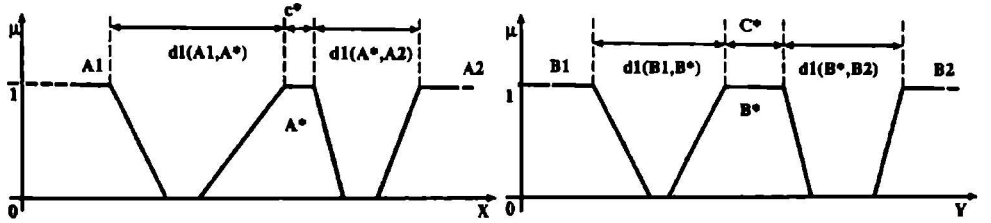


Figure 4. Antecedent and consequent distances and core lengths [13]

The length of the core of the conclusion (C*) (Fig. 4.) is calculated by multiplying the core length of the observation (c*) by the ratio of the distances of the consequent (d1(B1,B2)) and antecedent sets (d1(A1,A2)). The position of the core

is determined by the FEFRI introducing a so-called dissimilarity measure. The latter characterises the relation between two lengths, namely a fuzziness value and a distance between two fuzzy sets. The conservation of the relative fuzziness means that the left (right) fuzziness of the approximated conclusion in proportion to the flanking fuzziness of the neighbouring consequent should be the same as the (left) right fuzziness of the observation in proportion to the flanking fuzziness of the neighbouring antecedent. On the basis of the literatures [3, 13] it can be stated that the method fulfils the *conditions* 1, 3, 5.a and 8.

3.1.6. Modified α-cut based Interpolation

The modified α-cut based interpolation (MACI) [20] represents each fuzzy set by two vectors describing the left (lower) and right (upper) flank using the technique published by Yam [25]. The vectors contain the break points in case of piece-wise linear membership functions or endpoints of predefined (usually uniform distributed) α-cuts in case of smooth membership functions. The graphical representation of the vectors describing the right flanks of the sets can be seen on the figure 5. The antecedent and consequent sets are represented separately. The result will fulfil the *condition* 1 if B* is situated inside of the rectangle and above of the line l. This goal is reached through a coordinate transformation where Z_0 is substituted by the line *l*. The approximated conclusion will be crisp only if the consequent sets of the rules taken into consideration are singletons, as well.

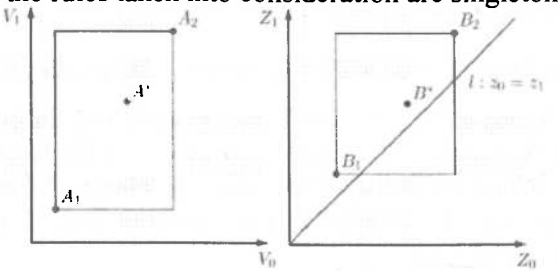


Figure 5. Graphical representation of the vectors [22]

Although this method is not conserving the linearity, the deviance is smaller than in the case of the KH interpolation [20] and the stability experienced at the KH method [19] remains. The estimated conclusion always yields fuzziness if the consequent sets of the rules taken into consideration have fuzziness [15]. The method can be used in multi-dimensional case, too [20]. It can be proven that the technique fulfils the *conditions* 1-4, 5.b, 6, 8 and 9 with the constraint that the sets should be convex and normal. Its generalized version [18] can be used in case of non-convex fuzzy sets, too.

3.1.7. The Improved Multidimensional Modified α-cut based Interpolation

The improved multidimensional modified α-cut based interpolation (IMUL) introduced by Wong, Gedeon and Tikk [24] combines the advantages of the MACI and the fuzziness conservation technique proposed by Kóczy and Gedeon in [3]. This method was developed for the case of multidimensional antecedent universe. The fuzzy sets are described by vectors containing the characteristic points, and the coordinate transformation introduced by MACI is used during the determination of the core of the approximated consequent.

The fuzziness of the observation plays a decisive role in the calculation of the flanking edges and beside this the relative fuzziness of the sets adjacent to the observation and adjacent to the approximated consequent are taken into consideration, as well. It can be proven that the technique fulfils the *conditions* 1-4, 5.a, 6, 8 and 9.

3.1.8. The HCL Interpolation

The interpolation developed by Hsiao, Chan and Lee (HCL) [4] for the case of triangle shaped convex and normal fuzzy sets combines the KH method with the interpolation of the slopes of the flanking edges.

The basic idea is that the slopes of the approximated conclusion can be calculated with the same linear combination of the respective (left or right) slopes of the consequents of the neighbouring rules as the linear combination which describes the relation between the respective flanking edges of the antecedents of the same rules and the flanking edges of the observation.

The method produces the estimated conclusion in three steps. First the two endpoints of the support are determined by means of the KH interpolation. After this the peak point of the triangle is calculated employing the relation between the slopes presented above.

The HCL interpolation cannot be classified clearly as an α-cut based technique because it is not based on the resolution and extension principles. It uses only one (usually $\alpha=0$) α-cut during the calculations. Its advantage is that it results a valid (interpretable) convex and normal fuzzy set having a little higher computational complexity than the KH method.

As a disadvantage can be mentioned that it is applicable only for the case of triangle shaped convex and normal fuzzy sets, not even crisp sets are allowed. Another drawback is the restriction expressing that the same linear combination have to describe on the left and the right side the relation between slopes of the respective edges of the antecedent sets and the slope of the respective edge of the

observation. It can be proven that the method satisfies the *conditions* 1, 3, 4 and 5.b.

3.2. Fuzzy Interpolation in the Vague Environment

The fuzzy interpolation in the vague environment (FIVE) introduced by Kovács and Kóczy [e.g. 14] puts the problem of rule approximation in a virtual space in the so-called vague environment whose conception is based on the similarity (indistinguishability) of the objects. The similarity of two fuzzy sets in the vague environment is defined by their distance weighted with the so-called scaling function, which characterizes the vague environment. The scaling function describes the shapes of all the terms in a fuzzy partition.

The challenge during the employment of this method is to find approximate scaling functions for both the antecedent and the consequent universes, which give good descriptions in case of non-Ruspini partitions, too. Scaling functions for the case of triangle and trapezoid shaped fuzzy sets are given in [14]. In consequence of the creation of the vague environments of the antecedent and consequent universes, the vague environment of the rule base is established, as well. In this environment each rule is represented by a point. If the observation is a crisp set, the conclusion, which will be crisp, can be also determined employing any interpolative or approximate technique.

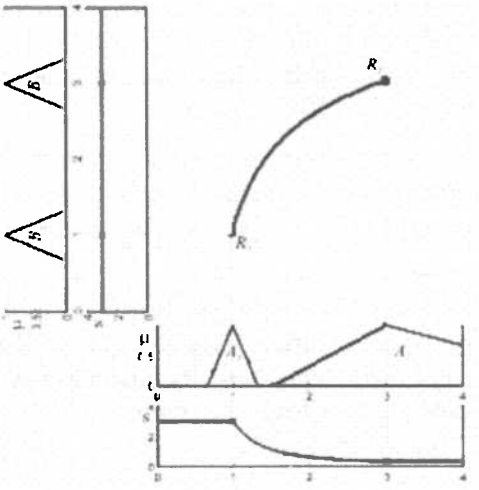


Figure 6. FIVE

The possibility of creation of the antecedent and consequent vague environments in advance ensures the fastness and hereby the applicability of the method for real-time tasks. Thus, only the interpolation of the points describing the rule base has to be made during the functioning of the system. In case of fuzzy observations the

antecedent environment should be created taking into consideration the shape of the set, which describes the input.

Figure 6. presents the partitions, the scaling function and the curve built from the points defined by the existent two rules and the points interpolated for the case of a one dimensional antecedent universe supposing crisp observations. It can be proven that the method satisfies the *conditions* 1-4, 5.a, 6 and 8.

3.3. The Generalized Methodology

Baranyi, Kóczy and Gedeon proposed in [1] a generalized methodology for the task of the fuzzy rule interpolation. In the centre of the methodology stands the interpolation of the fuzzy relation. A reference point, which can be identical with e.g. the centre point of the core, is used for the characterization of the position of fuzzy sets. The distance of fuzzy sets is expressed by the distance of their reference points. The interpolation consists of two steps.

In the first step an interpolated rule is produced, whose antecedent part has at least a partial overlapping with the observation and whose reference point has the same abscissa as the reference point of the observation. This task is divided into three stages. First with the help of a set interpolation technique the antecedent of the new rule is produced. Next the reference point of the conclusion is interpolated going out from the position of the reference points of the observation and the reference points of the sets involved in the rules taken into consideration. The applied technique can be a non-linear one, too. Hereupon the consequent set is determined similarly to the antecedent one. Several techniques are suggested in [1] for the task of set interpolation (e.g. SCM, FPL, FVL, IS-I, IS-II). In this paper the solid cutting method is presented in section 3.3.1. If λ_a denotes the ratio, in which the reference point of the observation divides the distance between the reference points of the neighbouring sets into two parts and λ_c denotes the similar ratio on the consequent side, the function $\lambda_c=f(\lambda_a)$ defines the position of the reference point of the consequent set. Through the selection of the function f() a whole family of linear ($\lambda_c=\lambda_a$) and non-linear interpolation techniques can be derived. This is also a possibility for parameterisation (tuning) of the methodology, which ensures the adaptation to the nature of the modelled system.

The approximated rule is considered as part of the rule base in the second step. The conclusion corresponding to the observation is produced by the help of this rule. As the antecedent part of the estimated rule generally does not fit perfectly to the observation, some kind of special single rule reasoning is needed. Several techniques are suggested in [1] for this task (e.g. FPL, SRM-I, SRM-II). As a precondition for all of these methods, it should be mentioned that the support of the antecedent set has to coincide with the support of the observation. Generally this is

not fulfilled. In such cases the fuzzy relation (rule) obtained in the previous step is transformed first, in order to meet this condition. For this task, in section 3.3.2, a solution is presented, which was originally suggested in [1].

Owing to the modular structure of the methodology in both of the steps one can choose from many potential methods if some conventional elements (e.g. distance measure) are used consequently. Based on the analysis in [1] and [15] the methodology can be characterized as follows. *Conditions* 1-4, 5.a and 8 are satisfied applying any of the suggested methods. In case of triangle shaped fuzzy sets the *condition* 7 is also fulfilled by those techniques. *Condition* 9 is also satisfied if SCM or FPL is used in the first step and FPL is used in the second step.

3.3.1. The Solid Cutting Method

The key idea of the solid cutting method (SCM) [2] developed by Baranyi et al. is to define vertical axes at the reference points of the two antecedent sets (A_1 and A_2) that flank the observation (A*) and after that to rotate these sets by 90° around the vertical axes. The virtual space created in such a manner is determined by the orthogonal coordinate axes S, X and μ. The rotated sets will be situated in parallel plane to the plane μxS (Fig. 7.).

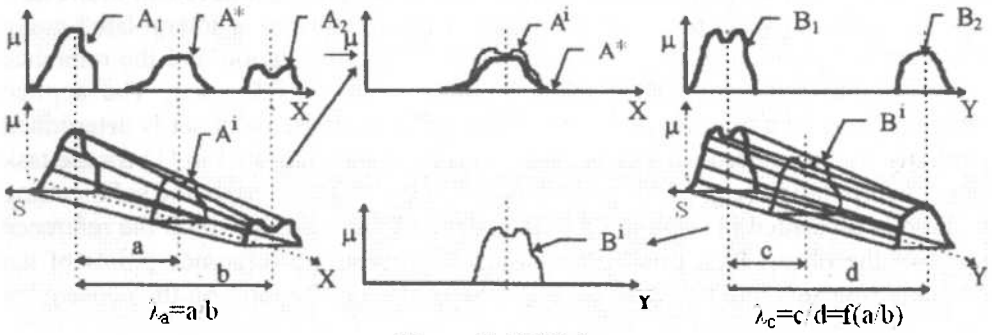


Figure 7. SCM [2]

In the next step a solid is generated fitting a surface on the contour and support of the sets. After this the solid is cut by the reference point of the observation with a plane parallel with μxS. Turning back the cross section by 90° one will obtain the antecedent set (A^i) of the estimated rule. The consequence (B^i) of the new rule is determined similarly by knowing the two consequent sets and the reference point.

3.3.2. Single Rule Reasoning Based on Transformation of the Fuzzy Relation and the Fixed Point Law

As mentioned in section 3.3, the support of the antecedent set (A^i) of the interpolated rule does not overlap generally with the observation (A*). Therefore,

the second step of the generalized methodology breaks down into two stages usually. First e.g. the technique "Transformation of the Fuzzy Relation" (TFR) transforms (stretches or shrinks) the interrelation area [1] of the new rule proportionally in order to ensure the needed coincidence of the supports. Secondly the transformed rule is fired applying e.g. the "Fixed Point Law" (FPL).

The TFR transforms the antecedent (A^i) and consequent (B^i) sets separately, but in similar way. Further on only the transformation of the set A^i is presented. First an interrelation function [1] is generated between the observation and the antecedent set in such manner that the endpoints of the support of A* are mapped to the endpoints of the support of A^i and the reference point of A^i is mapped to the reference point of A* (Fig. 8.). The rectangle defined by the endpoints of the supports of the sets is called the interrelation area.

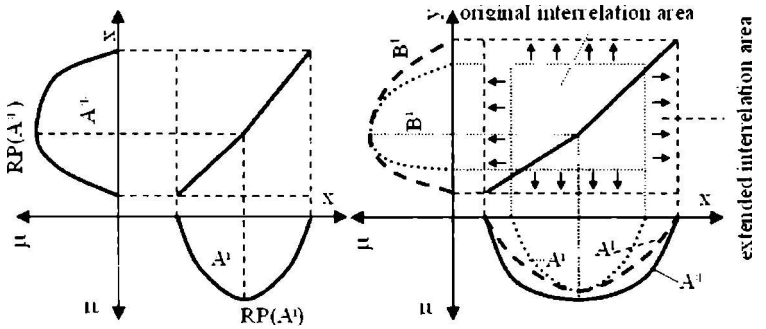


Figure 8. The two interrelation functions

The interrelation function is considered piece-wise linear containing two lines that connect the three characteristic points defined above. Next an interrelation function is generated, which describes the mapping between the points of the two sets (A^i and B^i) participating in the interpolated rule. The aim of the first stage is to modify proportionally the interrelation area of this mapping in such manner to reach the coincidence between the support of the observation and the horizontal side of the rectangle. So the support of the transformed set (A^t) is going to be the same as the support of A* The membership value of each point in A^t is equal to the membership value of its interrelated point in A^i.

In the second stage an interrelation function is generated between A* and A^t similar to the interrelation function defined in the first stage. Next, following the ideas of FPL the difference between the membership values of each interrelated point pair is calculated. This deviation is used by the determination of the approximated conclusion from the transformed consequent B^t taking into consideration the interrelation between A^t and B^t.

3.4. Interpolation with Generalized Representative Values

The method IGRV proposed by Huang and Shen [6] follows an approach similar to the generalized methodology. In the first phase, a representative value (RV) is determined for each used set. Its task is the same as the function of the reference point in the generalized methodology. It can be calculated by different formulas depending on the demands of the application. The centre of gravity played this role in the first variant of the method [5], which was developed for triangle shaped fuzzy sets. In case of an arbitrary polygonal fuzzy set the weighted average of the x coordinates of the node (break) points is suggested as RV The definition mode of the RV influences only the position of the estimated rule, but not the shape of the sets involved in the rule. Further on the Euclidean distance of the RVs of the sets are considered as the distance of the sets.

The antecedent of the approximated rule is determined by its α-cuts in such a manner that two conditions have to be satisfied. First its representative value has to coincide with the RV of the observation. Secondly the endpoints of the α-cuts of the observation have to divide the distance of the respective (left or right) endpoints of the α-cuts of the neighbouring sets in such proportion as the representative value of the observation divides the distance of the RVs of these sets. Following the same proportionality principle the RV and the shape of the consequent of the approximated rule are determined.

In the second phase, the similarity of the observation and the antecedent part of the new rule is characterized by the scale and move transformations needed to transform the antecedent set into the observation (Fig. 9.). The method was developed primordially for the case of polygonal shaped fuzzy sets. It is applicable in the case of multidimensional antecedent universes, too. In terms of classification, it can be considered as an α-cut based technique, because the scale and move transformation ratios are calculated for each level corresponding to node (break) points of the shape of sets.

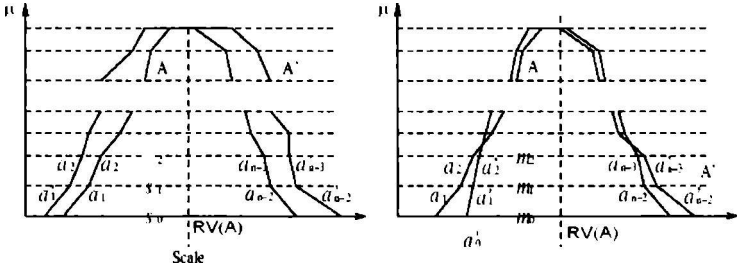


Figure 9. Scale and move transformations [6]

The method is well applicable in case of polygonal shaped sets, but the checking and constraint applications done at each α-level for the sake of the conservation of convexity increase the computational complexity of the technique.

The method can be tuned at two points. First one can choose the formula for the representative value. Secondly, the method for the calculation of the resulting transformation ratios in the case of multidimensional antecedent universes can be chosen. On the grounds of the analysis in [6] it can be stated that the method satisfies the *conditions* 1, 2, 3, 4, 5.a, 8, and 9.

Conclusions

Inference systems based on conventional (compositional) fuzzy inference methods in case of a sparse rule base cannot produce a result for all the possible observations. In such cases, where the fuzzy rule base could turn to be sparse, the system should adopt an approximate reasoning technique for the estimation of the conclusion. The surveyed fuzzy interpolation methods can be classified into two fundamental groups depending on whether they are producing the result in one or two steps. The first part of this paper gives a brief application oriented survey related to the condition structures can be expected to be fulfilled by the various fuzzy interpolation methods. The second part of the paper enumerates some of the main fuzzy interpolation methods emphasizing their basic ideas, significant characteristics and the conditions they are fulfilling from the above condition structure.

Table 1. Summary of the comparison

Method	**1**	**2**	**3**	**4**	**5.a**	**5.b**	**6**	**7**	**8**	**9**
KH			X	X		X			X	
Stabilized KH			X	X		X	X		X	
GK	X		X		X				X	
KHG	X		X		X				X	
MACI	X	X	X	X		X	X		X	X[3]
Generalized MACI	X	X	X	X		X	X		X	X
IMUL	X	X	X	X	X		X		X	X[3]
HCL	X		X	X		X				
FIVE	X	X	X	X	X		X		X	
GM with any techniques	X	X	X	X	X			X[1]	X	
GM with SCM/FPL and FPL[2]	X	X	X	X	X			X[1]	X	X
IGRV	X	X	X	X	X				X	X[3]

[1] only for triangle shaped fuzzy sets
[2] SCM or FPL in the first step and FPL in the second step
[3] the sets should be convex and normal

Table 1 contains the brief summary of the conditions that can be considered in accordance with the literature as fulfilled by the studied methods, where the columns represent the conditions, the rows indicate the methods and the cells containing an "X" denote the fulfilled conditions.

This paper has not aimed the presentation of the methods developed especially for hierarchical fuzzy rule bases although they could be very important in case of several practical application types. This topic will be covered by the next part of the survey.

REFERENCES

[1] BARANYI, P., KÓCZY, L. T., GEDEON, T. D.: *A Generalized Concept for Fuzzy Rule Interpolation.* In IEEE Transaction On Fuzzy Systems, ISSN 1063-6706, Vol. 12, No. 6, pp. 820-837, 2004.

[2] BARANYI, P., KÓCZY, L. T.: *A General and Specialised Solid Cutting Method for Fuzzy Rule Interpolation.* In J. BUSEFAL, URA-CNRS. Vol. 66. Toulouse, France, pp. 13-22, 1996.

[3] GEDEON, T. D., KÓCZY, L. T.: *Conservation of fuzziness in the rule interpolation,* Intelligent Technologies, International Symposium on New Trends in Control of Large Scale Systems, vol. 1, Herlany, pp. 13-19., 1996.

[4] HSIAO, W.-H., CHEN, S.-M., LEE, C.-H.: *A new interpolative reasoning method in sparse rule-based systems*, Fuzzy Sets and Systems 93, pp. 17-22, 1998.

[5] HUANG, Z. H., SHEN, Q.: *A New Interpolative Reasoning Method Based on Centre of Gravity*, in Proceedings of the 12th International Conference on Fuzzy Systems, Vol. 1, pp. 25 - 30, 2003.

[6] HUANG, Z., SHEN, Q.: *Fuzzy interpolation with generalized representative values*, in Proceedings of the UK Workshop on Computational Intelligence, pp. 161-171, 2004.

[7] JENEI, S.: *Interpolation and Extrapolation of Fuzzy Quantities revisited (I). An Axiomatic Approach.* Soft Computing, ISSN: 1432-7643, 5, pp. 179-193, 2001.

[8] KÓCZY, L. T., HIROTA, K.: *Ordering, distance and closeness of fuzzy sets*, Fuzzy Sets and Syst., vol. 59, pp. 281-293, 1993.

[9] KÓCZY, L. T., HIROTA, K.: *Rule interpolation by α-level sets in fuzzy approximate reasoning*, In J. BUSEFAL, Automne, URA-CNRS. Vol. 46. Toulouse, France, pp. 115-123, 1991.

[10] KÓCZY, L. T., KOVÁCS, SZ.: *Linearity and the CNF property in linear fuzzy rule interpolation.* In Proc. 3rd IEEE Int. Conf. Fuzzy Systems (FUZZ-IEEE '94), Orlando, FL pp. 870-875, 1994.

[11] KÓCZY, L. T. KOVÁCS, SZ.: *Shape of the fuzzy conclusion generated by linear interpolation in trapezoidal fuzzy rule bases*, in Proc. 2nd Eur. Congr. Intelligent Techniques and Soft Computing, Aachen, Germany, pp.. 1666-1670, 1994.

[12] KÓCZY, L. T., KOVÁCS, SZ.: *The convexity and piecewise linearity of the fuzzy conclusion generated by linear fuzzy rule interpolation*, In J. BUSEFAL 60, Automne, URA-CNRS. Toulouse, France, Univ. Paul Sabatier, pp. 23-29, 1994.

[13] KÓCZY, L.T., HIROTA, K., GEDEON, T. D.: *Fuzzy rule interpolation by the conservation of relative fuzziness*, Technical Report TR 97/2. Hirota Lab, Dept. of Comp. Int. and Sys. Sci., Tokyo Inst. of Techn., Yokohama, 1997.

[14] KOVÁCS, SZ., KÓCZY, L.T.: *Application of an approximate fuzzy logic controller in an AGV steering system, path tracking and collision avoidance strategy*, Fuzzy Set Theory and Applications, Tatra Mountains Mathematical Publications, Mathematical Institute Slovak Academy of Sciences, vol.16, pp. 456-467, Bratislava, Slovakia, 1999.

[15] MIZIK, S.: *Fuzzy Rule Interpolation Techniques in Comparison*, MFT Periodika 2001-04. Hungarian Society of IFSA, Hungary, 2001. http://www.mft.hu

[16] SHEPARD, D.: *A two dimensional interpolation function for irregularly spaced data*, Proc. 23rd ACM Internat. Conf., 517-524, 1968.

[17] SHI, Y., MIZUMOTO, M. WU, Z. Q.: *Reasoning conditions on Kóczy's interpolative reasoning method in sparse fuzzy rule bases*, Fuzzy Sets Syst., vol. 75, pp. 63-71, 1995.

[18] TIKK, D., BARANYI, P., GEDEON, T. D., MURESAN, L.: *Generalization of the Rule Interpolation method Resulting Always in Acceptable Conclusion*, Bratislava, Slovakia, Tatra Mountains, Math. Inst. Slovak Acad. Sci., vol. 21, pp. 73-91, 2001.

[19] TIKK, D., BARANYI, P., YAM, Y., KÓCZY, L. T.: *Stability of a new interpolation method*, in Proc. IEEE Conf. Syst. Man. and Cybern. (SMC '99), Tokyo, Japan, pp. III/7-III/9, 1999.

[20] TIKK, D., BARANYI, P.: *Comprehensive analysis of a new fuzzy rule interpolation method*, IEEE Trans Fuzzy Syst., vol. 8, pp. 281-296, June 2000.

[21] TIKK, D., JOÓ, I., KÓCZY, L. T., VÁRLAKI, P., MOSER, B., GEDEON, T. D.: Stability *of interpolative fuzzy KH-controllers*. Fuzzy Sets and Systems, 125(1) pp. 105–119, January 2002.

[22] TIKK, D.: *Investigation of fuzzy rule interpolation techniques and the universal approximation property of fuzzy controllers*, Ph. D. dissertation, TU Budapest, Budapest, 1999.

[23] Vass, Gy., Kalmár, L., Kóczy, L. T.: *Extension of the fuzzy rule interpolation method*, in Proc. Int. Conf. Fuzzy Sets Theory Applications (FSTA '92), Liptovsky M., Czechoslovakia, pp. 1-6, 1992.

[24] Wong, K. W., Gedeon, T. D., Tikk, D.: *An improved multidimensional α-cut based fuzzy interpolation technique*, in Proc. Int. Conf Artificial Intelligence in Science and Technology (AISAT'2000), Hobart, Australia, pp. 29–32, 2000.

[25] Yam, Y. Kóczy, L. T.: *Representing membership functions as points in high dimensional spaces for fuzzy interpolation and extrapolation.* Technical Report CUHK-MAE-97-03, Dept. Mech. Automat. Eng., The Chinese Univ. Hong Kong, Hong Kong, 1997.

[26] Kóczy, L.T., Hirota, K.: *Modular rule bases in fuzzy control*, FUZZ-IEEE 93, Aachen, pp. 606-610, 1993.

[27] Kóczy, L.T., Hirota, K.: *Interpolation in structured fuzzy rule bases*, FUZZ-IEEE 93, San Francisco, pp. 803-808, 1993.

Production Systems and Information Engineering
Volume 3 (2006), pp. 57-70

FUZZY BASED LOAD BALANCING FOR J2EE APPLICATIONS

PÉTER MILEFF
University of Miskolc, Hungary
Department of Information Engineering
mileff@ait.iit.uni-miskolc.hu

KÁROLY NEHÉZ
University of Miskolc, Hungary
Department of Information Engineering
nehez@ait.iit.uni-miskolc.hu

[Received November 2005 and accepted January 2006]

Abstract. The growth of Internet services during the past few years has increased the demand for scalable distributed computing systems. E-commerce systems concurrently serve many clients that transmit a large, number of requests. An increasingly popular and cost effective technique to improve server performance is *load balancing*, where hardware and/or software mechanisms decide which server will execute the client request. Load balancing mechanisms distribute client workload among server nodes to improve overall system responsiveness. Load balancers have emerged as a powerful new technology to solve this.

This paper focuses on a new generation of adaptive/intelligent dynamic load balancing technique, which based on J2EE technology and can be practical in J2EE application servers. The paper discusses in detail both the theoretical model of load balancing and its practical realization. The effectiveness of the new balancing method will be demonstrated through exact measurement results compared with former traditional non-adaptive methods.

Keywords: Distributed systems, Adaptive Load Balancing, J2EE Application server, JBoss

1. Introduction

As the number of concurrent requests is increased on a standalone server, so the application exceeds the pre-estimated respond time, because the work load is too much on the server machine. At this time, there are two options to solve this problem: using faster machines or using multiple machines parallel. The first solution can be expensive and limited by the speed of a standalone machine. The second choice is more straightforward: deploy the same application on several

machines and redirect client requests to those machines. The system is transparent from outside, which means that client applications perceive a standalone very-fast server with one accessible IP address (see Figure 1). To achieve the performance and transparency, load balancing algorithms must be utilized.

Load balancing can improve system performance by providing better utilization of all resources in the whole system, which consists of computers connected by local area networks. The main objective of load balancing is to reduce the mean response time of requests by distributing the workload [5].

1.1. Theoretical possibilities of realizing load balancing on OSI Layers

The OSI model was developed as a framework for developing protocols and applications that could interact seamlessly. The OSI model consists of seven layers and is referred to as the 7-Layer Networking Model [2]. Each layer represents a separate abstraction layer and interacts only with its adjoining layers. Load balancing mechanism can be realized on the Layer 3 - 7. OSI levels 3 and 4 can be supported balancing mechanisms via network router devices. On layers 5 and 7, 'URL Load Balancing' can be achieved. A lively example of 'URL Load Balancing' can be the following: the URL may be static (such as *http://www.xxx.net/home)* or may be a cookie embedded into a user session. An example of URL load balancing is directing traffic to *http://www.xxx.net/documents* through one group of servers, while sending *http://www.xxx.net/images* to another group. URL load balancing can also set persistence based on the "cookie" negotiated between the client and the server.

1.2. Network-based load balancing

This type of load balancing is provided by network router devices and domain name servers (DNS) that service a cluster of host machines. For example, when a client resolves a hostname, the DNS can assign a different IP address to each request dynamically based on current load conditions. The client then contacts the designated server. Next time a different server could be selected for its next DNS resolution. Routers can also be used to bind a TCP flow to any back-end server based on the current load conditions and then use that binding for the duration of the flow. High volume Web sites often use network-based load balancing at the *network* layer (layer 3) and *transport* layer (layer 4). Layer 3 and 4 load balancing (referred to as "switching" [1]), use the IP address/hostname and port, respectively, to determine where to forward packets. Load balancing at these layers is limited, however, by the fact that they do not take into account the content of client requests. Higher-layer mechanisms such as the so-called layer 5 switching described above perform load balancing in accordance with the content of requests, such as pathname information within a URL.

1.3. Operating System - based load balancing

This type of load balancing is provided by distributed operating systems via *clustering*, *load sharing*, or *process migration* mechanisms. For instance Microsoft provides a new clustering possibility: Microsoft Cluster Server (MSCS) This special Microsoft software provides services such as failure detection, recovery, and the ability to manage the servers as a single system. Clustering is a cost effective way to achieve high-availability and high-performance by combining many commodity computers to improve overall system processing power. Processes can then be distributed transparently among computers in the cluster. Clusters generally employ load sharing and process migration. Balancing load across processors – or more generally across network nodes – can be achieved via *process migration* mechanisms, where the state of a process is transferred between nodes. Transferring process state requires significant platform infrastructure support to handle platform differences between nodes. It may also limit applicability to programming languages based on virtual machines, such as Java.

1.4. Middleware-based load balancing

This type of load balancing is performed in middleware products, often on a per-session or per-request basis. For example, layer 5 switching has become a popular technique to determine which Web server should receive a client request for a particular URL. This strategy also allows the detection of "hot spots," *i.e.*, frequently accessed URLs, so that additional resources can be allocated to handle the large number of requests for such URLs.

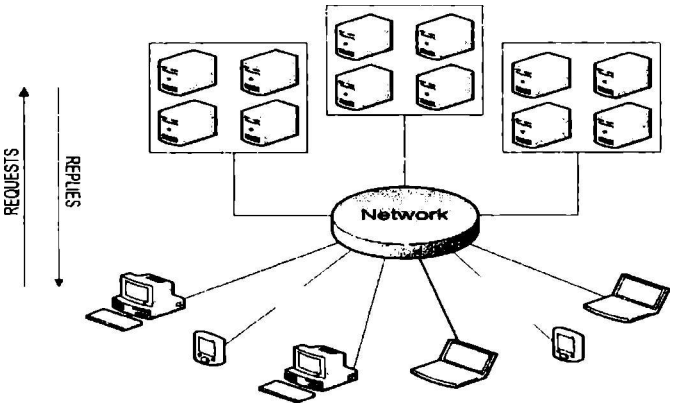


Figure 1. Horizontal load balancing

Middleware-based load balancing can be used in conjunction with the specialized network-based and OS-based load balancing mechanisms outlined above. It can also be applied on top of consumer level (COTS) networks and operating systems, which helps reduce cost. In addition, middleware-based load balancing can provide

semantically rich customization possibilities to perform load balancing based on a wide range of application-specific load balancing conditions, such as run-time I/O vs. CPU overhead conditions.

2. The practical approach of balancing problems

After we have surveyed the theoretical bases of load balancing, we direct our attention to a more practical scope of the problem.

A dynamic load balancing can be either *preemptive* or *non-preemptive*. A non-preemptive mechanism transfers only jobs that have just arrived, while a preemptive mechanism transfers jobs at any time, even when the jobs are in execution. Because preemptive mechanism are more costly than non-preemptive one and most of the benefit that can potentially be achieved through dynamic load balancing can be achieved using non-preemptive transfer only, non-preemptive transfers are usually used. Various proposed dynamic balancing methods are based on several policies. Three important ones among them are the transfer policy, the location policy and the selection policy, which decide when, where and what jobs should be transferred respectively. Much work [2][4] has been published on the design of transfer and location policy but very few on the selection policy.

Balancing policy: When designing a load balancing service it is important to select an appropriate algorithm that decides which server node will process each incoming request. For example, applications where all requests generate nearly identical amounts of load can use a simple Round-Robin algorithm, while applications where load generated by each request cannot be predicted in advance may require more advanced algorithms. In general, load balancing policies can be classified into the following categories:

- *Non-adaptive* – A load balancer can use non-adaptive policies, such as a simple Round-Robin algorithm or a randomized algorithm, to select which node will handle a particular request.
- *Adaptive* – A load balancer can use adaptive policies that utilize run-time information, such as CPU and disk I/O utilization, network loading.

This paper presents a new adaptive load balancing method, which efficiency are verified with help of many simulations.

2.1. Problem of real-time load balancing

Client requests arrive over the network and start a new process in memory. Each process runs separated from one another and rivals in gaining available resources. The objective of load balancers is to distribute these processes among the individual server instances in such a way that response time of processes will be minimal. Because the characteristic of the running tasks can be very various, so it

is essential to use an adaptive load balancing algorithm, which tries to distribute tasks in an intelligent way using *load information* of the nodes. This is a very difficult objective, because balancer must conform to the given job. If it could be known in advance what type of task will be arrive, then the scheduling algorithm could easily choose the most suitable server for the task. However, the type of tasks knows in general only the client. So the traditional algorithms like Round-Robin or Random access can be usable only with a certain type of tasks.

Leland and Ott [4] analyzed 9.5 million UNIX processes and found that there are three type of processes: CPU intensive processes use great amount of CPU cycles but do a little I/O operations; I/O intensive processes do a great deal of I/O but use a little CPU cycles; canonical processes do a little I/O and use a little CPU cycles. The amount of processes using great amount of CPU cycles and doing a great deal of I/O is extremely small.

Cabrera[5] analyzed 122 thousand processes running on VAX11/785 and found that mean lifetime of processes is 400 ms, the lifetime of 78% of processes is shorter than one second, 97% of processes terminate within 8 seconds. The author concluded that only long live jobs should be candidates for load balancing due to the overhead costs involved. If the running time of the job is rather short, then load balancing can loose its importance.

3. Concept of an Intelligent Load Balancer

Creating an efficient Load Balancer is a very difficult objective. Of course, there are many theoretical load balancing methods, but many times the practical model does not make these implementation and efficiency possible. Finding suitable and optimal method for balancing, it is essential to have the deepest knowledge of the specific system. Our aim was to develop a new load balancer for JBoss application servers, because only three types of load balancers are available in JBoss cluster: Round Robin, First Available, and Random balancer.

Before we examine the theoretical model of the new Load Balancer, we make a short overview of JBoss cluster.

3.1. The JBoss cluster

JBoss is an extensible, dynamically configurable Java based application server which includes a set of J2EE compliant components. JBoss is an open source middleware, in the sense that users can extend middleware services by dynamically deploying new components into a running server.

A cluster is a set of nodes. These nodes generally have a common goal. A node can be a computer or, more simply, a server instance (if it hosts several instances). In JBoss, nodes in a cluster have two common goals: achieving Fault Tolerance and

Load Balancing through replication. These concepts are often mixed. JBoss currently supports the following clustering features [9]:

Automatic discovery. JBoss cluster nodes automatically discover each other when they boot up with no additional configuration. Nodes that join the cluster at a later time have their state automatically initialized and synchronized by the rest of the group.

Fail-over and load-balancing features for:

- JNDI,
- RMI (can be used to implement your own clustered services),
- Entity Beans,
- Stateful Session Beans with in memory state replication,
- Stateless Session Beans

- HTTP Session replication with Tomcat (3.0) and Jetty (CVS HEAD)

Dynamic JNDI discovery. With its JMX-based Microkernel architecture JNDI clients can automatically discover the JNDI context.

- Cluster-wide replicated JNDI tree. It is replicated across the entire cluster. It requires no additional configuration and boots up with a cluster-enabled JBoss configuration. Remote JBoss JNDI clients can also implicitly use multicast to discover the JNDI tree.

Farming. JBoss farming takes this hot-deployment feature cluster-wide. Copying a deployable component to just one node's deployment directory causes it to be deployed (or re-deployed) across the entire cluster. Removing a component from just one node's deployment directory causes it to be undeployed across the entire cluster.

- Pluggable RMI load-balance policies. We used this feature to develop our load balancer.

JBoss uses an abstraction framework to isolate communication layers like JavaGroups. This was done so that other third-party group communication frameworks could be incorporated into JBoss seamlessly and easily. This framework also provides the tools and interfaces to write own clusterable services and components to plug into the JBoss JMX backbone.
Utilizing these flexibilities of the JBoss system we developed a new load balancer, which will be presented in details.

3.2. Architecture of the balancer

Before going into the details, first we examine the theoretical model, which is shown in Figure 2:

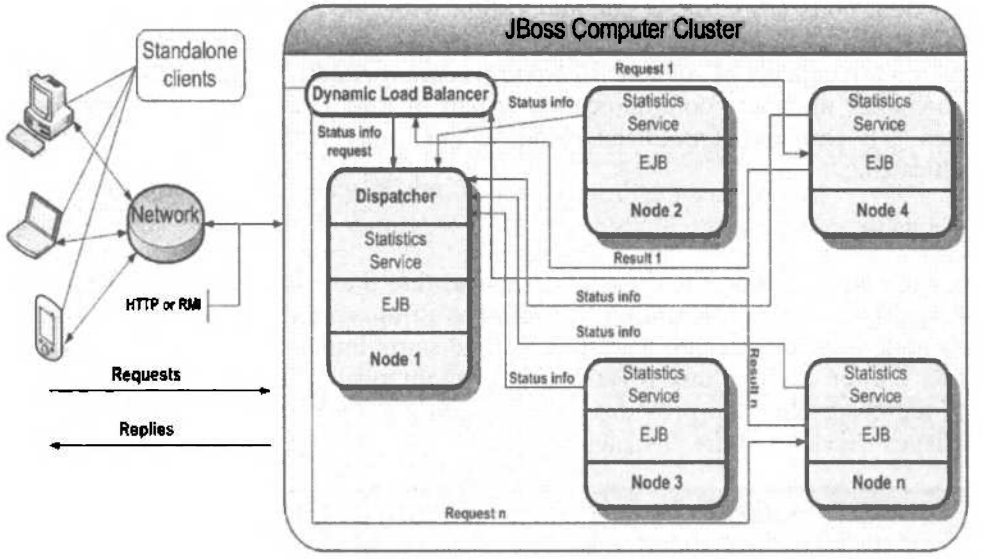


Figure 2. JBoss Load Balancer Architecture

The theoretical functionality of the balancer is the following: Standalone clients initiate requests over the network through HTTP protocol or RMI to the JBoss cluster. The JBoss cluster can be a complex of homogeneous or inhomogeneous computers [9]. JBoss application server runs on these in cluster mode. Of course, more clients can initiate a request at the same time to the cluster, so the cluster must fulfill more than one request parallel. Incoming requests are received and directed to the compliant node by the intelligent load balancer. So its objective is to choose the most ideal node, based on the collected *load information* by the Dispatcher MBean. To choose the ideal node is not an easy task. The main objective of the balancer is to realize a more effective task-division, which response time can be better than former algorithms. In the following, we show a detailed explanation of the practical realization.

3.3. Components of the Balancer

The architecture of our Balancer essentially can be divided into three individual components: the *Statistics Service*, the *Dispatcher*, and the *Scheduler* as well. The individual units are in close communication with one another (within one JVM), none of them can operate without the others. At present, component connections work on the concept of *Remote Method Invocation (RMI)*, but the further objective is to change the entire communication or part of it to a new TreeCache method of JBoss [7]. Utilizing TreeCache response time may be shorter because it uses multicasting.

3.3.1. Statistics Service

We can consider from the description above, that Statistics Service is responsible for load information. Naturally, this unit has to be started on each node. When a new node joins the cluster, Statistics Service starts immediately on it, because the JBoss cluster deploys this MBean [9][1] automatically. This service attempts to find the Dispatcher and provides data to it. *Figure 3* shows the architecture of the Statistics Service and the Dispatcher:

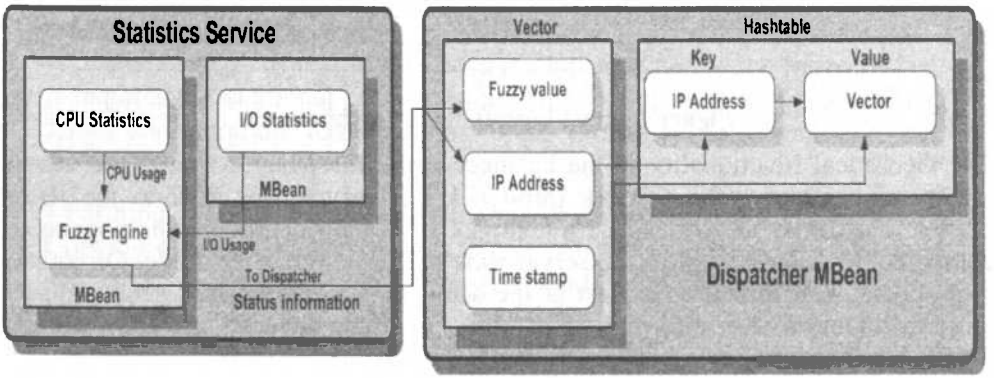


Figure 3. Elements of Statistics Service

Figure 3 shows that Statistics Service consists of three subcomponents: *CPU -*, *I/O Statistics* and *Fuzzy Engine*. The functionality of these arises from those names: CPU Statistics provides CPU usage and I/O Statistics provides information about I/O usage of a specific node. CPU Statistics and Fuzzy Logic components are represented collectively an MBean *(Managed Bean)*, however I/O Statistics is an another separate MBean. In JBoss system, each MBean can be considered as services. The sufficient node-information is essential to the compliant operating of the balancer. In fact, Java classes are running in a virtual machine on each host, therefore it does not make it possible to query the load information directly from the operating system. For this reason we had to evolve individual methods and had

to utilize operating system specific resources. Nevertheless these resources are operating system dependent.

The current version of the balancer works on MS Windows Systems, but further objective is to create Linux/Unix version too. Since Java 1.5 appeared on the market, it become possible to measure CPU average usage with Java Management Extension technology, using the built in OperatingSystemMXBean class. It has a function called *getProcessCpuTime()*, which can query the CPU time of the specific JVM in nanoseconds, from which the average CPU usage can be computed. The CPU usage can be query direct from the operating system as well, but in this case the efficiency of the balancer can degrade to a great extent. The reason of this is that: MS Windows operating system updates the data of the *Performance Monitor* every 1000 milliseconds (one second), which makes impossible to schedule short tasks. JBoss system can work with 50 ms sample time, but in this instance data acquisition is fulfilled in every 100 ms.

Acquiring I/O information is much harder task. Getting the required information we need to call operating system level methods via JNI *(Java Native Interfaces)* technology, which makes possible to merge the C/C++ and the Java programming language. However operating system is a limiting factor again, because data are only updated in every 1000 milliseconds. If client I/O requests are not so frequent, this limit is enough in practice.

Before we change to the consideration of the Fuzzy Engine, it is necessary to make a mention of a relevant feature of the statistics collector MBeans. All the nodes send information to the Fuzzy Engine, when the average usage of these, is smaller than 100%. This is the most essential condition of the operating of the balancer that will be detailed below.

The Fuzzy Engine is responsible for the part of adaptivity of the balancer. It is integrated in the Statistics Service and gathers information sent by I/O and CPU services and deducts a *fuzzy* value between 0 and 1 supported by a preset *Fuzzy Engine*. This fuzzy value will be sent to the Dispatcher that stores it in a hashtable. Current version of Balancer use three fuzzy linguistic variables: one for I/O and an other for CPU utilization and the third one indicates the service capability of a server node. First two variables are considered as input variables and third one as output variable. Both input variables are defined with three membership functions. Output server capability is defined with six membership functions. Further aim is to fine the shape of membership functions using a fuzzy-neuro engine. In Figure 4, all membership functions of fuzzy variables can be seen.

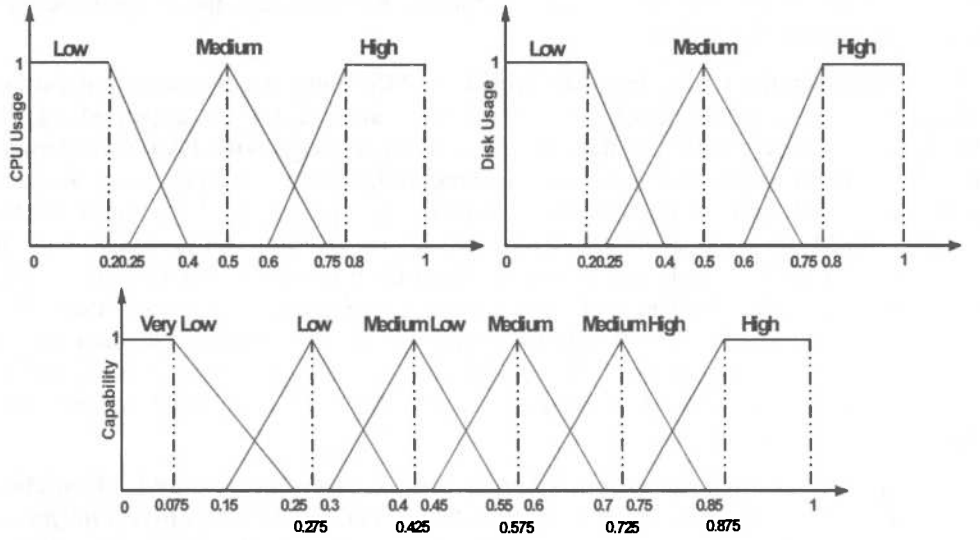


Figure 4. The Linguistic Variables of the Fuzzy Engine

3.3.2. The Dispatcher

The Dispatcher is the second most important part of the Load Balancer. It is also realized by MBean. Its objective is to store status information sent by the nodes in hashtable structure. Figure 3 shows the architecture of the Dispatcher. The sent and forwarded information consist of two parts: fuzzy values and the IP address of the specific node. IP address is essential to identify the nodes. The information gets into a hashtable entry as a vector, together with the time of arrival (*time stamp*). As *Figure 3* shows, the key of the hashtable is the IP address, because it is always individual. By the discussion of the Statistics Service we have mentioned, that there is a condition, whereas a node only send information to the Dispatcher, when its load is fewer than 100%. This effects in Dispatcher that, the belonging stored information of the hashtable entry will not be updated. The balancer will make a decision based on the timestamp value, which information is current and which is not.

The Dispatcher is deployed only on one node in the cluster. It makes no difference on which one, but starting on the fastest node is the best. The connection between the Statistics Services and the Dispatcher is dynamic. At startup time, each node finds and stores the network address of the node, on which the Dispatcher runs.

3.3.3. The Balancer

After preparation of data, the work of the balancer is no more so difficult. However we have to pay attention at the optimal implementation, because the least mistake can also cause big response time decrease. The balancer is a java class implemented a *CustomLoadBalancePolicy* interface, which is functionally part of the JBoss base interfaces.

Its theoretical workflow is the following: The balancer makes decision on the bases of the status information collected from the server nodes. It considers those information valid, which arrived within 150 ms. The highly loaded nodes do not send any information to the Dispatcher, so naturally the balancer does not give to one of them a new task. The balancer will choose the node with the best fuzzy engine value. However in case of a big loaded cluster it can often occur that all of the nodes are loaded fully and none of them makes a sign. Nevertheless, at this time the balancer have to choose one of them, but the question is which one.

Many solution methods have sprung up, because this case needs more consideration. Such method is needed, which can efficiently distribute the work among the highly loaded nodes. The first solution is the random distribution. It can be good, or can be very bad because of the random distribution. For instance if random balancer gives the work to a node, which is slower than the others, and of course also loaded on 100%, the response time of the system will be very low. We implemented this method as *Random Intelligent Balancer*, the results can be seen in *Table 1*. The method is proved a little better, which gives the work to that node, which average non-response time is the least, if every node are out of time constraint (*Average Intelligent Balancer*).

A very important element of the balancer is the following: in current version of the balancer a node can only get a work twice one after the other, if its CPU usage does not correspond to the stored value at the giving out of the previous work and also this value is more little, than the value of all the nodes. This condition came into the balancer therefore, because when almost more clients all at ones give their requests parallel, then without this condition the same node receive the request of more clients, because the requests are so close to one another, that the data of the balancer could not update so quickly.

4. Test and results

The testing process has been carried out on a JBoss cluster, consisting 7 homogeneous PC-s. Each machine had Pentium III 733 MHz CPU with 256 MByte RAM. Machines were connected via 100Mbps Ethernet network. Utilized operation system was Windows 2000 SP5. Application server version was JBoss 3.2.5 'WonderLand'.

Simulated client requests were carried out with a generic professional simulation environment: Apache JMeter [8]. During testing process, server machines where slowed-down randomly with a special Loader-MBean emulating I/O or CPU load. Loader-MBean is used for emulating other client requests and other applications that are parallel launching on the server nodes.

We have started simulations with one client and then we increased the number of clients to seven. In the course of all simulation we have tested all algorithms three times then we represented these average results in Figure 5. The diagram shows properly that the results of the Round-Robin in every case fell short of the results of the Intelligent Balancer.

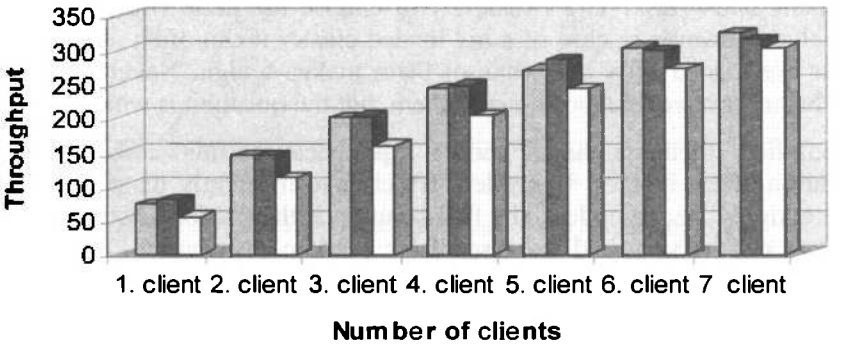


Figure 5. Test results

If we examine the results, we can see that the value of the Throughput is raised with the increasing number of the clients, although it is not in direct ratio. The more clients initiate request to the cluster, the more clients share the CPU. Exactly, for this reason there is no point about increasing the number of clients like number of nodes in the course of simulations, because at this time scheduling loses its importance.

Of course, it depends on the type of the tasks requested the clients and on that, in what extent they require the resources. In the course of seven homogeneous nodes the optimal distribution is, if all of them get one task. Artificial loads run in random time on the nodes independently from the client requests, which load the nodes for a period of time and to a certain extent. It is possible, if there are many client requests at the same time, then the cluster will be overloaded. At that time every node are at maximum load. Whereas at such time scheduling is impossible, therefore the best solution is that, if we distribute the tasks optimal among the nodes till then, while scheduling will be become possible. The Balancer does it in

two ways: with random node-choosing (Random Intelligent Balancer) and with using average response time. One node could not get two tasks one after another.

The Figure 5 shows the results both of the algorithms. It is easy to see that increasing the number of clients - which means that more task get into the system – the response time of Round-Robin and the Intelligent Balancer approach better and better approximate the theoretical maximum.

In case of inhomogeneous nodes certainly we can reach much better response time, but of course it depends on the inhomogenity of the nodes. The following table summarizes, how much speed increase can be achieved utilizing the new Balancer compared with Round-Robin algorithm. Results highly depend on the type of tasks: a task to what extend claims the capacity of a node. In our test environment, execution time of a task was 500 ms on a non-loaded server node. Client requests followed each other within 500ms time interval and plus-minus 200ms uniform random time. The aim of random interval is to simulate realistic non-predicted client requests. Based on the test results, it is clear that our intelligent balancer algorithm has better performance than Round-Robin algorithm.

Table 1. Balancing Algorithms comparison

Balancer Type	Speed Improvement / client						
	1. client	2. client	3. client	4. client	5. client	6. client	7. client
Random Intelligent Balancer	25%	23%	20%	16%	10%	10%	7%
Average Intelligent Balancer	30%	23%	21%	18%	14%	9%	4%

During tests intelligent load balancer and only Round-Robin algorithm was compared, because we experienced that in our simulations Round-Robin algorithm was definitely better than other classic methods like: First Available and Random balancer algorithms. Thus our aim was to outstrip this traditional non-adaptive method.

5. Conclusion

An intelligent fuzzy-based Load Balancer Application and its test results have been presented in this paper. Continuing work will focus on further developing and implementing more flexible XML based configuration possibilities and redesign communication between server nodes and the dispatched session bean utilizing the new JBoss TreeCache introduced by the latest JBoss version 4.0.

Acknowledgements

The research and development summarized in this paper has been carried out by the Production Information Engineering and Research Team (PIERT) established at the Department of Information Engineering and supported by the Hungarian Academy of Sciences. The financial support of the research by the afore-mentioned source is gratefully acknowledged.

REFERENCES

[1] LINDFORS, J.,FLEURY, M., THE JBOSS GROUP: JMX: *Managing J2EE with Java Management Extensions*. SAMS Publishing Inc., 2002.

[2] BASNEY, J., LIVNY, M.: "Deploying a High Throughput Computing Cluster," *High Performance Cluster Computing*, vol. 1, May 1999.

[3] O'RYAN, C., KUHNS, F., SCHMIDT, D. C., OTHMAN, O., PARSONS, J.: "The Design and Performance of a Pluggable Protocols Framework for Real-time Distributed Object Computing Middleware", in *Proceedings of the Middleware 2000 Conference*, ACM/IFIP, Apr. 2000.

[4] SCHMIDT, D., STAL, M., ROHNERT, H. BUSCHMANN, F.: *Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects*. Wiley, 2000.

[5] CABRERA, L.M.: *"The influence of workload on load balancing strategies"*, in Proc. Summer USENIX Conf., pp. 446-458, June 1986.

[6] LELAND, W., OTT, T.: *"Load balancing heuristics and process behavior"*, in Proc. ACMSIGMETRICS Conf. Measurement and Modeling of Computer Syst., May 1986.

[7] SHIRAZI, J.: *Java Performance Tuning, Second Edition*, O'Relly, 2003.

[8] JMETER GENERIC SIMULATION ENVIRONMENT, http://jakarta.apache.org/jmeter, 2005. (Apache Jakarta JMeter)

[9] JBOSS – LEADING J2EE OPEN SOURCE APPLICATION SERVER, www.jboss.org, 2005.

[10] FUZZY LOGIC SYSTEMS: http://www.seattlerobotics.org/encoder/mar98/fuz/flindex.html, 2005.

[11] KOPPARAPU, C.: *Load Balancing Servers, Firewalls, and Caches*, Wiley, 2002.

Production Systems and Information Engineering
Volume 3 (2006), pp. 71-82

COLLABORATIVE INVENTORY CONTROL POLICIES IN SUPPLY CHAINS

PÉTER MILEFF
University of Miskolc, Hungary
Department of Information Engineering
mileff@ait.iit.uni-miskolc.hu

KÁROLY NEHÉZ
University of Miskolc, Hungary
Department of Information Engineering
nehez@ait.iit.uni-miskolc.hu

[Received November 2005 and accepted January 2006]

Abstract. The inventory control is a critical problem of the management of supplier companies for several decades. In recent years numerous new supply chain and inventory control models have been developed to support management decisions. In this paper, we investigate the classical one-customer and one-supplier problem with an analytical, event oriented model. Our basic aim is to determine an optimal inventory holding and production policy for suppliers, which means determining of an optimal and a critical inventory stock-level. The expected (average) cost of supplier using the optimal policy will be minimized under stochastic customer demands. We examine this problem by means of an own simulation method and analysis of the results will also be discussed.

Keywords: Inventory Control (IC), Supply Chain Management (SCM), Stochastic Demand

1. Introduction

Since the early 90's, the business environment of companies on the field of mass production has importantly altered. The demand rate for their products remained on high level but a lot of new requirements appear on the market. The lifecycle of products have become shorter. Customer needs for new forms, designs, special packing or better product properties have greatly increased. Generally these companies assemble and bundle their products from components originated from suppliers. This development process caused changes in the business, engineering and logistic relations. The former, simple buying-selling (named "cool") relation has become more and more "warm" This means that, the cooperative and collaborative methods and activities have become the main object in SCM development. The fast

evolution of the IT technology plays an important role in this process. Independent in many aspects, locally-separated companies real-time, network-similar collaboration is not realizable without effective computer network information system.

The whole productive-marketing chain of the mass production is fairly long. The customer demands appear in shopping centres, which give orders to supply (logistics) centres. The centres transmit these demands to end-product manufacturers. OEM companies forward orders to dozen suppliers in the supply chain. This process initiates inside orders, start lots and order raw material from their own suppliers. These multi-stage information, decision and physical (producing and transporting) chains have not eliminable delay, which leads to delays and instabilities, back orders, overstocks and become source of unusable loss. These large, collaborative supply systems necessitate the more increased information technology support of business and technical process. There are complex ERP systems and auxiliary SCM modules and standalone SCM applications are available on the market, which support the above mentioned planning, decision, executive and information processes.

Relation of the marketing organizations, OEM and the supplier companies can be very various in practices. This motivates widely examination of the available models and further investigation of effective decision supporting and planning methods.

If we analyse barely the relation between the end-manufacturer and suppliers, even if strategy, tactical and operative collaborative areas can be separated. The stochastic market demands greatly influence activity of the mass production companies. In this paper we examine the possibility of the supplier inventory policy in case of non-deterministic demands.

We assume that the estimations concerning to future (forecasts) are solved, furthermore orders, acknowledgements, demands of delivery and the organization of the transport operation, synchronization of the planning process are also solved on the tactical level. We suppose that the supplier network on the strategy level is complete and the contracted as well as the computerized communication conditions are given to the realization of the business processes.

In the near past, based on the demand of a significant Hungarian mass production company we have examined the model of the supplying inventory respondent for stochastic demands with an event oriented analytical method. This stochastic model is the first step in the process of making the whole supply chain performance better. The attained results have been verified by simulation.

2. Related Studies

In the last few years many publications and studies have been published in the inventory control area, which validates the up-to-dateness of the subject. The most

important events related to the evolution of inventory control models are fully summarized in the paper of Hans-Joachim Girlich and Attila Chikán [1999]. In the late 1950's, the Optimal Inventory Policy problem was analyzed by two important economists: Arrow [Arrow et al., 1951] and Marschak [Arrow et al, 1951, Chikán 1999]. Karlin's presentations solved this problem with her dynamic programming method („The Structure of Dynamic Programing Models") [Karlin, 1955]. Thirty-six years later, Alistair Milne [Milne, 1966] emphasized that one of the best papers in the area of production decisions and inventory analysis area was the study of Arrow, Karlin and Scarf entitled "Studies in the Mathematical Theory of Inventory and Production" [Karlin, 1958].

E. Schneider's mathematical models deal with uncertainty-loaded problems of inventory control. E. Shaw in the "Elements of a Theory of Inventory" [Chikán, 1999] created a two-period uncertainty loaded model. In the 90's (S,s) type dynamic inventory control policies were published. The mathematician A. Markov laid strong foundations for the mathematical background for these models. In the meantime, John von Neumann and Oskar Morgenster's famous book, the "Theory of Games and Economic Behavior" [Neumann, 1940] became known, which gave a new direction to the approach of inventory problems. The paper of Dvoretzky, Kiefer and Wolfowitz [Dvoretzky et al., 1953] examined the (S,s) type policy in the case of a fixed time interval and penalty cost. Nowadays the analysis of inventory-holding problems has become an important part of the management of supply chains. Many excellent publications have been published related to this theme [Lal and Staelin (1984), Monohan (1984), Lee and Rosenblatt (1986), Dada and Srikanth (1987) and Weng (1995)], which work with the deterministic demand model [Girlich and Chikán, 1999].

In recent years further models have been published in the area of collaborative planning (Aviv, DATE), forecast and Vendor Management [Aviv and Federgruen, 1998], and information sharing within the supply chain [Gavirneni et al., 1999]. Nowadays in the explanation of the supply chain problems, the most prominent results are linked with the name of G.P. Cachon [Cachon, 1999, 2003]. For laying the foundation of the inventory policies, successful game theory results have sprung up. Stockpiling in the management of supply chains plays an important role nowadays. With the rapid evolution of information technology, ERP (Enterprise Resource Planning) and SCM (Supply Chain Management) applications systems are gaining in significance. Dynamic systems with many products are manageable with operations research models or constraint programming methods. However, solutions based on analytical results and heuristics have a great part in "what if" type investigations and in the case of quick decisions.

3. Analytical Approaching Method to the Solution of the Problem

In successful inventory control models, critical inventory and cost-optimal inventory policy are realized by decisions. In the course of these, decisions are made about the starting time and the quantity of the production. Naturally the individual decisions involve many responsibilities which consequences are appeared in producing, logistical and business costs.

In the course of non-reusable and overstock product producing stock finance and inventory holding costs and in case of non-sufficient product producing penalty (back-order) costs are appeared. Modelling these latter is very difficult. Of course, the different models can possess different objective functions, and by means of this common interest can be realized between the end-product manufacturer and the supplier (common cost function). As constraint the strict non-admittance of the „lack" (e.g. short cycle JIT) can be appeared. In this paper we consider a model, which in general allows the risk of the back-order, but the frequency of these can be reduced to an optional small level by increasing with the penalty costs.

The literature decomposes the explanation of the penalty costs into three areas. According to the first explanation the supplier pays penalty cost in the course of back-orders, which means lost business. This case can be noticed by the simple „cool" buying-selling relation. The second explanation of the penalty cost shows that in the course of the back-order at the supplier there are not lost business, only penalty cost.

This approach supposes already some kind of „warm" relation between the business partners. The third area of the explanation possesses the feature of the first two explanations. In case of unsatisfied order, the big back-order volumes lead not only to penalty costs, but to losing business too. This relation of the business partners is likewise between the „cool" and „warm" relation. In this paper we refer the explanation of the penalty cost of the model to the second type. We have chosen an event-oriented model for solving the inventory control problem of the supplier, which is based on cyclic demand of delivery and transport. The optimal stockpiling policy holds the supplier related costs at the minimal level on a long view. By minimizing the costs the profit maximization can be attained [Hayriye Ayhan].

We suppose the demand in the model is known as a random probability variable with its distribution function (in the simplest cases it is uniform distribution between a pre-defined D_{min} és D_{max} values). The model can be applied for other distributions too. The demands arrive as pre-known, fixed periodicity to the supplier. The knowledge of the distribution function is needed for the first step of the method. At this step in respect of a well-defined time interval we determine an optimal inventory level retailed on a long view in terms of the costs. In the later as the second step, we aim a less inventory level than this, using the information of the demand of delivery. To achieve this, a critical inventory level was determined, where the production and the non-production

costs respecting to one time interval are equal. The third important step of the method is the decision, when a decision is born about starting a production run. If the current inventory level is less than the critical level, than production must be started, otherwise nothing has to do.

Our model examines the problem from the side of supplier. After the solution of the model we make known the steps performing the optimal policy.

3.1. The Cost Function

On the basis of the problem outlined in the model the supplier cost function regarding one time period can be formulated as a function of the parameters in the following manner:

$$K(q) = c_f + c_v(q - x) + pE[max(D - q, 0)] + hE[max(q - D, 0)]. \quad (1)$$

Where the individual parameters are the following:

c_f – fix cost. This cost is always exist, when the producing of one series are started. [Ft / production]

c_v – variable cost: This cost type means the production cost of one product. [Ft / product]

p – penalty cost (or back order cost). If there is less raw material in the inventory then as much as satisfy the demands, this is the penalty cost of the unsatisfied orders. [Ft / product]

h – inventory and stock holding cost. [Ft / product]

D – It means that the demand from the receiver for the product, which is an optional probability variable. [number / period]

$E[x]$ – Expected value of the x stochastic variable.

q – The product quantity in the inventory. The decision of the inventory control policy concerns the product quantity being in the inventory after the product decision. This parameter includes the initial inventory as well. If we don't produce anything, then this quantity equals with the initial, i.e. concerning the existing inventory.

x – Initial inventory. We assume that the supplier possesses x product in the inventory at the beginning of the demand of delivery period.

m – It means the effective producing quantity in the current time period. Its value is the difference between the optimal and the remained quantity in the last time period.

Z – Summarized average manufactured product referred to a given time period.

The first part of the equation expresses, that starting the production of every series carry some fixed costs, which expresses the starting cost of a new production run. The second part shows the variable costs of the products, which will be produced. Because we assume that x product is available in the inventory, therefore one technologic decision will result producing of $m=q-x$ product.

The third part of the cost function is the so-called penalty cost (back-order), which symbolizes the costs issuing from unsatisfied demands. The $max(x,0)$ function performing in the cost function will be different from zero, if the demand is bigger than the quantity in the inventory. Of course it is possible events, when back-order is not allowable. This can be considered such as the p parameter of the model gets a high value. The last part of the equation gives the holding cost, which is arisen at that time when the demand was less than the quantity of the end-product in the inventory. If the demand is more than the produced quantity of the end-product, then of course there are no additional charges, because the inventory will be empty after filling the orders.

3.2. Determining of Optimal Stock Level

On the basis of the above mentioned cost function the determination of the optimal inventory level is a minimization problem.

$$\frac{dK(q)}{dq} = \frac{d}{dq}\left(c_f + c_v(q-x) + pE\left[max\left(D-q,0\right)\right] + hE\left[max\left(q-D,0\right)\right]\right) = 0 \quad (2)$$

From the upper relation for the $q = S$ optimal value (after a long derivation) we got an indirect solution. The complication rises from the handling of the operators of E expected value and max function. On a long view the amount of cost-optimal end-product can be calculated on the basis of the following relation:

$$F(S) = \frac{p - c_v}{p + h}, \quad (3)$$

where $F(D)$ is the distribution function of the demand.
The value of S expresses the amount of the end-product should be in the inventory when the demand appears. It is easy to see, if there are not available the demand meeting amount, then necessarily should not be started a production, because this process carries such fixed cost, which makes the production of the small volume expensive. Consequently it is conceivable, that there certainly exist a critical amount, which is smaller then the optimal (S) amount, but choosing this quantity it is more profitable to sustain the risk of the back-order. The name of that point, where the cost of the decision about producing and the decision about non-producing and the

decision where we rather undertake the risk of the back-order is equal, is the critical inventory level.

After these, the objective is to determine the critical level, which is probably smaller than the long-term cost-optimal inventory level. If the products in the inventory are less than this, only than is it profitable to increase the stock in hand to the optimal level. In the following we show how this level can be determined:

Let us introduce with nomination $L(q)$ the truncated, risk cost function, which is sum of the back-order and the inventory cost.

$$L(q) = pE[max(D-q,0)] + hE[max(q-D,0)]. \tag{4}$$

Suppose that the initial inventory level is less then the optimum level, namely $x < S$. If we increase this level to the optimum, then the total cost is:

$$K(S) = c_f + c_v(S-x) + L(S) \tag{5}$$

If the producing is not started, then on the other hand we have to calculate with only the initial inventory (x), so with $L(x)$. If the $L(x) \le K(S-x)$ condition is realized, then we don't have to produce, because the fix and the variable costs would increase the supplier cost. Determination of the critical inventory level can be attained with the $L(x) \le K(S-x)$ connection as follows:

$$\begin{aligned} L(x) &\le c_f + c_v(S-x) + L(S), \\ L(x) &\le c_f + c_v S - c_v x + L(S), \\ L(x) + c_v x &\le c_f + c_v S + L(S) \end{aligned} \tag{6}$$

Let s the critical inventory level. Our equation is modified as follows:

$$L(s) + c_v s = c_f + c_v S + L(S), \tag{7}$$

from where s can be determined. Understanding this solution, consequently if the inventory level of the supplier decreases under the critical level (s), then $m = S - x$ of quantity product must be produced. Otherwise it must not be produced, because the cost of the production is higher, than the cost of non-production.

4. Simulation Method for Checking the Policy

Henceforth we verified the solved work with help of 52 weeks simulation. We assume that the relation of the supplier and the customer is collaborative in the model. It means that in case of emerging of lack the supplier is liable for supplying the defect of the previous period in the next technologic cycle. Of this sanction and risk sharing are

expressed by the contracting of clientele in the value of p. We applied MAPLE mathematical software package completion the simulation. The starting data of the illustrating example simulation are the following:

The demand of the product follows uniform distribution, in *10* number/week and *20* number/week interval. The back-order cost is $p = 60$ unit in case of every element, and the variable cost is $c_v = 10$ unit. The holding cost is $h = 5$ unit/period. The fix part of the production cost is $c_f = 30$ unit/series. The following figure shows the results of the simulation:

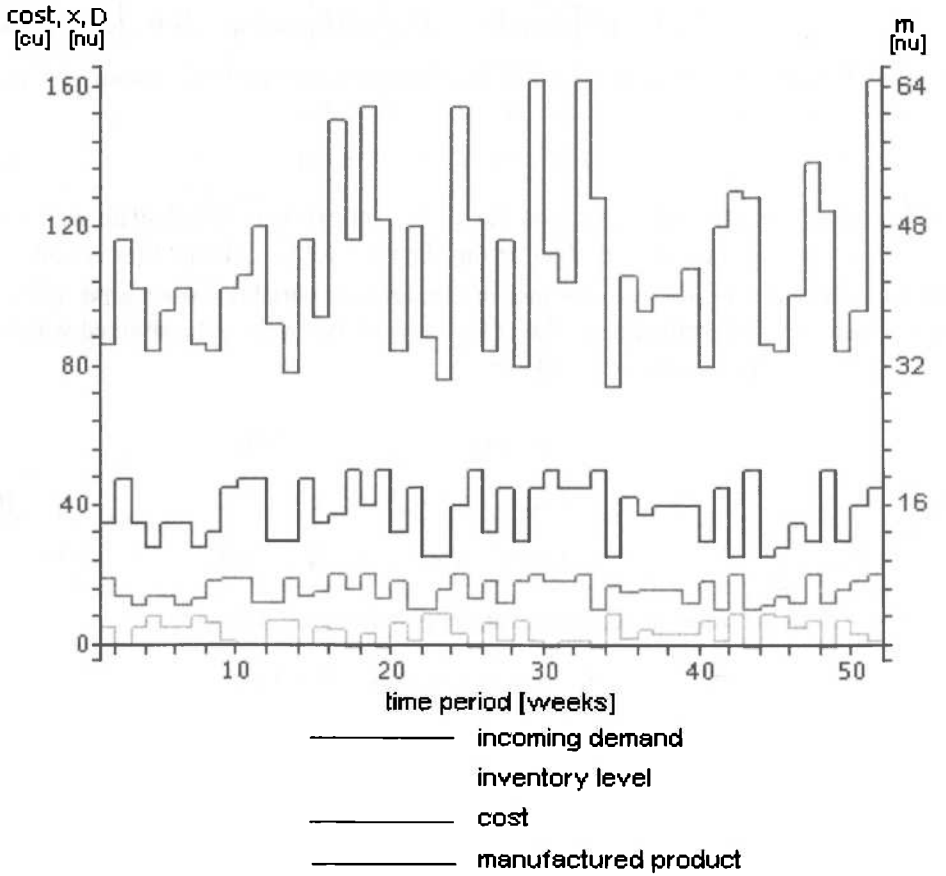


Figure 1. The simulation results of the supplier problem

Lack and with this penalty cost occur that time, when the supplier inventory level is negative, which means back-order quantity too. This happens when the stock is between the optimal and the critical inventory level, because at that moment the production is not started. If the incoming demand is bigger at this time, than the

inventory level, therefore of course lack is arisen. At the real task the lack is not, or minimal allowable for one part of products. This can be built into the model, if we set a high value to p. The figure 2 shows how decreases the frequency of paying penalty cost to zero, while the penalty cost is increased.

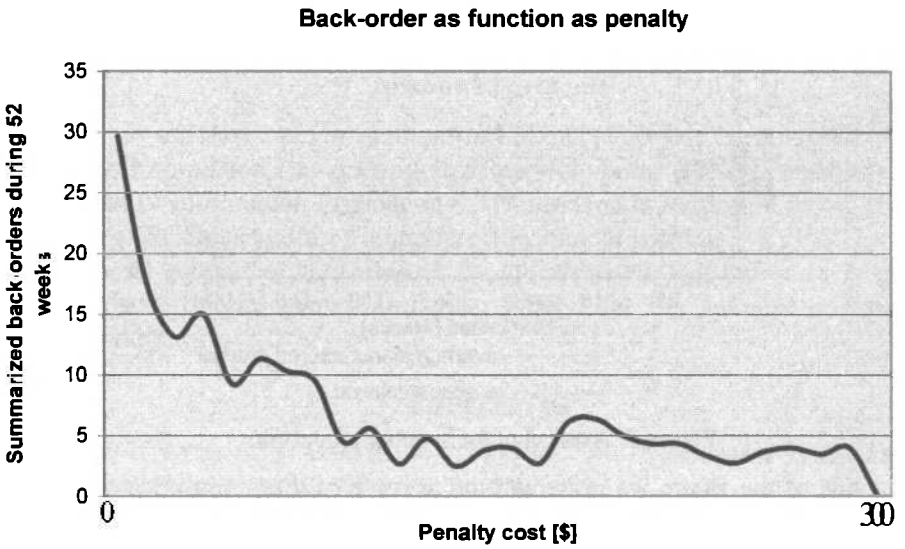


Figure 2. Back-orders and the penalty cost

We formed the results from averages of several simulations, where the demands accordance with uniform distribution randomly was changed. It can be seen that increasing the penalty cost there are such cases, where the frequency of the lack does not decrease. The reason of this is the values of the demands according to uniform distribution, which are changed randomly within the demand bounds. The other consequence of increasing the penalty costs – which we justified with help of further simulations - is that the weekly average of the produced products (m) approximate asymptotic-wise to the weekly average of demands. In case of the same demands and parameters we investigated to what value approximates the average value of the number of products produced by the supplier. The next figure shows the results of the 150 weeks simulation:

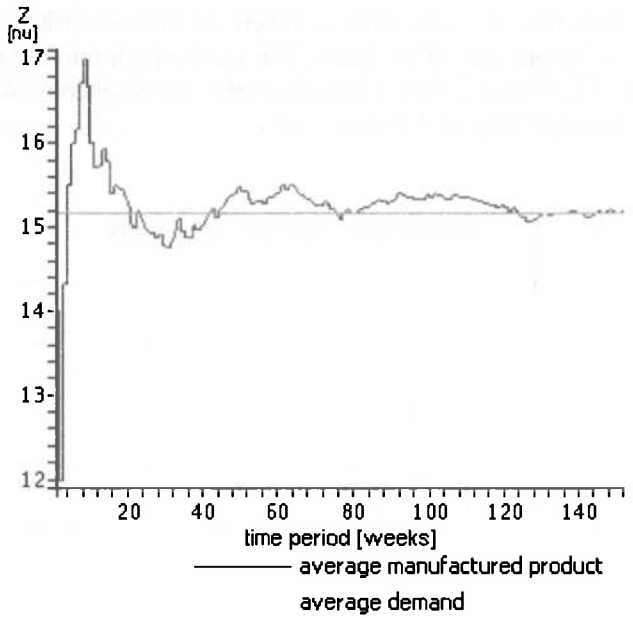


Figure 3. Average of the supplier production

On the y-axis of the figure we presented the average of the produced products as a function of full of weeks (Z). Well can be seen that progressing in time, the full of week average of the produced product better and better approximates to the average of the demands examined to the complete time interval, while it does not attain that. Of course with changing the penalty cost other transient processes can be realized too.

5. Conclusions

In the present paper we examined the problem in case of the collaborative relation of one supplier and one customer. Improving the well-known models from the literature we optimized the cost function of the supplier as a function of the parameters, which does not eliminate the possibility of back-order. Understanding the problem as a non-linear optimization problem, we determined the optimal inventory level. To apply the correct inventory policy we introduced and defined the critical quantity of the inventory. To control the results we verified the correctness of the model with a 52 weeks simulation. With the help of this simulation we presented that the model with changing the parameters can be made suitable for attaining an optional low value of the lack respectively for transient observation of the changing inventory level. The model runs quickly on the simulator, therefore it is suitable for fast testing of different policies and decision alternatives. Henceforth we aim at the expansion of decision work to the more periods. The objective of common investigation of several weeks is:

determining all those week-pairs, where the aggregated production cost is less than the separately production cost. Thus with help of the received week-pairs, the total cost of the producing becomes still smaller. Further aim is to create a game theory model for the problem, which simulation results we would compare the obtained effects.

It belongs to our aim to integrate the forecast information of the expected demand into the model, as well as to examine how historical data and uncertain forecast influences in time the conformation of inventory level.

Acknowledgements

The research and development summarized in this paper has been carried out by the Production Information Engineering and Research Team (PIERT) established at the Department of Information Engineering. The research is supported by the Hungarian Academy of Sciences and the Hungarian Government with the NKFP VITAL Grant. The financial support of the research by the aforementioned sources is gratefully acknowledged. Special thanks to Ferenc Erdélyi for his valuable comments and review works.

REFERENCES

[1] HAYRIYE AYHAN, JIM DAI, R. D. FOLEY, JOE WU: Newsvendor Notes, ISyE 3232 Stochastic Manufacturing & Service Systems, 2004.

[2] CACHON, GÉRARD P.: Competitive Supply Chain Inventory Management, Quantitative Models for Supply Chain Management (International Series in Operations Research & Management Science, 17), Chapter 5, 2003.

[3] CACHON, GÉRARD 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] TAYLOR, A. DAVID: Supply Chains A Managers Guide, Addison Wesley, 2003.

[5] HANS-JOACHIM GIRLICH, CHIKÁN ATTILA: The Origins of Dynamic Inventory Modelling under Uncertainty, International Journal of Production Economics Volume 71, Issues 1-3 pp, 1999.

[6] GARDNER, L. DANIEL: Supply Chain Vector: Methods for Linking the Execution of Global Business Models With Financial Performance, J. Ross Publishing , pp. 17-156, 2004.

[7] MAX MULLER,: Essentials of Inventory Management, American Management Association, pp. 17-143, 2002.

[8] WEISSTEIN, W. ERIC: CRC Concise of Encyclopedia of Mathematics, CRC Press, London, pp. 438-523, 1999.

[9] 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.

[10] 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.

[11] MILEFF, P., NEHÉZ, K.: *Applying Analytical Methods in Inventory Control Problems*. Proceedings of microCAD 20th International Scientific Conference, Hungary, pp. 128-135, 2006.

[12] BRAMEL, J., SIMCHI-LEVI, D.: *The Logic of Logistics: Theory, Algorithms, and Applications for Logistics Management*. Springer PLACE of publication, Chapter 8-9, 1997.

[15] ARROW, K.J., HARRIS, T., MARSCHAK, J.: *Optimal inventory policy*. Econometrica 19, pp. 250 – 272, 1951.

[16] KARLIN, S.: *The structure of dynamic programing models*. Naval Research Logistics Quarterly 2, pp. 285 – 294, 1955.

[17] ARROW, K.J., KARLIN, S., SCARF, H.: *Studies in the Mathematical Theory of Inventory and Production*. Stanford University Press, 1958.

[18] MILNE, A.: *The mathematical theory of inventory and production: The Stanford Studies after 36 years*. In Workshop, August 1994, Lake Balaton. ISIR, Budapest, pp. 59 – 77, 1996.

[19] NEUMANN, J., MORGENSTERN, O.: *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[20] DVORETZKY, A., KIEFER, J., WOLFOWITZ, J.: *On the optimal character of the (s; S) policy in inventory theory*. Econometrica 21, pp. 586 – 596, 1953.

Production Systems and Information Engineering
Volume 3 (2006), pp. 83-96

CONVERSION POSSIBILITIES OF STORAGE ZONES OF DISTRIBUTION WAREHOUSES IN CASE OF CHANGING STRUCTURES AND VOLUME OF ORDER PICKED PRODUCTS

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

LÁSZLÓ KOVÁCS
University of Miskolc, Hungary
Department of Material Handlings and Logistics
kovacs@snowwhite.alt.uni-miskolc.hu

GYÖRGY KOVÁCS
University of Miskolc, Hungary
Department of Material Handlings and Logistics
altkovac@uni-miskolc.hu

RICHÁRD BÁLINT
University of Miskolc, Hungary
Department of Material Handlings and Logistics
altrichi@snowwhite.alt.uni-miskolc.hu

[Received November 2005 and accepted May 2006]

Abstract. The paper examines a warehouse system which is including some storage buildings and divided into storage zones based on product groups in case of changing structure and volume of order picked products. The products of a commission are collected from different zones of different storage buildings. Warehouses are through-stores: loading in is completed on one side and only the order picking is completed on the other side. The inner structure of warehouses and types of products provide the possibility of relocation and conversion of different zones in case of changing structure and volume of commissions to reduce order picking work and cost, and improve productivity. This study presents a mathematical description of an order picking system taking the structure of commissions, location of zones, conversion possibilities of different product-zones and deterministic and stochastic changing order picking demands into consideration. The paper presents an optimisation process to determine location of different storage zones.

Keywords: storing, conversion of resources, optimisation of order picking system

1. Introduction

Today in the world of globalization not only the production but the sourcing and distribution is getting globalised. Distribution warehouses and logistic service centres with significant storage capacity are coming alive in large number. The profitability of operation of these big warehouses are greatly influenced by their capacity and the enhanced delivery lead time of the orders. One feature of the globalisation is the dynamic change in volume and structure of the transit commodities. The competitiveness of distribution warehouses are predominantly influenced by their flexibility to the dynamically alternating requirements. The authors tried to elaborate a general mathematical model and method for a common appeared storage problem.

The purpose of this study is to point at the efficiency of storage activity to maintain the most advantageous cost level i.e. the profitability in case of changing structure and volume of order picked commodities.

With other words to achieve the following targets:

to assure the required volume of products be stored,
to maximise the order picking capability,
to minimise the order picking lead time and labour demand.

We have not found any theoretical basis, mathematical models and methods for theoretical problems of resource conversion of order picking distribution stores during our literature mining. But lot of authors focused on the importance of the utilization of different dynamically changing logistics resources, changing in time and in space (*A. Chikan*, 1994; *R. Cooper, R. Kaplem*, 1991; *W. Domschke, A. Scholl*, 1993; *B. Gubenko*, 1996; *R. Jünemann*, 1989; *B. Kulcsár*, 1998; *H.-Chr. Pfohl*, 1996; *P. Schönsleben*, 1998; *P. Michelberger., L. Szeidl, P. Várlaki*, 2001; *D. Simchi-Levi, X. Chen, J. Bramel*, 1997).

In the above mentioned literature there are not available mathematical models and methods which can be applied in case of order picking stores. Virtual enterprise and virtual logistics company can provide a new and good opportunity for conversion and utilization of logistics resources (*Camarinha-Matos, L. M.*, 2004). The solution of the problem requires the knowledge of flexibility improving methods and procedures for flexible and integrated production systems and integrated logistics systems (CIL) (*J. Cselényi, T. Bányai*, 2002; *J. Cselényi, B. Illés*, 2004).

The research activity was based on the above mentioned literature and according to the demand of Hungarian enterprises (mostly mechatronical companies) and service companies. Research topics for determination and conversation of optimal logistics resources were defined for PhD students of the "József Hatvany" Doctoral

School for Information Science, Engineering and Technology at the University of Miskolc. Results of this research activity were published in papers of *R. Bálint, J. Cselényi*, 2002; *J. Cselényi, K. Dunai, Á. Gubán, R. Bálint,* 2003.

2. The Limitations of the Examined Objective

At the distribution warehousing systems the optimal tackling methodology of the variability of structure and volume is greatly depends on how the order picking system can be modelled.

The study taken into account the possibilities provided by the virtual enterprises detailed in paper of *Camarinha-Matos, L. M.*. Warehouses joined into a virtual logistics network provide the conversion possibilities of resources, because the utilization of different warehouses and storage zones is known in given time intervals. Study of *J. Cselényi, B. Illés* discuss the integration of CIM-CIL which is not only relating to technological processes, activities completed at one work place, but relating to storage activities of cooperating enterprises and companies using common convertible resources.

In the study we wish to analyze those models of storage systems which are highly productive and the best known in practice.

The features of the examined model of distribution warehouse order picking system are:

the warehousing system contains several buildings, some of them a system of isolated pavilions, others are block lay-out,
each building has several input and output loading docks,
the storage process in each building is implemented in zones separated by longitudinal passages which may be block or lined installations,
order picking process is implemented in the passages,
loading units of the same size are deposited in each in each storage zone, loading unit is homogenous, but in a given zone different type of commodities can be stored / each type of commodities, which can be stored in loading unit forming equipment that is typical in the given zone /,
whereas the storage implemented in several buildings and within a building in several zones it may well occur that a given type of loading unit forming equipment can be found in more than one zones, consequently the same sort of commodity may be found in several zones,
the warehouses can be head or transit stores, the order picking process may be performed manual or powered,
the model assists in preparing strategic decisions by the mathematical analysis of historic data of the previous T_0 period and taking into account the long term

predicted data resulting the conversion of the existing logistic resources and order picking strategies,
the model is suitable, in case of a given storage capacity and lay-out respectively to realise a common order picking task with regards the variation in structure and volume of commodities to be order picked and to define the optimal conversion possibilities, comprising:

- the selection of order picking passages,
- the re-storage of commodities:
  - » into another storage zone within the same building,
  - » from one building into another building's certain storage zone,
- tackling of variation in capacity requirement / increase-decrease / of a certain commodity,
- the structure and volume of order picked goods in the outgoing shipments.

The next step is the mathematical description of the data model, which is necessary to solve the above outlined problem.

3. Mathematical Description of the Data Model

The analysis is limited to

$$n = n_1 + n_2 + n_3 \tag{1}$$

storage zones in 3 buildings. This limitation does not mean any theoretical restriction, merely is given in the interest of a less extensive and easier mathematical description.

Given the storage capacity of the individual storage zones in LU / loading units /:

$$\overline{r_0} = \left[r_{0j} \right]_{(j=1\cdots n)\ \text{storage zones}} \tag{2}$$

The matrix, which expresses the possibility the placement of individual commodities in the storage zones:

$$F = \left[f_{ij} \right]_{(i=1\cdots p)\ \text{type of commodities}}, \tag{3}$$

where f_{ij}=1or 0.

If f_{ij}=0, then the *i-th* commodity can not be placed in the *j-th* storage zone, because there a different kind of loading unit forming equipment is applied.

If f_{ij}=1, then the *i-th* commodity can be placed into the *j-th* storage zone.

In the forthcoming period / let say in 1 year / the following order picked varieties must be composed:

$$K = \left[k_{\delta i}\right]_{\substack{(\delta=1\cdots w)\\(i=1\cdots p)}}, \tag{4}$$

where $k_{\delta i}$ is the number of *i-th* commodity in the kind of order picked varieties.

The number of annual requirement from the different kind of order picked varieties is:

$$z = \left[z_{\delta}\right]_{(\delta=1\cdots w)}, \tag{5}$$

where $z_{\delta} = \left[\frac{pcs}{year}\right]$.

The number of annual requirement for the order picked varieties from the *i-th* commodity is:

$$a_i = \sum_{\substack{\delta=1\\(i=1\cdots p)}}^{w} z_{\delta} k_{\delta i} \tag{6}$$

In the previous period the requirements for order picked varieties were $K^*, \overline{z}^*, \overline{a}^*$ respectively. The commodities in the store were deposited in the light of the above and expressed by the matrix *R*:

$$R^* = \begin{array}{c} 1 \\ i \\ p \end{array} \left[\begin{array}{c|c|c} & \begin{array}{c} j \\ \\ r_{ij}^* \end{array} & \\ I. & II. & III. \end{array}\right] \tag{7}$$

where:

$j = 1\cdots n_1$ Zones in store No. I.

$j = (n_1+1)\cdots(n_1+n_2)$ Zones in store No. II.

$j = (n_1+n_2+1)\cdots(n_1+n_2+n_3)$ Zones in store No. III.

r_{ij}^* is the number of the loading units containing the *i-th* commodity in the *j-th* storage zone.

In the previous period it was valid, that:

$$\sum_{i=1}^{p} r_{ij}^{*} \le r_{0j}. \tag{8}$$

The available free storage capacity in the individual storage zones is:

$$\Delta r_{0j}^{*} = r_{0j} - \sum_{i=1}^{p} r_{ij}^{*} \tag{9}$$

The number of commodities in the homogenous loading units is:

$$E = \left[e_{i\mu} \right]_{\substack{i=\cdots p \\ \mu=\cdots \omega}} \tag{10}$$

where $e_{i\mu}$ is the number of the *i-th* commodity in the μ-*th* loading unit forming equipment.

The feature of the E matrix is, that every row should contain only elements, those differ from 0.

In those j zones in the *i-th* row of the F matrix, where f_{ij}=1 everywhere identical loading unit forming equipment can be found, in addition in the *i-th* row of the E matrix $e_{i\mu}$>0.

The annual number of the *i-th* commodity in the storage zones at the current placement is:

$$\overline{h^{*}} = [h_i^{*}] \ (i = 1 \ldots p), \tag{11}$$

where:

$$h_i^{*} = c_i^{*} \sum_{j=1}^{n} e_{ij} r_{ij}^{*}, \tag{12}$$

and c_i^{*} is the circulating velocity of the *i-th* commodity in the previous period.

An important element of the data model is the matrix B, that describes the composition from order picked commodities in the outgoing particular freights:

$$B = [b_{k\delta}], \tag{13}$$

where

k=1 …u: freight type,

δ=1…w: type of commission variety,

$b_{k\delta}$: the number of δ type commission variety in the *k-th* freight type.

The number of annual outgoing freight types is:

$$y = [y_k], \tag{14}$$

where k= 1…u,

y_k : the annual outgoing number of the *k-th* freight: $y_k = \left[\frac{freight}{year}\right]$.

Different kind of analysis can be done based on the available data model.

4. A Couple of Analysis Based on the Data Model

It can be analyzed, if the required quantity of commodities to be order picked in the forthcoming period can be stored in the available transit warehouses or the missing capacity need to be replaced with regards the followings:

temporary capacity must be established,

capacity need to be hired,

if the anticipated increase in store capacity requirement may be predicted for a long term period the feasibility of building new stores or extending the existing buildings should be analyzed.

The available storage capacity for the μ-*th* loading unit forming equipment is sufficient, if:

$$\frac{\sum_{j\in\Theta_\mu} r_{oj}}{\sum_{j\in V_\mu} \frac{a_i}{e_{i\mu}} \frac{1}{c_i^*}} \geq 1, \tag{15}$$

where

$\mu = 1...\omega$,

Θ_μ is the set of the those *j-th* storage zones, where the μ-*th* type loading unit forming equipment can be stored,

V_μ : the set of those *i-th* commodities which can be stored in the μ-*th* type of loading unit forming equipment.

It is easy to see, if the (15) condition does not exist, the (15) condition can easily be restored by the circulation velocity of the c_i^* commodities.

If the (15) condition is satisfied, that makes out the available free capacity of the *μ-th* type loading unit forming equipment measured in "loading unit" number:

$$r_{o\mu} = \sum_{j \in \Theta_\mu} r_{oj} - \sum_{i \in V_\mu} \frac{a_i}{e_{i\mu}} \frac{1}{c_i^*}. \tag{16}$$

Based on the available data model it is also can be analyzed, that how many annual order picked varieties can be completed in the individual storage zones.

The output of the order picking process can be increased, the delivery lead time and labour demand and costs can be reduced by increasing the number of order picked varieties in one particular passage. The requirement of it is the availability of commodities in the given volume.

Within a given period the sub-proportion of the annual order picked varieties can also be analyzed with regards the *i-th* commodity from the *δ-th* commission varieties to be placed into the *j-th* zone:

$$\varphi_{\delta i}^j = \frac{z_\delta k_{\delta i}}{r_{ij} e_{i\mu} c_i^*}, \tag{17}$$

where r_{ij} and c_i are corresponding to data of the forthcoming period. Last, but not least we will refer to the principles and methodology of the modification of r_{ij} to r_{ij}^*

In the *j-th* storage zone the Φ^j order picking varieties sub proportion matrix can be generated:

$$\Phi^j = [\varphi_{\delta i}^j], \tag{18}$$

where δ=1…*w*, commission variety,

i=1…*p*, type of commodity.

Analyzing the Φ^j matrix, if:

$\varphi_{\delta i}^j \le 1$, then in the *j-th* zone the *i-th* commodity for the *δ-th* commission variety is fully available / 100 % /,

$\varphi_{\delta i}^{j} > 1$, then in the *j-th* storage zone only a fragment of the *i-th* commodity for the δ-*th* commission variety is available, i.e. the δ-*th* commission variety may be completed only in fragment,

$\varphi_{\delta i}^{j} = \infty$ may occur, if r_{ij}=0 i.e. in the *j-th* zone the *i-th* commodity is not available.

The Φ^{j} matrix may contribute to the solution of conversion task to be described in the next chapter providing important data or information.

5. Conversion Strategies Applying the Data Model

If the K^{*} matrix, the $\bar{z}^{*}$ and $\bar{a}^{*}$ vectors relating to the previous period alter to K matrix and $\bar{z}$ and $\bar{a}$ vectors for the next period then the conversion of the existing logistic resources and order picking strategies become necessary. The conversion may include:

the alteration of the R^{*} matrix, comprising,

- the alteration of the commodities and storage capacity in the individual storage zones, that may be created by,
  - » relocation of the given volume storage capacity of a commodity existing in the previous period,
  - » in the consequence of increase of the volume of a given commodity,
  - » in the consequence of decrease of the volume of a given commodity,
- the relocation of the particular storage zones into an other building,
- the selection of the optimal passage for order picking process,
- the definition of the optimal loading entries / channels /.

Further on – in consequence of the available limited volume we provide only strategic principles to select the optimal order picking passages.

The selection of the optimal order picking passages /storage zones/ may be defined in some consequtive steps.

Step 1.

Analyzing those rows of Φ^{j} matrix, at all elements where $k_{\delta i} > 1$, the following is true:

$$1 \geq \varphi_{\delta i}^{j} \geq 0, \tag{19}$$

(i=1...p), $i \neq \rho$, *if* $k_{\delta i} = 0$,

then neglected the requirements of the identical commodities in other commission varieties the δ-*th* commission variety can be completed from the *j-th* zone, but further investigation is required.

If the (19) relation is valid only in one row of Φ^j matrix, i.e. valid only for one type loading unit, then must be analyzed, that

- if

$$r_{ij} < r_{ij}^{*}, \tag{20}$$

(i=1…p),

then in the previous period provided available place in the *j-th* zone for the *i-th* commodity is adequate.

- if

$$r_{ij} > r_{ij}^{*}, \tag{21}$$

(i=1…p), but

$$r_{ij}^{*} + \Delta r_{oj} \geq r_{ij}, \tag{22}$$

then the required storage capacity surplus at r_{ij} can be stored into the free places of the *j-th* zone.

In these cases all the required conditions are given in the *j-th* zone to complete all number of the δ-type commission variety.

If exists even one type of commodity, in which case no one condition of the above is satisfied, then the δ-type commission variety can only partly be completed.

Step 2.

If (19) relation is valid in more rows of Φ^j (N_{j2} defines the set of those commission varieties, where (19) exists) then should be satisfied:

$$\varphi_{\delta 2}^{j} = \frac{\sum_{\delta \in N_{j2}} z_\delta k_{\delta i}}{r_{ij} e_{i\mu} c_i} \leq 1, \tag{23}$$

(i=1…p),

in addition either (20) or (21) and (22) should be exist.

If for the whole set of N_{j2} the (23) and the relating conditions are not satisfied then the set should be reduced to N_{j2}^{*} so, that at the required conditions let

$$\sum_{\delta \in N_{j2}^{*}} z_{\delta} \rightarrow Max. \tag{24}$$

be true.

If for the whole set of N_{j2} the (23) and the relating conditions are not satisfied, then we analyze how big is the defiance in the whole set of commission variety:

$$\Delta\varphi_{i2}^{j} = 1 - \varphi_{i2}^{j} \tag{25}$$

The (25) also refers to, if the reduction of N_{j2} set is advisable by principles governed other than (23) what kind of latitude is available.

Steps 3. ,4., 5.

In the referred paras we have analyzed that completing the 75%, 50% and 25% of the annual order picked varieties requirement (ε=0.75, 0.50, 0.25) in which cases satisfied the

$$\varphi_{ik}^{j} = \frac{\sum_{\delta \in N_{j\varepsilon}} \varepsilon z_{\delta} k_{\delta i}}{r_{ij} e_{i\mu} c_{i}} \leq 1 \tag{26}$$

and related conditions.

Step 6.

Types of δ commission varieties in the *j-th* storage zone should be analyzed which can not be completed at all in lack of one or more kind of commodities.

$M_{\delta i}^{j}$ represents that subset in the δ commission variety-set, which one can not be completed in lack of *i-th* commodity.

$M_{\delta i}^{j}$ subset includes those *i-th* kind of commodities, which satisfies the following conditions:

$$k_{\delta i} > 0 \text{ and } r_{ij} = 0. \tag{27}$$

It can be calculated, that in the *j-th* passage which maximum proportion of the *δ-th* commission variety can be completed:

$$\eta_{\rho}^{j} = \frac{\sum_{i \in M_{\delta i}^{j}} z_{\delta} k_{\delta i}}{z_{\delta} k_{\delta i}}. \tag{28}$$

Step 7.

In this step we summarize the results of the previous 6 steps. It is necessary to do so, to point at how to tackle exactly the optimum order picking process of a commission variety set which is required for the application of an opportunity set.

The summary includes the breakdown per zones relating to the individual commission varieties:

which kind of commission varieties can be completed without limits,

which kind of commission varieties can be completed with limits:

- which kind of commission varieties – members of those subsets – which can be completed in maximum number,
- how big is the required storage capacity to be freed by relocating the not involved commodities to complete the whole set,

identifies those commission varieties, of which the 75%, 50% and 25% of the annual requirement may be completed,

identifies those commission types, of which no one commission variety can be completed in the given storage zone and in what percentage of the commission variety can be completed.

Step 8.

This step describes the optimization of the individual order picking varieties i.e. in which zone or zones should be completed the commission varieties so, that we fix up the next two relations, described in chapter 1.:

minimising the P labour requirement, let

$$P \rightarrow min.$$

so, that the required order picking performance requirement let be balanced,

the performance requirement of order picking let minimum be, what we wish to be limited to:

- the movement of commodities between the zones,
- converted transportation from the individual zones,
- vehicle loading only by the order picking process governed by the freights.

The algorithm of optimization will be provided in the next scheduled study on this topic.

6. Conclusion

The study presents the rearrangement of order picking strategies and logistic resources which frequently may occur in distribution warehouses in the consequence of alteration structure and volume of order picked varieties. With other words how to rearrange the task-complex in the interest to find the optimal solution.

We have summarized the required data for the optimal decision making, the methodology of data collecting, elaborated the mathematical connexion that is required to provide an adequate data model. We have expounded those investigations and examinations which help to create an algorithm of optimisation with regards the functions, conditions and parameters to be optimised.

REFERENCES

[1] BÁLINT, R., CSELÉNYI, J.: *Die Bestimmung der optimalen Grössen von konvertierbaren logistischen Kapazitäten in logistischen Zentren.* MicroCAD 2002 International Scientific Conference in Section Material Flow Systems, Logistical Informatics. Proc. pp. 1-6. University of Miskolc, Hungary. ISBN 963 661 515 2., 2002.

[2] CAMARINHA-MATOS, L. M. ed.: *Virtual Enterprises and collaborative networks.* ISBN 1-4020-8138-3, Kluwer Academic Publisher, 2004.

[3] CHIKÁN, A.: *Corporate Economy.* Közgazdasági és Jogi Könyvkiadó, Budapest, 1994. (in Hungarian)

[4] COOPER, R., KAPLAN, R.: *Profit Priorities from Activity-Based Costing.* Harvard Business Review, May-June 1991.

[5] CSELÉNYI, J., BÁNYAI, T.: *Development and relationship of CIM and CIL, in: Production Processes and Systems.* A Publication of the University of Miskolc, Volume 1, HU ISSN 1215-0851, pp. 137-143, 2002.

[6] CSELÉNYI, J., DUNAI, K., GUBÁN, Á., BÁLINT, R.: *Capacity optimisation of non convertible logistic sources to be developed through regularly stepped specific cost functions and in line with capacity needs based on uniform distribution.* MicroCAD 2003 International Scientific Conference in Section Material Flow Systems, Logistical Informatics. Proc. pp. 77-82. University of Miskolc, Hungary. ISBN 963 661 547 0, 2003.

[7] CSELÉNYI, J., ILLÉS, B.: *Logistic Systems I.* University Press, Miskolc, 2004. (in Hungarian)

[8] DOMSCHKE, W. SCHOLL A. VOß S.: *Produktionsplanung - Ablauforganisatorische Aspekte.* Springer Verlag, Heidelberg, ISBN 3 540 56585 X, 1993.

[9] GUBENKO, B. K.: *Logistics.* Mariupol, 1996.

[10] JÜNEMANN, R.: *Materialfluss und Logistik.* Springer Verlag, 1989.

[11] KULCSÁR, B.: *Industrial Logistics.* LSI Oktatóközpont, A mikroelektronika Alkalmazásának Kulturájáért Alapítvány, Budapest, 1998. (in Hungarian)

[12] MICHELBERGER, P., SZEIDL, L., VÁRLAKI P.: *Applied Process Statistics and Time Series Analysis.*, Typotex Kiadó, Budapest, Hungary. ISBN 963 9132 44 6, 2001. (in Hungarian)

[13] PFOHL, H.-CHR.: *Logistiksysteme*, Berlin, 1996.

[14] SCHÖNSLEBEN, P.: *Integrates Logistik Management.* (Planung und Steuerung von umfassenden Geschäftprozessen.) ISBN 3-540-6329 22-5 Springer Verlag Berlin Heidelberg New York, 1998.

[15] SIMCHI-LEVI, D., CHEN, X., BRAMEL, J.: *The logic of logistics.* Springer series in operation research, 1997.

Production Systems and Information Engineering
Volume 3 (2006), pp. 97-108

THE TIME FACTORS OF MAINTENANCE LOGISTICS

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

[Received November 2005 and accepted May 2006]

Abstract. In the realization of maintenance processes, a vital role is played by logistical activities, i.e. maintenance logistical processes. The way in which the maintenance process is carried out has a basic influence on purchaser satisfaction. Maintenance as an influencing factor of customer satisfaction appears in three basic areas: the production and service process; the maintenance and servicing of the product for the customer; and the administration of maintenance services.
Dominant activities of maintenance logistics are:
- supplying the materials, components, tools and services,
- operating the supply chain of maintenance logistics,
- storage management of the maintenance logistics,
- maintenance inverse logistics.

In the field of maintenance logistics, the time factors and important parameters are explored and mathematically formulated, and the optimal time for starting the maintenance procedure is determined by the transit time.

Keywords: logistics, maintenance logistics, maintenance process, time factor

1. Introduction

For customers expectations the maintenance is an important respect. During the maintenance activity the logistics has got a determined importance [1], [2]. From the total transit time of the maintenance approximately 90% goes on the logistical type activities. The scientific literature does not deal particularly with their topic. In the first step my intention is to compose a model by which the time parameters of the maintenance can be given. After that the specific parameters will be determined on which these time factors are dependent.

The transit time of the maintenance activity is primarily determined by the completion of the maintenance logistical process and by the logistical strategies applied. This transit time has a substantial effect on customer satisfaction. The author investigated which time parameters have an important influence on determining the maintenance transit time.

The basic condition for initiating the maintenance activity is the existence of the following for the item to be maintained:

the necessary type of materials in suitable quality and quality,
the necessary type of parts in suitable quantity and quality,
the necessary type of equipment in suitable quantity and quality,
the services needed for maintenance and the necessary staff.

The assurance of the above conditions is the task of maintenance logistics.

2. Time Factors

For the time parameters of maintenance logistics, two main factors are considered, as can be seen in (2.1):

$$t_a = t_M + t_H \tag{2.1}$$

where

t_M - transit time for the order process of the necessary items, and

t_H - time to fulfilment of order, i.e. the total time of the logistical activities related to the item to be maintained.

The ordering process of the necessary items (materials, parts, equipment, service, staff) can be seen in Figure 1.

3. Process of Order of the Maintenance Necessities

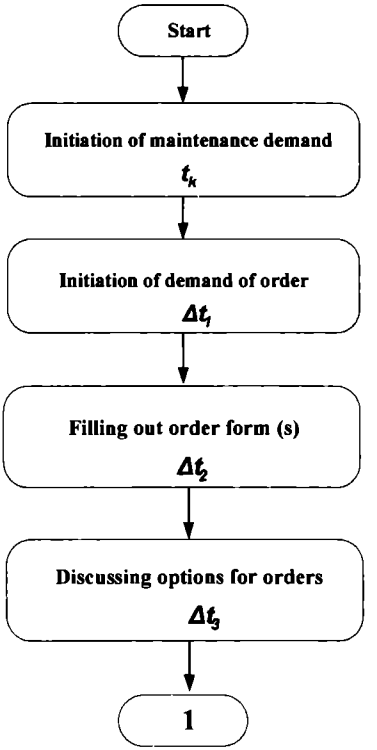
On base of Figure 1 the activities and time necessities of the ordering process are the followings:

failure occurred at a given moment,
after time Δt_1 on ordering demand is occurring for a given constituent,
after Δt_2 time the order has been written,
the order judged by the responsibilities for which Δt_3 is needed,
decision is born about the order is time Δt_4

- if it is negative, then no maintenance activity.

request for price offers to be concerned for which time Δt_5 is needed,
evaluation of the offers, time necessity Δt_6,
necessary time for selection of the optimal price offer is Δt_7,
time necessity for proceeding of the order Δt_8,

– handling time of prove of getting the order is Δt_g.



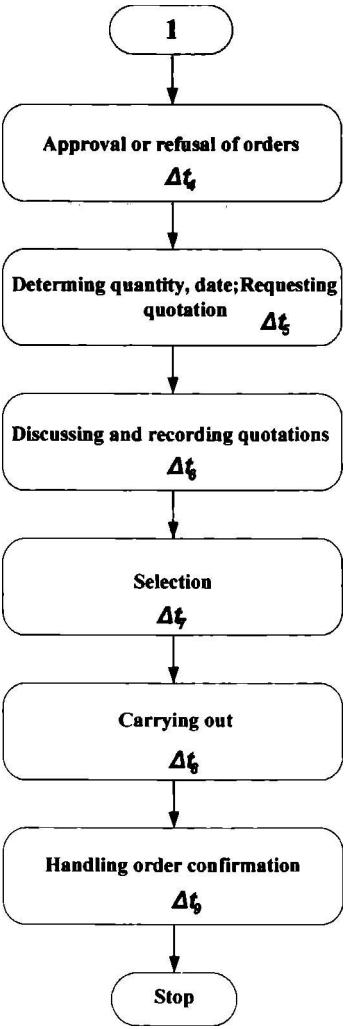


Figure 1. The order process for necessary maintenance items

Transit time of the ordering process of the necessities can be understood on base of

- for the materials necessary for the maintenance *(a)*
- for the constituents necessary for maintenance *(b)*
- for the facilities necessary for the maintenance *(c)*
- for the services necessary for the maintenance *(d)*

$$t_{M\theta}^{i_\theta, j_\theta} = \sum_{k_\theta=1}^{n_0} \Delta t_{k_0}^{i_\theta, j_\theta}, \tag{3.1}$$

where:

$$\theta = \{a, b, c, d\}$$

i_θ identifier of the transporter,

j_θ identifier of the product or service to be purchased,

k_θ identifier of the time increment of the ordering process,

n_θ maximum number of time increment elements taken into account, i.e. the number of activity elements in the order process,

a index regarding materials,

b index regarding constituents,

c index regarding facilities,

d index regarding services.

By using (3.1) the transit time can be given for the materials, parts, equipment and services which are necessary for the ordering of maintenance. It is advisable to specify the following:

$$t_{M\theta}^{i_0, j_0} \quad \sum_{k_\theta=1}^{n_0} \Delta t_{k_0}^{i_0, j_0} \rightarrow min\,. \tag{3.2}$$

4. Logistical Features of Order Fulfilment in Maintenance

In the following section the time factors connected with the fulfilment of the order for the necessary item are investigated. Time factors are related to the place where the needs can be satisfied and to the item needing maintenance, and include:

time for carrying out logistical activities,
waiting time for the necessary equipment.

Types of logistical activities in the field of maintenance logistics can include:

different kind of storage activities,
activities in connection with stores (adding and removing),
activities in connection with breakdown and forming of unit packages,
order picking,
transport,
loading and unloading,

- classification into:

- parts re-usable after overhauling,
- parts for recycling,
- pieces to be treated as waste.

The time factors for order fulfilment of maintenance are given in Figure 2, taking these types of activities into account.

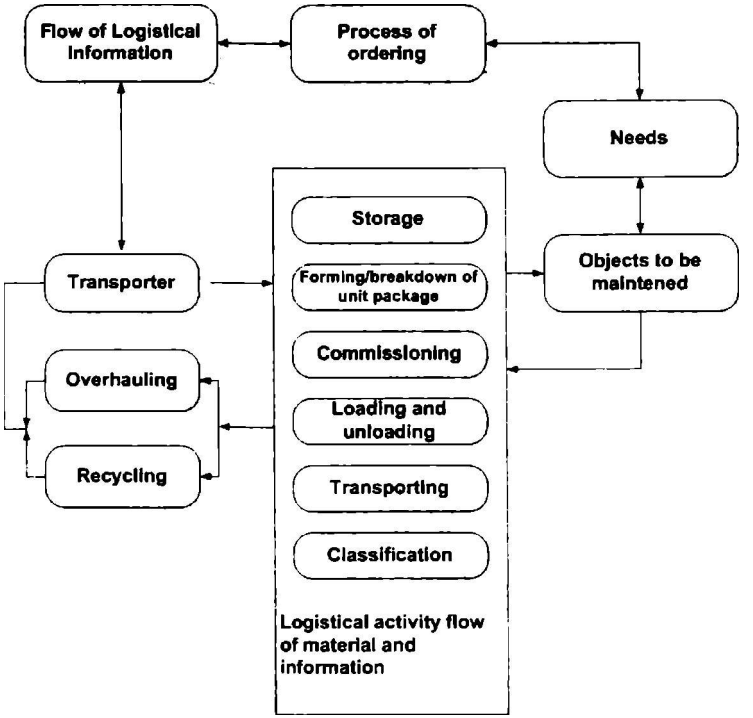


Figure 2. Logistical features of order fulfilment necessary for maintenance

In the system of order fulfilment for maintenance, the delivery of new materials and equipment is always taken into account, because for example necessary materials such as oil or glue cannot be re-used, and equipment needed is considered not to have to be overhauled because of the maintenance demand.

The need for parts can be satisfied from the viewpoint of logistics in three fundamentally different ways:

transport of either new or overhauled parts from the transporter's warehouse,

manufacture of the part, then transport, or

removal of a given part from its original place and its replacement after repair,

If the transport is from storage (a new or overhauled part) then the time to fulfilment of the order is:

in the case of parallel activities

$$t_{HR1} = \max_{i_r} \left\{ t_{HR1_{i_r}} \right\} \tag{4.1}$$

and in the case of serial activities

$$t_{HR1} = \sum_{i_r=1}^{n_{i_r}} t_{HR1_{i_r}} \tag{4.2}$$

$$t_{HR1_{i_r}} = t_T^{i_r} + t_E^{i_r} + t_K^{i_r} + t_R^{i_r} + t_S^{i_r}, \tag{4.3}$$

where:

t_{HR1} — time for order fulfilment for a given maintenance task in case of delivery from storage,

$t_{HR1}^{i_r}$ — time to order fulfilment of a transported part for transporter i_r for the given maintenance activity,

$t_T^{i_r}$ — storage time needed in case of supply between transporter i_r and the item to be maintained,

$t_E^{i_r}$ — time needed for forming and breaking down unit packages in case of supply between transporter i_r and the item to be maintained,

$t_K^{i_r}$ — time needed for order picking in case of supply between transporter i_r and the item to be maintained,

$t_R^{i_r}$ — the time needed for loading and unloading in case of supply between transporter i_r and the item to be maintained,

$t_S^{i_r}$ — the time needed for transport in case of supply between transporter i_r and the item to be maintained.

The time needed to carry out logistical activities:

$$t_{\theta}^{i_r} = \sum_{j_r=1}^{n_{j_r}^{i_r}} \sum_{\kappa_\beta=1}^{n_{\kappa\beta}^{i_r}(j_r)} t_{\beta_{j,\kappa\beta}}^{i_r}$$
$$i_r = 1,2 \ldots n_{i_r} \qquad (4.4)$$
$$\beta = \{T, E, K, R, S\},$$

where:

i_r	identifier of the supplier,
n_{i_r}	maximal number of suppliers,
j_r	index regarding necessities,
n_{j_r}	maximal number of necessities,
κ_β	index regarding logistics services,
n_{κ_β}	maximal number of given logistics service,
T	index regarding storage activity,
E	index regarding loading unit formation and disassembling,
K	index regarding order picking,
R	index regarding loading in,
S	index regarding transportation.

Summarizing the content of (4.4):

$$t_{HRI} = \max_{i_r} \left\{ \sum_{j_r=1}^{n_{j_r}(i_r)} \sum_{\kappa_t=1}^{n_{\kappa_t}(i_r,j_r)} t_{T_{i_r,j_r,\kappa_t}} \sum_{j_r=1}^{n_{j_r}(i_r)} \sum_{\kappa_e=1}^{n_{\kappa_e}(i_r,j_r)} t_{E_{i_r,j_r,\kappa_e}} \right.$$
$$\left. + \sum_{j_r=1}^{n_{j_r}(i_r)} \sum_{\kappa_k=1}^{n_{\kappa_k}(i_r,j_r)} t_{K_{i_r,j_r,\kappa_t}} + \sum_{j_r=1}^{n_{j_r}(i_r)} \sum_{\kappa_r=1}^{n_{\kappa_r}(i_r,j_r)} t_{R_{i_r,j_r,\kappa_r}} + \sum_{j_r=1}^{n_{j_r}(i_r)} \sum_{\kappa_s\ 1}^{n_{\kappa_s}(i_r,j_r)} t_{S_{i_r,j_r,\kappa_s}} \right\}, \qquad (4.5)$$
$$i_r = 1, \ldots n_{ir}.$$

It can be prescribed as an object function that (4.5) should be minimal:

$$t_{HRI} = \max_{i_r} \left\{ \sum_{j_r=1}^{n_{j_r}(i_r)} \sum_{\kappa_t=1}^{n_{\kappa_t}(i_r,j_r)} t_{T_{i_r,j_r,\kappa_t}} + \sum_{j_r=1}^{n_{j_r}(i_r)} \sum_{\kappa_e=1}^{n_{\kappa_e}(i_r,j_r)} t_{E_{i_r,j_r,\kappa_e}} + \sum_{j_r=1}^{n_{j_r}(i_r)} \sum_{\kappa_k=1}^{n_{\kappa_k}(i_r,j_r)} t_{K_{i_r,j_r,\kappa_t}} + \right. \tag{4.6}$$

$$\left. + \sum_{j_r=1}^{n_{j_r}(i_r)} \sum_{\kappa_r=1}^{n_{\kappa_r}(i_r,j_r)} t_{R_{i_r,j_r,\kappa_r}} + \sum_{j_r=1}^{n_{j_r}(i_r)} \sum_{\kappa_s=1}^{n_{\kappa_s}(i_r,j_r)} t_{S_{i_r,j_r,\kappa_s}} \right\} \rightarrow min$$

On base (4.6) it can be stated, that the objective function in case of constituent supply from maintenance transport storage is dependent on:

the number of applied transporters $\left(n_{i_r}\right)$,

the number of the type of constituents to be transported $\left(n_{j_r}\right)$,

- on the number of the stockings $\left(n_{\kappa_t}\right)$,
- on the number of the construction and re-assembly of unit consignments $\left(n_{\kappa_r}\right)$,
- on the number of the applied order picked units $\left(n_{\kappa_c}\right)$,
- on the number of applied loadings $\left(n_{\kappa_k}\right)$,

per transporters and type of constituents being involved in the supply.
the time demand for each logistical activity for a given transporter and part.

The actual time needed for logistical activities is composed of two parts:

the actual technological time of the logistical activity,
the waiting times for equipment that is necessary for the given logistical activity and perhaps also the waiting time for the object to be maintained.

Each of the elements in (4.3) can be written as

$$t_{\alpha}^{i_r,j_r} = t_{\alpha}^{'i_r,j_r} + t_{\alpha}^{''i_r,} \tag{4.7}$$

where:

$t_{\alpha}^{i_r,j_r}$ is the actual time for a given maintenance activity for logistical activity α for product j_r from transporter i_r,

$t_{\alpha}^{'i_r,}$ is the technological time for a given maintenance activity for logistical activity α for

$t_{\alpha}^{''i_r,}$ product j_r from transporter i_r, is the waiting time for a given maintenance activity for logistical activity α for product j_r from transporter i_r.

The following equations can be associated with equation (4.6):

$$t_{\alpha}^{i_r,j_r} = t_{\alpha}^{'i_r,} + t_{\alpha}^{''i_r,j_r} \rightarrow min \tag{4.8}$$

that is, if

$$t_{\alpha}^{''i_r,} \rightarrow 0 \tag{4.9}$$

then

$$t_{\alpha}^{'i_r,} \rightarrow min \tag{4.10}$$

The basic principles of the optimal selection of the maintenance logistical services, shown in (4.9) and (4.10):

the waiting time for logistical activities should be zero as possible,
the logistical technology time for given constituent and transporter should be minimal.

If a part needed for the maintenance activity is supplied by the overhaul of a removed part at a different location, then when determining the time to order fulfilment, the following times should be taken into account:

the time needed for the logistical activity between the object to be maintained and the location of the overhaul $\left(t_B\left(i_r, j_r\right)\right)$,

the actual overhaul time $\left(t_F\left(i_r, j_r\right)\right)$,

In this case the time to order fulfilment is:

$$t_{HR2} = t_{HR1} + t_B\left(i_r j_r\right) + t_F\left(i_r j_r\right). \tag{4.11}$$

If the maintenance activity is proceeded by the manufacture of the necessary part, the total manufacturing time $\left(t_G\left(i_r, j_r\right)\right)$, should be taken into consideration:

$$t_{HR3} = t_{HR1} + t_G\left(i_r j_r\right). \tag{4.12}$$

The necessary times of demand for the materials and equipment can also be written in a similar way to (2.1)

Let us denote, using (1), the points in time related to maintenance activities at which the necessary items are available:

$$t_{a_A} = t_{MA} + t_{HA}; \quad \text{for materials,} \tag{4.13}$$

$$t_{a_R} = t_{MR} + t_{HR}; \quad \text{for parts} \tag{4.14}$$

$$t_{a_E} = t_{ME} + t_{HE}; \quad \text{for equipment} \tag{4.15}$$

t_{a_S} ; the point in time at which the service is available.

Let us denote the point in time when the demand for maintenance is initiated by t_k. Then, a possible point for initiation of maintenance activity for the item to be maintained is:

$$t_{IND} = t_k + max\left\{t_{a_A}; t_{a_R}; t_{a_E}; t_{a_S}\right\} \tag{4.16}$$

On foundation of the time necessity the logistical process model of the maintenance activity has been worked out in the paper. Using the model it turns out,

what kind of time parts are involved by the maintenance logistical activity,
what kind of connection is between the unique time parts and the main logistical parameters,
that different transit times might be occurred on base of the different logistical parameters,
that the determination of the optimal transit time can be done by using a multi-parameter solution mass.

The problem for searching the optimal variation is not be object of this paper, but the time parameters which have to be investigated by which approximately 90% of the total time necessity can be influenced are given.

REFERENCES

[1] KURT, M.: *Tashenbuch Instandhaltungslogistik Qualität steigern.* Carl Hanser Verlag München, Wien, 1999.

[2] MAGGARD, B.,N.: *Instandhaltung, die funktioniert.* Verlag moderne Industrien, Landsberg 1995.

[3] ILLÉS, B.: *Information flow in a logistical system of maintenance,* Gépgyártástechnológia XXXVIII, 6, pp. 55-57, 1998. (in Hungarian)

[4] ILLÉS, B.: *Logistical management of maintenance,* Gépgyártástechnilógia XXXIX, 3, pp. 1-6, 1999. (in Hungarian)

[5] ILLÉS, B.: *Logistical consequences of maintenance activities* LOGINFO, Magyar Logisztikai Egyesület, 2, pp. 19-22, 1999. (in Hungarian)

[6] CSELÉNYI, J., ILLÉS, B., KOTA, L.: *Virtuelle Zentrale zur Disposition der Inspektion räumlich verteilter Objekte,* Magdeburger Schriften zur Logistik, MSL-Heft 1, pp. 61-66, 2002.

[7] ILLÉS, B., CSELÉNYI, J.: *Disposition von Personal, Material und Dienstleistungen bei räumlich verteilten Wartungsobjekten,* Conferencia Ceintifica Internacional de Ingenieria Mecánika, Universidad Central „Marta Abreu" De Las Villas, Santa Clara (Cuba), COMEC 2004, on CD-ROM, 6 pages.

Production Systems and Information Engineering
Volume 3 (2006), pp. 109-119

HYPOTHESIS-BASED SEARCH IN PARTLY-OBSERVABLE SYSTEMS

TAMÁS BÁKAI
University of Miskolc, Hungary
Department of Information Engineering
iitrifle@gold.uni-miskolc.hu

[Received November 2005 and accepted May 2006]

Abstract. Nowadays the growing demands are the dominant concept in most parts of life. To satisfy these demands, planning of more complex and flexible problem-solving systems is required. In the last decades many new technology and technique were developed to handle the growing demands [6][7][8][9]. Apparently the object-oriented modelling methodology was the most efficient between them. However, nowadays the growing demands of the market gradually outgrow the abilities of the pure object-oriented concepts. One of the main reasons of this is that the decomposition techniques can not handle efficiently the numerous sub-systems with varying objective functions and constraints. The common problems appear in the untreatable complexity and the missed deadlines. The use of artificial intelligence means new concepts in the field the development processes. The agent based programming gives the possibility to describe the functionality of the required system not only by using actions-reactions but by defining the goals and constraints in the system. The machine-learning helps to determine the connections, relations and logical behaviour in the dynamism of the modelled system and helps to reveal the effects of the non-modelled systems into the modelled system. This paper shows a method for revealing and handling the effects of a non-modelled system according to the observed behaviour of the modelled system.

Keywords: learning systems, partly-observable systems, identification, cause-effect relations

1. Modelling the Observable-Systems

The basic concept of the machine learning is based on the observations of a functioning system and the knowledge gathering from the observed data. The observed systems according to their perceptibility can be categorized as observable and partly-observable systems [1]. In the learning system developed at the University of Miskolc, the observed system can be modelled using the object-oriented concept. The objects model the main and separate elements of the

observed system. The attributes describe the properties of the objects and the values represent the concrete cases of the attributes. The activities model the connections between the objects of the system. In this system each object has to have one or more attributes and zero or more activities and each attribute has to have two or more values [2]. The structure of the connections between the objects, attributes, values and activities can be described by a tree model. Each element represents a node of the tree.

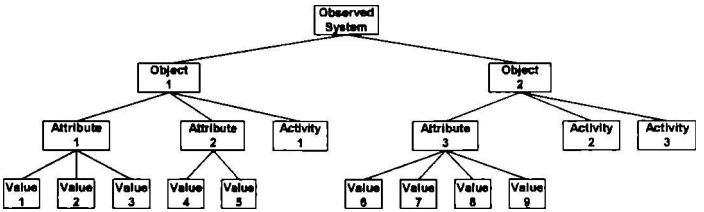


Figure 1. Structure of the connections between the objects, attributes, values and activities of the observed system

The activities and the values in this concept have no sub-elements therefore these are named leaf-nodes. Each leaf-node has one parent-node. The parent-node of a leaf-node represents the node the actual leaf-node belongs to directly. The parent-node of a value is the attribute the value belongs to and the parent-node of an activity is the object the activity belongs to. Each attribute has one marked value among its values at each time moment. This marked value represents the state of that attribute at that time moment. So the marked value is named active-value. The state of the observed system at time t can be described by the set of its active-values at time t.

The observed system has one marked activity among its activities at each time moment. This activity represents the event in the observed system of that time moment. The marked activity is named active-activity.

In the practice, the sampling frequency of the observing system is much higher than the frequency of the changes of states of the observed system. Therefore those complex events that generate changes of state can be separated into the sequence of single events. The event which occurs at time t_k represents the transient signal for the t_k change of state process and after reaching the t_{k+1} state of the observed system the active-event becomes inactive again.

Each leaf-node has one super-parent-node, too. The super-parent-node of a leaf-node represents the node whose one possible value the actual leaf-node represents. The super-parent-node of a value represents a possible value of the attribute the actual value belongs to; therefore the super-parent of each value is the attribute the actual value belongs to. The super-parent-node of an activity represents a possible

value of the events in the observed system therefore the super-parent-node of each activity is the observed system itself.

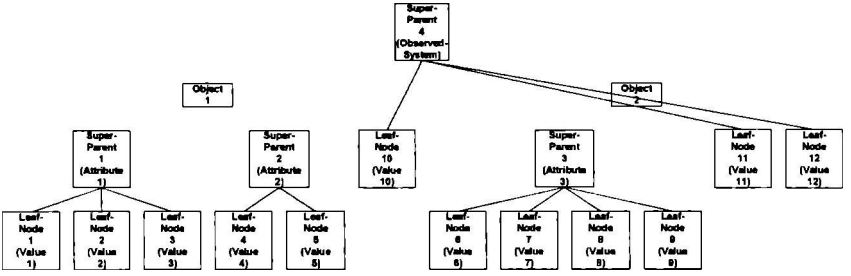


Figure 2. Structure of the super-parent and the leaf-node elements

The state of the observed system at time t can be described by the set of active-values at time t and the event which occurs at time t can be described by the active-activity at time t. The dynamic behaviour of the observed system can be modelled by its changes of state. The t_k change of state contains three parts:

1, the state of the observed system at time t (source-state of the change of state),
2, the event at time t,
3, the state of the observed system at time $t+1$ (destination-state of the change of state).

The changes of state are stored in a history database in chronological order. The t-th entry of the history contains the state of the observed system at time t_k and the event which occurred at time t_k.

2. Meaning of inconsistence in the changes of state processes

Denote each state of the observed system with uppercase letters and each event of the observed system with lowercase letters of the alphabet. Denote same letters the states with the same set of active-values and the same events and denote different letters the states with different set of active-values and the different events. According to this a possible state of the history can be as follows:

Table 1. Example of a possible pattern of the history database

time sequence	history
k	A, a
k+1	B, c
k+2	C, a
k+3	A, b
k+4	B

Four types of changes of state can be distinguished:

Table 2. The main different types of the changes of state

1. same source-state and same event result the same destination-state	*2.* same source-state and same event result different destination-state
3. different source-state or different event result the same destination-state	*4.* different source state or different event result different destination-state

In the changes of state of *types 1*, *3* and *4* the source-state and the event determine unambiguously the destination-state. The changes of state of these types are consistent. In the changes of state of type *2* the source-state and the event do not determine unambiguously the destination-state. The changes of state of these types are inconsistent.

If an observed system contains inconsistent changes of state then the cause-effect relations in the observed system can not be revealed unambiguously. This case indicates that the observed system is partly observable only.

3. Properties of the inconsistent changes of states

If in the changes of state processes of the observed system two changes of states become inconsistent then this proves that the observed system contains non-modelled elements. Modelling of these non-modelled elements can distinguish the same source states or the same events of the changes of states of type *2*. Therefore these changes of states could be converted into the changes of states of type *4*. According to the behaviour of the non-modelled elements the inconsistent changes of states can be divided into two main groups namely the Non-Observable-State-Space (NOSS) based and the sequential characteristic based inconsistent changes of states. For example if the observed system is a logical gate-circuit with an existing but non-modelled enabling input [3][10] then the observed changes of

states contain NOSS-based inconsistence. If the observed system is a container filling-emptying system which has one input and one output tap and three sensors which indicates the level of liquid at full, half and empty states, then the observed changes of states contain Sequential-Characteristic based inconsistence. To eliminate the different types of inconsistence two elimination methods was developed.

To eliminate the NOSS-based inconsistence, an automatically generated state-space with objects, attributes, values and activities is needed. The dynamic behaviour of this NOSS-based system can be determined indirectly according to the observable changes of state. The correct static structure and dynamic behaviour of the NOSS-based system can not be determined correctly at any time. The only thing the learning system can do is to determine the NOSS-based system in order to eliminate the inconsistence of the changes of states until the actual state of the system. Each newly observed change of state may modify the complete structure and behaviour of the automatically generated NOSS-based system.

To eliminate the sequential based inconsistence retrospection is required in the sequence of the observed changes of states. For example regarding to the container filling-emptying system the destination state of the changes of states can be determined unambiguously taking into consideration not only the actual state of the system but some sequence of states in the past from the actual state. This method is named *n-steps-deep-retrospection*. In the previous container filling-emptying example only one retrospection step is sufficient to eliminate the inconsistence in the changes of states.

$$\begin{aligned} (empty, half) &\Rightarrow full \\ (full, half) &\Rightarrow emty \end{aligned} \tag{1}$$

Detailed analysis is required to determine which method gives the best performance for eliminating the inconsistent changes of states of the actual observed system. This kind of research gives the developing possibilities of this system today. According to the latest results both methods can eliminate both types of inconsistence but they are different from each other according to their need for additional information.

4. The inconsistence elimination ability of The NOSS-based method

The NOSS-based elimination method assumes that the inconsistence in the changes of states is based on non-modelled objects, attributes, values or activities of the observed system. According to this concept the actual state of the system contains an observable and a non-observable part. In this point of view two states are equivalent with each other if not only their observable but also their non-

observable parts are equivalent with each other. Using this concept gives the possibility to distinguish the equivalent source-states of the changes of states of *type 2* with non-equivalent non-observable state space parts associated to their source states. Therefore the changes of states of *type 2* can be converted into the changes of states of *type 4*.

There is a dominant difference between the observable and non-observable parts of the states according to their knowableness. Reaching the t-th state of the observed system the observable part of this state defines correctly and invariably the observed state. On the other hand, generating the non-observable part for the t_k state gives nothing else than the changes of states will be consistent until the t_k state. However, many other non-observable state space combinations can give the same result. Choosing between them means an optimization task after each change of state step of the observed system. The constraint of this optimization is to define a non-observable state space for each state of the observed system in order to eliminate the inconsistence of the changes of states. The objective function of this optimization is to minimize the additional information need for the definition of the non-observable state space. To satisfy the constraint of this optimization the forbidden states of the non-observable state space has to be defined for each state of the observed system. The forbidden states represent the permanent information about the behaviour of the non-observable parts of the states. The forbidden states for each state are defined in the Non-Observable-State-Space-Constraint (NOSSC) of each state.

Denote with *!i* in the NOSSC associated to the j-th state of the observed system that the non-observable part of the j-th state can not be equivalent with the non-observable part of the i-th state.

Similarly to the observable part of the observed system, the static structure of the non-observable part is modelled by objects, attributes, values and activities. These elements may be attached to an observable or a non-observable element of the modelled system. To determine the concrete need for each type of these elements and to reveal their necessary connections with each other requires deeper analysis in the static structure and dynamic behaviour of the observed system. At this time the non-observable part of each state is modelled using only one object (namely: *NOObject*), one attribute (namely: *NOAttribute*) and as many values (namely: *0*, *1*, *x*) as many non-equivalent states of the non-observable parts require.

5. The forbidden equivalences of the Non-Observable Parts

The elimination of the inconsistent changes of states of the observed system using the NOSS-based method means that in the changes of states with equivalent source-state and event but with non-equivalent destination-state the equivalence of

the source-states has to be eliminated. The source-states of the inconsistent changes of states are in forbidden equivalence. Because the observable part of each state holds permanent information and can not be modified, therefore a non-observable part is needed for eliminating the equivalence of these source-states. Figure 3 shows an example of the NOSS-based inconsistence elimination.

time	observable state space
...	...
i	A, a
i+1	B, ?
...	...
j	A, a
j+1	C

$$A \xrightarrow{a} B$$

$$A \xrightarrow{a} C$$

3.a. Example of inconsistent changes of states

time	observable state space	NOSSC	non-observable state space
...	...	...	...
i	A, a	?	0
i+1	B, ?	?	?
...	...	....	...
j	A, a	!i	1
j+1	C	....	...

$$A0 \xrightarrow{a} B$$

$$A1 \xrightarrow{a} C$$

3.b. Example of the elimination of the inconsistent changes of states

Figure 3. Example of NOSS-based inconsistence elimination

In this example the changes of states (*i*-th and the *j*-th) are inconsistent with each other and the properly generated non-observable part eliminates their inconsistence by taking their source states different. The *?* sign denotes the unimportant elements of the state according to this example.

Because the destination state of each changes of state is the source state of the next changes of state simultaneously, therefore each NOSS-based inconsistence elimination step may cause a side effect. A side effect appears each time if the previous states of the modified states are equivalent with each other. If the states *i*-

1 and *j-1* are equivalent with each other then before the non-observable part of states *i* and *j* become modified, the type of these previous changes of states were of *type 1*. After the non-observable part of states *i* and *j* will be modified the type of these previous changes of states become of *type 2*. These mean that these previous changes of states inconsistent with each other. To eliminate the occurred side effect, the equivalence of states *i-1* and *j-1* has to be eliminated too. This step may cause a new side effect too, if the previous states of the modified states are equivalent with each other. To eliminate all of the side effects occurred, the equivalence elimination method has to be performed until either the previous states of the modified states are non-equivalent with each other or one of the modified states is the first state of the observed system. Table 3. shows an example of side effects elimination steps.

Table 3. Example for eliminating the inconsistence and its side effects

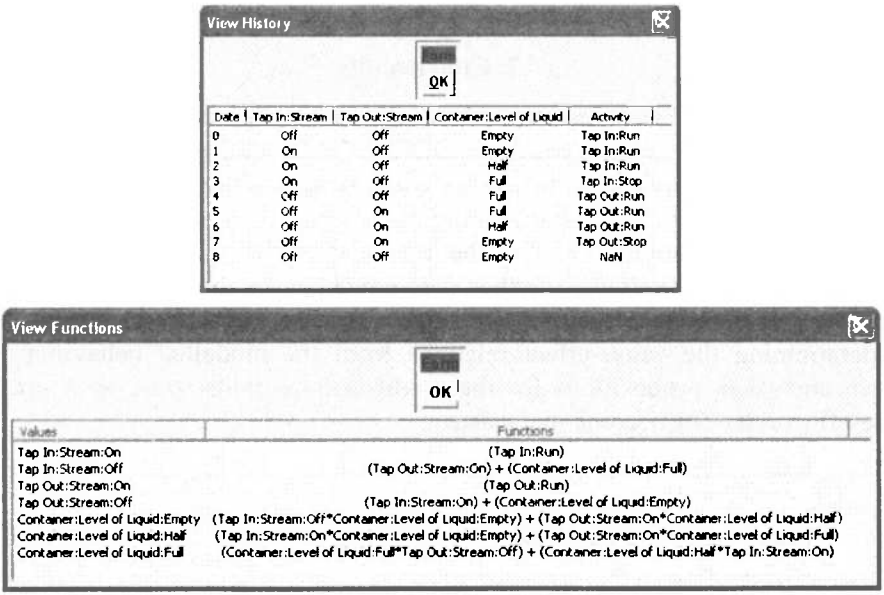
time	observable state space	NOSSC
0	?	?
1	?	?
...	...	...
i-k-1	B, a	?
i-k-1	C, ?	?
i-k	F, ?	?
i-2	D, c	?
i-1	B, a	?
i	A, a	?
i+1	B, ?	?
...	...	...
...	...	!0
...	...	...
j-k-1	B, a	!i-k-1
j-4	A, ?	!i-k-1
j-k	E, ?	!i-k
j-2	D, c	!i-2
j-1	B, a	!i-1
j	A, a	!i
j+1	C, ?	?

Because the state *j-k* (which belongs to the *k*-th inconsistence elimination step) is not equivalent with the state *I-k* no more side effects occur and the inconsistence elimination steps could be finished. If the state *j-k-1* (which belongs to the *k-1*

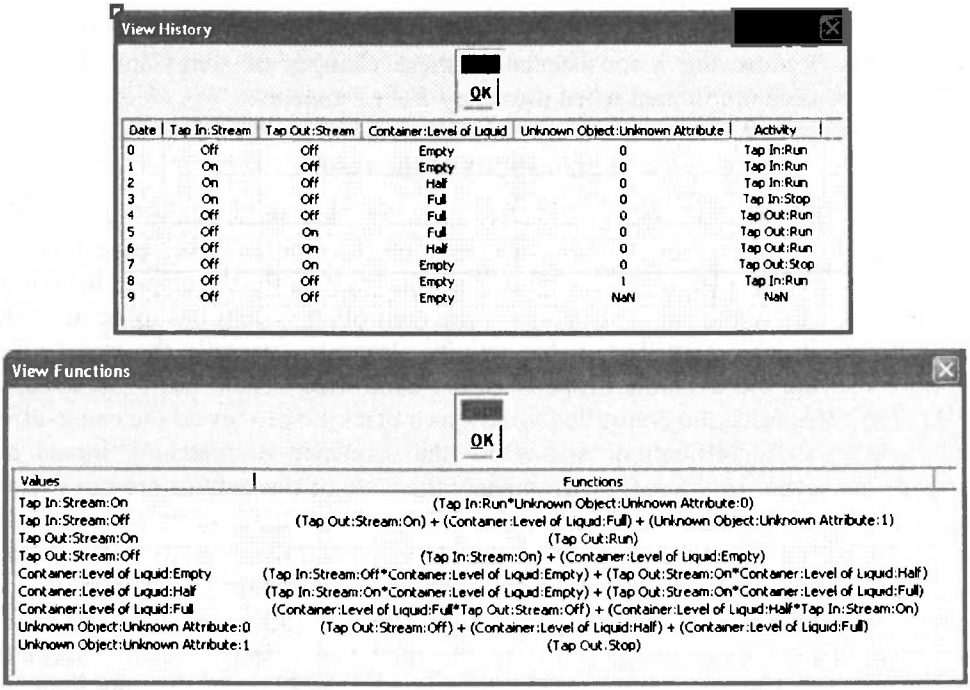
inconsistence elimination step) is equivalent with the state *I-k-1* then it results no side effect because the inconsistence of these changes of states and their side effects have been eliminated when the state *j-k-1+1* appears.

6. Applicability of the results

Nowadays the market has gradually growing demands against automation processes. In order to satisfy these demands more complex developing tools are required [7]. When planning a complex controller system the developers have to do two things. Firstly the static structure of the controlled system has to be modelled by defining its object-attribute-value-activity elements; secondly the cause effect relations giving the dynamic property of the controlled system has to be revealed [4] [5] [6]. Regarded the controlled system as a black-box to reveal the cause-effect relations is an identification task where the sequence of matching inputs and outputs are given. In the real environments the state of the outputs are not defined by the state of its matching inputs only. The cause of this lays on the sequential behaviour of the controlled system on the one hand and the wrongly defined static structure on the other. To divide the modelling of the controlled system into an observable and a non-observable part can give the possibility to reveal and handle the logical and the sequential parts of the modelled system and to correct the structural deficiencies. Figure 4 shows some screenshots of this implemented modelling tool.



4.a, Logical functions of the controlled system contains no non-observable elements



4.b, Logical functions of the controlled system contains non-observable elements

Figure 4. The revealed logical functions of the system identification method

7. Conclusions

The use of the NOSSC-based method for eliminating the inconsistent changes of states of the observed system results in an NOSSC for each state. The NOSSC of a state contains the states from which the equivalency of the actual state has to be eliminated. To determine the non-observable part of a state according to its NOSSC means the determination of the less value of the *NOAttribute* which is not the value of the *NOAttribute* of either state contained by the NOSSC of the actual state. This inconsistence elimination method gives a helpful tool for the modeller for determining the cause-effect relations from the modelled behaviour of the system and takes propositions for the modifications of its static structure if the cause-effect relations become inconsistent.

REFERENCES

[1] RUSSELL, S. J. NORVIG, P.: *Artificial Intelligence in Modern Approach.* Panem Könyvkiadó, Budapest, 2000. in Hungarian)

[2] KONDOROSI, K. LÁSZLÓ, Z., SZIRMAY-KALOS, L.: *Object Oriented Software Development.* ComputerBooks, Budapest, 1999. (in Hungarian)

[3] BÁNHIDI, L., OLÁH, M.: *Automation for Engineers.* Tankönyvkiadó, Budapest, 1992. (in Hungarian)

[4] VENKATESH, K., ZHOU, M.: *Object-Oriented Design of FMS Control Software Based on Object Modeling Technique Diagrams and Petri Nets*, Journal of Manufacturing Systems, pp. 118-136, 1998.

[5] CASTILLO, L., SMITH, J. S.: *Formal Modeling Methodologies for Control of Manufacturing Cells*: *Survey and Comparsion.* Journal of Manufacturing Systems, Vol 21/No 1, pp. 40-57, 2002.

[6] GAERTNER, N., THIRION, B.: *GRAFCET an Analysis Pattern for Event Driven Real-time Systems*, Plop Conference, p. 11, 1999.

[7] TÓTH, T.: *Design and Planning Principles, Models and Methods in Computer Integrated Manufacturing.* Miskolci Egyetemi Kiadó, 1998. (in Hungarian)

[8] BÁKAI, T.: *New Methods and Tools for Supporting the Logical Control Design of Machine Lines in the Manufacturing Industry*, 6th International Carpathian Control Conference, Miskolci Egyetemi Kiadó, pp. 279-286, 2005.

[9] BÁKAI, T.: *Identification of PLC Based Control Systems*, microCAD 2005 International Scientific Conference, pp. 13-18, Miskolci Egyetemi Kiadó, 2005.

[10] AJTONYI, I., GYURICZA, I.: *Programmable Control Systems, Networks and Systems.* Műszaki Könyvkiadó, Budapest, 2002. (in Hungarian)

Production Systems and Information Engineering
Volume 3 (2006), pp. 121-137

MODELING AND SOLVING OF THE EXTENDED FLEXIBLE FLOW SHOP SCHEDULING PROBLEM

GYULA KULCSÁR
University of Miskolc, Hungary
Department of Information Engineering
kulcsar@ait.iit.uni-miskolc.hu

FERENC ERDÉLYI
Production Information Engineering Research Team (PIERT) of the
Hungarian Academy of Sciences
University of Miskolc, Hungary
Department of Information Engineering
erdelyi@ait.iit.uni-miskolc.hu

[Received October 2005 and accepted April 2006]

Abstract. This paper discusses the production control problems of customized mass production, which can be described as combining make to stock and make to order type production at the same time. In order to solve scheduling and resources allocation issues, a new computer model for customized mass production will be presented. The focus has been set to determine alternative routes and machines allocation for feasible scheduling. We have developed a computer framework to check different approximate heuristic algorithms, the result of which will be summarized in this paper.

Keywords: shop floor scheduling (assigning, sequencing, simulation), alternative routes, parallel machining, constraint, due date

1. Introduction

In essence there are two kinds of manufacturing: make to order (MTO) and make to stock (MTS). The production of unique or complex goods falls into the first category. MTO production can be characterized by individual or customer specified products usually designed from a set of firm level components, and small batch production on universal machines at job-shop environment.

Make to stock manufacturing is used in mass production, where the finished products are delivered from a warehouse when customer requests and purchases them. Mass production technology usually has automated manufacturing and/or assembly lines, highly skilled or specialized workers, big lot sizes, automated quality checking, automated packing operations, and relatively high material stock level.

Certain manufacturers may use both procedures: in addition to satisfying their accepted orders they can utilize their manufacturing resources for producing goods for stock. In this customized mass production paradigm the firms plan their production partially for external direct orders, arriving from logistic or shopping centers but to reach better delivery dates they must make forecast for manufacturing semi-finished products and buying materials with long external lead time [5]. In this business environment there are lots of uncertainties originated from incoming orders and unavailability of machines, equipment or human resources. For this reason, real time production data collection, fast interactive information management and effective scheduling of tasks are the most important tools to achieve production goals. These are the main functions of Manufacturing Execution Systems [7].

2. Scheduling Problems in Customized Mass Production

2.1. Classification of Shop Floor Scheduling Models

Scheduling is the allocation of a set of well-defined resources to a set of given tasks subject to some pre-determined constraints, in order to satisfy a specific objective. In order to formulate a scheduling problem, the specification $\alpha/\beta/\gamma$ is typically used [1], where:

- α machine environment,
- β processing characteristics and constraints, and
- γ objective functions.

Production of parts is carried out in batches. The batch size may vary from one part (manufactured in job shops) to millions of parts (manufactured in production lines). Depending on how the jobs are executed at the shop floor (i.e. the sequence, in which jobs visit machines), we can classify manufacturing systems as one of:

- serial systems (also called flow shops) or
- non-serial systems (job shops).

The following figures (Figure 1 and Figure 2) summarize some typical flow types.

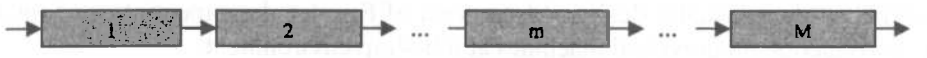


Figure 1. Flow Shop Scheme

In flow shop structure, there are machines *(m 1,... M)* in sequence. Unlimited intermediate storage between two successive machines is usually assumed. All jobs have the same routing. Each job has to be processed on each one of the *m* machines. Permutation flow shop means that the queues in front of each machine operate according to FIFO (First In First Out discipline).

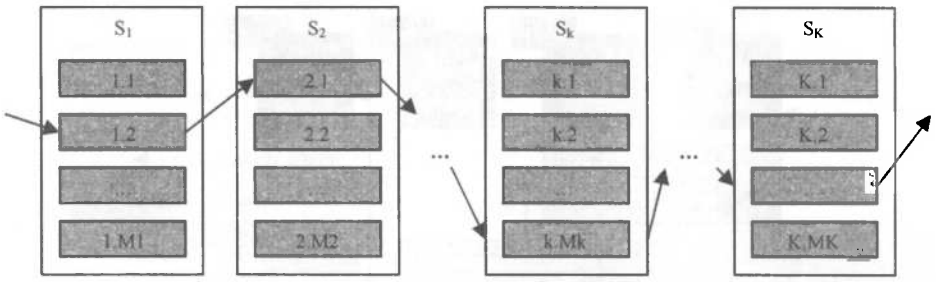


Figure 2. Flexible Flow Shop Scheme

The flexible flow shop environment has stages, at stage S_k *(k=1, ..., K)* there are M_k identical machines in parallel. There is usually unlimited intermediate storage between two successive stages. Each job has to be processed at each stage on any of the machines.

In this paper, we are focusing on extended flexible flow shop machine environments with alternative routes, parallel machines and unlimited intermediate storages. The following sections will deal with an extended flow shop environment with four stages (see Figure 3).

2.2. Extended Flexible Flow Shop Scheduling Problem

The problem is inspired by a real case study concerning a Hungarian firm specialized in lighting products. It is a customized mass production approach so that an order-book for a given time period corresponds to different products to be produced in required quantity. We concentrate on generating short-term production schedule of the manufacturing processes.

This section covers the scheduling problem in detail, the related entities of the system, and how they relate to each other. The whole scheduling model can be described as follows. We present the machine environment (α), the processing characteristics and constraints (β) and the objective functions (γ).

Product Type: The product type can be defined as the combination of components that a machine is capable of handling. The combination uses the AND operator and the OR operator to combine various lists of components.

Production Orders: There are production orders. A production order includes the type of the final product to be manufactured, the required quantity and the defined due date. In order to satisfy a production order, the components are taken through various processes before finally becoming the final product. Manufacturing includes a set of steps that involve the actual production of the final product.

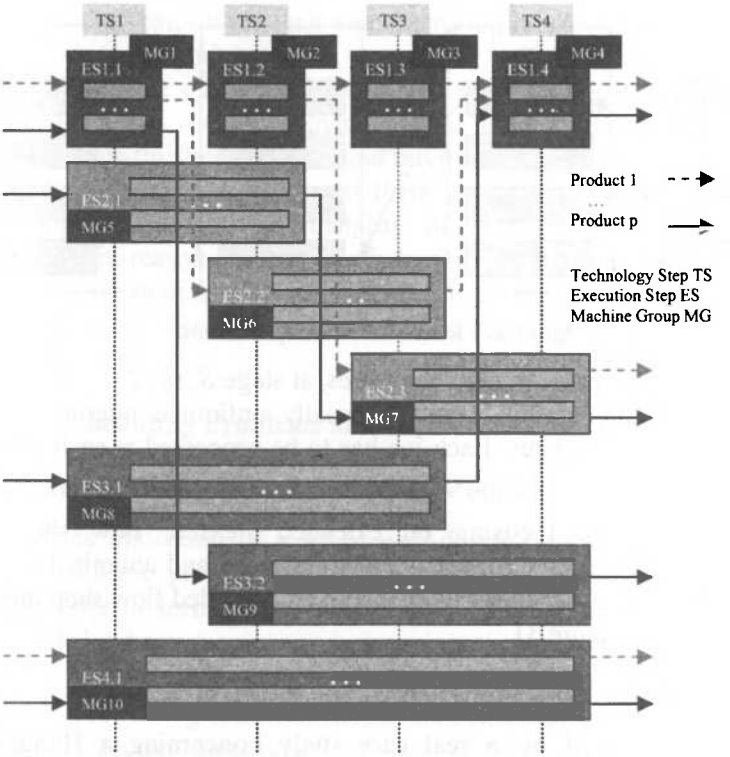


Figure 3. Extended Flexible Flow Shop Scheme

Technology Steps: The steps under the technology processes are termed as technology steps. Typically, in manufacturing we have four technology steps like preparation, assembly, quality checking and packaging. Preparation is the first step of the technology processes when defined properties of certain components have to be modified. Assembly is the technology step in which the components are assembled together, quality checking has two parts: a forced wait time when the quality of the product is observed and ascertained before going through packaging finally. A technology step may include some operations, but we suppose that no pre-emption is allowed at the level of the technology steps.

Execution Steps, Execution Routes: There is a very important concept namely execution step in this environment. The execution step is a well-defined set and sequence of technology steps (Figure 4). It describes which technology steps to be processed on the same production line. If the execution step includes two technology steps (i.e.: TS1 and TS4), it has to include all technology steps, which are between these two steps (TS1, TS2, TS3, TS4), as well. Moreover, the sequence of execution steps is called execution route. So each execution route

includes all technology steps required in order that the final product can be produced. An execution route can include one or more execution steps, but the common part of included execution steps, which is a set of technology steps, has to be an empty set (Figure 5).

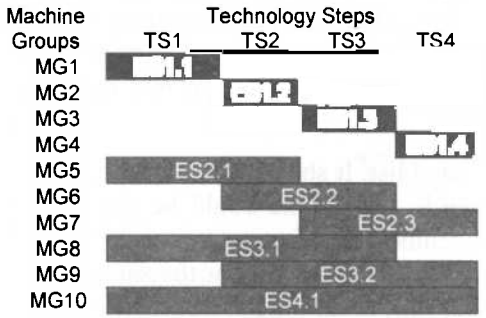


Figure 4. Execution Steps

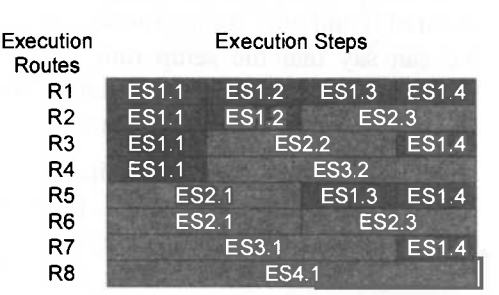


Figure 5. Execution Routes

Pallets: Each product is normally packaged on standard sized pallets. Each pallet consists of a pre-defined number of the finished products. Even though the physical pallets come into existence only after packaging is over, for convenience reasons, we start looking at the logical pallets right from the beginning. Hence, we schedule pallets (job). A production order is first identified to be consisting of a particular number of pallets and the production order will be closed when all of these pallets have gone through all the technology steps.

Machines: In our scheduling model, a machine is a production line that consists of a group of workplaces, which are lined in a sequence; the output of the first workplace becomes the input of the second one and so on. Typically these production lines are inseparable unites. Hence when it comes to scheduling, then the production line should be considered as one unit and not the individual workplace.

Production Rate and Machine Capacity: During manufacturing, we have to schedule units (pallets) on production lines (machines). In order to estimate the process time taken by a pallet on a machine, we need to know the production rate and the capacity of that machine to process that pallet. Production rate is normally specified as quantities producible per time unit on that machine. The capacity of a machine could vary based upon what is the final product it is producing and what is the effective shift time in the calendar.

Process Time: The production rate of the machine is required while computing the process time of a pallet on the machine. Process Time means the processing time of a pallet spending on a machine (in time unites).

$$\text{Process Time} = \text{Scheduled Quantity} / \text{Production Rate}. \quad (1)$$

Setup Time: This is one of the most important properties of the machine (production line). This does not affect the producibility of a final product directly. By definition, a setup time (changeover time) gives the time delay to changeover from one product type to another product type. In our current model, the setup is required if and only if the product type of the last pallet different from the next one. We can say that the setup time value only depends on the product type to be processed, so it is allowed to define multiple value setup time for one machine. Setup time is specified in time units.

Machine Group: Each machine has an associated list. It shows all technology steps can be executed on a machine. In other words, a machine could be potentially capable of executing an execution step. A machine group is a set of machines that can execute the same execution step (Figure 4). The machines in the same group are parallel machines with different production rate, setup time and capacity.

Alternative Execution Routes: A given final product can be produced differently, because there are different execution routes on which the required components are taken through before becoming the final product. These alternative routes differ in the execution steps. In addition, each execution route may include parallel machines assigned to one or more execution step. In our model, there is a dynamic list which describes the available execution routes at a given time period for each final product.

Component availability: In this issue we use a simplification of the original problem. It means that we do not focus on all components availability; instead we suppose all of the required material available in the needed quantity from the CST (Constrained Start Time) of the pallets on the machine. CST of the pallet specifies the earliest time when the first execution step of the pallet can start from the aspect of the component (material) availability.

Objectives: A scheduling objective is a measure to evaluate the quality of a certain schedule. In real-life situations, there are many (delivery capability, machine utilization rate, stock or WIP level, they are usually conflicting) objectives. For delivery capability, one can distinguish two types of objectives:

- due date related objectives and
- non due date related objectives.

For due date related objectives, we assume that there are jobs J_i $(i=1,...,N_J)$. Each job J_i has due date d_i and release date r_i. The due date represents the commitment of the company with a customer. The release date implies the availability of components from the beginning. We denote the finishing time of job J_i by C_i. The following definitions may be defined for each job:

Lateness of a job: $L_i = C_i - d_i$. (2)

Tardiness of a job: $T_i = \max(0, L_i)$ (3)

Earliness of a job: $E_i = \max(0, -L_i)$ (4)

With each of these functions F_i we get some possible objectives. So the most important objectives may be as follows:

Maximum: $\gamma = \max(F_i)$. (5)

Total: $\gamma = \sum_i F_i$ (6)

Average: $\gamma = \frac{\sum_i F_i}{n}$ (7)

Number of late jobs: $\gamma = |\{i \mid T_i > 0\}|$. (8)

Usually, not all of the jobs are equally important. Weights w_i can be assigned to each job representing the relative importance of the jobs. Some measures that take into account the different weight of the jobs are as follows:

Weighted maximum: $\gamma = \max(w_i F_i)$. (9)

Weighted total: $\gamma = \sum_i w_i F_i$ (10)

Weighted average: $\gamma = \frac{\sum_i w_i F_i}{n}$ (11)

The most common objective functions, which are non due date related, are as follows:

Makespan: $\gamma = \max(C_i)$. (12)

Total flow time: $\gamma = \sum_i C_i$ (13)

Weighted total flow time: $\gamma = \sum_i w_i C_i$ (14)

It is well known, that the optimal solution can be quite different if the chosen objective changes. Depending on the fixed objectives, each decision maker wants to minimize a given criterion. On one hand, the commercial manager is interested in satisfying orders by minimizing the lateness. On the other hand the production manager wishes to minimize the work in process by minimizing the maximum flow time.

3. Solution of the Scheduling Problem

In this section we outline our approach to solve problem described in section 2.2. We show the developed data model and the basic steps of our methods. Then we present a computer application of this solution.

3.1. Basic Data Structures

In our model, we use indexed arrays in order to accelerate the calculation. In these arrays, there are no full length identifiers and attributes of entities (i.e.: jobs, machines, routes and so on), instead there are indexes, which are non-negative integer values assigned to the entities, to point to the position of the target object in the base array. Therefore, in any of indexes of a given array, we can use any value of the same array or another array. In order to indicate an element of one-dimensional or two-dimensional array, we use the following formulations:

- ARRAY_NAME[ROW_INDEX]
- ARRAY_NAME[ROW_INDEX][COLUMN_INDEX]

If an array element is a data structure made up of fields, we use the dot operator to refer to a specified data field by using the field name:

- ARRAY_NAME[ROW_INDEX].FIELD_NAME
- ARRAY_NAME[ROW_INDEX][COLUMN_INDEX].FIELD_NAME

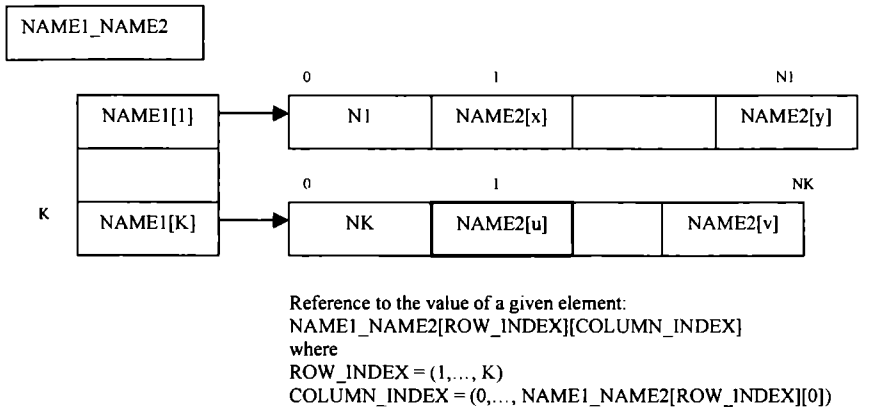


Figure 6. General Structure of two-dimensional arrays

General structure of two-dimensional arrays with variable number of row elements can be seen on Figure 6. It shows the association of two different type arrays (NAME1 and NAME2). The actual array can be specialized from the general structure in such a way that basic array corresponds to NAME1 and related array corresponds to NAME2.

3.2. Data Model

Using the above formulations, we can describe the entities of the scheduling model detailed in section 2.2 as follows:

In the system, there are different final products p (p $1,\ldots, N_P$) which may be produced. There is an order book for a given time period. It has production orders o (o $1,\ldots, N_O$). Each production order o includes the type of the final product $O[o].P$, the required quantity $O[o].Q$ and the defined end time $O[o].ET$ (due date).

At the shop floor, pallets can be moved. Each pallet consists of a pre-determined number $NP[p]$ ($p = 1,\ldots, N_P$) of the finished products p. Each production order o is identified to be consisting of a particular number of pallets. We schedule pallets, one pallet means one job. So we have jobs i ($i = 1,\ldots, N_J$) altogether. Each job has four attributes: $J[i].P$ means the final product p, $J[i].Q$ means the quantity of the products, $J[i].CST$ means the constrained start time and $J[i].CET$ means the constrained end time.

Each job i has to visit four technology steps $TS[t]$ ($t=1,\ldots, 4$) in the same sequence. The workshop contains ten possible machine groups mg ($mg = 1,\ldots, 10$) connected to each other in a given configuration (Figure 3.). Each machine group mg contains a pre-defined number of machines. There is a two-dimensional array named MG_M which describes the list of machine groups with the machines which belong to them. The structure of MG_M is inherited from the general structure shown on Figure 6. Machine group corresponds to NAME1 and machine corresponds to NAME2.

In a given machine group mg, each machine can process the same execution step which is one of the well defined execution steps es (es $1,\ldots, 10$). We have machines m ($m=1,\ldots, N_M$) altogether. Each machine m may have N_P different production rate $M\_PR[m][p]]$ ($m =1,\ldots, N_M$ *and* p $1,\ldots, N_P$). Similarly, each machine m may have N_P different setup time $M\_ST[m][p]$ ($m =1,\ldots, N_M$ *and* $p = 1,\ldots, N_P$), according to the definitions of the setup time and production rate in section 2.2.

Jobs can be moved on eight possible execution routes r ($r = 1,\ldots, 8$) (see Figure 5.). Each route r includes a sequence of machine groups. These assignments are defined in an array named R_MG, which is a specialization of the structure shown on Figure. 6. Machine group corresponds to NAME1 and machine group corresponds to NAME2.

In our model, utilizing what has gone before, we can determine an array P_R which describes the available execution routes in the actual time period for each final product p. The array P_R can also be specialized from the general structure in such

a way that final product corresponds to NAME1 and execution route corresponds to NAME2.

We suppose the shop floor has already been loaded, so the actual state of the system has to be known in order to calculate start time and end time of each job on each assigned machine. It means that the effect of the last confirmed schedule has to be available. These data can be obtained from array *M_ENGAGED* which shows the earliest time of each machine when the machine is available, *M_ENGAGED[m]* ($m = 1,..., N_M$).

Additional arrays have been defined to store the result of the scheduling. There is a special array named *J_A* which includes the route and machines assigned to jobs in the following way: *J_A[i][am]* ($i = 1,..., N_J$ *and* $am = 0,...,$ *R_MG[J_A[i][0]][0]*). Where:

- *i* means a job,
- *J_A[i][0]* means the assigned route,
- *R_MG[J_A[i][0]][0]* means the number of machines in the assigned route,
- *J_A[i][am]* ($am=1,...,$ *R_MG[J_A[i][0]][0]*) means the sequence of assigned machines.

There is an array named *MWLOAD* which shows the sequence of jobs on machines. This structure *MWLOAD[m][ai]* ($m= 1,...,N_M$ *and* ai $1,...,$*MWLOAD[m][0]*) is a specialization of the general structure (Figure 6.). In this case, machine corresponds to NAME1 and job corresponds to NAME2.

- *m* means a machine,
- *MWLOAD[m][0]* means the number of jobs on machine *m*,
- *MWLOAD[m][ai]* ($ai = 1,...,$ *MWLOAD[m][0]*) means the sequence of jobs to be processed on machine *m*.

Finally, we have defined an array named *MSTET* which stores the calculated times, which are as follows: *ST* start time, *SetT* setup time, *PT* process time and *ET* end time of the jobs *MWLOAD[m][ai]* ($ai \quad 1,...,$ *MWLOAD[m][0]*) on each machine *m* ($m = 1,...,N_M$).

3.4. Calculation Model

The calculation means numerical simulation of the production to calculate the time data of the operations. Inputs are jobs *i*, machines *m*, their assignments *J_A*, sequences of jobs on machines *MWLOAD*, abilities of machines *M_PR*, *M_ST* and availabilities of machines *M_ENGAGED*. Simulation of job *i* on an intermediate machine requires, among other things, the end time of job *i* on the previous machine and the shop floor environment has lots of junctions of the possible routes.

So we have to define the machine group sequence in which the calculation can be performed.

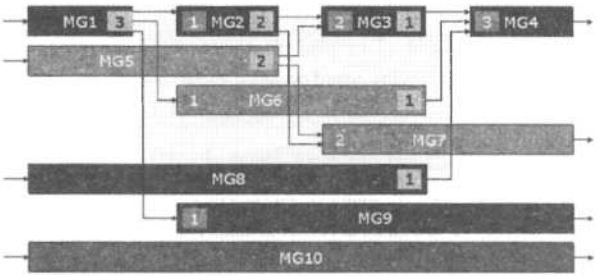


Figure 7. Precedence constraints between machine groups

Precedence constraints can be presented as a simple directed graph (Figure 7), in which the vertices are the machine groups and the edges show the required sequences of machine groups. For each machine group (MG), indegree and outdegree can be defined. Indegree means the number of inward directed edges from a given vertex and outdegree means the number of outward directed edges from a given vertex. At each machine group, the maximum of indegree and outdegree shows the number of possible routes which can include the given machine group. Possible sequence of the machine groups have been obtained by using the non-increasing indegree order of the machine groups.

$$D_{IN}(4) > D_{IN}(3) > D_{IN}(7) > D_{IN}(2) > D_{IN}(6) > D_{IN}(9) > D_{IN}(1) > D_{IN}(5) > D_{IN}(8) > D_{IN}(10). \quad (15)$$

The sequence is fixed in *PRI_MG* array which includes the priority of each machine group. The priorities are as follows:

Priority:	{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
Machine Group:	{4, 3, 7, 2, 6, 9, 1, 5, 8, 10}

The main steps of calculation method can be seen on the flow chart (Figure 8.). The method goes in non-increasing priority order of the machine groups, and it calculates time data *(ST, SetT, ProcT, ET)* of jobs on each machine which is in the machine group. Start time data of jobs play the main role in calculation. This value of a given job *i* on an assigned machine *m* is determined by the following values:

- the constraint start time of the job *(J[i].CST)*,
- end time of the job on the previous machine *(MSTET[prev_m][i_on_prev_m].ET)*,
- end time of the previous job on the machine *(MSTET[m][ai -1])*,
- setup time of the job on the machine *(M_ST[m][J[i].P])* and
- the availability of the machine *(M_ENGAGED[m])*.

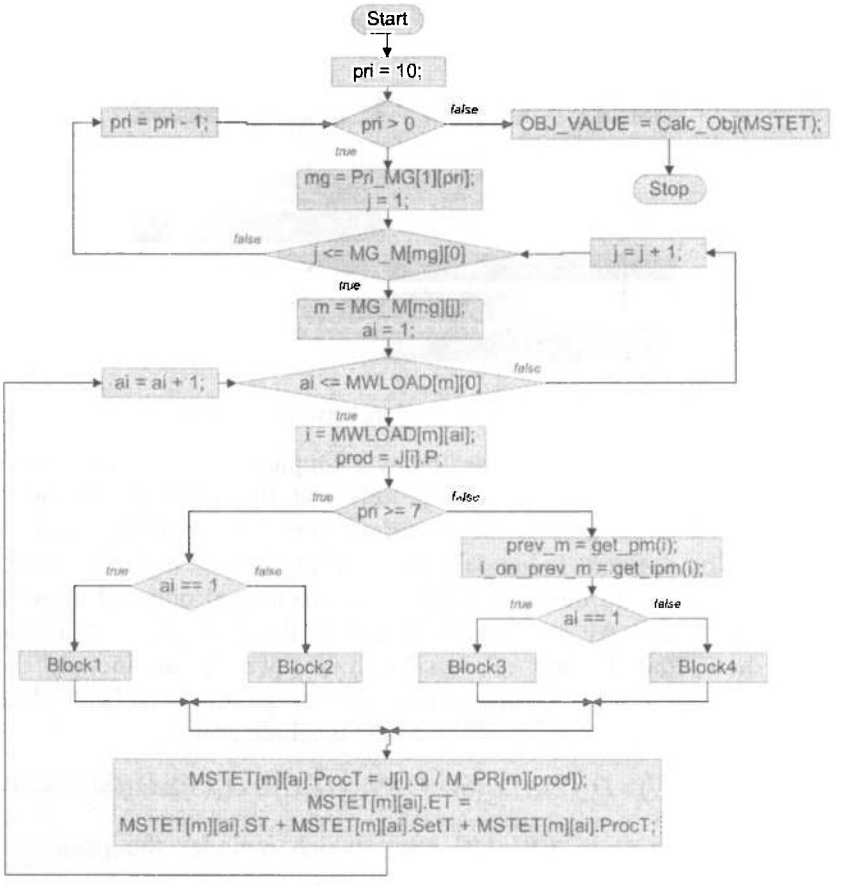


Figure 8. Flow Chart of Simulation

In point of view of start time and setup time four cases can be distinguished:

Block1: first job on first machine.

```
MSTET[m][ai].SetT = M_ST[m][prod];
MSTET[m][ai].ST = max( J[i].CST, M_ENGAGED[m] );
```

Block2: non first job on first machine.

```
if ( prod != J[MWLOAD[m][ai -1]].P ) MSTET[m][ai].SetT = M_ST[m][prod];
else   MSTET[m][ai].SetT = 0;
MSTET[m][ai].ST = max( MSTET[m][ai -1].ET, J[i].CST );
```

Block3: first job on non first machine.

```
MSTET[m][ai].SetT = M_ST[m][prod];
MSTET[m][ai].ST = max( MSTET[prev_m][i_on_prev_m].ET - MSTET[m][ai].SetT,
                       M_ENGAGED[m] );
```

Block4: non first job on non first machine.

```
if ( prod != J[MWLOAD[m][ai -1]].P )  MSTET[m][ai].SetT = M_ST[m][prod];
else   MSTET[m][ai].SetT = 0;
if ( MSTET[prev_m][i_on_prev_m].ET <= MSTET[m][ai -1].ET)  MSTET[m][ai].ST = MSTET[m][ai -1].ET;
else  MSTET[m][ai].ST = max( MSTET[prev_m][i_on_prev_m].ET - MSTET[m][ai].SetT,
                             MSTET[m][ai -1].ET);
```

Typical examles of working Block 4 can be seen on the Figure 9.

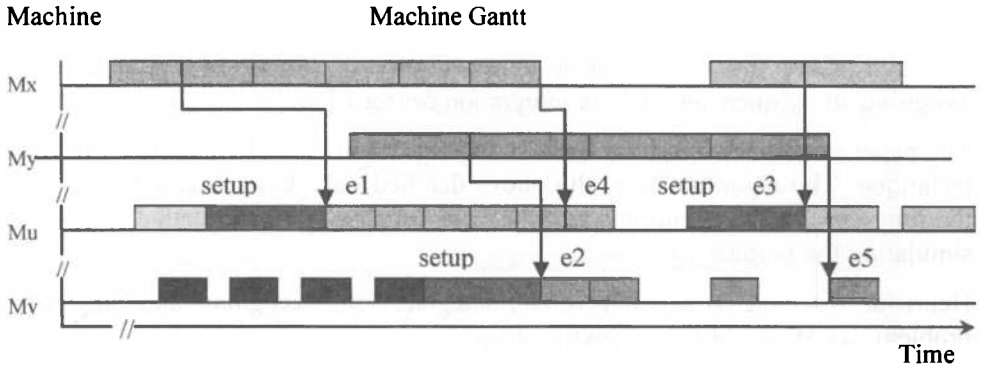


Figure 9. Working Block4

The outputs of the time calculation are *MSTET* array, which includes fixed times, and *OBJ_VALUE*, which stores the evaluated value of the chosen objective function.

3.5. Heuristic Algorithms

Our scheduling problem is difficult to solve because of its combinatorial nature. In order to define a schedule for the production of each job, it is necessary for each job i ($i = 1,...,N_J$):

1. to assign to one of the possible routes: $J\_A[i][0] = P\_R[J[i]\ P][ar]$ ($ar \in \{1,..., P\_R[J[i]\ P][0]\}$),
2. to assign to one of the possible machines at each possible machine group according to selected route: $J\_A[i][amg] = MG\_M[amg][am]$ ($amg = 1,..., R\_MG[J\_A[i][0]][0]$ and $am \in \{1, ..., MG\_M[amg][0]\}$),
3. to fix its position in the queue of each selected machine: $MWLOAD[J\_A[i][amg]][ai]$ ($amg = 1,..., J\_A[i][0]$ and $ai = 1,..., MWLOAD[J\_A[i][amg]][0]$),
4. to fix its starting time on each selected machine. $MSTET[J\_A[i][amg]][ai]$ ($amg = 1,..., J\_A[i][0]$ and $ai = 1,..., MWLOAD[J\_A[i][amg]][0]$).

Different heuristic procedures have been developed to solve the problem. These procedures are integrated into the scheduling engine (SE). At present, SE includes two kinds of classes of heuristic algorithms, which are as follows:

- Constructive algorithms,
- Iterative improvement algorithms.

The basic approach of our heuristic algorithms consists of tree steps:

1. Assigning: SE creates the *J_A*.
2. Sequencing: SE creates the *MWLOAD*.
3. Simulation: SE calculates the *MSTET*.

The algorithms differ from each others in the decision making in issues of assigning and sequencing and the integration degree of steps.

The paper outlines one of these scheduling algorithms based on heuristic insertion technique. The algorithm uses the above detailed calculation method to evaluate the time data of the operations and the value of the chosen objective function by simulating the production.

Heuristic Inserting Algorithm (HIA) integrates the assigning and sequencing problem. It consists of the following steps:

Step 1: Order the sequence of production orders according to the Earliest Due Date (EDD) rule, then in this sequence let the secondary consideration be the number of jobs of the production orders. Create the list of the jobs.

Step 2: Get the next job from the list.

Step 3: Enumerate all possible routes and machines for the job. Assign the job to each possible route in increasing order of the number of machine groups. Assign the jobs to each possible combination of available machines at each machine group of the route in decreasing order of production rates.

Step 4: On each first machine, insert the job into each position which is between two different batches. (On a machine, a batch means a sequence of jobs which is related to the same production order.)

Step 5: On the further machines of the route, the jobs flow through the system in order of arrival (First In First Out, FIFO).

Step 6: In each case use the calculation method.

Step 7: After simulation select the best solution according to the chosen objective function.

Step 8: Update data. If the list is empty, the algorithm is finished else go to Step 2.

3.6. Computer Application

We developed a computer application which consists of a problem generator, production simulator, scheduling engine and a database system. The main goal of this framework is that it can be an extremely useful tool supporting future studies of alternative scheduling algorithms.

The application uses sample data sets created by problem generator. The generator produces random problem instances with sizes and characteristics specified by user and then it writes them into the database. The generated data are well-defined random values, but the user can directly change certain data.

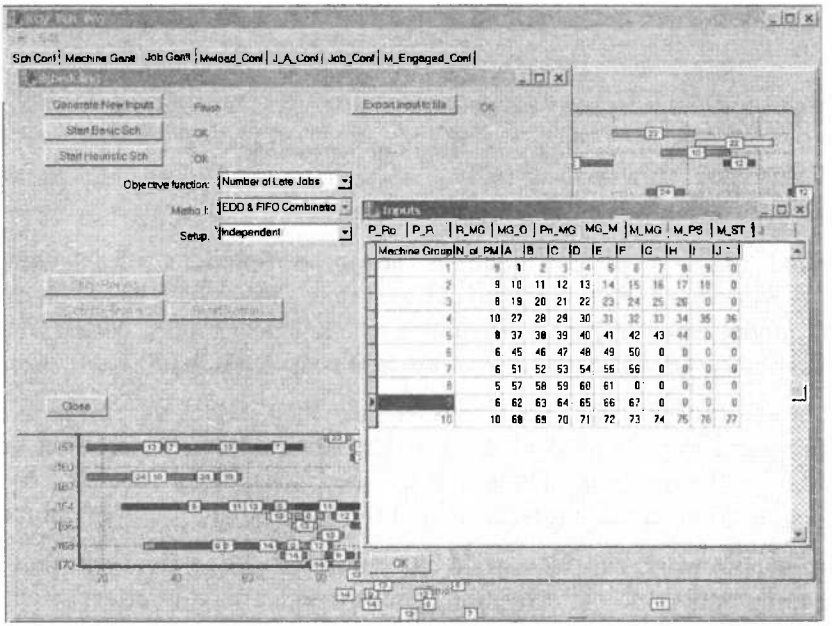


Figure 10. User Interface of Scheduler Engine

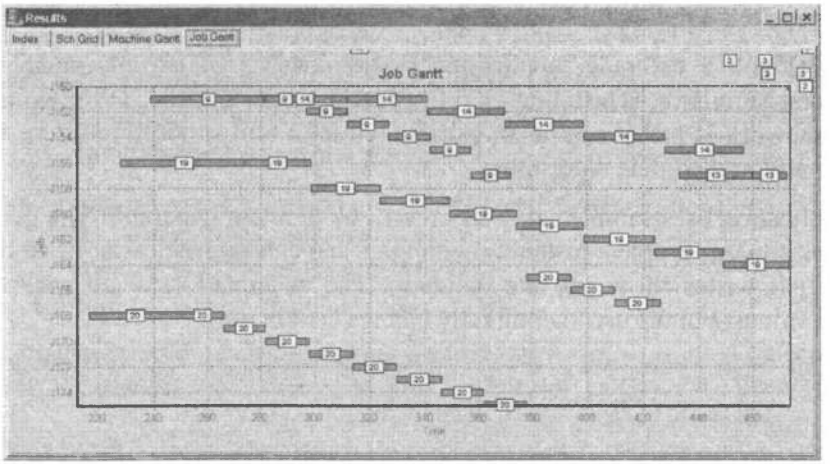


Figure 11. Result of Scheduling

4. Conclusions

Customized mass production requires advanced functions of Manufacturing Execution Systems (MES). The conventional flow shop model has to be extended to a new model which supports alternative technological routes, parallel machines and where setup and job characteristics are also considered.

In this paper, some possible extensions of flow shop model for customized mass production have been described. A new scheduling approach based on heuristic methods to solve extended flexible flow shop scheduling problems has been introduced. A computer program developed for this problem has been outlined.

Future research work will be carried out on investigating heuristic procedures, which can be applied to our scheduling problem and studying effect of change in the machine environment.

Acknowledgements

The research and development summarized in this paper was partially supported by the Hungarian Academy of Sciences (HAS) within the framework of Production Information Engineering Research Team (PIERT) established at the Department of Information Engineering of the University of Miskolc (Grant No. MTA-TKI 06108). The results are also connected with the NODT project entitled "VITAL" (National Office for Development and Technology founded by the Hungarian Government, Grant No.: 2/010/2004, project leader: László Monostori DSc). The authors would like to express their thanks for the financial support.

REFERENCES

[1] BRUCKER, P.: *Scheduling Algorithms*, Springer-Verlag, Berlin, 1998.

[2] KIS, T., ERDŐS, G., MÁRKUS, A., VÁNCZA, J.: *A Project-Oriented Decision Support System for Production Planning in Make-to-Order Manufacturing*, ERCIM News, No. 58, pp. 66-67, 2004.

[3] KOVÁCS, A., VÁNCZA, J.: *Completable Partial Solutions in Constraint Programming and Constraint-based Scheduling*, Principles and Practice of Constraint Programming, Springer LNCS 3258, pp. 332-346, 2004.

[4] KOVÁCS, L.: *The Methodology of Data Base Design and Management*, ComputerBooks, Budapest, 460 p. (In Hungarian), 2004.

[5] KULCSÁR, GY., HORNYÁK, O., ERDÉLYI, F.: *Shop Floor Decision Supporting and MES Functions in Customized Mass Production,* Conference on Manufacturing Systems Development - Industry Expectations, Wroclaw, Poland, pp. 138 – 152, 2005.

[6] KURNAZ, A., COHN, Y., KOREN, Y.: *A Framework for Evaluating Production Policies to Improve Customer Responsiveness*, CIRP Annals, Volume 54/1, pp. 401-406, 2005.

[7] MCKAY, K., N. WIERS, V., C.: *Unifying the Theory and Practice of Production Scheduling*, Journal of Manufacturing Systems, Vol. 18, No. 4, pp. 241-248, 1999.

[8] ZWEBEN, M., FOX, M.,S.: *Intelligent Scheduling*, Morgan Kaufmann Publishing, San Francisco, 1994.

Production Systems and Information Engineering
Volume 3 (2006), pp. 139-155

NEW APPROACHES IN SOLVING MACHINE-PART GROUPING PROBLEMS

TIBOR TÓTH
University of Miskolc, Hungary
Department of Information Engineering
toth@ait.iit.uni-miskolc.hu

ATTILA KÖREI
University of Miskolc, Hungary
Department of Information Engineering
matka@gold.uni-miskolc.hu

[Received March 2006 and accepted May 2006]

Abstract. Machine-part grouping problems arise in a production plant when forming a new production system or reorganizing an existing one. Application of Group Technology principles can help in finding the optimal layout and manufacturing system. In order to satisfy the basic principle *similar things should be done similarly* parts are assigned to different families based on their processing requirements and machines are separated into groups to process specific part families. The machine-part cell formation problem is a widely researched area and numerous algorithms have been developed to solve it. This paper provides a survey of the latest results related to clustering methods, artificial intelligence approaches and some mathematical techniques. In addition an abstract algebraic method based on the theory of concept lattices is also outlined.

Keywords: group technology, cell formation problem, similarity coefficients, genetic algorithms, mathematical programming, concept lattices.

1. Introduction

The philosophy of group technology (GT) plays an important role in the design of manufacturing cells. The basic concept of GT is to identify and exploit the similarity between parts, machines and manufacturing processes. GT is a disciplined approach to grouping items by their attributes. Parts having similar processing requirements are arranged into part families, and the machines processing them are grouped into cells. The advantages of applying group technology principles are reduced setup time, queuing time and material handling time, shorter lead times, reduced tool requirements and improved product quality.

The adoption of GT concepts yields an efficient production system and a significant reduction can be expected in overall manufacturing costs.

Cellular manufacturing (CM) can be regarded as one of the major applications of group technology. CM requires the identification of groups of similar parts and the associated machines which form cells. The determination of part families and machine cells is called the cell formation (CF) problem.

During the last three decades the CF problem has been widely researched and numerous methods have been developed for solving it. These methods are classified by several review papers: we follow the taxonomy proposed by Shafer [16]. The main aspect of his classification is the methodology the cell formation procedures are based on. From studying these methodologies, cell formation techniques can be arranged into the following six groups:

manual methods,

classification and coding approaches,

algorithms for sorting machine component matrix,

statistical cluster analysis,

artificial intelligence methods,

mathematical techniques.

In this paper we give a short survey of the latest results in solving CF problems. Some of these methods can be regarded as new approaches and others are improved versions of an older technique. We focus on the last three group of Shafer's taxonomy, and in the next three sections the main aspects and the latest results from these fields are reviewed. In Section 5 a new mathematical approach is described. This method is based on Formal Concept Analysis which is a prospering field of applied lattice theory.

2. Statistical Cluster Analysis

Application of a clustering technique requires the development of a measure quantifying the similarity or dissimilarity between two objects (parts or machines). Using the appropriate similarity measure the necessary production data can be incorporated in the early stages of the machine-part grouping procedure. Some of these factors are operation sequences, within-cell machine sequences, processing requirements of parts, production volumes, unit operation time, alternative process routings, etc. A number of similarity coefficients have been developed for taking into consideration the different production factors and goals during the CF process. The calculation of these coefficients combined with a clustering algorithm is called

a similarity coefficient based clustering method (SCM) and generally it consists of the following three steps:

a) Form the initial machine-part incidence matrix $[a_{ij}]$, where an entry "1" ("0") indicates that machine i is used (not used) to process part j. (Here i is the machine index (i=1,2,...,m), and j is the part index (j=1,2,...,n).
b) Choose a similarity coefficient and calculate for each pair of parts or machines the corresponding values. These numbers are stored in a similarity matrix whose elements represent the sameness between two parts or machines.
c) Based on the values of the similarity matrix a clustering algorithm expands the hierarchy of similarities among all pairs of parts (machines). Using the obtained tree or dendogram the part families (machine groups) can be identified.

Before applying some clustering technique we have to select a similarity coefficient which indicates the degree of similarity between object pairs. The most frequently employed coefficient is the Jaccard similarity coefficient which is defined for parts i and k in the following way:

$$J_{ik} = \frac{N_{ik}}{N_i + N_k - N_{ik}},$$

where N_{ik} is the number of machines that parts i and k have in common in their production, and N_i, N_k mean the number of machines processing part i and k respectively. Besides the Jaccard similarity coefficient numerous other similarity coefficients have been proposed in the literature. One of the most comprehensive reviews of the topic is given by Yin and Yasuda [21], who developed a new taxonomy to classify the various similarity and dissimilarity coefficients. Besides this classification they attempted to explain why similarity coefficient based methods are more flexible than other approaches. First, similarity coefficient methods apply cluster analysis to CF procedures. Clustering techniques are fundamental methods in group technology, being a basic approach for estimating similarities. Fitting well to the main idea of GT, a similarity coefficient based method can be a more effective way to solve CF problems. Another reason for the preference of SCM methods is that they are more suitable for certain principles which are generally accepted in solving complex problems:

(i) decomposition of the problem into small conquerable problems and
(ii) decomposition of the solution into small tractable stages.

These principles are satisfied by a clustering method because it consists of three steps as mentioned in the beginning of this section. These steps are independent of

each other, which makes it possible to reselect the similarity coefficients when extending the problem to incorporate additional production factors.

It is worth mentioning that the use of similarity coefficients is often combined with other techniques, mostly artificial intelligence methods for solving machine-part grouping problems. For example Tóth and Molnár [17] have developed two algorithms for forming part-groups and inserting new parts in existing groups. Starting from the similarity matrix of the parts they used fuzzy classification to solve the problems. In [10], Jeon and Leep proposed a new similarity coefficient, which considers alternative routes, and based on these coefficients the part families are identified by using genetic algorithm. Adenso-Díaz et al. in [1] suggested weighted similarity coefficients and Tabu search for determining machine cells.

3. Artificial Intelligence Methods

With the increasing speed and capacity of today's computers, researchers frequently apply artificial intelligence methods to solve the CF problem. The most commonly used techniques are pattern recognition, fuzzy reasoning, neural networks and genetic algorithms. Another reason for successful application of these methods is the NP-completeness of the cell formation problem, i.e. there is no algorithm of polynomial complexity to solve it. This means that methods using heuristics can be more suitable to solve the CF problem than other exact approaches.

In this section we focus on the methods using genetic algorithms. The basic ideas of these methods are discussed and some recent results are referred to.

Genetic algorithm (GA) is a heuristic search technique which was introduced by Holland in 1975 [9]. It is based on an analogy to natural selection and Darwin's evolution concepts. First a chromosome structure is to be defined to represent the solutions of the optimisation problem. After generating an initial solution population (which is done mostly randomly) some members of the population are selected to be parents to produce offspring. The selection is based on the so-called fitness function: the higher an individual's fitness value the more likely that individual is to be selected, satisfying the principle of survival of the fittest. The less fit members are replaced by new ones, who are produced by the parents using genetic operators: crossover and mutation. Crossover combines the best parts of parent chromosomes in order to exploit promising areas of the search place. Mutation is a small random modification of the chromosome that increases the diversity of the population and explores new regions of the search place. The process is repeated until a termination criterion is reached.

The main questions before implementing genetic algorithm are the following:

how to encode the structure of the chromosomes for representing solutions,
how to generate the initial population,
how to choose a good fitness function,
how to define the genetic operators,
how to choose the parameters according to the crossover and mutation operator, the population size, rate of individuals to be selected, and maximum number of iterations.

There are several possibilities for encoding chromosomes. Most of the studies use an integer codification to represent solutions. For example in a machine cell formation problem the following chain

m_1	m_2	m_3	m_4	m_5
1	2	1	2	1

represents a solution where two cells are formed, the first contains the machines m_1, m_3, m_5 while machines m_2, m_4 belong to the second cell. The main drawback of this representation that it induces redundancy since the cell indices can be permuted, so the chromosome

m_1	m_2	m_3	m_4	m_5
2	1	2	1	2

represents the same solution as the previous one. The redundancy grows very quickly with the number of cells, making the search for good solutions even more difficult. Boulif and Atif [2] avoided this difficulty by choosing binary coding for the chromosomes This method is based on the graph theory model of the CF problem, where the nodes represent machines and an edge of the graph indicates whether there is inter-machine traffic between the two vertices of this edge. An edge is encoded by 0 if the traffic between its two vertices is intracellular (expressing that the machines corresponding to the two vertices are in the same cell). The intercellular edges are denoted by 1. Using this reduced alphabet the GA algorithm can be implemented in an efficient way, and the search for good solutions becomes easier.

Although genetic algorithm is one the most popular methods for solving CF problems, we have to mention some disadvantages of this technique. The main problem with the standard GA approach is its weakness, which means that it does not incorporate problem-specific knowledge. The other drawback is related to the standard encoding scheme (both with integer and binary coding); which can cause unexpected effects when applying GA operators. To overcome these problems De Lit, Falkenauer and Delchambre propose a grouping genetic algorithm (GGA) to

solve the cell formation problem [6]. GGAs are a special class of genetic algorithms introduced by Falkenauer in 1992 [7] modifying the standard GAs to better match the structure of grouping problems. There are two main differences between GGA and the classic genetic algorithm: GGA uses a group oriented encoding scheme and special genetic operators suitable for the chromosomes. For solving a machine-part cell formation problem the GGA applies the following chromosome representation:

$$p_1 p_2 p_3 ... p_P \mid m_1 m_2 m_3 ... m_M \mid g_1 g_2 g_3 ... g_G$$

where p_i is the group to which part i is assigned, m_j is the group to which machine j is assigned and g_k is an existing group number. P denotes the number of parts, M denotes the number of machines in the problem and G is the number of groups in the solution. The main characteristic of the genetic operators of the GGA is that they work with the group part of the chromosomes rather than items [6]. In [3], Brown and Sumichrast compared the performance of a GGA against the performance of a standard GA approach in three different grouping problems. Their second problem was the machine-part cell formation problem and they used grouping efficacy to compare solution quality. Grouping efficacy can be computed by the following formula:

$$GE = 1 - \frac{e_0 + e_v}{e + e_v},$$

where e is the total number of ones in the original machine-part incidence matrix, e_v is the number of voids and e_0 is the number of exceptional elements in the solution. (Voids occur when a part does not require one of the machines in its group and exceptional elements arise when a part requires a machine from another group.) Using this performance measure the authors tested both solution techniques and they concluded that GGA outperformed the standard genetic algorithm in solving the machine-part cell formation problem, and is indeed an efficient technique even for large-sized problems.

4. Mathemathical Techniques

Mathematical techniques include methods related to graph theory, combinatorial analysis and mathematical programming. One of the widely researched exact methods is the integer programming approach. In group technology Kusiak was the first who adopted a linear programming method for part-family formation [12]. The suggested p-median model uses n^2 decision variables as follows:

$$y_{ij}= \begin{cases} 1, \text{ if part } i \text{ belongs to part family } j \\ 0, \text{ otherwise} \end{cases}$$

$i=1,2,...,n$, where n is the number of parts. Denote the desired number of part-families by p and compute the similarity coefficients s_{ij} between parts i and j in the following way:

$$s_{ij}^{(p)} = \sum_{k=1}^{m} \delta(a_{ki}, a_{kj}) \qquad i \neq j, \qquad i, j=1,2,...,n.$$

$$s_{ii}^{(p)} = 0 \qquad\qquad i=1,2,...,n.$$

Here a_{ki} is the element of the kth row and ith column of the machine-part incidence matrix and δ is the Kronecker function: $\delta(a_{ki}, a_{kj})=1$ if $a_{ki} = a_{kj}$, and 0 otherwise.

The objective is to maximize the sum

$$\sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}^{(p)} x_{ij}$$

satisfying the following conditions:

(1) $\sum_{j=1}^{n} y_{ij} = 1$ for all $i = 1,...,n$,

(2) $\sum_{j=1}^{n} y_{jj} = p$,

(3) $y_{ij} \leq y_{jj}$ for all $i = 1,...,n$, $j = 1,...,n$

Condition (1) ensures that each part belongs to exactly one part family, and (2) specifies the required number of part families. Constraint (3) ensures that part i is grouped into the part family represented by j, if this family exists.

The part family formation method described here can be readily adapted to form machine cells first. In this case we have to compute similarity coefficients between machines, for example we can use the definition suggested by Wei and Kern [19]:

$$s_{ij}^{(m)} = \sum_{k=1}^{n} \Gamma(a_{ik}, a_{jk}), \text{ where } \Gamma(a_{ik}, a_{jk}) = \begin{cases} n\text{-}1 & \text{if } a_{ik} = a_{jk} = 1, \\ 1 & \text{if } a_{ik} = a_{jk} = 0, \\ 0 & \text{if } a_{ik} \neq a_{jk}. \end{cases}$$

These coefficients are applied by Won and Lee [20], who suggested two modified versions of the *p*-median model. They started from an extended *p*-median model, where the decision variables are the following:

$$x_{ij} = \begin{cases} 1, \text{ if machine } i \text{ is clustered into cell } j \\ 0 \text{ otherwise,} \end{cases}$$

and the objective function is $\sum_{i=1}^{m}\sum_{j=1}^{m} s_{ij}^{(m)} x_{ij} \longrightarrow \max,$

subject to

$$(4)\ \sum_{i=1}^{m} x_{ij} = 1 \text{ for all } j = 1,...,m,$$

$$(5)\ \sum_{j=1}^{m} x_{jj} = p,$$

$$(6)\ \sum_{j=1}^{m} x_{ij} \geq L x_{jj} \text{ for all } i = 1,...,m,$$

$$(7)\ \sum_{j=1}^{m} x_{ij} \leq U x_{jj} \text{ for all } i = 1,...,m.$$

Conditions (4) and (5), similarly to the original formulation, ensure that each machine is assigned exactly to one cell and the number of machine cells is prescribed. In constraints (6) and (7) the number of machines grouped in to cell *i* is limited: at least *L* machines should be assigned to cell *i* only if cell *i* is formed, and *U* is the maximum number of machines allowed in each cell.

Since in most practical problems parts outnumber machines ($n>m$), the extended formulation with its m^2 binary variables leads to a smaller linear integer

programming problem compared to Kusiak's model. In spite of this reduction the extended model needs further improvement because of the difficulties in its implementation. The problem is how to choose the optimal median number p. In order to avoid this problem the entire model is tested for p=2,...,m, and then the best solution is selected. This type of implementation can hardly be carried out, because running the entire model for varying values of p causes too much computation time even on a medium sized CF problem. To overcome these difficulties Won and Lee introduced a special set of machines that have a high probability of serving as medians or seed machines for clustering. They developed an algorithm for determining the candidate set of median machines and after this with the modification of constraints, speedier implementation was achieved by excluding a large amount of binary variables.

In addition Won and Lee proposed another modified formulation of the model which makes further reduction of the number of variables possible and they presented remarkable test results, applying their method on large-sized CF problems. (The applications of integer programming approaches in CF problems containing 40 or more machines are rarely reported in the literature because these methods have so far required enormous computation time.)

Another opportunity to modify the original p-median model is the linear assignment model proposed by Wang [18]. The basic idea of the group formation algorithm is the selection of the p most dissimilar parts or machines. These items probably will be assigned to different groups since they have dissimilar design or manufacturing features. These group representatives can be determined recursively by using the similarity coefficients. With the knowledge of the group representatives the model can be formulated both for part family and machine cell formation and it contains far fewer decision variables compared with the p-median model (pn instead of n^2 or pm instead of m^2). The reduction in number of variables encouraged Wang to test his method on medium sized CF problems (the maximal number of machines was 40) and it proved an efficient method in a comparative study [18].

5. Solving Cell Formation Problems with Concept Lattices

In this section a new mathematical approach for solving machine-part cell formation problem is presented. The method has been developed at the University of Miskolc, in a collaborative project between the Department of Information Engineering and the Institute of Mathematics. This abstract algebraic method is related to Formal Concept Analysis, which can be regarded as a field of applied lattice theory. For the sake of completeness the basic elements of Formal Concept

Analysis are briefly introduced: for details see the fundamental work of Ganter and Wille [8].

The theory of Formal Concept Analysis is based on the theory of complete lattices. An ordered set $L=(L,\leq)$ is called a *lattice* if for any two elements x and y the supremum $x\vee y$ and the infimum $x\wedge y$ always exist. L is called a *complete lattice* if the supremum $\vee S$ and the infimum $\wedge S$ exist for any subset S of L. Every complete lattice has a largest (unit) element and a smallest (zero) element. The elements a and b are *neighbours* if $a<b$ and there is no element c fulfilling $a<c<b$ This relation is denoted by $a\prec b$ The neighbours of the zero elements are called the *atoms* of the lattice. A complete lattice in which every element is the supremum of atoms is called an *atomistic lattice*.

One of the basic notions of formal concept analysis is the term of formal context. A *formal context* $K=(G,M,I)$ consists of two sets G and M and a relation I between them. The elements of G are called the objects and the elements of M are called the attributes of the context (these traditional notations come from the German words *Gegenstand* and *Merkmal)*. The binary relation $I\subseteq G\times M$ is defined as follows: $(g,m)\in I$ if and only if the object $g\in G$ has the attribute $m\in M$. A small context can be represented by a cross table.

Observe that a machine-part incidence matrix can be considered as an analogous structure to the formal context. A given machine-part grouping problem corresponds to the formal context (G,M,I) where G is the set of machines, M contains the parts and I is determined by the incidence matrix with the following relation: $(g,m)\in I$ if the part m visits the machine g (Table 1).

For a set $A\subseteq G$ we define the set of the common attributes for the objects belonging to A: $A'=\{m\in M\mid (g,m)\in I,\ \forall g\in A\}$. Correspondingly for a set $B\subseteq M$ we define the set of the objects possessing all the attributes in B: $B'=\{g\in G\mid (g,m)\in I,\ \forall m\in B\}$.

Let $A\subseteq G$ be a set of objects and $B\subseteq M$ be a set of attributes. The pair $C=(A,B)$ is called a *formal concept* of the context (G,M,I) if the conditions $A'=B$ and $B'=A$ hold true. In this case A is called the *extent* and B is called the *intent* of the concept C with the notation $A=Ext(C)$ and $B=Int(C)$. For example ({1,4,5,10}, {a,b,c,o,p}) is a concept of the context K represented in Table 1.

Table 1. A machine-part incidence matrix with 11 machines and 22 parts. It corresponds to the formal context $K=(G,M,I)$, where G contains 11 objects, M contains 22 attributes and I is determined by the 1-s in the table

K		Parts																				
		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
Machines	1	1	1	1								1				1	1				1	1	1
	2					1			1			1	1							1			
	3					1							1	1						1			
	4	1	1	1								1				1	1				1	1	1
	5	1	1	1				1								1	1				1	1	1
	6								1				1							1			
	7				1	1				1					1			1	1				
	8					1				1	1		1	1	1			1	1	1			
	9				1					1					1			1	1				
	10	1	1	1				1	1			1	1			1	1		1				
	11				1					1	1							1	1				

Denote by $L(G,M,I)$ the set of all concepts of the context (G,M,I) and introduce a partial order between the elements of it: $(A_1,B_1)\leq(A_2,B_2)$ if and only if $A_1 \subseteq A_2$ (then $B_2 \subseteq B_1$ also holds true). It can be verified that the lattice $(L(G,M,I),\leq)$ is a complete lattice and it is called the *concept lattice* of the context (G,M,I). The lattice $L(G,M,I)$ can be represented by a Hasse diagram, using the notion of neighbourhood. The elements of the lattice are depicted by circles in the plane. If x and y are concepts with $x \prec y$, the circle corresponding to x and the circle representing y are joined by a line segment. >From such a diagram the order relation can be read off as follows: $x < y$ if and only if the circle representing y can be reached by an ascending path from the circle representing x. The diagram of the concept lattice originating from the context K is presented in Fig. 1. The concept $c_1 = \{G,\varnothing\}$ represents the unit element of the concept lattice, its extent is equal to the full set of the objects, while the zero element is $c_0 = \{\varnothing,M\}$.

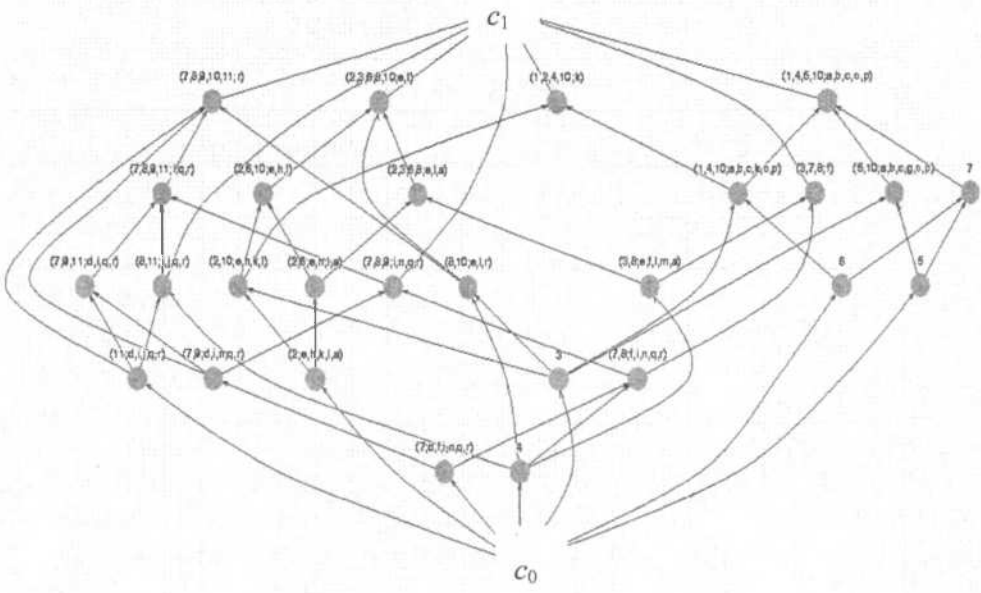


Figure 1. The concept lattice of the context K

A *classification system* S of the concept lattice $L(G,M,I)$ can be defined as a system of concepts where the extents of the concepts give a partition of the object set G. Formulating this definition we obtain

$$S = \{ (A_i, B_i) \mid i \in I \}$$

where $(A_i, B_i) \in L(G,M,I)$, $G = \bigcup A_i$, $A_i \cap A_j = 0$ if $i \neq j$.

For example the concepts ({2,3,6,8},{e,l,s}), ({1,4,5,10},{a,b,c,o,p}), ({7,9,11}, {d,i,q,r}) form a classification system of the concept lattice in Fig. 1.

The set of all classification systems of $L(G,M,I)$ is denoted by $Cls(L)$. We define an ordering relation between classification systems: $S_1 < S_2$ if the partition induced by S_1 refines the partition induced by S_2. It can be proved that the pair $(Cls(L), <)$ is a complete lattice which is called the *classification lattice* of $L(G,M,I)$. The 0-element of $Cls(L)$ is denoted by S_0, that is $S_0 = \wedge\{S : S \in Cls(L)\}$.

If the initial context consists of too many objects and attributes we obtain a large sized concept lattice and it is not simple to select those concepts from it that form a classification system. In this case using the results of Radeleczki [14] we can

determine the classification systems by means of a box lattice, which is a simpler structure than the original concept lattice.

The 0-element of $L(G,M,I)$ and any elements of a classification system are called *box elements* of $L(G,M,I)$. The set of the box elements are denoted briefly by $B(L)$. Restricting the partial order given on $L(G,M,I)$ to the set $B(L)$ we obtain a lattice again, the *box lattice* of $L(G,M,I)$. It can be verified that $(B(L),\leq)$ is an atomistic complete lattice, and its atoms are the elements of the finest classification system S_0. The box lattice originating from context K is shown in Fig 2.

The determination of the classification systems is carried out by means of box lattices in the following steps (more detailed discussion can be found in [11]).

a) Having started from the given context we determine the atoms of the box lattice.
b) We generate the further box elements using the observation that in a box lattice every element is a supremum of atoms.
c) Choosing the maximal disjoint systems of the box lattice we get the required classification systems (a maximal disjoint system cannot be extended by further box elements saving the property that the intersection of any two elements is the zero element).

Note that the determination of the classification systems gives a basis for formation of machine cells. The elements of a machine cell are usually characterized by the common parts processed by them. This means, using the notion of formal concept analysis, that $G_i''=G_i$ for every $i\in I$, where the sets G_i give a partition of the machine set G. In this case every block G_i is the extent of some concept of the concept lattice $L(G,M,I)$, because the pairs (G_i, G_i') satisfy the equations defining a concept. In other words the formation of machine cells can be solved by finding the suitable classification systems of $L(G,M,I)$.

Following the steps described above we have determined all of the classification systems of the concept lattice represented in Fig. 1. After applying step a) the following 7 atoms are obtained (the concepts are identified by their extents):

$a_1 = \{1,4\}$, $a_2 = \{5\}$, $a_3 = \{2,6\}$, $a_4 = \{3,8\}$, $a_5 = \{7,9\}$, $a_6 = \{10\}$, $a_7 = \{11\}$

The further box elements were determined by step b):

$d_1 = \{1,4,5,10\}$, $d_2 = \{7,9,11\}$, $d_3 = \{2,3,6,8,10\}$, $d_4 = \{5,10\}$, $d_5 = \{2,6,10\}$,

$d_6 = \{1,4,10\}$, $d_7 = \{2,3,6,8\}$, $d_8 = \{1,4,5\}$.

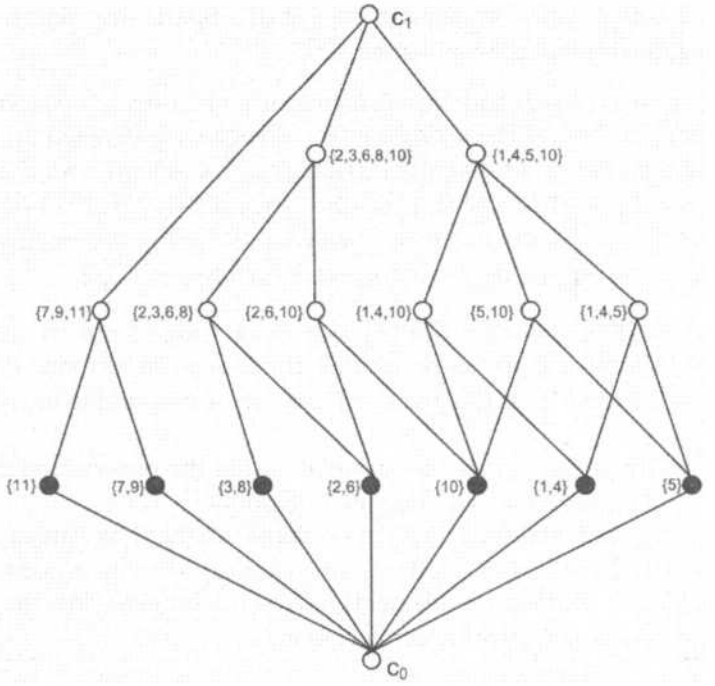


Figure 2. The box lattice of the context *K*. Atoms are represented by black circles, the further box elements are white. This lattice is much simpler than the concept lattice

Using step c) all of the classification systems were generated. In order to avoid uninteresting partitions two restrictions are assumed: every machine cell has to consist of at least three elements and the maximum number of bottleneck machines (i.e. machines processing parts from more than one family) is one. With these constraints four classification systems are formed (Table 2).

Table 2. Machine cells with the box lattice method

Machine cells			Bottleneck machine
{7,9,11}	{2,3,6,8}	{1,4,5,10}	
{7,9,11}	{2,3,6,8,10}	{1,4,5}	
{7,9,11}	{2,3,6,8}	{1,4,5}	{10}
{7,9,11}	{2,3,6,8}	{1,4,10 }	{5}

The context in Table 1 is borrowed from Cheng's study [5], where 12 algorithms for forming machine groups were compared. At the end of the examination there were three methods left yielding satisfactory results. These algorithms are the following: average linkage clustering (ALC) based on similarity coefficients [15], ZODIAC algorithm developed by Chandrasekharan and Rajagopalan [4] and a branch and bound algorithm (B&B) by Kusiak, Boe and Cheng [13]. The resulting machine cells are listed in Table 3.

Table 3. The results of three machine cell formation methods from Cheng's comparative study

Method	Machine cells			Bottleneck machines	
ALC	{7,9,11}	{2,3,6,8}	{1,4,5,10		
ZODIAC	{7,8,9,11	{2,3,6}.	{1,4,5,10		
B&B	{7,9,11}	{2,3,6}	{1,4,5}	{8}	{10}

It can be seen that these results are very similar to the decompositions obtained by the box lattice method, and our approach offers several opportunities to form the machine cells. The next step in improving the box lattice method is to solve the problem of constraints, namely how to build reasonable restrictions into the algorithm in order not to determine uninteresting classification systems.

Acknowledgements

This research was supported by the Hungarian National Foundation for Scientific Research (Grant No. T046913).

REFERENCES

[1] ADENSO-DÍAZ, B., LOZANO, S. EGUÍA, I.: *Part-machine grouping using weighted similarity coefficients.* Computers & Industrial Engineering, 48, pp. 553-570, 2005.

[2] BOULIF, M., ATIF, K.: *A new branch-&-bound-enhanced algorithm for the manufacturing cell formation problem.* Computers & Operation Research, 33, pp. 2219-2245, 2006 (available online from March 2005).

[3] BROWN, E. C., SUMICHRAST, R, T.: *Evaluating performance advantages of grouping genetic algorithms.* Engineering Applications of Artificial Intelligence, 18, pp. 1-12, 2005.

[4] CHANDRASEKHARAN, M. P., RAJAGOPALAN, R.: *ZODIAC-An algorithm for concurrent formation of part-families and machine-cells.* International Journal of Production Research, 25(6), pp. 835-850, 1987.

[5] CHENG, C. H.: *Algorithms for Grouping Machine Groups in Group Technology.* Omega, 20(4), pp. 493–501, 1992.

[6] DE LIT, P., FALKENAUER, E., DELCHAMBRE, A.: *Grouping genetic algorithms: an efficient method to solve the cell formation problem.* Mathematics and Computers in Simulation, 51, pp. 257-271, 2000.

[7] FALKENAUER, E.: *The grouping genetic algorithm-widening the scope of the GAs.* JORBEL-Belgian Journal of Operations Research, Statistics and Computer Science, 33, pp. 79-102, 1992.

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

[9] HOLLAND, J. H.: *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Application to Biology, Control and Artificial Intelligence.* Ann Arbor, MI: University of Michigan Press, 1975.

[10] JEON, G., LEEP, H. R.: *Forming part families by using genetic algorithm and designing machine cells under demand changes.* Computers & Operations Research, 33, pp. 263-283, 2006 (available online from April 2005).

[11] KÖREI, A.: *Solving cell formation problems with concept lattices.* Machine Engineering, 5(3-4), pp. 37-46, 2005.

[12] KUSIAK, A.: *The Generalized Group Technology Concept.* International Journal of Production Research, 25(4), pp. 561-569, 1987.

[13] KUSIAK, A., BOE J. W., CHENG, C. H.: *Designing cellular manufacturing systems. Branch-and-bound and A* approaches.* IIE Transactions, 25(4), pp. 46-56, 1993.

[14] RADELECZKI, S.: *Classification systems and their lattice*, Discussiones Mathematicae General Algebra and Applications, 22, pp. 167-181, 2002.

[15] SEIFODDINI, H., WOLFE, P. M.: *Application of the similarity cofficient method in group technology.* IIE Transactions, 18(3), pp. 271-277, 1986.

[16] SHAFER, S. M.: *Part-Machine-Labour Grouping: The Problem and Solution Methods.* In: Group Technology and Cellular Manufacturing, State-of-the-Art Synthesis of Research and Practice, Edited by Suresh N.C. Kay J.M., Kluwer Academic Publishers, 1998.

[17] TÓTH, T., MOLNÁR, M.: *Computarized Similarity of Parts Supporting Group Technology.* Production Systems and Information Engineering, 1, pp. 71-82, 2003.

[18] WANG, J.: *Formation of machine cells and part families in cellular manufacturing systems using a linear assignment algorithm.* Automatica, 39, pp. 1607-1615, 2003.

[19] WEI, J. C., KERN, G. M.: *Commonality analysis: A linear cell clustering algorithm for group technology.* International Journal of Production Research, 27(12), pp. 2053-2062, 1989.

[20] WON, Y., LEE, K. C.: *Modified p-median approach for efficient GT cell formation.* Computers & Industrial Engineering, 46, pp. 495-510, 2004.

[21] YIN, Y., YASUDA, K.: *Similarity coefficient methods applied to the cell formation problem: A taxonomy and review.* International Journal of Production Economics, 101, pp. 329-352, 2006 (available online from March 2005).