Designing Reliable Communication for Heterogeneous Computer Systems

Miroslaw Hajder, Janusz Kolbusz, and Roman Korostenskyi

University of Information Technology and Management in Rzeszow
miroslaw.hajder@gmail.com,{jkolbusz,rkorostenskyi}@wsiz.rzeszow.pl

Abstract. This study describes the network design solution to the problem of connecting heterogeneous computer systems based on analysis of multipartite hypergraphs. To do this proposes a mathematical model of reliability for the two modes of operation of the system: with redundancy communication subsystem and the division of communication load. As the evaluation criteria applied solutions expected changes in processing capacity, latency communication and system reliability. Solution design task is sought in the collection Pareto optima, which describes a method for selecting a particular solution in case of equivalence with respect to the vector of the objective function.

Key words: Multipartite hypergraphs, architecture connections, system reliability model, design methodology

1 Introduction

Another important feature of modern processing systems is the variety of offered services. Nowadays, in the same network they are different, often incompatible with communication services (eg. Isochronous and synchronous transfer). This issue requires a change of quality of provided services, through the dynamic allocation of independent communication channels to users or services present in the network specifically in particular, this applies to multimedia services rendered in real-time or services comprising critical infrastructure. Modern distributed systems are also characterized by high dynamics of changes in operating parameters. Load elements of computing and communications is changing rapidly, which prevents the design and execution of the network to meet even medium-term requirements of the users. Another disadvantage occurring in communication subsystems ubiquity of traffic is bursty traffic, obstructing, and sometimes preventing proper functioning of the network. Currently, the solution to the above problems is the use of load leveling, both communications and computing. Moreover, an effective solution to most of these problems can be also providing a flexible reconfiguration of connections, preferably at the logical level, without having to modify the hardware architecture. In this way, links can be dynamically adapted to the current traffic pattern.

Effective methods of reconfiguring connections should be seen in the use of modern communication technologies, especially those that allow to realize it at the logical level and which additionally improve the utilization of physical communication channels. An interesting issue is the construction of multi-channel network of bus-sharing bus logic dynamic range conversion of a set of buses to which the user is attached. Because of that it becomes possible to adapt to the existing network architecture in the traffic patterns. However, the use of such architecture requires solving a specific design task, which for obvious reasons should be characterized by an acceptable time complexity and memory. Because of the combinatorial nature of the task it is difficult to meet.

These issue may include the design task to build large class of system configuration tasks. This task is mostly decomposed into three basic subtasks: a. The selection of system components; b. their deployment; c. determining the connection between them. In previous work, component selection subtask is solved inter alia by implemented using for this purpose methods seeking the shortest path [1], [2] Backpack block [2], [3] the clustering multipartite graph [4], mullioned clique [5], morphological analysis. Subsequently, a solution subtasks arrangement of the components, currently is the most frequently used variants task assignment [2], [6], [7], [8]. To determine the connections between system components there is the most commonly used a method of agglomeration [9], [10] and the method solving the task of building the optimal hierarchy [11].

2 Architecture Connections and System Reliability Model

The Fig. 1 shows a distributed system consisting of K_N node calculation - decoration, each of which has a K_l retunes elements of the transceiver and the K_B communication buses. Buses B_i $(i = 1, \ldots, K_B)$ are logical and can be implemented on the basis of methods of reproduction of wave-go in one physical bus D_F. The addition of logical channels to any node N_i physical bus B_F is done by using a physical connection channels *l* and distributors of bus channels *C*.

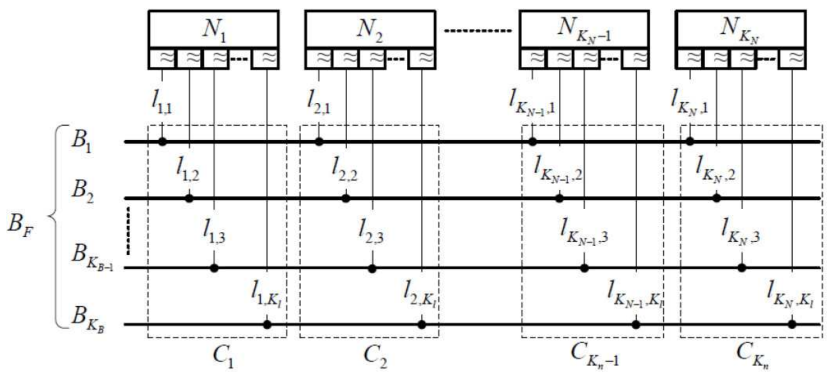

Fig. 1. Trunk generalized computing system architecture

Interconnect architecture of the analyzed system can be dynamically reconfigured by tunable elements transmitter - receiver. If you change the traffic pattern elements of the transmitter - receiver extent of the wave changes, and indeed attach themselves to another logical bus. The system can operate in two modes: a. redundancy communication subsystem; b. the communication load sharing. To evaluate the operating characteristics of the system in one of these modes, the probabilistic model suggested combinatorial, and in particular evaluating the reliability R. In this model, the kit: the transceiver - receiving (≈), the physical connection cable (*l*), and a manifold physical channel (*C*) are treated as a single device connection. Let p_{io} - the probability of performance transceiver - the receiving node computational p_l - probability of physical fitness connecting channel; p_c - the probability of the distributor channel efficiency Trunk, the p_{fk} - probability of physical fitness BUS channel. Then, the probability of p_{ku} efficiency of connecting the node to the logical channel BUS is defined as: $p_{ku} = p_{io}p_l p_c$ and p_{uku} probability of the merger selected node with other nodes calculation is equal to $p_{uku} = p_{io}p_l p_c p_{fk}$.

Trunk consider the reliability of a distributed system with connections complete (each compute node is connected to each bus logic) and equal rights computational nodes, which is the most general example of this class of systems. The probability $p_{we}(k_{we})$ efficiency connection sets providing connection to the bus channel not less than $k_{we}^{\min}$ compute nodes with their total number K_N determined by the expression:

$$p_{we}(k_{we}) = p_{fk} \sum_{i=k_{we}^{\min}}^{K_N} C_k^i p_{ku}^i (1 - p_{ku})^{k_{we}-i} \tag{1}$$

Let K_B to be the amount of bus logic (ie. The maximum multiplicity system interface), p_{we}^{sp} - likely performance computing node. Then, using expression (1), according to the proposed condition for the system in redundancy, reliability *R* is defined by the following formula:

$$R = \sum_{j=k_{we}^{\min}}^{K_N} C_{K_N}^j (p_{we}^{sp})^j (1 - p_{we}^{sp})^{K_N-j} \sum_{k=1}^{K_B} C_{K_B}^k p_{we}(j)^k (1 - p_{we}(j))^{K_B-k} \tag{2}$$

Consider the current system of equal rights computational nodes working in load sharing mode of communication. Let $W(k_{we}, k_m, \sigma)$ to be the amount of system efficiency states consisting of k_{we} compute nodes, connected with k_m canals system, which can be selected with the existence of σ denials transceiver components. In order to determine the amount of $W(k_{we}, k_m, \sigma)$ state performance of the system in the event of damage, etc., it is proposed to use the methods of include - exemption. $H_1(\sigma)$ number of states malfunction of the entire system in case of refusal not less than one minimum section for a system with equal rights nodes is equal to:

$$\begin{aligned} H_1(\sigma) = K_N C_{(K_N-1)K_B}^{\sigma-K_B} + C_{K_N}^2 C_{(K_N-1)K_B}^{\sigma-K_B} \textstyle\sum_{\alpha=1}^{K_B-1} C_{K_B}^{\alpha} = \\ C_{(K_{Ne}-1)K_B}^{\sigma-K_B} K_N + C_{K_N}^2 \textstyle\sum_{\alpha=1}^{K_B-1} C_{K_B}^{\alpha} \end{aligned} \tag{3}$$

Then, the value of *W* is equal to $W(k_{we}, k_m, \sigma) = C_{k_{we}k_m}^{\sigma} + \sum_{i=1}^{i_\sigma} (-1)^i H_i(\sigma)$, where: $C_{k_{we}k_m}^{\sigma}$ - total number of states σ denials system for transmitting elements - receiving; i_σ - the maximum number taken into account when assessing the minimum cross sections; $H_i(\sigma)$ - the number of states of a computing system failure, refusal to transmit elements - receiving no less than the minimum i sections. The probability $P_{ku}\left(k_{we},k_m,\sigma\right)$, k_{wu} ensures consistency nodes by k_m bus for refusals σ is defined by the expression:

$$P_{ku}\left(k_{we},k_m,\sigma\right)=W\left(k_{we},k_m,\sigma\right)p_{ku}^{k_{we}k_m-\sigma}\left(1-p_{ku}\right)^{\sigma} \tag{4}$$

Using the expression (4), the likelihood of $P_{ku}\left(k_{we},k_m\right)$ ensures the consistency of computing nodes k_{we}, k_m buses, with denials of the existence of σ can be written as:

$$P_{ku}\left(k_{we},k_m\right)=\sum_{\sigma=0}^{(k_{we}-1)k_m}P_{ku}\left(k_{we},k_m,\sigma\right) \tag{5}$$

Using the expression (5), it will determine the likelihood of consistency $P_{uku}\left(k_{we}\right)$, k_{we} compute nodes:

$P_{uku}\left(k_{we}\right)=\sum_{l=k_m^{\min}}^{K_B}C_{K_B}^{l}p_{fk}^{l}\left(1-p_{fk}\right)^{K_B-l}P_{ku}\left(k_{we},l\right)$,

where: $k_m^{\min}$ - the minimum required number of buses needed to provide the required bandwidth. In this way, the reliability of calculation system is equal to:

$$R=\sum_{n=k_{we}^{\min}}^{K_N}C_{K_N}^{n}\left(p_{we}^{sp}\right)^{n}\left(1-p_{we}^{sp}\right)^{K_N-n}P_{uku}\left(n\right) \tag{6}$$

For computing system client-server mode redundancy it will determine the likelihood of $P_{ks}\left(K_K,K_S\right)$ efficiency BUS communication channel to which, through the operational elements transmitter - receiver including no less than $k_k^{\min}$ customers with their total number of K_K and $k_s^{\min}$ servers with the total number of K_S:

$$\begin{gathered}P_{ks}\left(K_K,K_S\right)=p_{fk}\sum_{i=k_k^{\min}}^{K_K}\sum_{j=k_s^{\min}}^{K_S}C_{K_K}^{i}p_{ku}^{i}\left(1-p_{ku}\right)^{K_K-i}\cdot\\ C_{K_S}^{j}p_{ku}^{j}\left(1-p_{ku}\right)^{K_S-j}\end{gathered} \tag{7}$$

Let $p_k^{sp}$and p_s^{sp} be the likelihood of efficiency for clients and servers nodes, respectively. Then, using the expression (7), reliability Rcan be written as:

$$\begin{gathered}R=\sum_{m=k_k^{\min}}^{K_K}C_{K_K}^{m}\left(p_k^{sp}\right)^{m}\left(1-p_k^{sp}\right)^{K_K-m}\cdot\\ \sum_{n=k_s^{\min}}^{K_S}C_{K_K}^{n}\left(p_s^{sp}\right)^{n}\left(1-p_s^{sp}\right)^{K_S-n}\cdot\\ \sum_{l=1}^{K_m}C_{K_m}^{l}\left(P_{ks}\left(K_K,K_S\right)\right)^{l}\left(1-P_{ks}\left(K_K,K_S\right)\right)^{K_m-l}\cdot\end{gathered} \tag{8}$$

Let's consider the reliability of client-server system with a complete blend of computing and communication subsystem working in load sharing mode. Let $W\left(k_s,k_k,k_m,\sigma\right)$to be the number of states efficient system consisting of servers k_s, $k_k$clients, $k_m$bus, in the presence of σ denials transceiver components. For the client-server system, the number of failure conditions the $H_1\left(\sigma\right)$ no less than a minimum cross section can be determined using the expression:

$$H_1(\sigma) = (k_k + k_s)\, C^{\sigma-k_m}_{k_m(k_k+k_s-1)} + k_k k_s C^{\sigma-k_m}_{k_m(k_k+k_s-1)} = \\ = C^{\sigma-k_m}_{k_m(k_k+k_s-1)} \left(k_k + k_s + k_k k_s \sum_{\alpha=1}^{k_m-1} C^{\alpha}_{k_m} \right), \tag{9}$$

a number of states proper functioning as:

$$W(k_k, k_s, k_m, \sigma) = C^{\sigma}_{k_m(k_k+k_s)} + \sum_{i=1}^{i_\sigma} (-1)^i H_i(\sigma), \tag{10}$$

Where: $C^{\sigma}_{k_m(k_k+k_s)}$- the total number of computing system states that may occur at σ refusals. The probability $P_{ks}(k_s, k_k, k_m, \sigma)$ ensures consistency k_s servers and k_k clients using k_m bus in case of refusal elements σtransmitter - receiver can be written as:

$$P_{ku}(k_s, k_k, k_m, \sigma) = W(k_s, k_k, k_m, \sigma)\, p_{ku}^{(k_s+k_k)k_m-\sigma} (1-p_{ku})^{\sigma}. \tag{11}$$

The probability of $P_{ku}(k_s, k_k, k_m)$ servers to ensure coherence k_s and k_k clients using k_m bus in the event of a refusal elements transmitter - receiver has been defined as:

$$P_{ku}(k_s, k_k, k_m) = \sum_{s=0}^{(k_s+k_k)k_m} P_{ku}(k_s, k_k, k_m, \sigma), \tag{12}$$

and the probability $P_{ku}(k_s, k_m)$ of consistency k_s servers and k_k clients as:

$$P_{ku}(k_s, k_k) = \sum_{k_m=1}^{K_m} C^{k}_{K_m} p_{fk}^{k_m} (1-p_{fk})^{K_m-k_m} P_{ku}(k_s, k_k, k_m). \tag{13}$$

Using the expression (11), (12), and (13) the sought reliability R will be written as:

$$R = \sum_{k=k_k^{\min}}^{K_k} C^{k}_{K_k} (p_k^{sp})^k (1-p_k^{sp})^{K_k-k} \cdot \\ \sum_{s=k_s^{\min}}^{K_s} C^{s}_{K_s} (p_s^{sp})^s (1-p_s^{sp})^{K_s-s} P_{ku}(s,k) \tag{14}$$

The above-described methodology we will use for further connections to network design measuring system.

3 Task Design and Its Solution

We will consider hypergraph $H = (V, E)$ comprising a set $V = \{v\}$ of vertices and a set of $E = \{e\}$ of edges, which represent a subset of the set *V*, i.e. $e \subseteq V$. Hypergraph *H* is a *k*-regular if each of its edge $e \in E$ consists of *k* vertices. On the other hand, hypergraph *H* is *l*-partite graph, if the set of its vertices is divided into *l* subsets$V_1, V_2, \ldots, V_l$, in such a manner that the vertices of each of the edges $e = (v_1, v_2, \ldots, v_l) \in E$ belong to different parts of the graph, ie. $v_i \in V_i$, where $i = 1, \ldots, l$. For the determination of *l*- partite hypergraphs we will use record form $H = (V_1, V_2, \ldots, V_l)$.

Let's consider the *l*- partite hypergraph $H = (V_1, V_2, \ldots, V_l)$. In this graph, the part $a = \left(V_1^A, \ldots, V_i^A, \ldots, V_l^A, E_A\right)$, for $i = 1, \ldots, l$ and $V_l^A \subseteq V_l$, where any two edges $e_1, e_2 \in E_A$ overlap in one and the same vertex $v \in V_1^A$ and do not overlap at any vertex $v \in V_l^A$, will be called star. This means that the cardinality of $V_1^A$is 1, and the vertex $v \in V_1^A$, will be called the center of the star. We distinguish the simple and complex stars. If any pair of edges $e_1, e_2 \in E_A$ covers only in one vertex $v \in V_1^A$, then the star is called simple. Otherwise, a star will be called complex. The number of edges of the star will be called degree. For the edge $e = (v_1, v_2, \ldots, v_l) \in E$ of the star vertices$v_1$and $v_l$we will call end. In turn, the vertices $v_2, \ldots, v_{l-1}$ will be determined as internal. Vertices set of the part of graph $V_2, \ldots, V_{l-1}$ are composed of empty pairs of disjoint sets $V_i\,(v_j)$, $v_j \in V_j$, where: $i = 2, \ldots, l-1$, $j = i+1$.

For hypergraph $H = (V, E)$ its subhypergraph $H_1 = (W, U)$ we'll be called hypergraph for which set of the vertices *W* is the vertices subset *V* of hypergraph *H,* ie. $W \subseteq V$and the edge set *U* is the edges subset *E* of the hypergraph *H,* wherein if the $(x, y) \in E$ and $x, y \in W$, then $(x, y) \in U$. Hypergraph cohesion component will be called the set of its vertices, such as any of two of its elements there is a path between them, but there is no path leading from the vertex of belonging to this collection to any other vertex outside. If there is in the subhypergraph $H_1 = (V_1, E_1)$ of hypergraph *H* consistency of each component there is a star with center at some vertex $v \in V_1$, the H_1we will call the coverage of hypergraph *H* stars.

Fixed design task was to find such a connection structure that will ensure the maximization or minimization of operating parameters, such as communication delay, errors in access to the communication medium as a result of his occupation, performance computational processing nodes and others. Such a task can be solved by seeking the cover at least the stars of trigeminal graph. Let's consider the task.

Input data. As an input data we use the design task:

1. $B = \{b\}$ - A set of logical communication bus dedicated by physical channel under communication using any of the methods of reproduction communication;
2. $F = \{f\}$ - A set of access protocols, which describes the functioning logical BUS communication [12];
3. $N = \{n\}$ - A set of customers of the system, using BUS communication channels.

Customers are divided into groups $d \in D$ taking their communication requirements into account, where $D = \{d\}$ - is a set of types of communication requirements. Elements of the set *D* shall be as follows: $d = 0$ – service streams sensitive to errors and latency; $d = 1$ – support for latency-sensitive flows; $d = 2$ – service streams are sensitive to transmission errors; $d = 3$ – handling sensitive streams not being sensitive to these factors.

The definition of design task. Each of the compute nodes $n \in N$ should be assigned a set of *M* bus *b,* which is a subset of *B,* ie. $M \subseteq B$, each of which will operate on the basis of one of the access protocols $f \in F$.

The mathematical model. The task design is iteratively solved. In each of the steps and for each of the compute nodes there is sought one bus $b \in B$ providing for node communication services using Access Protocol $f \in F$. To avoid multiple of includes a node on the same bus at each successive step from the available buses the client is excluded from those for whom it has already been connected.

The mathematical model is based on the 3-partite hypergraph $H = (V, E) = (X, Y, Z, E)$. Busses from the set $B = \{b\}$ correspond to the vertices of the first part *x* $(x \in X)$. Each of them (at the same time each logical bus) is assigned to the label $\eta\,(x)$

determining the transmission characteristics of the bus, in the simplest case, the number of nodes measured, that it can handle.

The *f* elements of the set *F* bus access protocols correspond to the vertices *y* of the second part of hypergraph *H* $(y \in Y)$, and the elements *n* from *N* compute nodes correspond to the vertices $z$of the third part of hypergraph $(z \in Z)$.The set of edges $E = \{e\}$ includes all three vertices (x, y, z)such that $x \in X\_$, $y \in Y$, $z \in Z$. There are permitted only those edges, for which a selected bus the client can handle $b_i \in X$ using a communication protocol $f_l \in Y_2$. Collection $E = \{e\}$ of all edges is determined essentially by a set of all admissible triples $e = (x, y, z)$. Taking account of the value of the parameter $n(x)$ for $x \in X$in hypergraph $H = (V, E) = (X, Y, Z, E)$ permissible step design task solution will be any of his sub hypergraph $\beta = (V_\beta, E_\beta)$ for$V_\beta \subseteq V$ and $E_\beta \subseteq E$of which each component represents the simple consistency star of stage with the center in the apex $x \in X$. As $S = S(H) = \{s\}$we denote the set of all feasible solutions of tasks covering hypergraph *H* stars.

Each of edges $e \in E$ of hypergraph $H = (V, E)$ there is assigned three scales describing the following characteristics solutions:

1. $\omega(e) = \phi(x, y, z)$ - Expected customer conversion processing performance in a system in which the client is supported in communication *with* the bus *X,* which uses a communication protocol *y.* To evaluate the performance characteristics of the proposed process there were used in [13], [14], [15]. Described measures therein are modified so that they reflect the change in performance which are due to changes in the architecture of calls. Because of the nature of the bus network connecting the primary measure of performance is the number of nodes, for which there is available transportation network [16].
2. $\xi(e) = \phi(x, y, z)$ - Expected change in the communication delay on demand of the customer for these conditions. The level of changes in the delay is determined on the basis of the stochastic model using the method described in [17].
3. $\psi(e) = \phi(x, y, z)$ - Expected change in system reliability for client-server preserved the conditions of point. 1. To determine the reliability of the method was applied changes presented in §2.

Solution design task. The rating of the solutions will be shown as a multi-criteria. The proposed set of criteria is obviously exemplary, and his selection depends on the needs of the designer, in particular concerning the nature of the future operation of the network merger. For the case of under consideration there are the three functions described below.

Let's consider the set $A = \{a\}$ of acceptable solutions of design task. For each of them we define the following characteristics assessing the quality of solutions:

1. Criterion performance computing: $\Phi_1(a) = \max_{a \in A} \min_{e \in E_a} \omega(e)$, where: E_a– sub of edge of hypergraph *H* belonging to solution *a.* Using this criterion, we strive to maximize the minimum level of performance (computing or communication) system.
2. Communication delay criterion: $\Phi_2(a) = \min \sum_{e \in E_a} \xi(e)$, which provides network search junction with minimal delay summary. For systems with varying levels

of validity of nodes, the value $\xi(e)$ of expected changes in the communication delay is called using vertex priority.

3. The criterion of reliability: $\Phi_3(a) = \max \sum_{e \in E_a} \psi(e)$. This criterion provides a network architecture search for which the total reliability is maximum likewise in the case of a delay criterion communication Φ_2.

The possibilities of the above method is not limited to the application of the summation as a *min* or *max.* To assess the quality of solutions it can be used any of the method of folding (convolution) parameters, including methods taking the weight of each sub-parameters into account. These sub-criteria are related by a function to form $\Phi(a) = (\Phi_1(a), \Phi_2(a), \Phi_3(a))$. Multi-criteria objective function $\Phi(a)$defines a set A feasible solutions, a set of Pareto A^pcomposed of Pareto optima a^p. If two solutions $a_1, a_2 \in A$ vector objective function $\Phi(a)$ are equivalent, then the set of A^p is secreted full set of alternatives A^A, which is, in fact, a maximum system vectorially different optima Pareto.

4 The Research, Results and Further Work

The work approach used to create a multi-channel design methodology destined for fieldbus communication service systems, client-server computing. The existing methodology is focused on providing definite level of computing capacity of the entire system, regardless of its reliability parameters. In the presented in the work version, methodology seeks the optimal solution with respect to multi-criteria objective function that one of the criteria is reliability. Depending on how sub-criteria ties and a set of restrictions proposed methodology also allows you to specify the connection architecture, characterized by: **a.** maximum reliability with certain: the minimum efficiency and maximum delay network communication links; **b.** the minimum communication delay with the reduction in the minimum reliability and performance; **c**. maximum computing performance of a specified acceptable level of reliability and delays. There was also tested solution, the aim of which was to design load leveling various communication buses. For each of the solutions sought there are limited the maximum cost of construction. We analyzed the network of connections complete and partial, flat and hierarchical.

Simulation studies of obtained architectures for computing model client-server based on the methodology presented in [16] showed that the use of multi-channel communication systems can flexibly adapt to the current needs of the computing system. Increasing performance computing system with unchanging resources, obtained by reconfiguring its connections reached 260%. Deviations in terms of the burden of communication channels does not exceed 31% and the probability of rejecting a service request has fallen nearly 8-fold. The algorithm of the interconnection network is characterized by polynomial time complexity, which can react in real time to any change in traffic patterns.

Further research will focus on finding effective methods of searching for coverage of celebrities simple multipartite hypergraph, which will allow the use of graphs in the design process of any of valor, and this will introduce further design criteria.

References

1. T. Yu, Y. Zhang, and K. J. Lin, "Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints," *ACM Transaction on The Web*, vol. 1, no. 1, May 2007.
2. M. R. Garey and D. S. Johnson, *Computers and Interactability. The Guide to the Theory of NP-Completeness*. San Francisco: W.H. Freeman & Company, 1979.
3. H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. Berlin: Springer, 2004.
4. C. Chekuri and O. Hudry, "Optimal clustering of multipartite graphs," *Discr. Appl. Math.*, vol. 156, no. 8, pp. 1330-1347, 2008.
5. M. Dawande, P. Keskinocak, and J. M. Swaminathan, "On bipartite and multipartite clique problems," *Journal of Algorithms*, vol. 41, no. 2, pp. 388-403, 2001.
6. A. Scarelli and S. C. Narulla, "A multicriteria assignment problem," *Journal of Multi-Criteria Decision Analysis*, vol. 11, no. 2, pp. 65-74, 2002.
7. E. Cela, *The Quadratic Assignment Problem*. Dordrecht: Kluwer Academic Publishers, 1998.
8. T. Oncan, "A survey on the generalized assignment problem," *INFOR*, vol. 45, no. 3, pp. 123-141, 2007.
9. A. K. Jain, M. N. Murty, and P.J. Flynn, "Data clustering: a review.," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264-323, 1992.
10. N. Jardine and R. Sibson, *Mathematical Taxonomy*. London: John Wiley & Sons, 1971.
11. S. Mishin, "Optimal organizational hierarchies in firms.," *J. of Business Economics and Management*, vol. 8, no. 2, pp. 79-99, 2007.
12. R. Dutta, A. E. Kamal, and A. E. Rouskas, *Traffic Grooming for Optical Networks*.: Springer, 2009.
13. B. R. Haverkort, *Performance of computer communication systems : a model-based approach*. Chichester: John Wiley & Sons Ltd, 1999.
14. M. Hajder, H. Loutskii, and W. Streciwilk, *Informatyka. Wirtualna podroz w swiat systemow i sieci komuterowych*. Rzeszow: Wydawnictwo Wyzszej Szkoşy Informatyki i Zarzadzania w Rzeszowie, 2002.
15. S. Sahni and V. Thanvantri, "Parallel Computing: Performance Metrics and Models," *IEEE Parallel and Distributed Technology*, vol. 4, pp. 43-56, Jan. 1996.
16. M. Hajder, "Improving the efficiency of multi-channel backbones client-server," Visnyk of National Technical University of Ukraine, vol. 37, pp. 37-48, 2002.
17. M. Hajder and M. Kielbus, "Matematyczny model opoznien w sieci z komutacja pakietow," in *XV Konferencja Sieci i Systemy Informatyczne*, Lodz, 2007, pp. 47-50.