CONVEX EXTENSION OF DISCRETE-CONVEX FUNCTIONS AND APPLICATIONS IN OPTIMIZATION OF PARALLEL AND DISTRIBUTED SYSTEMS

Tiit Riismaa
Institute of Cybernetics at Tallinn University of Technology
Akadeemia tee 21, Tallinn, 12618, Estonia
E-mail: tiitr@ioc.ee

A method of description and optimization of the structure of multi-level processing systems is presented. The set of feasible structures for such class of systems is defined. The representation of this set is constructed in terms of the graph theory. A recursive algorithm is constructed to solve the general optimization problem on the structures. For the reduced statement two types of variable parameters are defined: for the level size and for the relations of adjacent levels. Two different classes of iteration methods are developed. For solving the reduced problem the recursive algorithm is constructed, where index of level is the index of recursion. Also a numerical method of local searching is developed. On each step of the iteration the calculation of the value of objective function is required only on some vertices of some kind of unit cube. The considered approach is illustrated by modelling and optimization of the structure of multi-level processing system.

1. Introduction

Large-scale problems can be decomposed in many different ways. The current approach for describing and optimizing the structure of hierarchical systems is based on a multi-level partitioning of given finite set in which the qualities of the system may depend on the partitioning. Examples of problems of this class are aggregation problems, structuring of decision-making systems, database structuring, the problems of multiple distribution or centralization, multi-level tournament systems, multi-level distribution systems, optimal clustering problems.

In a multi-level distribution system each element is a supplier for some lower level elements and a customer for one higher-level element. The zero-level elements are only customers and the unique top-level element is only a supplier. The choice of optimal number of suppliers-customers on each level is a mathematically complicated problem.

The multi-level tournament system [2] is a relatively simple special case of a multi-level processing system. To consider a tournament system, the number of games (pair-wise comparisons) is a quadratic function of the number of participants. This is a very quickly increasing function. If the number of participants is large, the number of games is very large. This is a reason why the multi-level approach is useful for the selection of the winner. From the tournaments of the first level the winners are distributed between the tournaments of the next level. The second level tournaments winners are going to the third level, until the winner is selected. Suppose the goal is to minimize the number of all games. If the price for all games is the same, the solution of the problem is well known. Each tournament has two participants and only one game is played. If the prices of games for different levels are different or constraints to the number of levels are active, a relatively complicated nonlinear integer programming problem arises.

The optimal structuring procedure considered in this paper is based on the full set of hierarchical trees of a feasible structure that could be composed from the given set of elements [14]. In the context of this statement an element is considered as a logical part of the processing system that is carrying out an identifiable mission and obeys the necessary functionality and autonomy [1, 3, 5, 6, 7].

The hierarchical structure is considered as a hierarchical tree. The variable parameters are also parameters that describe this tree. The feasible set of structures is a set of hierarchical trees. The arising optimization problem is an integer mathematical programming problem. In general, the solving process is exposed as a systematized selection of all possible feasible variations.

The assembling problem as well as a broad class of design and implementation problems, such as component selection in production systems, reconfiguration of manufacturing structures, optimization of the hierarchy of decision making systems, multi-level aggregation, creation and cancellation of levels [15], etc. can be mathematically stated as a multi-level selection problem [9, 19, 21, 22, 23].

The main difficulty from the point of view of optimization is that the number of subsets of partitioning is a variable parameter. For corresponding optimization problem it means that upper limit of summation, the number of summands (integer valued parameter) is a variable parameter. The search for a solution to this nonlinear integer programming problem is considered mathematically complicated.

2. Discrete approach of optimal multi-level paralleling

Consider all *s*-levels hierarchies, where nodes on level *i* are selected from the given nonempty and disjoint sets and all selected nodes are connected with selected nodes on adjacent levels. All oriented trees of this kind form the feasible set of hierarchies [12, 14, 16, 17].

The illustration of this formalism is given in Fig.1.

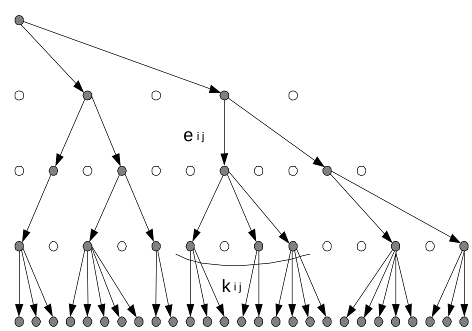

Figure 1 - Feasible set of structures

Suppose Y_i is an adjacent matrix of levels *i* and $i-1$ $(i=1,...,s)$. Suppose m_0 is the number of 0-level elements (level of object).

Theorem 1. All hierarchies with adjacent matrixes $\{Y_1,...,Y_s\}$ from the described set of hierarchies satisfy the condition

$$Y_s \cdot \ldots \cdot Y_1 = (\underbrace{1,\ldots,1}_{m_0}) \qquad (1)$$

The assertion of this theorem is determined directly [12].

3. The graph theory statement of general problem of structure optimization

The general optimization problem is stated as a problem of selecting the feasible structure which corresponds to the minimum of total loss given in the separable-additive form:

$$\min\left\{\sum_{i=1}^{s}\sum_{j=1}^{m_i}\mathrm{h}_{\mathrm{ij}}\left(\sum_{r=1}^{m_{i-1}} d_{jr}^{\,i}\, y_{jr}^{\,i}\right)\middle|\, Y_s\cdot\ldots\cdot Y_1=(\underbrace{1,\ldots,1}_{m_0})\right\}\text{ over } Y_1,\ldots,Y_s. \qquad (2)$$

Here $h_{ij}(\cdot)$ is an increasing loss function of *j*-th element on *i*-th level and d_{jr}^{i} is the element of $m_i\times m_{i-1}$ matrix D_i for the cost of connection between the *i*-th and (*i*-1)-th level [4, 16].

The meaning of functions $h_{ij}(k)$ depends on the type of the particular system.

By the optimization of the structure of multi-level tournament system, the loss inside the *j*-th tournament on *i*-th level is

$$h_{ij}(k_{ij})=d_j^i k_{ij}(k_{ij}-1),$$

where k_{ij} is the number of participants of *j*-th tournament of *i*-th league.

By complexity optimization of hierarchically connected subsystems, the loss inside the *j*-th set of partitioning on *i*-th level may be defined as follows:

$$h_{ij}(k)=\sum_{q=1}^{k} a_{ijq}\,\frac{k!}{q!(k-q)!}.$$

In this case the value of the function $h_{ij}(k)$ describes the number of all nonempty subsystems inside the *j*-th set of partitioning on *i*-th level.

Mathematically this problem is an integer programming problem with a non-continuous objective function and with a finite feasible set. For solving this kind of nonlinear integer programming problems only non-effective classical methods are known.

4. Recursive algorithm for solving the general problem of structure optimization

In this part the algorithm that selects the feasible structures corresponding to the minimum of total loss will be introduced.

Denote

$$N_1=\left\{Y_1\middle|\sum_{j=1}^{m_1} y_{jr}^1=1(r=1,\ldots,m_0)\right\},\qquad N_i=\left\{Y_i\middle|Y_i\cdot\ldots\cdot Y_1=(\underbrace{1,\ldots,1}_{m_0})\right\}(i=2,\ldots,s).$$

The set N_i contain all i-th elements Y_i of feasible presentations $\{Y_1,\ldots,Y_s\}$ $(i=1,\ldots,s)$.

Define the functions

$$f_1(Y_1)=\left\{\sum_{j=1}^{m_1} h_{1j}\left(\sum_{r=1}^{m_0} d_{jr}^1 y_{jr}^1\right)\middle|\sum_{j=1}^{m_1} y_{jr}^1=1\right\},$$

$$f_i(Y_i)=\min\left\{f_{i-1}(Y_{i-1})+\sum_{j=1}^{m_i} h_{ij}\left(\sum_{r=1}^{m_{i-1}} d_{jr}^i y_{jr}^i\right)\middle| Y_{i-1}\in N_{i-1}; Y_i\cdot\ldots\cdot Y_1=(\underbrace{1,\ldots,1}_{m_0})\right\}$$

over Y_{i-1} $\quad (i=2,\ldots,s)$.

The function $f_i(Y_i)$ represents the minimum loss on levels $1,\ldots,i$ for adjacent matrix Y_i.

Define the sets

$$P_i(Y_i)=Arg\min\left\{f_{i-1}(Y_{i-1})+\sum_{j=1}^{m_i} h_{ij}\left(\sum_{r=1}^{m_{i-1}} d_{jr}^i y_{jr}^i\right)\middle| Y_{i-1}\in N_{i-1}; Y_i\cdot\ldots\cdot Y_1=(\underbrace{1,\ldots,1}_{m_0})\right\}$$

over Y_{i-1} $\quad (i=2,\ldots,s)$;

$$\tilde{P}_s(Y_s^*)=Arg\min\{f_s(Y_s)|Y_s\in N_s\}.$$

The set $P_i(Y_i)$ is a set of adjacent matrixes Y_{i-1} minimizing the loss on levels $1,\ldots,i$ for this adjacent matrix Y_i.
The solution in terms of defined sets is sequential:

Step 1. For each $Y_1\in N_1$ compute the function $f_1(Y_1)$ and construct the set
$P_1(Y_1)=Arg\min\{f_1(Y_1)|Y_1\in N_1\}$.

Step i. For each $Y_i\in N_i$ construct recursively by the index i the sets
$P_i(Y_i)$ $\quad (i=2,\ldots,s-1)$.

Step s. For each $Y_s\in N_s$ construct the set $\tilde{P}_s(Y_s^*)=Arg\min\{f_s(Y_s)|Y_s\in N_s\}$.

Now the following set is constructed:

$$U_s^*=\left\{[Y_1^*,\ldots,Y_s^*]\middle| Y_s^*\in\tilde{P}_s(Y_s^*), Y_{s-1}^*\in P_s(Y_s^*),\ldots,Y_1^*\in P_2(Y_2^*)\right\}.$$

This set contain formally all presentations of optimal structures from the feasible set.

On step i for each adjacent matrix Y_i find the optimal adjacent matrixes Y_{i-1} must be founded. It means that this minimization method is very general, but time-consuming and applicable for small-dimensional problems. For this reason we further slightly restrict the class of problems by simplifying the functional dependencies in the objective function.

5. Reduced problem of structure optimization

Now an important special case is considered where the connection cost between the adjacent levels is the property of the supreme level: each row of the connection cost matrices between the adjacent levels consists of equal elements.
There is a possibility to change the variables and to represent the problem so that

$$d^i_{jr} = 1; \quad i = 1,...s; \quad j = 1,...,m_i; \quad r = 1,...,m_{i-1}.$$

Now the total loss depends only on sums $\sum_{r=1}^{m_{i-1}} y^i_{jr} = k_{ij.}$, where k_{ij} is the number of edges beginning in the *j*-th node on *i*-th level.

Recognize also that $\sum_{j=1}^{m_i} k_{ij} = p_{i-1}, \ i = 1,...,s$, where p_i is the number of nodes on *i*-th level. If to suppose additionally that $h_{i1}(k) \le \cdots \le h_{im_i}(k)$ for each integer k, the general problem (2) transforms into the two mutually dependent phases:

$$\min\left\{ \sum_{i=1}^{s} g_i(p_{i-1}, p_i) \,\middle|\, (p_1,..., p_{s-1},1) \in W^s \right\} \text{ over } p_1,..., p_{s-1} \tag{3}$$

where

$$g_i(p_{i-1}, p_i) = \min\left\{ \sum_{j=1}^{p_i} h_{ij}(k_{ij}) \,\middle|\, \sum_{j=1}^{p_i} k_{ij} = p_{i-1} \right\} \text{ over } k_{i1},...,k_{ip_i}$$
$$W^s = \{(p_1,..., p_{s-1},1) \,|\, 1 \le p_i \le p_{i-1}\}. \tag{4}$$

Free variables of the inner minimization (4) are used to describe the connections between the adjacent levels. Free variables of the outer minimization (3) are used for the representation of the number of elements at each level.

6. Recursive algorithm for solving the reduced problem of structure optimization

Consider the functions that represent the minimum loss on levels 1, 2,…, *i*, if there are exactly p_i nodes on *i*-th level:

$$G_1(p_1) = g_1(p_0, p_1),$$

$$G_i(p_i) = \min\{G_{i-1}(p_{i-1}) + g_i(p_{i-1}, p_i) \,|\, p_{i-1} \in A_{i-1}\} \text{ over } p_{i-1}$$

and

$$A_i = \{1,..., m_i\}, \quad i = 1,..., s.$$

For each $p_i \in A_i$ define the sets
$$B_i(p_i) = Arg\min\{G_{i-1}(p_{i-1}) + g_i(p_{i-1}, p_i) \,|\, p_{i-1} \in A_{i-1}\}$$
$$B_1(p_1) = \{p_0\}$$

and for each $p_{i-1} \in B_i(p_i)$, $p_i \in A_i$ the sets

$$T_i(p_{i-1}, p_i) = \{(k_{i1},...,k_{ip_i}) = Arg\min\left(\sum_{j=1}^{p_i} h_{ij}(k_{ij}) \middle| \sum_{j=1}^{p_i} k_{ij} = p_{i-1}\right)\}.$$

For each $p_i \in A_i$ denote

$T_i(p_i) = \bigcup T_i(p_{i-1}, p_i)$, over all $p_{i-1} \in B_i(p_i), i = 1,...,s.$

The solution in terms of defined sets is sequential:

Step 1. For each $p_1 \in A_1$ compute the function $G_1(p_1)$ and construct the set $T_1(p_0, p_1)$.

Step i. For each $p_i \in A_i$ construct recursively by the index i

$G_i(p_i)$, $B_i(p_i)$ and

$T_i(p_i) = \cup T_i(p_{i-1}, p_i)$ over $p_{i-1} \in B_i(p_i)$, $i = 2,...,s.$

As a result of the application of given steps we have for every $p_i \in A_i$

all $G_i(p_i)$, $B_i(p_i)$, $T_i(p_i)$, $i = 2,...,s$.

Now let us consider following sets:

$$V_s = \{(p_1^*,..., p_{s-1}^*,1) \mid p_{s-1}^* \in B_s(1), p_{s-2}^* \in B_{s-1}(p_{s-1}^*),..., p_1^* \in B_2(p_2^*)\}.$$

Each vector $(p_1^*,..., p_{s-1}^*,1) \in V_s$ defines a target set:

$$Q_s = \{\{T_1(p_0, p_1^*), T_2(p_1^*, p_2^*),...,T_{s-1}(p_{s-2}^*, p_{s-1}^*), T_s(p_{s-1}^*,1)\} | (p_1^*,...,p_{s-1}^*,1) \in V_s\}$$

Each element from the set Q_s corresponds to all feasible representations $\{Y_1^*, Y_2^*,...,Y_s^*\}$ that do not modify the minimal value of the cost function.

7. Mathematical properties of the reduced problem of structure optimization

This statement has some advantages from the point of view of the optimization technique. It is possible to adapt effective methods of the convex programming for solving outlined special cases. The function $f: X \to R$, $X \subset R^n$, is called discrete-convex [8,14] if for all $x_i \in X\,(i = 1,...,n+1); \lambda_i \geq 0(i = 1,...,n+1)$ and $\sum_{i=1}^{n+1} \lambda_i = 1; \sum_{i=1}^{n+1} \lambda_i x_i \in X$ holds $f(\sum_{i=1}^{n+1} \lambda_i x_i) \leq \sum_{i=1}^{n+1} \lambda_i f(x_i)$.

Important special case is $X = Z^n$, where $Z^n = \underbrace{Z \times ... \times Z}_{n}$ and $Z = \{..., -1, 0, 1, ...\}$.

The use of all n+1 elements convex combinations follows from the well-known theorem of Caratheodory [20].
The graph of a discrete-convex function is a part of the graph of a convex function.

The convex extension f_c of function $f : X \to R \left(X \subset R^n\right)$ is the majoring convex function $f_c : conv\, X \to R$, where $f_c(x) = f(x)$ if $x \in X$.

Theorem 2. The function $f : X \to R$ $(X \subset R^n)$ can be extended to convex function on *convX* if f is discrete-convex on X. The convex extension f_c of f is

$$f_c(x) = \min_{x_i, \lambda_i} \left\{ \sum_{i=1}^{n+1} \lambda_i f(x_i) \,\middle|\, x = \sum_{i=1}^{n+1} \lambda_i x_i ;\ \sum_{i=1}^{n+1} \lambda_i = 1;\ \lambda_i \geq 0 \left(i = 1, ..., n+1\right), x_i \in X \left(i = 1, ..., n+1\right) \right\}$$

Theorem 3. The convex extension f_c of f is

$$f_c(x) = \begin{cases} \max\limits_{a,b} \left\{ \langle a, x \rangle + b \,\middle|\, \langle a, y \rangle + b \leq f(y), y \in X \right\}, x \notin X \\ f(x), x \in X \end{cases}$$

The convex extension is so called point-wise maximum over all linear functions not exceeding the given function.
The convex function of a discrete-convex function is a piecewise linear function.
Each discrete-convex function has a unique convex extension.
The class of discrete-convex functions is the largest one to be extended to the convex functions.

Theorem 4. If $h_{ij}(k)(i = 1, ..., s;\ j = 1, ..., m_i)$ in (4) are discrete-convex functions then $\sum_{i=1}^{s} g_i(p_{i-1}, p_i)$ in (3) is a discrete-convex function.

The proof of this lemma is not complicated but needs a lot of secondary results, is long and is not presented here.

Given results enable to extend the objective function (3), (4) to the convex function.

8. Algorithm of local searching for the reduced problem of structure optimization

The particular choice of the variables (3), (4) enables to construct a class of methods for finding the global optimum. In this paper it is only declared that the objective function of such integer programming problem is a discrete-convex function.

The described class of methods is based on the known algorithms of convex programming [20] adapted for the integer programming.

Recall of (3) is denoted

$g\left(p_0, p_1, ..., p_{s-1}, 1\right) = \sum_{i=1}^{s} g_i\left(p_{i-1}, p_i\right)$ and $p = \left(p_1, ..., p_{s-1}, 1\right)$.

Consider following finite-step algorithm:

$$p_0^{(s)} = (\underbrace{1,\ldots,1}_{s}), \text{ and } p_k^{(s)} = p_{k-1}^{(s)} + x_k^{(s)}(q,t), \quad (k = 1,2,\ldots).$$

It is assumed that

1) $x_k^{(s)}(q,t)$ are lexicographically ordered by (q,t):

$$x_k^{(s)}(q,t) = \left(\overbrace{ \left(\underbrace{0,\ldots,0}_{t} \right), \left(\underbrace{1,\ldots,1}_{q} \right), 0,\ldots,0 }^{s} \right), \quad (q = 1,\ldots,s-t-1; t = 0,\ldots,s-2); \tag{5}$$

2) $p_0 \geq p_{k1}^{(s)} \geq \ldots \geq p_{ks}^{(s)} = 1;$ (6)

3) $x_k^{(s)}(q,t)$ is lexicographically the first that satisfies the condition

$$g\left(p_0, p_{k1}^{(s)},\ldots, p_{ks-1}^{(s)},1\right) \leq g\left(p_0, p_{k-11}^{(s)},\ldots, p_{k-1s-1}^{(s)},1\right) \tag{7}$$

Remark 1. Consider the vertices of the (s-1)-dimensional unit cube:

a) with vertices of integer coordinates, where the nearest vertex to the (s-1)-dimensional zero-point is $\left(p_{k-11}^{(s)},\ldots, p_{k-1s-1}^{(s)}\right)$;

b) with vertices satisfying condition (5).

The number of that kind of vertices (5) is $\frac{1}{2} \cdot s(s-1)$. The number of vertices (5), (6) is no more than $\frac{1}{2} \cdot s(s-1)$.

The condition (5) puts in order all vertices of described unit cube.

Remark 2. On the step *k* the value of goal function is computed on ordered vertices (5), (6) of unit cube until the first value satisfying (7) is found. If that kind of a value does not exist, the solution of problem (3), (4) has been found [14].

9. Academic example: optimization the structure of multi-level processing system

Consider the processing of *n* parts [11, 13]. In case of one processing unit the overall processing and waiting time for all n parts is proportional to n^2 and is a quickly increasing function. For this reason the hierarchical system of processing can be suitable. From zero-level (level of object) the parts will be distributed between p_1 first-level processing units and processed (aggregated, packed etc.) by these units. After that the parts will be distributed between p_2 second-level processing units and processed further and so on. From p_{s-1} (s - 1)-level the units will be sent to the unique s-level unit and processed finally. The cost of processing and waiting on level i is approximately

$$g_i(p_{i-1}, p_i) = (d_i\, l_{i-1}\, p_{i-1} / p_i)^2\, p_i + a_i\, p_i \quad (i = 1,\ldots,s).$$

Here l_i is the number of aggregates produced by one robot on level i (a number of boxes for packing unit), d_i is a loss unit inside the level i, and a_i is the cost of i-th level processing unit. The variable parameters are the number of processing units on each level $p_i(i=1,...,s)$.
The goal is to minimize the total loss (processing time, waiting time, the cost of processing units) over all levels:

$$\min\sum_{i=1}^{s}\left((d_i l_{i-1})^2\left(\left(p_i\left(\left[\frac{p_{i-1}}{p_i}\right]+1\right)-p_{i-1}\right)\left[\frac{p_{i-1}}{p_i}\right]^2+\left(p_{i-1}-p_i\left[\frac{p_{i-1}}{p_i}\right]\right)\left(\left[\frac{p_{i-1}}{p_i}\right]+1\right)^2\right)+a_i p_i\right)$$

over natural $p_i(i=1,...,s)$. Here $[p]$ is the integer part of p. The goal function of this discrete programming problem is discrete-convex. It is possible to extend this function to convex function (Lemma 1) and get a solvable convex programming problem using the method of local searching.

10. Concluding remarks

Many discrete or finite hierarchical structuring problems can be formulated mathematically as a multi-level partitioning procedure of a finite set of nonempty subsets. This partitioning procedure is considered as a hierarchy where to the subsets of partitioning correspond nodes of hierarchy and the relation of containing of subsets define the arcs of the hierarchy. The feasible set of structures is a set of hierarchies (oriented trees) corresponding to the full set of multi-level partitioning of given finite set.

Each tree from this set is represented by a sequence of Boolean matrices, where each of these matrices is an adjacency matrix of neighboring levels. To guarantee the feasibility of the representation, the sequence of Boolean matrices must satisfy some conditions – a set of linear and nonlinear equalities and inequalities.

Examples of problems of this class are aggregation problems, structuring of decision-making systems, database structuring, multi-level tournament systems, multi-level distribution systems.

The recursive algorithms considered in the paper are double-cycle optimization algorithms. The inner cycle increases the number of elements inside of the current level by one unit, and outer cycle on each step increases the number of levels by one unit. On each iteration step a one-parameter integer programming problem must be solved.

The formalism described in this paper enables to state the reduced problem as a two-phase mutually dependent discrete optimization problem and construct some classes of solution methods. Variable parameters of the inner minimization problem are used for the description of connections between adjacent levels. Variable parameters of the outer minimization problem are used for the presentation of the number of elements on each level.

The two-phase statement of optimisation problem guarantees the possibility to extend the objective function to the convex function and enables to construct algorithms for finding the global optimum. In this paper for finding the global optimum the method of local searching is constructed. On each step of iteration the calculation of the value of objective function is required only on some vertices of some kind of unit cube.

The approach is illustrated by a multi-level production system example.

11. References

[1] Berio, G., Vernadat, F. B., New Developments in Enterprise Modeling Using CIM-OSA. Computers in Industry, 40 (1999), 99 - 114.

[2] Laslier, J.-F., Tournament Solutions and Majority Voting. Springer, Berlin, Germany, 1997.

[3] Littover, M., Randvee, I., Riismaa, T., Vain, J., Recursive Integer Programming Technique for Structural Optimization. In: Proc. 4-th International Scientific Colloquium on CAx Techniques, Bielefeld, Germany, 1999, Druckerei Robert Bechauf, Bielefeld, Germany, 1999, 95 - 102.

[4] Littover, M., Randvee, I., Riismaa, T., Vain, J., Optimization of the Structure of Multi-Parameter Multi-Level Selection. In: Proc. 17th International Conference on CAD/CAM, Robotics and Factories of the Future, Durban, South Africa, 2001, University of Natal, Durban, South Africa, 2001, 317 - 322.

[5] Mesarovic, M. D., Macko, D. and Takahara, Y., Theory of Hierarchical Multi-Level Systems. Academic Press, New York, 1970.

[6] Mesarovic, M. D., Takahara, Y., General System Theory: Mathematical Foundations. Academic Press, New York, 1975.

[7] Miyawaki, F., Masamune, K., Suzuki, S., Yoshimitsu, K., Vain, J. Scrub nurse robot system - intraoperative motion analysis of a scrub nurse and timed automata based model for surgery. IEEE Transaction on Industrial Electronics, Special Section on Human Adaptive Mechatronics, **52**, 5 (2005), 1227 - 1235.

[8] Murota, K., Discrete Convex Analysis. SIAM, Philadelphia, 2003.

[9] Randvee, I., Riismaa, T., Vain, J., Optimization of Holonic Structures. In: Proc. Workshop on Production Planning and Control, Mons, Belgium, 2000, Ateliers de la FUCaM, Mons, Belgium, 2000, 61 - 66.

[10] Riismaa, T., Mathematical Modelling of Continuous Optimal Multi-Level Paralleling. In: Proc. Fifth IASTED International Conference on Modelling, Simulation, and Optimization, Oranjestad, Aruba, 2005, ACTA Press, Anaheim – Calgary – Zurich, 2005, 181 – 186.

[11] Riismaa, T. Multi-Level Continuous Paralleling. The Internatonal Journal of INGENIUM, 4 (2005), 485 - 492.

[12] Riismaa, T., Optimization of the Structure of Fuzzy Multi-Level Decision-Making System. In: Proc. International Conference on Modelling and Simulation of Business Systems, Vilnius, Lithuania, 2003, Technologija, Kaunas, Lithuania, 2003, 31 - 34.

[13] Riismaa, T., Optimization of the Structure of Multi-Level Parallel Processing. In: Proc. IASTED International Conference in Applied Informatics, Innsbruck, Austria, 2002, ACTA Press, Anaheim, Calgary, Zurich, 2002, 77 - 80.

[14] Riismaa, T., Description and Optimization of Hierarchical Systems. Automation and Remote Control, 12(1993), 146 - 152.

[15] Riismaa, T., Randvee, I., Description and Optimization of the Structure of Multi-Level Parallel Selection. In. Proc: 19th International Conference on CAD/CAM, Robotics and Factories of the Future, Kuala Lumpur, Malaysia, 2003, SIRIM Berhad, Selangor, Malaysia, 2003, 423 - 429.

[16] Riismaa, T., Randvee, I., Recursive Algorithms for Optimization of Multi-Level Selection. In: Proc. 18th International Conference on CAD/CAM, Robotics and Factories of the Future, Porto, Portugal, 2002, INESC Porto, Porto, Portugal, 2002, 333 - 340.

[17] Riismaa, T., Randvee, I., Vain, J., Optimization of the Structure of Multi-Level Parallel Assembling. In: Proc. 7th IFAC Workshop on Intelligent Manufacturing Systems, Budapest, Hungary, 2003, Elsevier, Oxford, England, 2003, 235 - 238.

[18] Riismaa, T., Vaarmann, O., Optimal Decomposition of Large-Scale Systems. In: Proc. International Workshop on Harbour, Maritime and Multimodal Logistics Modelling & Simulation, Riga, Latvia, 2003, Riga Technical University, Riga, Latvia, 2003, 385 - 394.

[19] Riismaa T., Randvee, I., Restructuration of Control Scenarios. In: Proc. 23rd EUROMICRO Conference New Frontiers of Information Technology, Budapest, Hungary, 1997, IEEE Computer Society, Washington, Bryssels, Tokyo, 1997, 281 - 287.

[20] Rockafellar R. T., Convex Analysis. Princeton University Press, New Jersey, 1970.

[21] Vain, J., Randvee I., Riismaa, T. Two Phase Technique For Assembly Line Balancing. In: Preprints of 16th IFAC World Congress, Prague, Czech Republic, 2005, CD-ROM, 2005.

[22] Vain, J., Randvee, I., Riismaa, T., Ernits, J. Solving line balancing problems with model checking. Proceedings of the Estonian Academy of Sciences. Engineering, **8**, 4 (2002), 211 - 222.

[23] Vain, J., Ernits, J.-P., Littover, M., Randvee, I., Riismaa, T. A Tool for Flexible Planning of Rescue Routes. In: Proc. IFAC Workshop on Control in Natural Disasters, Tokyo, Japan, 1998, Pergamon, Oxford, England, 1999, 85 - 90.