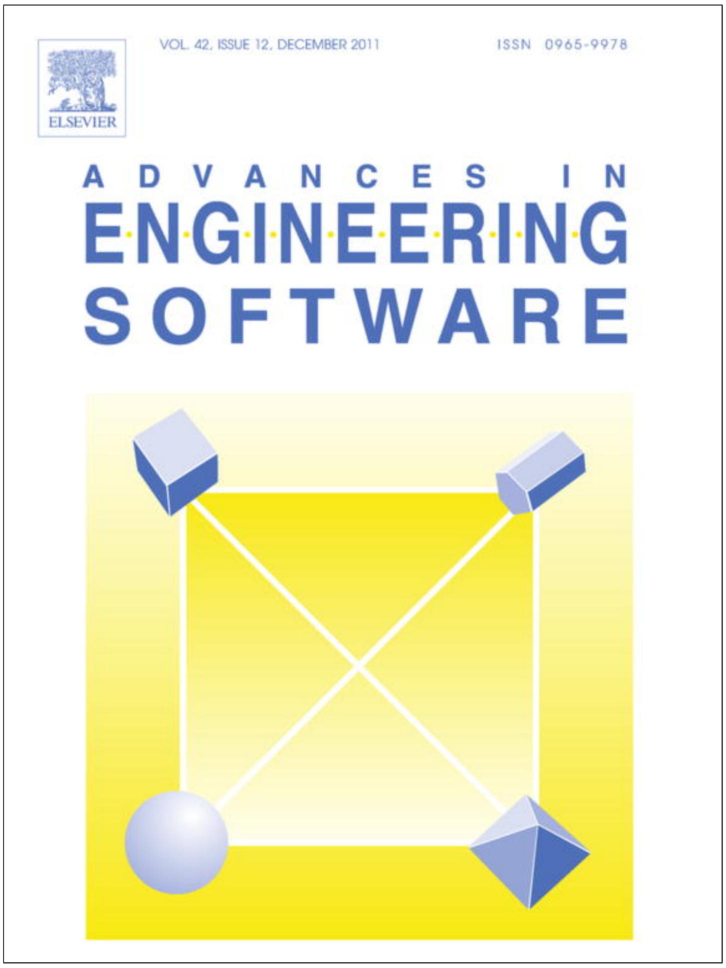
VOL. 42, ISSUE 12, DECEMBER 2011
ISSN 0965-9978
ELSEVIER
ADVANCES IN
ENGINEERING
SOFTWARE

Advances in Engineering Software 42 (2011) 1089–1098

Contents lists available at SciVerse ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Four-layer framework for combinatorial optimization problems domain ☆

Mark Sh. Levin

Inst. for Information Transmission Problems, Russian Academy of Sciences, 19 Bolshoj Karetny Lane, Moscow 127994, Russia

ARTICLE INFO

Article history:
Received 7 November 2010
Received in revised form 18 July 2011
Accepted 24 July 2011
Available online 23 August 2011

Keywords:
Problem solving environment
System architecture
Combinatorial optimization
Decision making
Problem structuring
System design

ABSTRACT

Four-layer framework for combinatorial optimization problems/models domain is suggested for applied problems structuring and solving: (1) basic combinatorial models and multicriteria decision making problems (e.g., clustering, knapsack problem, multiple choice problem, multicriteria ranking, assignment/allocation); (2) composite models/procedures (e.g., multicriteria combinatorial problems, morphological clique problem); (3) basic (standard) solving frameworks, e.g.: (i) Hierarchical Morphological Multicriteria Design (HMMD) (ranking, combinatorial synthesis based on morphological clique problem), (ii) multi-stage design (two-level HMMD), (iii) special multi-stage composite framework (clustering, assignment/location, multiple choice problem); and (4) domain-oriented solving frameworks, e.g.: (a) design of modular software, (b) design of test inputs for multi-function system testing, (c) combinatorial planning of medical treatment, (d) design and improvement of communication network topology, (e) multi-stage framework for information retrieval, (f) combinatorial evolution and forecasting of software, devices. The multi-layer approach covers 'decision cycle', i.e., problem statement, models, algorithms/procedures, solving schemes, decisions, decision analysis and improvement.

1. Introduction

Multi-layer approaches to computers and networks are well-known (e.g., [105,106]). In this article, a multi-layer approach is used for structuring and solving of applied modular problems. The significance of model-based problem solving environments, problem solving engines, and decision support systems has been increased (e.g., [1,29,34,51,77,87,91,104,109]). This is crucial in computer science, engineering design, mathematics, management, etc. In addition, it is reasonable to point out increasing importance of the stage of problem formulation/statement/structuring (e.g., [6,16,19,24,33,38,79,81,92,97,100,102]). Note the issues of problem formulation/statement/structuring are important for well-structured, ill-structured, and unstructured problems (e.g., [39,99–101,108]).

In the case of modular applied systems, examination processes (system analysis and modeling, problem structuring/solving) are based on analysis and modeling of combinatorial objects and relations over the objects. Fig. 1 depicts a "combinatorial world" frame. Here the basic kinds of combinatorial objects (or structural forms [42]) are as follows (e.g., [2,30,88,114]): element (item), set, multiset, chain (string), list, partition, order, star, ring, tree, parallel-series graph, hierarchy (and "layered structure", "cylinder"), grid, multigraph, hypergraph, etc.

Combinatorics is concerned with the study of arrangement, patterns, designs, assignment, schedules, connections, and configurations [93]. Thus, combinatorial optimization problems are targeted to the following basic goals (and their combinations): partitioning, routing, scheduling, assignment, location, placement, covering, packing, etc. (e.g., [22,26,30]). The second group of basic combinatorial problems consists of discrete multicriteria decision making problems: (a) choice of the best alternative, (b) ranking (e.g., linear ordering, group ranking to get a linear ordered alternative subsets), and (c) clustering (e.g., [36,46,83,94,117]). The third kind of considered combinatorial problems is targeted to design/approximation of structures (e.g., graphs, networks, hierarchies), for example: (i) direct design of a structure (e.g., an "optimal" tree), (ii) addition of edges/links and/or nodes (e.g., hot link assignment problems, graph augmentation problems), (iii) spanning problems (e.g., minimum spanning tree for a graph, minimum spanning Steiner tree, a "minimum" spanning k-connected structure) (e.g., [15,20,23,25,28,30,88,110–112]).

In general, there exist two basic approaches to modular system modeling: (1) usage of complex models (e.g., finite state machines, Petri nets), (2) combination of some more simple problems/models. Here the second approach above is used. Subproblems of several kinds are used as follows (e.g., [33,51,99–101]): (i) formalized, (ii) well-structured, and (iii) ill-structured (e.g., multicriteria ranking, expert judgment), and (iv) unstructured. Thus, a result of an applied problem analysis can be considered as a problem frame (or multi-problem scheme, problem "space") (e.g., [33,52,60,61,104]).

☆ Published draft: M.Sh. Levin, Towards four-layer framework of combinatorial problems. IEEE COMPSAC 2008, 873–878, 2008.

E-mail address: mslevin@acm.org

0965-9978/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2011.07.013

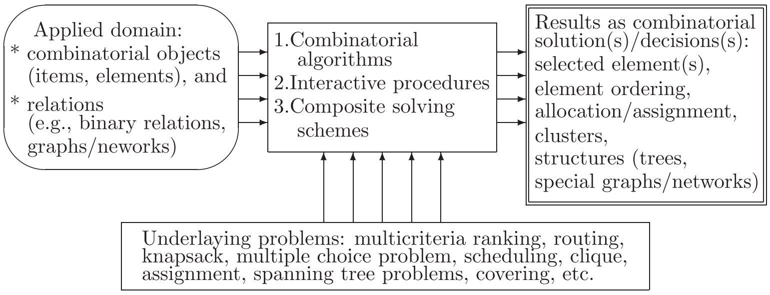

Fig. 1. Illustration for "combinatorial world".

Mainly, our four-layer composite framework is based on combinatorial optimization and decision making problems/models examined by the author (e.g., [46,51,56,58]). The framework is targeted to modular approaches in integrated multi-disciplinary applied domains. Concurrently, this multi-layer architecture can be useful for education/training as well ([55,60,61]). Further, the considered approach may be useful from the viewpoint of computational thinking [115].

2. Basic problems/models

Multicriteria ranking can be considered as follows. Let $\Omega = \{1,\ldots,i,\ldots,\lambda\}$ be a set of items (alternatives) which are evaluated upon criteria $\Upsilon = \{1,\ldots,j,\ldots,\delta\}$ and $z_{i,j}$ is an estimate (quantitative, ordinal) of item i on criterion j. The matrix $\{z_{i,j}\}$ is a basis to build a partial order on Ω, for example, through the following generalized scheme: (a) pairwise elements comparison to get a preference (and/or incomparability, equivalence) binary relation, (b) building a partial order on Ω. Here the following partial order (partition) as linear ordered subsets of Ω is searched for (Fig. 2):

$\Omega = \bigcup_{k=1}^{m}\Omega(k), |\Omega(k_1)\cap\Omega(k_2)| = 0$ if $k_1 \neq k_2$, $i_2 \preceq i_1\ \forall i_1 \in \Omega(k_1)$, $\forall i_2 \in \Omega(k_2)$, $k_1 \leqslant k_2$ (sorting problem) (e.g., [46,94,117,118]). Set $\Omega(k)$ is called layer k, each item $i \in \Omega$ gets priority r_i that equals the number of the corresponding layer. This problem belongs to a class of ill-structured problems by classification of Simon [101]. The list of basic techniques for multicriteria selection (sorting problem) is the following (e.g., [8,117,118]): (1) multi-attribute utility analysis, (2) multi-criterion decision making based on interactive procedures, (3) Analytic Hierarchy Process (AHP), and (4) outranking techniques.

Clustering problem is a basic scientific problem in many domains (e.g., [36,83,84]):

Divide an initial set of elements into groups (*subsets, clusters*) *to minimize the* "*distances*" (*or proximities*) *between elements in the clusters* (*i.e.*, "*intercluster distances*").

The following data can be used as initial information: (a) parameters of each element and/or (b) proximity ("distances") between elements. Basic clustering algorithms are described in [36]. Often polynomial heuristics are used (e.g., agglomerative algorithm). In our study, our modification of agglomerative algorithm is mainly used [53].

The basic *point-to-point shortest path problem* is the following (e.g., [12,30]). Given a directed connected weighted graph $G = (A,E)$ (A is the set of vertices/nodes, E is the set of arcs, there is a nonnegative weight $w(e)\ \forall e \in E$, two vertices are specified as source $a \in A$ and destination $b \in A$). The problem is:

Find a shortest (minimal) path from a *to* b*, where the total length of path* $\langle a,b\rangle$ *corresponds to the sum of arc weights in the path.*

The standard algorithm for this problem is the one developed by Dijkstra (e.g., [30]) which runs in $O(|E| + |A|\log|A|)$ [27].

The basic *knapsack problem* is ([30,41]):

$$\max\sum_{i=1}^{m} c_i x_i \text{ s.t. } \sum_{i=1}^{m} a_i x_i \leqslant b, \quad x_i \in \{0,1\},$$

where $x_i = 1$ if item (element) i is selected, c_i is a value ("utility") for item i, and a_i is a weight (or required resource). Often nonnegative coefficients are assumed. The problem is NP-hard [30]. In the case of *multiple choice problem* (Fig. 3), the items are divided into groups and it is necessary to select elements (items) from each group while taking into account a total resource constraint (or constraints):

$$\max\sum_{i=1}^{m}\sum_{j=1}^{q_i} c_{ij}x_{ij} \text{ s.t. } \sum_{i=1}^{m}\sum_{j=1}^{q_i} a_{ij}x_{ij} \leqslant b, \quad \sum_{j=1}^{q_i} x_{ij} = 1\ \forall i = \overline{1,m}, x_{ij} \in \{0,1\}.$$

Assignment/allocation problems are widely used in many domains (e.g., [3,10,11,13,21,30,85,89]). Simple *assignment problem* involves correspondence matrix $A = \|a_{ij}\|(i = \overline{1,n};\ j = \overline{1,n})$, where a_{ij} is a profit to assign element i to position j. The problem is (e.g., [30]):

Find the assignment $\pi = (\pi(1),\ldots,\pi(n))$ *of elements to positions which corresponds to a total effectiveness:* $\sum_i a_{i\pi(i)} \rightarrow \max$.

The problem can be solved efficiently. More complicated well-known model as *quadratic assignment problem* (QAP) includes interconnection between elements of different groups (each group corresponds to a certain position) (e.g., [13,22,89,90]). Let a nonnegative value $d(i,j_1,k,j_2)$ be a profit of compatibility between item j_1 in group J_i and item j_2 in group J_k. Also, this value of compatibility is added to the objective function. Thus, QAP (NP-hard) is:

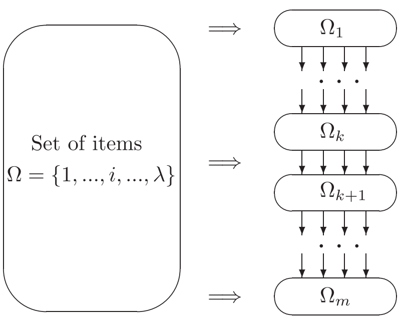

Fig. 2. Scheme of multicriteria ranking.

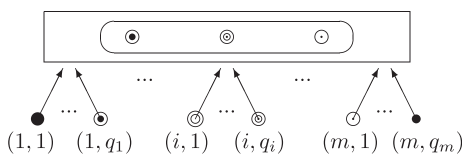

Fig. 3. Illustration for multiple choice problem.

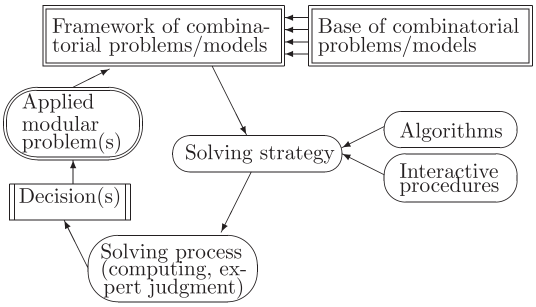

Fig. 4. 'Decision cycle' and support components.

$$\max \sum_{i=1}^{m} \sum_{j=1}^{q_i} c_{i,j} x_{i,j} + \sum_{l<k} \sum_{j_1=1}^{q_l} \sum_{j_2=1}^{q_k} d(l, j_1, k, j_2) x_{l,j_1} x_{k,j_2},$$
$$l = \overline{1,m}, \; k = \overline{1,m}$$
$$\text{s.t.} \sum_{i=1}^{m} \sum_{j=1}^{q_i} a_{i,j} x_{i,j} \leqslant b, \quad \sum_{j=1}^{q_i} x_{i,j} \leqslant 1 \; \forall i = \overline{1,m}, \; x_{i,j} \in \{0,1\}.$$

In addition, it is reasonable to point out other basic combinatorial optimization problems: routing, scheduling, traveling salesman problem (TSP), clique, independent set problem, packing problems, matching, augmentation problems, the longest path problem, hotlink assignment problems, graph coloring, etc. (e.g., [2,14,20,22,25,26,28,30,37,40,44,86,88]).

3. Decision cycle, four-layer framework

Generally, our approach is based on 'decision cycle' (Fig. 4) [51] with orientation to stages as follows: (a) analysis of the application problem(s) (i.e., applied domain), (b) construction of problem/model frame ("space"). Composite (modular) solving schemes (strategies) can be obtained concurrently as an integration of algorithms and procedures for the selected problems/models.

Our four-layer model/problem framework consists of the following layers (Fig. 5):

Layer 1. Basic combinatorial models and multicriteria DM-models: multicriteria ranking, clustering, knapsack problem, multiple choice problem, shortest path problem, clique problem, assignment/allocation, traveling salesman problem (TSP), graph approximation, etc.
Layer 2. Composite models/procedures, mainly, as multicriteria problem versions (clustering, knapsack problem, multiple choice problem, combinatorial synthesis based on morphological clique problem or multipartite clique problem, assignment/allocation, etc.).
Layer 3. Standard solving frameworks, e.g.: (i) Hierarchical Morphological Multicriteria Design (HMMD) (ranking, combinatorial synthesis) to design modular systems, (ii) design of a modular solving strategy (e.g., a partitioning/synthesis heuristic for complicated problems), (iii) system upgrade, (iv) multi-stage design (two-level HMMD), and (v) system evolution/forecasting, and (vi) special multi-stage composite framework (clustering, assignment/location, multiple choice problem).
Layer 4. Domain-oriented solving framework, e.g.: (a) design of modular software, (b) information search/retrieval, (c) planning a marketing strategy, and (d) network design/extension/imrovement.

Evidently, each element at layer $k (k = \overline{2,4})$ is based on (consists in) a combination of elements at layer $(k-1)$.

In addition, it is reasonable to point out several *basic conceptual operations*:

1. decreasing a problem dimension: *clustering*;
2. analysis and revelation of the most important elements (e.g., system parts, components): *selection/ranking*, *knapsack problem*;
3. simplification of a problem, for example: *spanning problem*, *graph approximation*;
4. composition (combination, integration, synthesis): *multiple choice problem*, *morphological clique problem*, *multipartite clique problem*, *shortest path problem*; and
5. planning: *scheduling*.

4. Composite models

Composite problems/models are mainly considered as multicriteria problem formulations.

In *multi-objective (multicriteria) shortest path problem* several parameters are associated with each arc. Here Pareto-efficient (nondominated) paths are searched for. The survey on

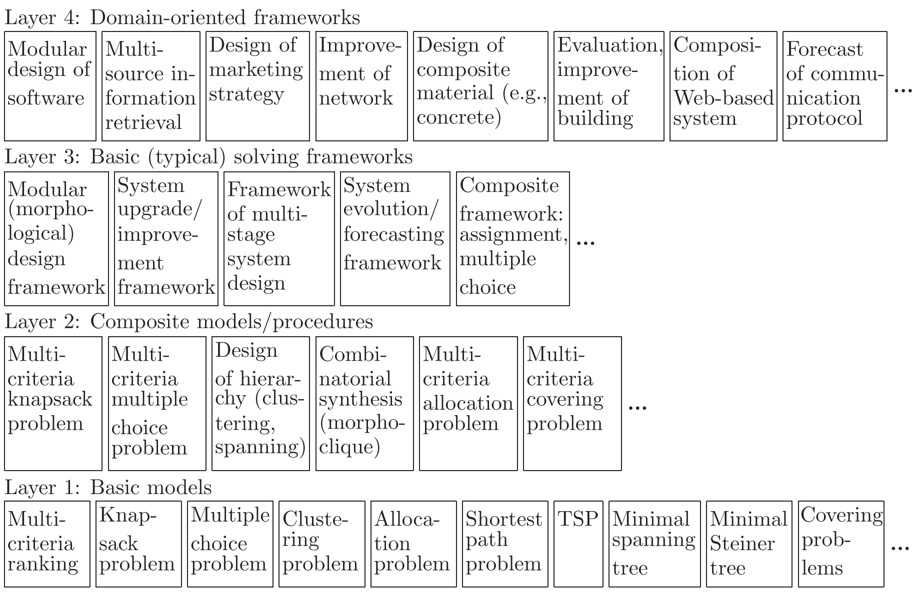

Fig. 5. Four-layer framework.

the multi-objective shortest path problems and algorithms are contained in [107]. The multicriteria shortest path problem is NP-hard (even in the bicriteria case) (e.g., [30]). The following algorithms are mainly used: (i) multicriteria Dijkstra's algorithm (label setting algorithm), (ii) multicriteria Ford-Bellman's algorithm, (iii) dynamic programming, (iv) fast approximation schemes, (v) genetic algorithms, and (vi) heuristics.

In *multi-objective multiple choice problem* multiple criteria description $\{c_{i,j}\}$ (i.e., $\forall(i,j)$) is used and vector objective function $(f^1, \ldots f^r)$ is ([73,103]):

$$\left(\max \sum_{i=1}^{m} \sum_{j=1}^{q_i} c_{ij}^1 x_{ij}, \ldots, \max \sum_{i=1}^{m} \sum_{j=1}^{q_i} c_{ij}^p x_{ij}, \ldots, \max \sum_{i=1}^{m} \sum_{j=1}^{q_i} c_{ij}^r x_{ij} \right).$$

Evidently, here it is necessary to search for Pareto-efficient (by the vector objective function above) solutions. The following solving schemes can be used (e.g., [73,103]): (i) heuristic based on multicriteria ranking of elements and step-by-step packing the knapsack, (ii) multicriteria ranking of elements to get their ordinal priorities and usage of approximate solving scheme (as for knapsack problem) based on discrete space of system quality (as for morphological clique problem), (iii) enumerative methods, and (iv) evolutionary algorithms.

The problem *design of a hierarchy* is a crucial one and can be based on hierarchical clustering of some initial system elements ('Bottom–Up' approach, Fig. 6) (e.g., [36,53]). In addition, it is possible to point out other approaches to this problem, for example (e.g., [56]):

(a) expert judgment as a serious partitioning of a system into subsystems;
(b) minimum spanning tree problems (e.g., [15,23,30,88,111]);
(c) multicriteria spanning Steiner tree problem (e.g., [75]); and
(d) usage of optimization models (e.g., linear programming) to design an "optimal" hierarchy (e.g., [110]) and other structure synthesis techniques (e.g., [112]).

Fig. 7 illustrates spanning tree problems.

Our combinatorial synthesis is based on *morphological clique problem* that has a standard brief description (e.g., [46,51]). The examined composite (modular, decomposable) system consists of components and their interconnection (IC) or compatibility. Basic assumptions are the following: (a) the system has a tree-like structure (generally, it is *morphological tree* model [52]); (b) a composite estimate for system quality is considered as integration of components (subsystems, parts) qualities and qualities of IC (compatibility) across subsystems; (c) monotonic criteria for the system and its components are examined; and (d) quality of system components and quality of IC are evaluated upon coordinated ordinal scales. The designations are: (1) design alternatives (DAs) for nodes of the model; (2) priorities of DAs ($r = \overline{1,k}$; 1 corresponds to the best one); (3) ordinal compatibility (IC) for each pair of DAs ($w = \overline{1,l}$; l corresponds to the best one). Let S be a system consisting of m parts (components): $P(1), \ldots, P(i), \ldots, P(m)$. A set of design alternatives is generated for each system part above. The problem is:

Find a composite design alternative $S = S(1)\star \ldots \star S(i)\star \ldots \star S(m)$ *of* DAs *(one representative design alternative* $S(i)$ *for each system component/part* $P(i), i = \overline{1,m}$) *with non-zero* IC *between design alternatives.*

A discrete space of the system excellence is based on the following vector (Fig. 8): $\overline{N}(S) = (w(S); \bar{n}(S))$, where $w(S)$ is the minimum of pairwise compatibility between DAs which correspond to different system components (i.e., $\forall P_{j_1}$ and $P_{j_2}, 1 \leqslant j_1 \neq j_2 \leqslant m$) in $S, \bar{n}(S) = (n_1, \ldots, n_r, \ldots n_k)$, where n_r is the number of DAs of the rth quality in S. As a result, we search for composite decisions which are nondominated by $\overline{N}(S)$.

The described problem is NP-hard [43]. Clearly, the compatibility component of vector $\overline{N}(S)$ can be considered on the basis of a poset-like scale (as $\bar{n}(S)$) as well. In this case, the discrete space of system excellence will be an analogical lattice [51]. The solving process can be based on two strategies [46]: (1) enumerative method, (2) dynamic programming. Note combinatorial synthesis can be based on versions of multipartite clique problem as well [18]. Figs. 8–11 illustrate the composition problem. In the numer-

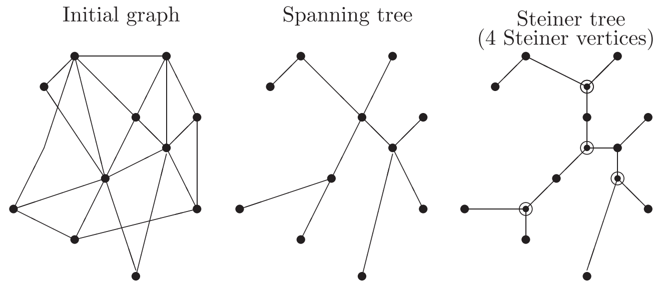

Fig. 7. Illustration for spanning trees.

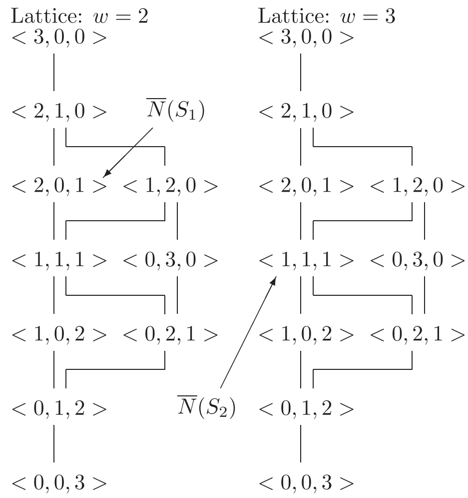

Fig. 8. Illustration of system quality space.

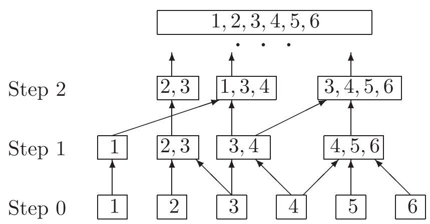

Fig. 6. Design of hierarchy by agglomerative algorithm.

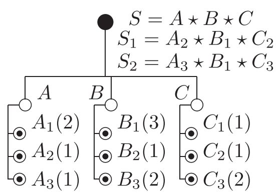

Fig. 9. Example of composition.

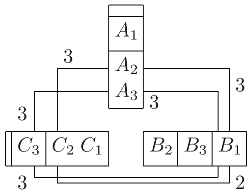

Fig. 10. Concentric presentation.

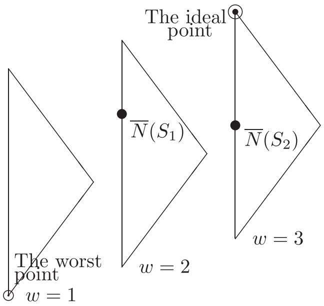

Fig. 11. Space of system quality.

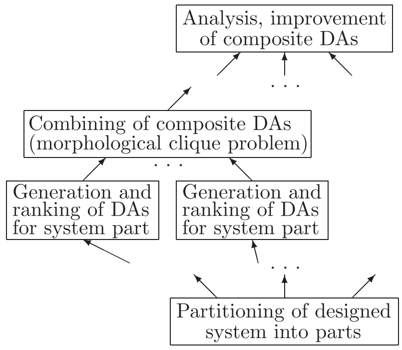

Fig. 12. 'Cascade-like' scheme.

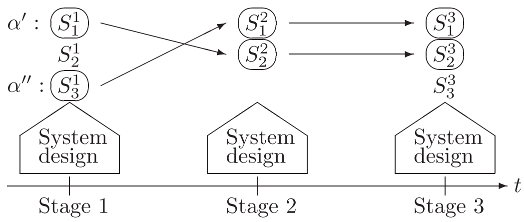

Fig. 13. Illustration for multi-stage (trajectory) design.

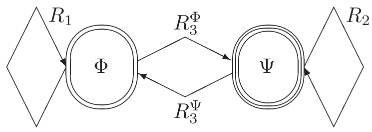

Fig. 14. Allocation over binary relations.

ical example (DAs priorities are shown in parentheses in Fig. 9, compatibility estimates are presented in Fig. 10), composite decisions are: $S_1 = A_2 \star B_1 \star C_2, \overline{N}(S_1) = (2; 2, 0, 1)$; $S_2 = A_3 \star B_1 \star C_3, \overline{N}(S_2) = (3; 1, 1, 1)$.

In our multi-layer framework, *multicriteria assignment problem* (and multicriteria *QAP* too) corresponds to the layer of composite models/procedures as well. Here the objective function is transformed into a vector function (e.g., [17,71,95,96]): $c_{i,j} \Rightarrow \overline{c_{i,j}} = \left(c^1_{i,j}, \ldots, c^r_{i,j}\right)$. Several solving approaches can be used to search for Pareto-efficient solutions, for example: (1) enumerative methods, (2) interactive methods, and (3) heuristics. From the practical viewpoint, the following generalizations of the assignment/allocation problems are prospective ones: (i) improvement of allocation solutions, (ii) re-allocation, (iii) extension of allocation solutions, and (iv) multi-stage allocation (or dynamical allocation, multidimensional assignment).

5. Basic solving frameworks

Our approach *Hierarchical Morphological Multicriteria Design* (HMMD) is described in ([46,51]). HMMD is based on *morphological clique problem*. A basic version of HMMD involves the following phases: (1) design of a tree-like system model; (2) generation of DAs for leaf nodes of the model; (3) hierarchical selection and combining of DAs into composite DAs for the corresponding higher level of system hierarchy; and (4) analysis and improvement of composite DAs (composite decisions). Fig. 12 illustrates the corresponding *cascade-like* design framework.

The *multi-stage design* approach (or *system trajectory design*) is based on HMMD ([46,51]) (Fig. 13). The *multi-stage design* process consists of the following: (1) structuring of stages; (2) combining of composite decisions for each stage (bottom hierarchical level, usage of HMMD); and (3) combining a multi-stage decision trajectory (up-level of the hierarchy, usage of HMMD). In Fig. 13, illustrative system trajectories are: $\alpha' = \langle S_1^1 \rightarrow S_2^2 \rightarrow S_2^3 \rangle$ and $\alpha'' = \langle S_3^1 \rightarrow S_1^2 \rightarrow S_1^3 \rangle$.

Allocation over binary relations scheme was described in [47] (Fig. 14). Here allocation is examined as mapping of a set of elements (Φ) into a set of *positions* (Ψ) [47]: $\Phi \Rightarrow \Psi$ while taking into account binary relations. The following binary relations are used: (a) "*proximity*": on ($\Phi \times \Phi$): R_1, on ($\Psi \times \Psi$): R_2; (b) "*correspondence*": on ($\Phi \rightarrow \Psi$): R_3^Φ, and on ($D \rightarrow \Phi$): R_3^Ψ. The problem is:

Find mapping (allocation) $X{:}\Phi \Rightarrow D$ *while taking into account the following*: (i) *the best realized* "*correspondence*" and (ii) *saving* "*proximity*" R_1 *on* R_2.

Evidently, the central problem formulation issue is the following: measurement of the above-mentioned *correspondence* and "*proximity*". This kind of allocation problem can be reduced to morphological design based on *morphological clique problem* (usage of HMMD) ([46,56]).

The solving process of some graph-based combinatorial problems (e.g., traveling salesman problem, Steiner tree problem) may be based on *partitioning/synthesis macroheuristic* (i.e., "divide and conquer" strategy) [46]: (1) clustering of the initial problem graph into subgraphs; (2) solving the problem for each obtained subgraph to get a set of solutions (as future design alternatives); and (3) composition of the global solution from the local solutions obtained at the previous stage (usage of HMMD).

Four-stage solving scheme ("2-sets and 4-problems framework") was suggested in ([57,66]). Here two interconnected sets of elements are examined and processed (Fig. 15): (i) *elements* and (2) *positions*. The solving scheme consists of the following stages: (1) clustering of *element* set (to decrease the dimension); (2) clustering of *position* set (to decrease the dimension); (3) assignment/allocation of *element* clusters into *position* clusters; and (4) selection for each "assignment" (i.e., pair *element* cluster-*position* cluster) an operation (action) from a specified set of operations (*multicriteria multiple choice problem* or *morphological clique problem*).

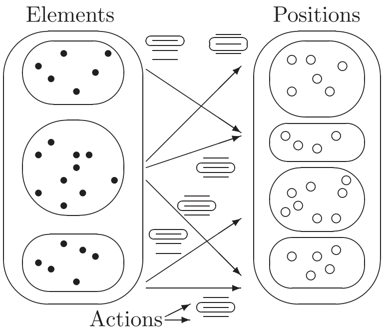

Fig. 15. Illustration for four-stage scheme.

In many applications, a solving strategy can be represented as a chain of information processing stages or a series–parallel graph of the stages (i.e., *series–parallel* solving strategy). In [46], a modular approach to multicriteria ranking corresponds to this situation. For each stage above, a set of local algorithms/procedures is generated (as design alternatives) including their multicriteria description and their compatibility. The design of the solving strategies is based on HMMD [46].

Our scheme for modeling of *combinatorial system evolution and forecasting* is [51]: (1) design of generalized system hierarchy, (2) design of the system hierarchy for each time stage (for each system generation); (3) definition of system changes (changes between neighbor system generations) as a set of change operations; (4) description of the system change operations: (i) multicriteria description, (ii) definition of some binary relations over the operations (e.g., relation of precedence or/and complementarity); and (5) formulation and solving of a forecasting problem as selection/composition of system change operations for the future (e.g., knapsack-like problems, morphological clique problem).

Design of k-connected network can be considered as well. Let $G = (A,E)$ be a graph (network), where A is a vertex (node) set, E is an edge set. Let us consider a special kind of k-connected network: (i) k "centers", where each "center" is $(k + 1)$-vertex clique, (ii) each other vertex has k edges (one edge for each "center"). Fig. 16 depicts 3-connected structure. A solving scheme to build k-connected structure is:

Stage 1. Selection of $k \times (k + 1)$ vertices for k "centers".
Stage 2. Clustering of the selected vertices to get k clusters as "centers" (each cluster consists of $k + 1$ vertices).
Stage 3. Connection of each other vertex with "centers": one connection for each "centers" (here multiple choice problem or its modifications can be used).

6. Domain-oriented solving frameworks

The significance of design problems for composition of software is increasing (e.g., [80,103]). The usage of HMMD for hierarchical *design of modular packaged software* is described in ([50,51]): (1) design of the system hierarchy, (2) generation of design alternatives (alternative software modules) for leaf node of the system model; and (3) 'Bottom–Up' design of the packaged software system. Analogically, this hierarchical design approach was used for other design applications: fusion of ordinal decisions ([48,51]), human–computer systems [49], composite material (e.g., concrete) [62], product life cycle ([51,62]), telemetry system [65], Web-hosting system [68], electronic shopping based on system composition [54], integrated security system [67], vibration conveyor [46], etc.

In [51], *combinatorial planning of medical treatment* is described as hierarchical combination of various treatment operations (medicine, physiotherapy, etc.). The process is based on HMMD.

Framework for evaluation and improvement of composite system is presented in ([51,63]). The framework is based on hierarchical system modeling, expert judgment, multicriteria ranking, combinatorial optimization models, hierarchical integration of ordinal decisions, modular design of a set of improvement actions (multiple choice problem or HMMD). A basic illustrative applied example is targeted to two-floor building ([51,63]). In ([46,50,51, 54,58,73,74]), other applications are described (e.g., information system, modular software, management system, car).

In [64], design of test inputs and their sequences in *multi-function system testing* is suggested (Fig. 17).

Combinatorial design of marketing strategy can be based on four-stage solving scheme (Fig. 15) [57]. Analogically, this approach was used in system testing planning [66] and in political marketing [69].

Network topology design and redesign problems are often crucial for network-like systems (communication, etc.). Topology design problems are often targeted to the following basic network kinds, for example (e.g., [32,82,116]): (i) hierarchical (multi-layer) networks and/or (ii) k-connected networks. Clearly, here the following approaches can be used: (a) direct design of hierarchies, (b) spanning structures (e.g., trees, etc.), (c) design of special kinds of structures (e.g., scheme above, Fig. 16).

The scheme for *improvement/upgrade of communication network* can be considered as follows ([73,76]): (1) clustering of network nodes (to decrease the problem dimension); (2) revelation of network bottlenecks for each network cluster (i.e., revelation of node(s), where it is necessary to improve electronic facilities), here multicriteria ranking model can be used; (3) generation of possible communication facilities for improvement/upgrade (expert judgment, multicriteria ranking); and (4) selection and assignment of

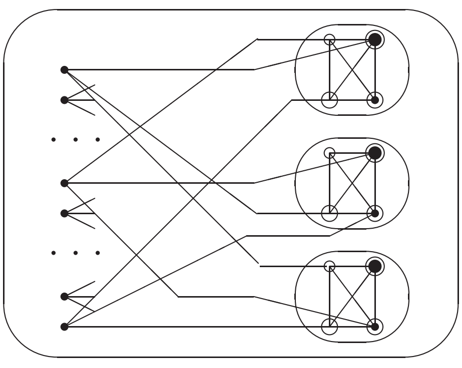

Fig. 16. Illustration for three-connected network.

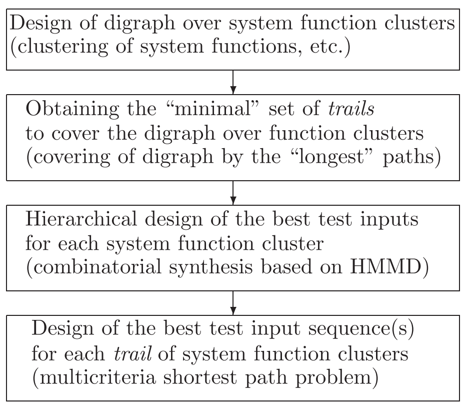

Fig. 17. Solving scheme for system testing.

communication facilities for each network bottleneck (multicriteria multiple choice problem). Generally network improvement problems can be based on the following basic actions (e.g., [58]): (i) addition and allocation of new nodes, (ii) improvement (upgrade) of old nodes, (iii) addition and allocation of new links, (iv) improvement (upgrade) of old links, and (v) building of a new topology (at various layers). Thus, the following solving schemes are examined (e.g., for an existing network topology):

Scheme 1: Addition of new nodes and/or links.
Scheme 2: Improvement of existing nodes and/or links.
Scheme 3: Integration of schemes 1 and 2.
Scheme 4: Design of a new network topology.

Note special combinatorial optimization problems have been examined for improvement/extension of communication networks or hierarchies, for example: hotlink assignment problems (e.g., [20,28]), graph augmentation problems (e.g., [25]).

The following approaches are considered for the above-mentioned solving schemes: (a) an engineering analysis, (b) multicriteria analysis and selection, (c) knapsack-like problems including multicriteria knapsack, (d) allocation problems (including basic assignment problem, quadratic assignment problem, some kinds of multicriteria assignment problems, etc.), (e) approximation/spanning/covering problems, and (f) morphological combinatorial synthesis problem.

A composite *multi-stage framework for multi-source information retrieval* is briefly described in ([45,46]) (Fig. 18): (1) design of a structural request consisting of several parts (components); (2) definition of correspondence between request components and data bases (information sources); (3) limitation of each data base for the search process (revelation of bounded "domains" in each data base); (4) searching for in each data base (in the bounded "domains"); (5) multicriteria selection of the most important data for each request component (with a resultant ordinal priority for each selected information item); (6) description of the request component ordinal compatibility (e.g., by some *keywords*); and (7) design/synthesis of the interconnected data as morphological clique or a "quasi clique" (components of the clique correspond to request parts).

A *framework for connection of users and access points* in last mile problem (e.g., communication networks, service networks) can be based on several combinatorial problems (e.g., [58,59,71]): multicriteria selection (connection alternative, access point), multicriteria knapsack problem, assignment/allocation, modular design of a connection system (multiple choice problem or HMMD).

Combinatorial evolution of standards for multimedia information is examined in [70] (Fig. 19): *MPEG-1* ⇒ *MPEG-2* ⇒ *MPEG-4* ⇒ Forecast. The scheme of analysis and forecasting is the following: (1) generalized structure of standard is considered as a morphological tree; (2) a set of change operations is generated as an integrated list of changes between *MPEG-1* and *MPEG-2*, *MPEG-2* and *MPEG-4*; (3) the set of change operations is evaluated upon a set of criteria and some binary relations (e.g., precedence, complementarity) are defined over the set; and (4) combinatorial problems (e.g., multicriteria ranking, clustering, multiple choice problem, morphological clique problem), which are based on the set of change operations above, are formulated and solved for forecasting. Similar combinatorial evolution approach has been used for modeling and forecasting of communication protocols ZigBee [72]. Here three ZigBee protocol generations were analyzed and three protocol forecasts were obtained: (i) expert judgment based forecast, (ii) knapsack problem based forecast, and (iii) multiple choice problem based forecast (Fig. 20). Analogically, combinatorial evolution approach was used for the following composite systems: DSS for multicriteria ranking 'COMBI-PC' [46], requirements to communication networks [52], and electronic devices [51].

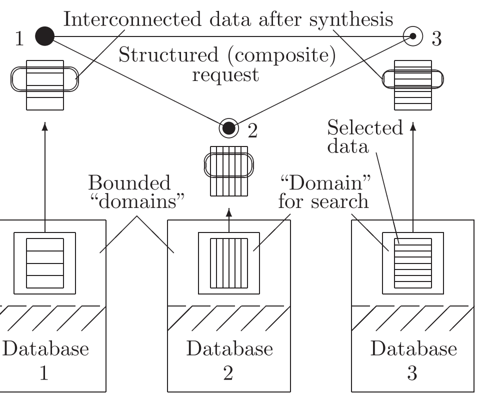

Fig. 18. Illustration for search, selection, synthesis.

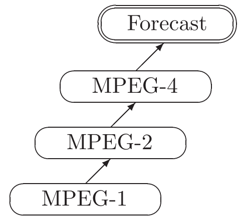

Fig. 19. Evolution of multimedia standard.

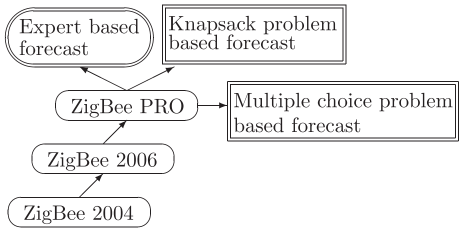

Fig. 20. Evolution of ZigBee protocol.

7. Systems architecture issues: brief discussion

Generally, a system architecture (systems architecture) defines the structure and/or behavior of a system (e.g., [9,31,51,78,98,113]). In recent years, the significance of systems architecture issues has been increased for various complex and multi-disciplinary systems. Evidently, modularity is a basic approach to decease complexity of system design and systems architecture problems (e.g., [4,5,7,9,35,51]).

On the other hand, it is necessary to take into account stages of system life cycle: design, redesign/improvement/modification, testing, maintenance, utilization, recycling. Thus, it is reasonable to examine composite systems architecture issues while taking into account system parts (software, mechanics, control, etc.) and stages of system life cycle.

It may be reasonable to consider the following types of systems (and corresponding systems architecture issues): (i) a one-discipline system (e.g., software engineering, electronics, information engineering, aerospace and mechanical engineering, civil engineering); (ii) a multi-disciplinary system including different parts (control, mechanics, software, algorithms, information, etc.). Note contemporary one-discipline systems involve various one-discipline parts.

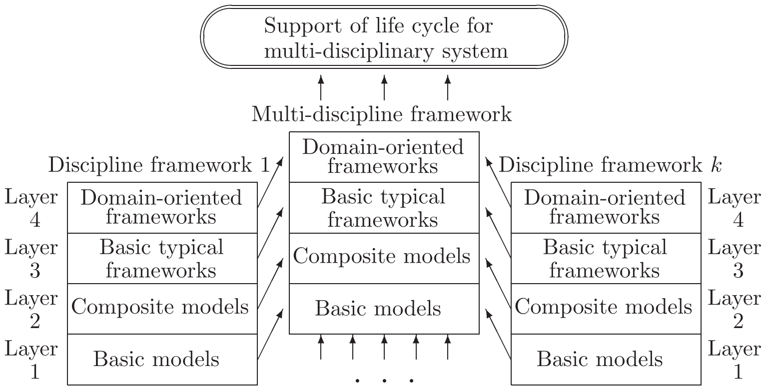

Fig. 21. Integration of multi-discipline framework.

In the paper, a composite framework for analysis, structuring, and solving of some combinatorial problems, which are useful at stages of system life cycle (design, testing, maintenance, etc.), has been described. Evidently, a real usage can require a combination of various problems (e.g., combinatorial optimization problems, continuous optimization problems, PDE based modeling, etc.). Here a four-layer framework for combinatorial problems domain is described. This four-layer framework can be considered as a composite component for an architecture to support a multi-disciplinary system (Fig. 21).

8. Conclusion

In the paper, a four-layer framework for combinatorial problems/models domain was suggested. The approach is targeted to applied problems structuring and solving. This is one of the first steps in the design of the solving environments of this kind.

In the future, it may be reasonable to consider the following research directions:

(I) Suggested four-layer framework for combinatorial optimization problems domain: (1.1) extension of the set of basic models (layer 1) and composite models (layer 2) by other significant combinatorial optimization problems (e.g., timetabling problems); (1.2) examination of dynamic and/or on-line problems (layers 1 and 2); (1.3) consideration of re-configuration combinatorial problems, e.g., re-assignment/re-allocation, re-scheduling, re-routing, re-coloring, re-packing, re-covering (layer 2); (1.4) usage of problems/models under uncertainty (e.g., probabilistic and/or fuzzy estimates) (layers 1 and 2); (1.5) implementation of the suggested architecture as a software tool or a decision support environment; (1.6) usage of formal methods for modeling and/or for composition of components, e.g., composite models/problems (layer 2) and basic solving frameworks (layer 3); (1.7) usage of AI techniques, e.g., for composition of components at the layer of standard (basic, typical) solving frameworks (layer 3); (1.8) study of approaches to design typical solving frameworks (layer 3) and their transformation into domain-oriented frameworks (layer 4); and (1.9) consideration of real world applied examples.

(II) Macro-architectural issues: (2.1) examination of approaches to integration of problem/discipline domain systems (architectures) into a multi-problem/multi-discipline domain system (architecture); (2.2) analysis of life cycle issues and/or "technological systems problems" (e.g., design, revelation of bottlenecks, changes and transformation, development, forecasting) for systems architectures; and (2.3) consideration of real-world applied multi-problem/multi-discipline domain systems.

(III) Education: (3.1) usage of the described four-layer framework in CS/engineering education, (3.2) organization of student multi-disciplinary team (s) to learn/study/design of multi-problem/multi-discipline domain systems (i.e., "learning a multi-problem/multi-discipline domain").

References

[1] Alba E, Luque G, Garcia-Nieto J. MALLBA: a software library to design efficient optimization algorithms. Int J Innovative Comput Appl 2007;1(1):74–85.
[2] Aho AV, Hopcroft JE, Ullman JD. The design and analysis of computer algorihtms. Reading, MA: Addison Welsey; 1976.
[3] Alevras D. Assignment and matching. In: Floudas CA, Pardalos PM, editors. Encyclopedia of optimization. Springer; 2009. p. 106–8.
[4] Baldwin CY, Clark KB. Design rules: the power of modularity. Cambridge: MIT Press; 2000.
[5] Bass L, Clements P, Kazman R. Software architecture in practice. 2nd ed. Reading, Mass: Addison-Wesley; 2003.
[6] Bowen K. The process of problem-formulation. EJOR 2004;128(2):258–65. 8.
[7] Brown AW. Large-scale, component-based development. Englewoos Cliffs (NJ): Prentice Hall PTR; 2000.
[8] Buede DM. Software review. Overview of MCDA software market. J Multi-Criteria Decis Anal 1992;1(1):59–61.
[9] Buede DM. The engineering design of systems: models and methods. 2nd ed. New York: J Wiley & Sons; 2009.
[10] Burkard RE. Communication network assignment problem. In: Floudas CA, Pardalos PM, editors. Encyclopedia of optimization. Springer; 2009. p. 394–6.
[11] Burkard R, Dell'Amico M, Martello S. Assignment Problems. SIAM, Providence; 2009.
[12] Cavendish D, Gerla M. Routing optimization in communication networks. In: Cheng MX, Li Y, Du D-Z, editors. Combinatorial optimization in communication networks. Springer; 2006. p. 505–47.
[13] Cela E. The quadratic assignment problem: theory and algorithms. Dordrecht: Kluwer Academic Publishers; 1998.
[14] Cheng MX, Li Y, Du D-Z, editors. Combinatorial optimization in communication networks. Springer; 2006.
[15] Cieslik D. Steiner minimal trees. Berlin: Springer; 1998.
[16] Cramer EJ, Dennis JE, Frank PD, Lewis RM, Shubin GR. Problem formulation for multidisciplinary optimization. SIAM J Opt 1994;4(3):754–76.
[17] Current J, Min H, Schilling D. Multiobjective analysis of facility location decisions. EJOR 1990;49(3):295–300.
[18] Dawande M, Keskinocak P, Swaminathan JM, Tayur S. On bipartite and multipartite clique problems. J Algorithms 2001;41(2):388–403.
[19] De S. An explanation subsystem to support decision problem structuring. IEEE Int Conf SMC 1995;3:2850–5.
[20] Douieb K, Langerman S. Dynamic hotlinks. Algorithmica 2008;50(2):208–22.
[21] Drezner T. Competitive facility location. In: Floudas CA, Pardalos PM, editors. Encyclopedia of optimization. Springer; 2009. p. 396–401.
[22] Du D-Z, Pardalos PM, editors. Handbook of combinatorial optimization, vols. 1–3. New York: Springer; 1999.
[23] Du D-Z, Lu B, Ngo H, Pardalos PM. Steiner tree problems. In: Floudas CA, Pardalos PM, editors. Encyclopedia of optimization. Springer; 2009. p. 3723–43.

[24] Eden C, Ackermann F. Where next for problem structuring methods. J ORS 2006;57(7):766–8.
[25] Eswaran KP, Tarjan RE. Augmentation problems. SIAM J Comput 1976;5(4):653–65.
[26] Floudas CA, Pardalos PM, editors. Encyclopedia of optimization. Springer; 2009.
[27] Fredman ML, Tarjan RE. Fibonacci heaps and their uses in improved network optimization algorithms. J ACM 1987;34(3):596–615.
[28] Fuhrmann S, Krumke SO, Wirth H-C. Multiple hotlink assignment. In: Brandstadt A, Le VB, editors. International conference on WG 2001. LNCS, vol. 2204. Springer; 2001. p. 189–200.
[29] Gallopoulos E, Houstis E, Rice JR. Computer as thinker/doer: problem solving environments for computer science. IEEE Comput Sci Eng 1994;1(2):11–23.
[30] Garey MR, Johnson DS. Computers and intractability, the guide to the theory of NP-completeness. San Francisco: W.H.Freeman and Company; 1979.
[31] Garlan D, Shaw M. Software architecture: perspectives on an emerging discipline. Englewood Cliffs (NJ): Prentice-Hall; 1995.
[32] Gavish B. Topological design of telecommunication networks-the overall design problem. EJOR 1992;58(2):149–72.
[33] Goel V, Pirolli P. The structure of design problem spaces. Cognitive Sci 1992;16(3):395–429.
[34] Gupta JND, Forgionne GA, Mora M. Intelligent decision-making support systems: foundations, applications and chellenges. New York: Sprigner; 2006.
[35] Huang CC, Kusiak A. Modularity in design of products and systems. IEEE Trans SMC-Part A 1998;28(1):66–77.
[36] Jain AK, Murty MN, Flynn PJ. Data clustering: a review. ACM Comput Surv 1999;31(3):264–323.
[37] Jensen TR, Toft B. Graph coloring problems. New York: Wiley; 1995.
[38] Joldersma C, Roelofs E. The impact of soft OR-methods on problem structuring. EJOR 2004;152(3):696–708.
[39] Jonassen DH. Instructional design models for well-structured and ill-structured problem solving learning outcomes. Educ Technol Rea Dev 1997;45(1):65–94.
[40] Johnson DS, Trick MA, editors. Cliques, coloring, and satisfiability, DIMACS series in discrete mathematics and theoretical computer science, vol. 26, AMS, Providence; 1996.
[41] Kellerer H, Pferschy U, Pisinger D. Knapsack problems. Berlin: Springer; 2004.
[42] Kemp C, Tenenbaum JB. The discovery of structural form. PNAS 2008;105(31):10687–92.
[43] Knuth DE, Raghunathan A. The problem of compatible representatives. SIAM J Disc Math 1992;5(3):422–7.
[44] Lawler EL, Lenstra JK, Rinnooy Kan AHG, Shmoys DB, editors. The traveling salesman problem. New York: Wiley; 1985.
[45] Levin MSh. Combinatorial design in hypertext. In: 5th International conference on information systems development, Poland; 1996. p. 255–63.
[46] Levin MSh. Combinatorial engineering of decomposable systems. Dordrecht: Kluwer Academic Publishers; 1998.
[47] Levin MSh. Allocation on binary relations: combinatorial models. In: 14th International conference on systems engineering ICSE'2000, UK: Coventry Univ.; 2000. p. 372–76.
[48] Levin MSh. System synthesis with morphological clique problem: fusion of subsystem evaluation decisions. Inform Fusion 2001;2(3):225–37.
[49] Levin MSh. Towards combinatorial planning of human–computer systems. Appl Intell 2002;16(3):235–47.
[50] Levin MSh. Modular system synthesis: example for composite packaged software. IEEE Trans SMC, Part C 2005;35(4):544–53.
[51] Levin MSh. Composite systems decisions. New York: Springer; 2006.
[52] Levin MSh. Combinatorial technological systems problems (examples for communication system). In: International conference on systems engineering and modeling ICSEM2007, Israel; 2007. p. 24–32.
[53] Levin MSh. Towards hierarchical clustering. In: Internatonal conference on CSR-2007, LNCS 4649, Springer; 2007. p. 205–15.
[54] Levin MSh. Morphological approach to electronic shopping. In: IEEE region 8 International Conference on Sibircon-2008, Novosibisrsk; 2008. p. 280–5.
[55] Levin MSh. Student research projects in system design. In: International conference on computer supported education CSEDU-2009, vol. 2, Lisbon; 2009. p. 67–72.
[56] Levin MSh. Combinatorial optimization in system configuration design. Autom Remote Control 2009;70(3):519–61.
[57] Levin MSh. Combinatorial scheme for design of marketing strategy. Bus Infomatics 2009;2(08):42–51 [in Russian].
[58] Levin MSh. Towards communication network development (structural systems issues, combinatorial models). In: IEEE region 8 international conference on "SIBIRCON-2010", Irkutsk, vol. 1; 2010. p. 204–8.
[59] Levin MSh. Connection of user's in last mile problem. IEEE Trans SMC-Part A 2011;41(2):370–4.
[60] Levin MSh. Course on system design (structural approach). Electronic preprint; 2011, 22pp., 1103.3845v1 [cs.SE].
[61] Levin MSh. Towards k-set problem frameworks in education. In: The 3rd International conference on computer supported education CSEDU-2011, vol. 2, The Netherlands; 2011. p. 99–104.
[62] Levin MSh, Nisnevich ML. Combinatorial scheme for management of life cycle: example for concrete macrotechnology. J Intell Manuf 2001;12(4):393–401.
[63] Levin MSh, Danieli MA. Hierarchical decision making framework for evaluation and improvement of composite systems (example for building). Informatica 2005;16(2):213–40.
[64] Levin MSh, Last M. Design of test inputs and their sequences in multi-function system testing. Appl Intell 2006;25(1):544–53.
[65] Levin MSh, Khodakovskii IA. Structural design of the telemetry system. Autom Remote Control 2007;68(9):1654–61.
[66] Levin MSh, Merzlyakov AO. Composite combinatorial scheme of test plannig (example for microprocessor systems). In: IEEE region 8 International Conference Sibircon-2008, Novosibisrsk; 2008. p. 291–5.
[67] Levin MSh, Leus AV. Configuration of integrated security system. In: 7th IEEE international conference on industrial informatics INDIN 2009, Cardiff; 2009. p. 101–5.
[68] Levin MSh, Sharov SYu. Hierarchical morphological design of Web-hosting system. J Integr Des Process Sci 2009;13(1):1–14.
[69] Levin MSh, Fimin AV. Combinatorial scheme for analysis of political candidates and their strategies. Inform Process 2009;9(2):83–92 [in Russian].
[70] Levin MSh, Kruchkov O, Hadar O, Kaminsky E. Combinatorial systems evolution: example of standard for multimedia information. Informatica 2009;20(4):519–38.
[71] Levin MSh, Petukhov M. Multicriteria assignment problem (selection of access points). In: 23rd International conference on IEA/AIE 2010, LNCS 6097, part II. Springer; 2010. p. 277–87.
[72] Levin MSh, Andrushevich A, Kistler R, Klapproth A. Combinatorial evolution of ZigBee protocol. In: IEEE region 8 international conference SIBIRCON-2010, Irkutsk, vol. 1; 2010. p. 314–9.
[73] Levin MSh, Safonov AV. Towards modular redesign of networked system. In: 2nd International congress on ultra modern telecommunication and control system and workshops ICUMT-2010, Moscow; 2010. p. 109–14.
[74] Levin MSh, Andrushevich A, Klapproth A. Improvement of building automation system. In: 24th International conference IEA/AIE 2011, LNCS 6704, Part II, Springer, Heidelberg; 2011. p. 459–68.
[75] Levin MSh, Nuriakhmetov RI. Electronic preprint; 2011, 11pp., 1102.2524v1 [cs.DS].
[76] Levin MSh, Safonov AV. Improvement of regional telecommunications networks. J Commun Technol Electron 2011;56(6):770–8.
[77] Luger GF. Artificial intelligence: structures and strategies for complex problem solving. 5th ed. Boston: Addison-Wesley; 2004.
[78] Maier MW, Rechtin E. The art of systems architecting. 2nd ed. Boca Raton (FL): CRC Press; 2000.
[79] Massey AP, Wallace WA. Understanding and facilitating group problem structuring and formulation: mental representations, interaction, and representation aids. DSS 1996;17(4):253–74.
[80] McKeenly PK, Sadjadi SM, Kasten EP, Cheng BHC. Composing adaptive software. IEEE Comput 2004;37(7):56–64.
[81] Mingers J, Rosenhead J. Problem structuring methods in action. EJOR 2004;152(3):530–54.
[82] Minoux M. Network synthesis and optimum network design problems: models, solutions, applications. Networks 1989;152(3):530–54.
[83] Mirkin BG. Clustering for data mining: a data recovery approach. Boca Raton (FL): Chapman & Hall/CRC; 2005.
[84] Mirkin B, Muchnik I. Combinatorial optimization in clustering. In: Du D-Z, Pardalos PM, editors. Handbook of combinatorial optimization, vol. 2. New York: Springer; 1999. p. 261–329.
[85] Murphey R. Frequency assignment problem. In: Floudas CA, Pardalos PM, editors. Encyclopedia of optimization. Springer; 2009. p. 1097–101.
[86] Nemhauser GL, Wolsey LA. Integer and combinatorial optimization. New York: Wiley; 1988.
[87] Ng Y-H, Chin S-F. Problem solving in a dynamic environment. Singapore: World Scientific Publishing Co.; 1994.
[88] Papadimitriou CH, Steiglitz K. Combinatorial optimization: algorithms and complexity. Englewood Cliffs, NJ: Prentice-Hall; 1982.
[89] Pardalos PM, Wolkowicz H, editors. Qudratic assignment and related problems. AMS, Providence; 1994.
[90] Pitsoulis L, Pardalos PM. Quadratic assignment problem. In: Floudas CA, Pardalos PM, editors. Encyclopedia of optimization. Springer; 2009. p. 3119–49.
[91] Rice JR, Boisvert RF. From scientific software libraries to problem-solving environments. IEEE Comput Sci Eng 1996;3(3):44–53.
[92] Ritchey T. Problem structuring using computer-aided morphological analysis. J ORS 2006;57(7):792–801.
[93] Roberts FS. Applied combinatorics. Englewood Cliffs (NJ): Prentice Hall; 1984.
[94] Roy B. Multicriteria methodology for decision aiding. Dordrecht: Kluwer Academic Publishers; 1996.
[95] Roy B, Slowinski R. Multi-criteria assignment problem with incompatibility and capacity constraints. Ann Oper Res 2006;147(1):287–316.
[96] Scarelli A, Narula SC. A multicriteria assignment problem. J Multi-Criteria Dec Anal 2002;11(2):65–74.
[97] Scheubrein R, Zionts S. A problem structuring from and for multiple criteria decision support system. Comput Oper Res 2006;33(1):18–31.
[98] Shenhar AJ. Systems engineering management: a framework for the development of a multidisciplinary discipline. IEEE Trans SMC-Part C 1994;24(2):327–32.
[99] Simon HA. The structure of ill-structured problems. Artif Intell 1973;4(3):181–201.

[100] Simon HA et al. Decision making and problem solving. Interfaces 1987;17(5):11–31.
[101] Simon HA, Newell A. Heuristic problem solving: the next advance in operations research. Oper Res 1958;6(1):1–10.
[102] Smith GF. Towards a heuristic theory of problem structuring. Manage Sci 1988;34(12):1488–506.
[103] Sousa JP, Poladian V, Garlan D, Schmerl B, Shaw M. Task-based adaptation for ubiquitous computing. IEEE Trans SMC-Part C 2006;36(3):328–40.
[104] Takahara Y, Yongmei L. Multi-level systems approach to solver system design. In: 18th International conference on systems engineering ICSEng 2005; 2005. p. 462–7.
[105] Tanenbaum AS. Computer networks. 4th ed. NJ: Prentice Hall PTR; 2002.
[106] Tanenbaum AS. Structured computer organization. 5th ed. NJ: Prentice Hall; 2006.
[107] Tarapata Z. On a multicriteria shortest path problem. Int J Appl Math Comput Sci 2007;17(2):269–87.
[108] Tomiyama T, Collaborative product development in ill-structured problem domain. In: 10th International conference on computer supported cooperative work in design CSCWD'06, 2006. p. 1–6.
[109] Turban E, Aronson JE, Liang T-P. Decision support systems and intelligent systems. 7th ed. NJ: Prentice Hall; 2004.
[110] Voronin AA, Mishin SP. Algorithms to seek the optimal structure of the organizational system. Autom Remote Control 2002;63(5):803–14.
[111] Voss S. Capacitated minimum spanning trees. In: Floudas CA, Pardalos PM, editors. Encyclopedia of optimization. Springer; 2009. p. 347–57.
[112] Waller RJ. The synthesis of hierarchical structures: techniques and applications. Decis Sci 1976;7(4):659–74.
[113] Williams BJ, Carver JC. Characterizing software architecture changes: a systematic review. Inform Softw Technol 2010;52(1):31–51.
[114] Wilson RJ. Introduction to graph theory. New York: Academic Press; 1972.
[115] Wing JM. Computational thinking. Commun ACM 2006;49(3):33–5.
[116] Wynants C. Network synthesis problems. Dordrecht: Kluwer Academic Publishers; 2001.
[117] Zopounidis C, Doumpos M. Multicriteria classification and sorting methods: a literature review. EJOR 2002;138(2):229–46.
[118] Zopounidis C, Doumpos M. Multicriteria sorting methods. In: Floudas CA, Pardalos PM, editors. Encyclopedia of optimization. Springer; 2009. p. 2379–96.