Chapter 1

# Towards System Configuration Design

**Mark Sh. Levin**
Institute for Information Transmission Problems, Russia
mslevin@acm.org

The significance of systems configurations has been increased in many applied domains (e.g., software, hardware, manufacturing systems, supply chain systems, solving strategies). In the paper several system configuration problems are investigated: (i) searching for (selection of) a set (structure) of system components, (ii) searching for a set of compatible system components, (iii) reconfiguration of a system as redesign of the system structure, (iv) multi-stage design and redesign of system configuration, and (v) design or redesign of the system configuration for multi-product systems. Several combinatorial models (including multicriteria statements) are under examination: problem of representatives, multiple choice problem, morphological clique problem (with compatibility of system components), and their modifications.

## 1.1 Introduction

In recent years, the significance of systems configurations for complex multi-component systems has been increased in many applied domains (Fig. 1): (1) manufacturing systems (e.g., [21]); (2) computer systems (e.g., [19]); (3) hardware (e.g., [22]); (4) software (e.g., [11], [12], [16], [20], [22]); (5) algorithm systems and solving strategies (e.g., [10], [16]); (6) communication systems (e.g., [15], [20]); (7) web-based services (e.g., [1], [17]); (8) family of industrial products (e.g., [7]); and (9) supply chains systems (e.g., [2], [3], [23]). In the main, the following approaches are used to system configuration design: (a) multicriteria multiple choice problem (e.g., [15], [22]); (b) hierarchical morphological design approach (e.g., [10], [11], [12]); (c) approaches based on fuzzy sets (e.g., [23]); and (d) AI techniques (e.g., [19]). In the article a set of basic design configuration problems is described (design of system configuration, reconfiguration, etc.). The problems are based on support combinatorial models (problem of representatives, multiple choice problem, and morphological clique problem).

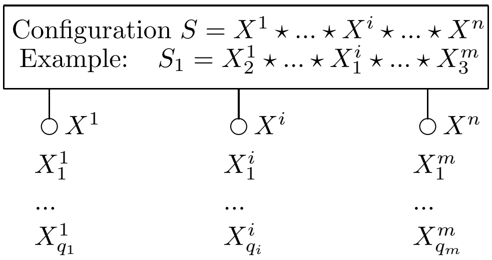

Fig. 1. Illustration of configuration

## 1.2 Support Combinatorial Problems

Problem of representatives is described in [6] (Fig. 2): there exists a set of initial element sets, to construct a set of representatives for each initial set. In [9] the problem is examined with binary compatibility of elements (generally it is NP-hard).

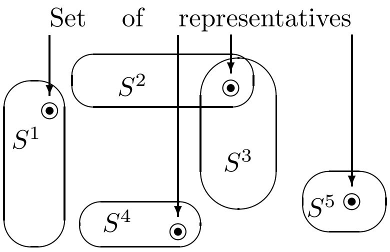

Fig. 2. Problem of representatives

Knapsack-like problems are widely used for system configuration design. The basic problem is (e.g., [5], [8], [18]):

$$\max \sum_{i=1}^{m} c_i x_i \quad s.t. \quad \sum_{i=1}^{m} a_i x_i \leq b, \; x_i = 0 \cup 1, \; i = 1, ..., m$$

and additional resource constraints $\sum_{i=1}^{m} a_{i,k} x_i \leq b_k;\; k = 1, ..., l;$ where $x_i = 1$ if item $i$ 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 ([5], [18]) and can be solved by the following methods: (i) enumerative methods (e.g., Branch-and-Bound, dynamic programming), (ii) approximate schemes with a limited relative error (e.g., [8], [18]), and (iii) heuristcs. For multiple criteria statements it is necessary to search for Pareto-effective solutions and analogical approaches can be used. In the case of a multiple choice problem, the items (i.e., elements) are divided into groups and we select elements from each group while taking into account a total resource constraint (or constraints):

$$\max \sum_{i=1}^{m} \sum_{j=1}^{q_i} c_{ij} x_{ij} \quad s.t. \quad \sum_{i=1}^{m} \sum_{j=1}^{q_i} a_{ij} x_{ij} \leq b, \quad \sum_{j=1}^{q_i} x_{ij} = 1, \ i = 1, ..., m, \ x_{ij} = 0 \cup 1$$

For multiple criteria description $\{c_{i,j}\}$ $\forall (i,j)$ (i.e., multi-objective multiple choice problem), the vector goal function $( f^1, ..., f^p, ..., f^r )$ is as follows [15]:

$$( \max \sum_{i=1}^{m} \sum_{j=1}^{q_i} c^1_{ij} x_{ij} , ..., \ \max \sum_{i=1}^{m} \sum_{j=1}^{q_i} c^p_{ij} x_{ij} , ..., \ \max \sum_{i=1}^{m} \sum_{j=1}^{q_i} c^r_{ij} x_{ij} )$$

Evidently, here it is necessary to search for the Pareto-effective (by the vector objective function above) solutions. In this case, the following solving schemes can be used (e.g., [15]): (i) enumerative methods (e.g., Branch-and-Bound, dynamic programming), (ii) heuristic based on multicriteria ranking of elements and step-by-step packing the knapsack, (iii) 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 excellence (it is described bellow).

Hierarchical Morphological Multicriteria Design (HMMD) on the basis of morphological clique problem generalizes multiple choice problem (via taking into account element compatibility) and morphological analysis (via ordinal evaluation of elements and their compatibility). A brief description of HMMD is a typical one as follows ([10], [11], [12], [13]). The examined composite (modular, decomposable) system consists of components and their interconnection (IC) or compatibility. Basic assumptions of HMMD are the following: *(a)* a tree-like structure of the system (generally, it is *morphological tree* model [13]); *(b)* a composite estimate for system quality that integrates components (subsystems, parts) qualities and qualities of IC (compatibility) across subsystems; *(c)* monotonic criteria for the system and its components; and *(d)* quality of system components and IC are evaluated on the basis of coordinated ordinal scales. The designations are: (1) design alternatives (DAs) for leaf nodes of the model; (2) priorities of DAs ($r = 1, ..., k$; 1 corresponds to the best one); (3) ordinal compatibility (IC) for each pair of DAs ($w = 0, ..., l$, $l$ corresponds to the best one). The basic phases of HMMD are: *1.* design of the tree-like system model; *2.* generation of DAs for leaf nodes of the model; *3.* hierarchical selection and composing of DAs into composite DAs for the corresponding higher level of the system hierarchy; and *4.* analysis and improvement of composite DAs (decisions). Let $S$ be a system consisting of $m$ parts (components): $P(1), ..., P(i), ..., P(m)$. A set of design alternatives is generated for each system part above. The problem is:

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

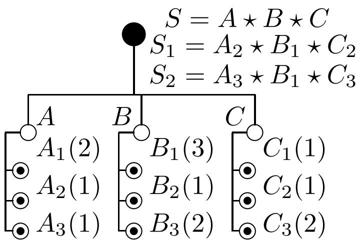


Fig. 3. Example of composition

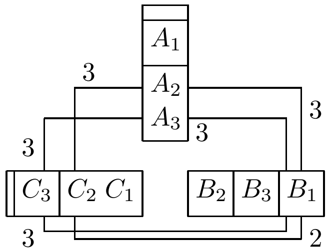


Fig. 4. Concentric presentation

A discrete space of the system excellence on the basis of the following vector is used: $N(S) = (w(S); n(S))$, where $w(S)$ is the minimum of pairwise compatibility between DAs which correspond to different system components (i.e., $\forall\ P_{j_1}$ and $P_{j_2}$, $1 \leq j_1 \neq j_2 \leq m$) in $S$, $n(S) = (n_1, ..., n_r, ...n_k)$, where $n_r$ is the number of DAs of the $r$th quality in $S$. As a result, we search for composite decisions which are nondominated by $N(S)$. The described problem is NP-hard. Clearly, the compatibility component of vector $N(S)$ can be considered on the basis of a poset-like scale as well (as $n(S)$ ). In this case, the discrete space of system excellence will be an analogical lattice. Figs. 3, 4, and 5 illustrate the composition problem. In the examples, composite decisions are: $S_1 = A_2 \star B_1 \star C_2$, $N(S_1) = (2; 2, 0, 1)$; $S_2 = A_3 \star B_1 \star C_3$, $N(S_2) = (3; 1, 1, 1)$.

The solving process can be based on two strategies [10]: (1) enumerative method, (2) dynamic programming.

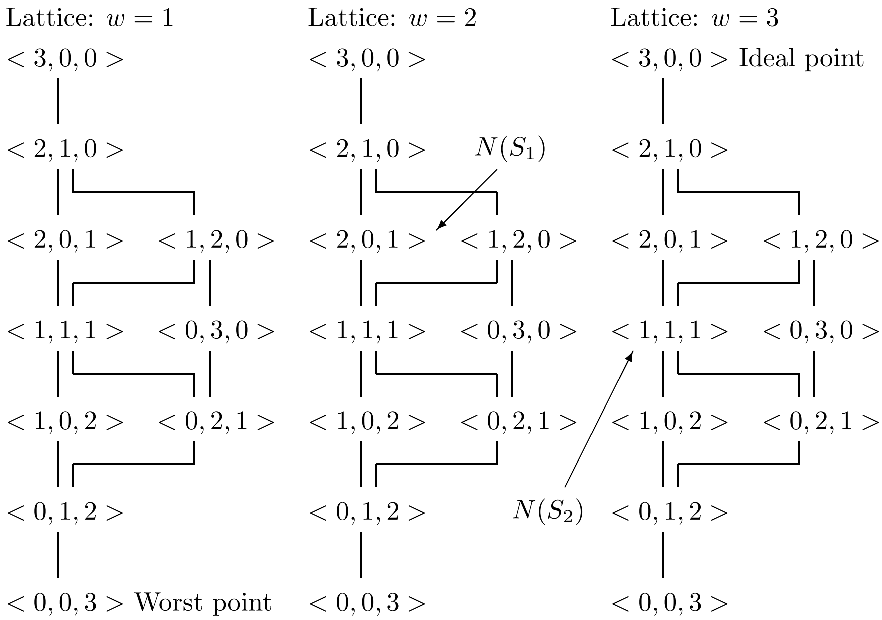


Fig. 5. Discrete space of system quality

## 1.3 Basic Configuration Problems

The system configuration problem consists in searching for (selection of) a set (structure) of system components (Fig. 1). It is reasonable to point out the following cases for the set of basic alternative elements or design alternatives DAs: (a) basic case (set of elements), (b) set of compatible elements, (c) intersection of element sets corresponding to different system parts, and (d) hierarchical multi-layer system model.

Out set of problems consists of the following:

**1.** Basic system configuration design problem: a system structure as a set of components (i.e., their realization) (**P**). The problem is: *searching for (selection of) a set (structure) of system components.* Here the following basic models can be used: (i) problem of representatives (Fig. 2) [6], (ii) multiple-choice problem (i.e., without element compatibility, Fig. 1) ([5], [8], [15], [18]).

**2.** Design of system configuration with compatible components (**G**). Here the following basic models can be used: (i) problem of compatible representatives [9], (ii) morphological clique problem (HMMD) ([10], [11], [12]) (Figs. 3 and 4).

**3.** Reconfiguration problem: design of a new system configuration based on the previous configuration (**R**).

Generally, a structure of system changes can be considered as follows (e.g., [10], [13]): *1.* components/design alternatives: *1.1.* improved old DAs, *1.2.* new DAs; *2.* improved compatibility; and *3.* new system structure. In reconfiguration problem, mainly cases 1.1, 1.2, and 3 are examined. For changing of system components (cases 1.1, 1.2) multiple choice problem can be used (e.g., [15]). Fig. 6 illustrates a system redesign (modification) process: from initial system $S_a$ to modified system (via changes of components): $S_a \Rightarrow S_b$.

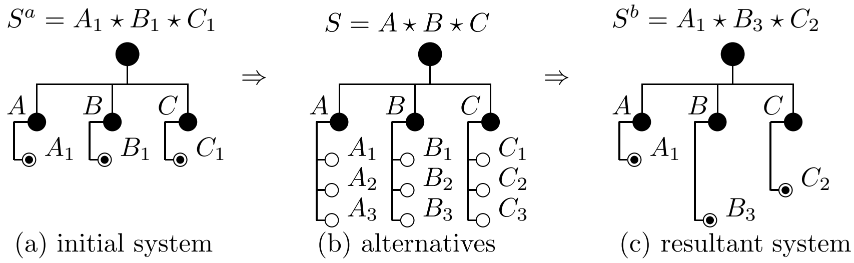


Fig. 6. System reconfiguration

HMMD can be used for more complicated problems when compatibility among component changes exists (e.g., [10], [11], [12]):

**4.** Multi-stage configuration problem for design of configuration trajectory (**T**): design of a system configuration for several series stages (i.e., design of *system configuration trajectory*): (a) design of an initial system configuration, (b) several series system reconfiguration problems.

Another macro-strategy consists in two-layer design process (i.e., multi-stage design): (i) design of a system configuration (or several configurations) at each stage (usage of multiple choice problem or HMMD), (ii) selection at each stage the best configuration while taking into account compatibility between the selected configurations (usage of HMMD). Here hierarchical morphological design has been used ([10], [11], [12]).

Fig. 7 illustrates a 3-stage system configuration trajectory.

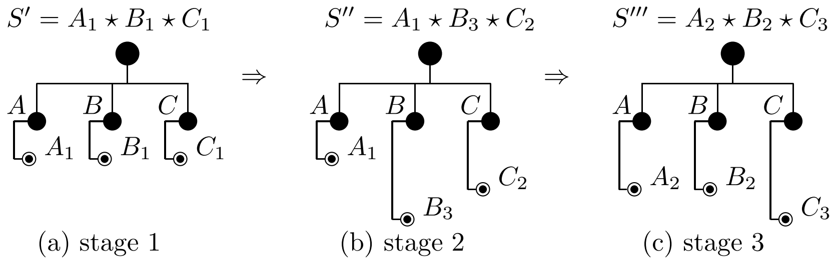


Fig. 7. Trajectory of configuration

**5.** Design of system configuration for multi-product systems (**M**).

The significance of product families is increasing. In [12] it is shown, HMMD can be used as a basis for the design process of the multi-product system. Fig. 8 illustrates a three-product system with one common module.

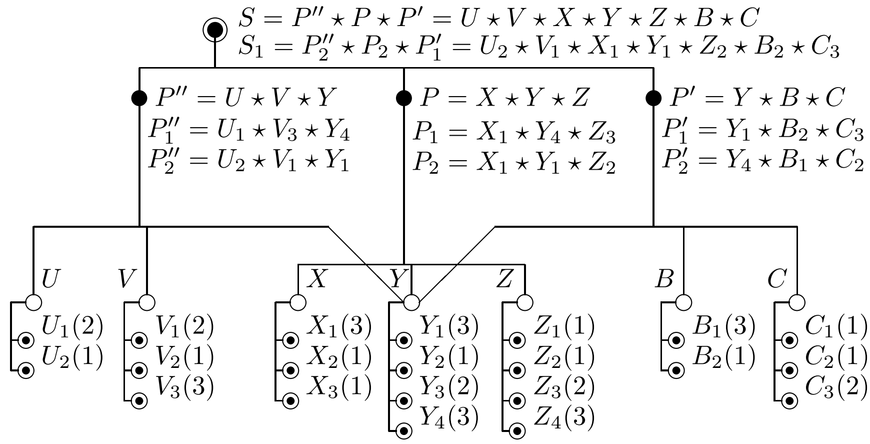


Fig. 8. Three-product system with one common module

## 1.4 Conclusion

In the paper, the design problem for system configurations is examined: searching for (selection of) a set (structure) of system components. For the case of compatible system components the problem is studied as well. Three problems are considered as additional statements: (a) reconfiguration of a system as redesign of the system structure, (b) multi-stage design and redesign of system configuration, and (c) design or redesign of the system configuration for multi-product systems. Mainly, two basic models are described: multicriteria multiple choice problem and morphological clique problem (with compatibility of system components).

Future research directions can include the following: (1) analysis of additional problems (e.g., design of system configuration as a multi-layer hierarchical structure via hierarchical clustering [14]), (2) usage of new design models (e.g., models under uncertainty), (3) investigation of various applications, (4) modeling and analysis of evolution (development) of systems configurations, (5) analysis of interconnection of the considered problems and research direction *software configuration management* (e.g., [4]), and (6) usage of the system configuration design problems in engineering and computer science education.

# Bibliography


[1] Benatallah, B., Sheng, Q.Z., and M. Dumas, "The self-serv environment for web services composition", *IEEE Internet Computing* 7(6) (2003) 40-48.

[2] Blomqvist, E., Levashova, T., Ohgren, A., Sandkuhl, K., Smirnov, A., and V. Tarassov, "Configuration of dynamic SME supply chains based on ontologies", in: Marik, V., Brennan R.W., Pechoucek M. (eds.) *Holonic and Multi-Agent Systems for Manufacturing.* LNAI 3593, Springer (2005) 246-256.

[3] Chandra, C. and J. Grabis, *Supply Chain Configuration: Concepts, Solutions, and Applications.* Springer (2007).

[4] Conradi, R. and B. Westfechtel, "Versions models for software configuration management", *ACM Comput. Survey* 30(2) (1998) 232-282.

[5] Garey, M.R. and D.S. Johnson, *Computers and Intractability. The Guide to the Theory of NP-Completeness*, W.H.Freeman & Company (1979).

[6] Hall, M., Jr., *Combinatorial Theory*, 2nd ed., Wiley (1986).

[7] Hotz, L., Wolter, K., and T. Krebs, *Configuration in Industrial Product Families.* IOS Press (2006).

[8] Keller, H., Pferschy, U., and D. Pisinger, *Knapsack Problems.* Springer (2004).

[9] Knuth, D. and A. Ratghunathan "The problem of compatible representatives", *SIAM J. on Discrete Mathematics* 5(3) (1992) 422-427.

[10] Levin, M.Sh., *Combinatorial Engineering of Decomposable Systems.* Kluwer (1998).

[11] Levin, M.Sh., "Modular system synthesis: example for composite packaged software", *IEEE Trans. on SMC - Part C* 35(4) (2005) 544-553.

[12] Levin, M.Sh., *Composite Systems Decisions.* Springer (2006).

[13] Levin, M.Sh., "Combinatorial technological systems problems (examples for communication system)", *ICSEM2007*, (2007) 24-32.

[14] Levin, M.Sh., "Towards hierarchical clustering", in: V. Diekert, M. Volkov, A. Voronkov, (Eds.), *CSR 2007*, LNCS 4649, Springer (2007) 205-215.

[15] Levin, M.Sh. and A.V. Safonov, "Design and redesign of configuration for facility in communication network", *Information Technologies and Computer Systems (Russian Academy of Sci.)* 4 (2006) 63-73 (in Russian).

[16] Li, L., Garibaldi, J., and N. Krasnogor, "Automated self-assembly programming paradigm: initial investigation", *Proc. of the Third IEEE Int. Workshop on Engineering of Autonomic and Autonomous Systems EASe2006*, IEEE Press (2006) 25-36.

[17] Madhusudan, T. and N. Uttamsingh, "A declarative approach to composing web services in dynamic environments", *Decision Support Systems* 41(2) (2006) 325-357.

[18] Martello, S., and P. Toth, *Knapsack Problem: Algorithms and Computer Implementation.* Wiley (1990).

[19] J. McDermott, J., "R1: a rule-based configurer of computer systems", *Artificial Intelligence* 19(2) (1982) 39-88.

[20] McKeenly, P.K., Sadjadi, S.M., Kasten, E.P., and B.H.C. Cheng, "Composing adaptive software", *IEEE Computer* 37(7) (2004) 56-64.

[21] Mehrabi, M.G., Ulsoy, A.G., and Y. Koren, "Reconfigurable manufacturing systems", *J. of Intell. Manufact.* 11(4) (2000) 403-419.

[22] Poladian, V., Sousa, J.P., Garlan, D., Schmerl, B., and M. Shaw, "Task-based adaptation for ubiquitous computing", *IEEE Trans. on SMC - Part C* 36(3) (2006) 328-340.

[23] Smirnov, A., Sheremetov, L., Chilov, N., and J.R. Cortes, "Soft-computing technologies for configuration of cooperative supply chain", *Applied Soft Computing* 4 (2004) 87-107.