Modular Systems Design via Multiobjective Optimization

Ivan Mustakerov and Daniela Borissova

Institute of Information and Communications Technologies at Bulgarian Academy of Sciences, Bulgaria, Sofia 1113, Acad. G. Bonchev, St. bl. 2,

E-mail: mustakerov@iit.bas.bg, dborissova@iit.bas.bg

Abstract: The design of modular technical systems should consider compatibility, functional restrictions and user requirements and is a complex combinatorial problem. The current paper proposes a design approach based on mathematical modelling and multicriteria combinatorial optimization tasks formulations. The proposed approach is illustrated on the example of personal computer configuration design. It takes into account the existing compatibility restrictions between the personal computer main modules and can be extended and modified to reflect different functional and users' requirements. The developed design modelling technique is used to formulate multiobjective discrete mixed-integer optimization tasks. The practical applicability of the developed approach is tested by numerical examples based on real personal computer modules data. The solutions of the formulated optimization tasks define combinations of Pareto-optimal modules satisfying the compatibility restrictions and given user requirements. The results of numerical experiments show the possibility for practical application of the proposed design approach.

Keywords: *modular systems design, multiobjective optimization, PC configuration design.*

1. INTRODUCTION

The purpose of modular systems design is to produce systems that satisfy the customers' needs, increase the probability of system success, reduce risk and reduce total-life-cycle cost. Customer requirement of product involves functionality, performance, appearance, price and dimension of the product, etc. In other words, the customer requirement is denotation of unsolved precept in customer domain and it includes the essential characters of the product in need [Quin and Wei, 2010]. The recent trends in system modularization facilitate the automation of design leveraging digital design repository in global manufacturing environment [Yoo and Kumara, 2010]. The essential benefits of modularity include decreasing of production costs and order lead-time, flexible designs to respond to the changing markets and technologies. Sometimes modularity is criticized for lack of performance optimization and excessive product similarity [Tzeng and Huang, 2008], [Heinrich and Jungst 1991], [Sanchez, 1993].

In general, the product variety optimization includes three degree of optimization problems – attribute assignment, module combination, and simultaneous design of both. When the possibilities of computational optimization for product variety design under fixed product architecture are explored, optimization is demanded to determine the choice of modules and their combination under fixed product architecture [Quin and Wei, 2010]. Using of optimization methods to solve engineering design problems is usually an interdisciplinary direction and includes selection of proper variables, constraints, objectives and models [Sobester et al., 2012]. Combinatorial optimization has proved to be a prospective method for engineering system design [Andersson, 2000], [Levin, 2009]

The current paper proposes an approach to designing modular systems based on multiobjective mixed-integer nonlinear optimization. Multiobjective optimization is used as a

AMO – Advanced Modeling and Optimization. ISSN: 1841-4311

tool to support the decision maker (DM) in respect of satisfaction of different requirements toward the system functionality. The proposed approach is illustrated by real life example of personal computer (PC) system configuration. As it is known different PC users have different requirements for computer system functionality. These requirements are reflected in the choice of main system modules – motherboard (MB), processor (CPU), memory (RAM), etc. An essential modular system design problem is the diversity of modules with specific compatibility restrictions/relationship to each other. For PC configuration example, MB manufacturers as ASUS, ABIT, MSI, Intel and Gigabyte offer many different MB types with specific requirements about the CPU, memory and other components. The same applies to CPU modules (produced by Intel or AMD for example) and is also true for RAM modules. In this respect, the example of the PC configuration is a good illustration of problems inherent to modular system design. There exists also a particular interest in optimization of computer system configuration. This is proved by publications demonstrating applying of different scientific approaches: heuristic-based methods [Tam and Ma, 2000], [Tam and Ma, 2002]; constraint and rule satisfaction problem [Jae-Kui et al., 1996], [Soininen et al., 2000]; weight constraint rule language [McDermott, 1982], grouping genetic algorithm [Kreng and Lee, 2004]. The current paper describes an approach to modular system design based on multiobjective combinatorial optimization as a flexible tool for satisfying multiple contradictory requirements.

2. SPECIFICS OF MODULAR SYSTEM DESIGN

Understanding of the designed system specifics is a must to proper formalization of the modular system design process. The ultimate challenge is the determination of optimal modules combination considering all of the given system specifications. Taking this into account, an algorithm for optimal modular system design is proposed on Fig. 1.

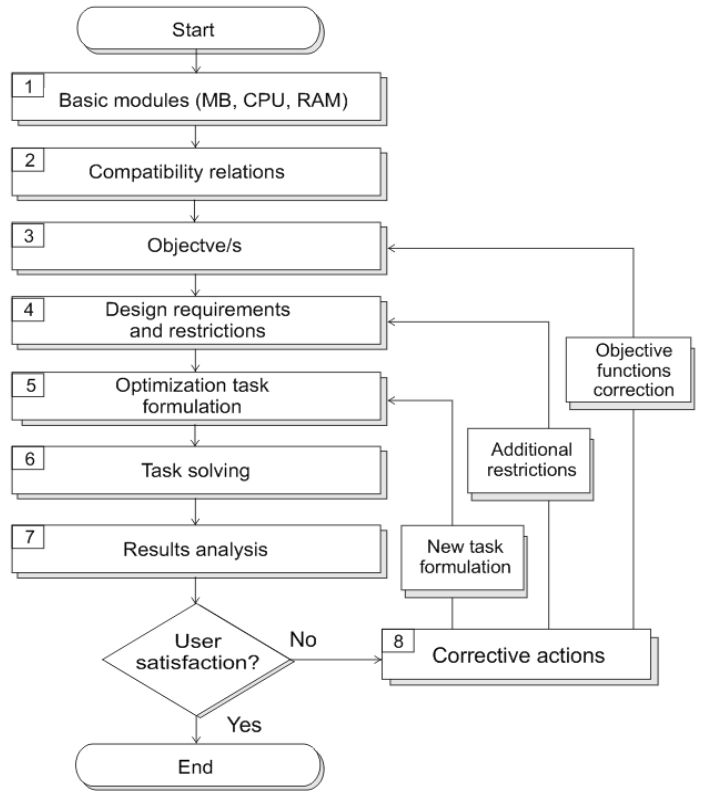

Start
1 Basic modules (MB, CPU, RAM)
2 Compatibility relations
3 Objectve/s
4 Design requirements and restrictions
5 Optimization task formulation
6 Task solving
7 Results analysis
User satisfaction?
No
Yes
8 Corrective actions
New task formulation
Additional restrictions
Objective functions correction
End

Fig. 1. Flowchart of the proposed algorithm for optimal modular system design

2.1. Basic modules to be considered in the design

The algorithm starts with procedure of defining of the sets of modules to be considered in design. The example of PC configuration design could be adequately illustrated by considering of **motherboard (MB)**, **processor (CPU)** and **memory (RAM)** as modules directly affecting properties of designed system.

2.2. Compatibility issues

Compatibilities restrictions are very important to consider in choice of modules to get properly working system. For the choice of MB, CPU and RAM modules the following should be considered:

- **motherboard** has strict requirements about the rest of the modules - specific type of CPU socket and type, certain chipset, number of slots and ports (PCI, AGP, IDE, SATA, e-SATA, USB, FireWire, LAN, RAID, etc.), specifications for the type and size of RAM.
- **processor** manufacturing companies offer CPUs with different sockets and functional specifications. For example, Intel product line[1] offers socket 478 for Pentium IV and Celeron processors; Socket 775 for Pentium, Celeron, Core™ 2 Duo, Quad and Extreme processors; Socket 1156 for Core™ i7 and i5 processors, Socket 1366 for Core™ i7. Other leading producer of CPUs – AMD[2] offers Socket A for Athlon™ and Duron™ processors, Socket AM2 for Athlon™ 64 and Semperon™ processors, Socket AM2+ for Phenom™ and Athlon™ 64 processors, Socket F for the Opteron™ and Athlon™ 64 7x processors. Therefore, the choice of particular type of CPU should be linked to the corresponding MB specifications.
- **random access memory** modules also have to be chosen accordingly MB specifications about memory type, slots number and maximal memory size that can be supported.
- **other** PC modules such as hard disk storage, video controller, RAID controller, optical storage devices, case, power supply, keyboard, etc., could also be considered in problem formulation but the choice of above basic modules is adequate to illustrate the proposed approach.

2.3. The design objectives

The optimization of any system design is in respect of given design objectives (optimization criteria). It is often necessary to consider simultaneously many and in most cases, contradictory criteria. This means, looking for multiobjective trade-off solutions (Pareto-optimal solutions). The multiobjective optimization can be adapted to reflect different decision maker's (DM) requirements by defining of proper objective functions for parameters of the designed system.

2.4. The design requirements and restrictions

Optimization of the system design means to consider all given functional, user and application environment requirements and restrictions. All of them should be specified and included in mathematical model. The user requirements are essential for the acceptance of the designed system.

[1] http://www.intel.com/content/www/us/en/intelligent-systems/previous-generation/intel-ixp4xx-intel-network-processor-product-line.html
[2] http://www.amd.com/us/products/desktop/processors/Pages/desktop-processors.aspx

2.5. Optimization problem formulation

Optimization problem formulation is based on mathematical model of the design process. The design process basically consists of two problems: the generation of design alternatives and decision making. Mathematical model proposed in the paper optimizes the modular design process by providing Pareto-optimal design alternatives.

2.5.1. Defining of mathematical model

The design of real-life modular technical systems is complex combinatorial problem. The combinatorial choice process can be modeled by using of binary integer decision variables associated with different parameters of the modules. For example, if $j \in J$ are indexes of different types of modules, $k \in K^j$ are indexes of parameters of module of type j , $i \in I_j$ are indexes of the various modules of the same type and each particular module i of type j has assigned a binary integer variable x_i^j, then choice of modules can be expressed by choice of its parameters as:

(1) $$\forall j \in J : (\forall k \in K^j : P^{j,k} = \sum_{i \in I_j} P_i^{j,k} x_i^j),\ \ x_i^j \in \{0,1\},\ \sum_{i \in I_j} x_i^j = 1$$

The essential considerations related to system design (for example, costs, compatibility issues and user requirements) have to be taken into account when modeling the design process. When choosing MB and CPU modules the *cost* parameter can be defined by equations:

(2) $$P^{MB\cos t} = \sum_{i \in I_{MB}} P_i^{MB\cos t} x_i^{MB},\ x_i^{MB} \in \{0,1\},\ \sum_{i \in I_{MB}} x_i^{MB} = 1$$

(3) $$P^{CPU\cos t} = \sum_{j \in I_{CPU}} P_i^{CPU\cos t} x_i^{CPU},\ x_i^{CPU} \in \{0,1\},\ \sum_{i \in I_{CPU}} x_i^{CPU} = 1$$

In the same way, other parameters of modules can be also defined as a result of the choice, as:

(4) $$P^{CPUclock} = \sum_{i \in I_{CPU}} P_i^{CPUclock} x_i^{CPU}$$

(5) $$P^{CPUcore} = \sum_{i \in I_{CPU}} P_i^{CPUcore} x_i^{CPU}$$

For the PC example, the selection of RAM modules is more complicated because includes not only type but also number of RAM modules. This is why other nonnegative integer decision variables x_i^{ram} are used for different types of RAM modules. The corresponding equations for RAM parameters *cost* and *size* using decision variables x_i^{ram} are:

(6) $$P^{RAM\cos t} = \sum_{i \in I_{RAM}} P_i^{RAM\cos t} x_i^{RAM},\ x_i^{ram} \in \mathrm{N}$$

(7) $$P^{RAMsize} = \sum_{i \in I_{RAM}} P_i^{RAMsize} x_i^{RAM},\ x_i^{RAM} \in \mathrm{N}$$

where **N** is the set of nonnegative integers.

2.5.2. Compatibility relationships

The design of the system cannot be acceptable, if not taken into consideration the compatibility relationships between modules. The compatibility relationships are described by definition of subsets of compatible modules. For the PC example, if subsets of CPUs compatible with particular type of motherboard *MB_i* are designated as I_{CPU}^i then the compatibility constraints about the decision variables assigned to MBs and to CPUs are:

(8) $$\forall i \in I_{MB} : x_i^{MB} \le \sum_{j \in I_{CPU}^i} x_j^{CPU}$$

where $I_{CPU}^i \subseteq I_{CPU}$ are indexes of the CPUs compatible with motherboard of type *i*.

The numerical experimentations showed that the specific of RAM modules choice requires two types of constraints:

- for compatibility of RAM modules and MBs

(9) $$\forall i \in I_{MB} : x_i^{MB} \sum_{k \in I_{RAM}^{i+}} x_k^{RAM} \ge 0$$

- for incompatibility of RAM modules and MBs

(10) $$\forall i \in I_{MB} : x_i^{MB} \sum_{k \in I_{RAM}^{i-}} x_k^{RAM} \le 0$$

where $I_{RAM}^{i+} \subseteq I_{RAM}$ and $I_{RAM}^{i-} \subseteq I_{RAM}$ are subsets of compatible and incompatible RAM modules with MB_i respectively.

Other types of restrictions about RAM modules and MBs compatibility reflect the limitation of maximal RAM size $P^{RAMsize}$ that can be supported by any particular type of MB_i:

(11) $$P^{RAMsize} \le \sum_{i \in I_{MB}} P_i^{MBram\max} x_i^{MB}$$

MB number of slots $P_i^{MB,RAMslots}$ available for RAM modules that is supported by MB_i:

(12) $$\sum_{i \in I_{RAM}} x_k^{RAM} \le P_i^{MBram\text{slots}}$$

and the requirement that at least one RAM module is needed to have functional computer configuration:

(13) $$\sum_{i \in I_{RAM}} x_k^{RAM} \ge 1$$

2.5.3. User requirements

This modeling approach allows easy introduction of different user requirements about the designed system parameters. For example, for RAM size, CPU clock frequency, CPU core number, etc., as:

(14) $$P^{RAMsize} \ge P^{RAM\min}$$

(15) $$P^{CPUclock} \ge P^{CPUclock\min}$$

(16) $$P^{CPUcore} \ge P^{CPUcore\min}$$

where $P^{RAM\min}$, $P^{CPUclock\min}$, $P^{CPUcore\min}$ are minimal values given by the DM.

Other requirements about the parameters of designed system can be easily introduced in similar way.

2.6. Optimization tasks formulation

The described approach for modular systems design is based on formulation of multiobjective optimization tasks. The tasks solutions will define Pareto-optimal choice of modules corresponding to the DM preferences. A realistic example of PC configuration design could be to minimize the *cost* of modules while maximizing the CPU *computational power* (core number, clock frequency) and RAM *size*. This leads to the following multiobjective formulation:

(16)
$$\begin{aligned}&\min\,(P^{MB,cost} + P^{CPU,cost} + P^{RAM,cost})\\&\max P^{CPU,core}\\&\max P^{CPU,clock}\\&\max P^{RAM,size}\end{aligned}$$

subject to (2) – (13) and possibly plus some of the constraints reflecting user requirements (14) – (16).

2.7. Result analysis

The optimization of any system design is connected with analysis of the final results. The proposed algorithm aims to assist the DM in taking of decision. The results of optimization tasks solution are analyzed by the DM and are accepted or not accepted. In the second case some corrective actions are applied on the next step.

2.8. Corrective actions

The proposed algorithm involves 3 types of corrective actions: 1) redefining the objectives or including of new ones; 2) redefining user design requirements; 3) new formulation of the optimization task. These actions will provide another alternative to the designed system and will assist the DM in making of design decision.

3. NUMERICAL TESTING

In order to verify the theoretical developments in the paper some real PC modules are used as shown in Table 1, 2 and 3.

Table 1. MBs data

#	MB type	CPU socket	Intel® CPU type	RAM type	RAM slots	RAM max, GB	RAM MHz	Price BGL
1	ASROCK P45XE-WIFIN/P45	LGA775	Core™ 2E, Core™ 2 Quad, Core™ 2 Duo, Pentium® Dual Core, Celeron® Dual Core, Celeron®	DDR2	4	16	1200, 1066, 800, 667	178.50
2	ASROCK G31M-S/G31	LGA775	Core™ 2E, Core™ 2 Quad, Core™ 2 Duo, Pentium® EE, Pentium® D, Pentium® 4 E, Pentium® 4, Celeron®	DDR2	2	8	800, 667	69.00
3	Gigabyte G31M-ES2C/G31	LGA775	Core 2 Quad, Core 2 Extreme, Core 2 Duo, Pentium EE, Pentium D, Pentium 4, Celeron	DDR2	2	4	1066, 800, 667	74.50
4	Gigabyte X48-DQ6 /X48	LGA775	Core 2 Quad, Core 2 Extreme, Core 2 Duo, Pentium EE, Pentium D, Pentium 4, Celeron	DDR2	4	8	1200, 1066, 800, 667, 533	227.50
5	ASUS P5S800-VM /SIS661FX	LGA775	Pentium4, Celeron	DDR	2	2	400, 333, 266	36.00
6	ASUS P7P55D/ PRO/P55	LGA1156	Core i5, i7	DDR3	4	16	1600, 1333, 1066	315.50
7	INTEL DX58SO/X58/BOX	LGA1366	Core i7	DDR3	4	8	1600, 1333, 1066	421.50

Table 2. CPUs data

#	CPU type	Socket	Clock frequency, GHz	Core number	Price
1	CELERON-D 347	LGA775	3.06	1	59.50
2	C DUO E5200 /800/2M BOX	LGA775	2.50	2	109.00
3	C2 DUO E7600 /1066/3M BOX	LGA775	3.06	2	251.00
4	C2 QUAD Q8200S/1333/BOX	LGA775	2.33	4	381.00
6	C i5-750 /8M/BOX/	LGA1156	2.66	4	371.00
5	C i7-860 /8M/BOX/	LGA1156	2.80	4	531.00
7	C i7-940 /8M/BOX/	LGA1366	2.93	4	979.50

Table 3. RAM modules data

#	RAM Type	RAM size, GB	Frequency, MHz	Price
1	DDR A-DATA	1	400	55.00
2	DDR2 KINGSTON	1	667	41.50
3	DDR2 KINGSTON	1	1066	64.50
4	DDR3 KINGSTON	1	1066	42.50
5	DDR2 KINGSTON	2	800	77.50
6	DDR3 A-DATA	2	1333	77.50
7	DDR3 HYPER X KINGSTON	2	1600	83.00

Note: Data in the tables are actual at the time of manuscript preparation

The compatibility between modules is described by the following sets:

- for MB and CPU compatibility:

$MB_1\text{-}CPU = \{CPU_1, CPU_2, CPU_3, CPU_4\}$ $CPU_1\text{-}MB = \{MB_1, MB_2, MB_3, MB_4, MB_5\}$
$MB_2\text{-}CPU = \{CPU_1, CPU_2, CPU_3, CPU_4\}$ $CPU_2\text{-}MB = \{MB_1, MB_2, MB_3, MB_4\}$
$MB_3\text{-}CPU = \{CPU_1, CPU_2, CPU_3, CPU_4\}$ $CPU_3\text{-}MB = \{MB_1, MB_2, MB_3, MB_4\}$
$MB_4\text{-}CPU = \{CPU_1, CPU_2, CPU_3, CPU_4\}$ $CPU_4\text{-}MB = \{MB_1, MB_2, MB_3, MB_4\}$
$MB_5\text{-}CPU = \{CPU_1\}$ $CPU_5\text{-}MB = \{MB_6\}$
$MB_6\text{-}CPU = \{CPU_5, CPU_6\}$ $CPU_6\text{-}MB = \{MB_6\}$
$MB_7\text{-}CPU = \{CPU_7\}$ $CPU_7\text{-}MB = \{MB_7\}$

- for MB and RAM modules compatibility:

$MB_1\text{-}RAM = \{RAM_2, RAM_3, RAM_5\}$ $RAM_1\text{-}MB = \{MB_5\}$
$MB_2\text{-}RAM = \{RAM_2, RAM_5\}$ $RAM_2\text{-}MB = \{MB_1, MB_2, MB_3, MB_4\}$
$MB_3\text{-}RAM = \{RAM_2, RAM_3, RAM_5\}$ $RAM_3\text{-}MB = \{MB_1, MB_3, MB_4\}$
$MB_4\text{-}RAM = \{RAM_2, RAM_3, RAM_5\}$ $RAM_4\text{-}MB = \{MB_6, MB_7\}$
$MB_5\text{-}RAM = \{RAM_1\}$ $RAM_5\text{-}MB = \{MB_1, MB_2, MB_3, MB_4\}$
$MB_6\text{-}RAM = \{RAM_4, RAM_6, RAM_7\}$ $RAM_6\text{-}MB = \{MB_6, MB_7\}$
$MB_7\text{-}RAM = \{RAM_4, RAM_6, RAM_7\}$ $RAM_7\text{-}MB = \{MB_6, MB_7\}$

3.1. Multiobjective optimization tasks

As the most real life optimization problems are multiobjective in nature, there are many developed methods that could be used to solve them [Andersson, 2000]. An easy to handle approach for solving multiobjective optimization problems is transforming a multiple-objective (vector) problem into single objective (scalar) problem [Miettinen and Makela, 2002]. The most common way of conducting multiobjective optimization is by *a priori* articulation of the decision maker (DM) preferences. That means that before the actual optimization is performed the different objectives are aggregated to one single figure of merit. The a priori information about the user's preferences for different objectives is specified by weight coefficients w_i [Marler and Arora, 2004]. The used linear normalization technique is:

$$f^* = \frac{f - f^{\min}}{f^{\max} - f^{\min}} \quad \text{(for maximizing objectives)}$$

$$f^* = \frac{f^{\max} - f}{f^{\max} - f^{\min}} \quad \text{(for minimizing objectives)}$$

where $f^{\min}$ and $f^{\max}$ are the minimum and maximum values each objective could take.

Accordingly to the *weighted sum* method the multiobjective problem (16) is transformed into a single objective formulation as follows:

(17) $$\max\left(w_1 PC^{cost*} + w_2 CPU^{core*} + w_3 CPU^{clock*} + w_4 RAM^{size*}\right),$$

where $\sum_{f=1}^{4} w_f = 1$.

The optimization task (17) s.t. (2) – (13) is solved three times with three different weight coefficients combinations expressing different DM preferences.

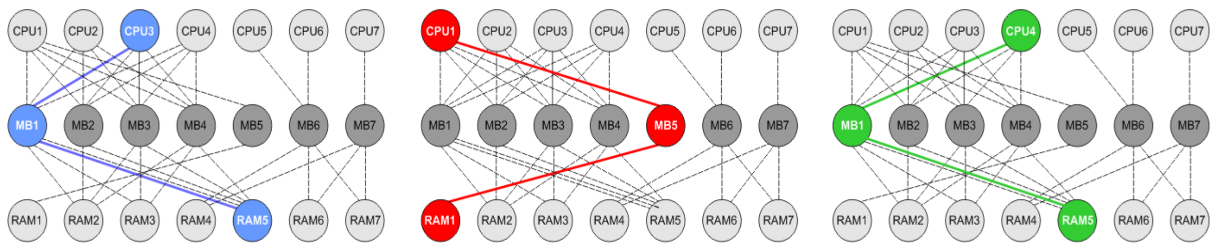
4. RESULTS AND DISCUSSION

The formulated optimization tasks are solved by LINGO[3] ver. 12 on a desktop PC with Intel® Celeron® 2.93 GHz CPU and 2 GB of RAM under MS© Windows XP operating system. The solution times for the described numerical examples are of order of seconds but obviously depend on the size of the problems. The LINGO solver uses *branch-and-bound* method for formulated here nonlinear mixed-integer problems. The Pareto optimal solutions of the three task runs are shown in Table 4.

Table 4. The solutions results

Task	**DM preferences**				**Solution results**			
	w_1 (*for PC cost*)	w_2 (*for CPU core*)	w_3 (*for CPU clock*)	w_4 (*for RAM size*)	**MB**	**CPU**	**RAM**	**PC cost**
1	0.25	0.25	0.25	0.25	MB_1	CPU_3	4 RAM_5 (4x2=8 GB)	739.50
2	0.70	0.10	0.10	0.10	MB_5	CPU_1	1 RAM_1 (1x1=1 GB)	150.50
3	0.15	0.35	0.20	0.30	MB_1	CPU_4	4 RAM_5 (4x2=8 GB)	869.50

The graphical representations of the solutions in Table 5 are illustrated on Fig. 2, where the dashed line shows the existing modules compatibilities.



a) *b)* *c)*

Fig. 2. Graphical representation of optimization tasks solutions
a) solution of Task 1; b) solution of Task 2; c) solution of Task 3

The results of all optimization tasks solutions define Pareto-optimal combinations of MB, CPU and RAM modules satisfying compatibility and user requirements and demonstrate the the applicability of the proposed approach to modular systems design. The solutions of multiobjective tasks show that the different sets of weight coefficients values (i.e. different user preferences) define different modules combinations. The Task 1 is an example of equivalent compromise between all objectives, the Task 2 looks for the cheapest combination of modules while the Task 3 stresses on the computational power of the system. The multiobjective optimization and using of weighted objectives can be seen as a flexible simulation tool for modular systems design. It allows better reflecting of user (decision maker's) preferences by adjusting of the different weight coefficients values.

[3] http://www.lindo.com

Other types of objective functions and solution methods could be used and investigated upon the suitability of the described approach to modular system design. It should be pointed out that the formulated multiobjective optimization tasks are real data examples used to illustrate numerically the proposed approach. The dimensions of real life problems could be much higher. For example, a middle size company for selling of PC modules has a list of more 250 different MBs, 120 CPU types and 70 different RAM modules. That defines a quite complex combinatorial problem with about 2,100,000 modules combinations not to mention considering of other modules. It could be anticipated the other real life examples of modular systems designs would lead to large scale problems demanding for proper mathematical methods to solve them. On the other side, the larger diversity of modules could be an advantage in the sense of defining of larger solution space, but on the other hand, the discrete combinatorial problems tend to increase the computational difficulties with increasing of the problem dimensions. The specifics of the proposed model formulations leads to a sparse constraints matrix and that could be a prerequisite for relaxation of the computational requirements. Taking into account the constantly growing power of the modern computers, it can be expected that the real problems sizes would not be obstacle to the practical applicability of the proposed approach. Further experimentation with other real life large scale problems is needed to justify that. It should be mentioned also that because of the model nonlinearity the solutions of the formulated optimization tasks theoretically could lead to local optimums. Again further investigations and practical experience would show if this is acceptable for the real life applications.

5. CONCLUSION

The design of real life modular technical systems can be a complex combinatorial problem. It should consider many different compatibility, functional and user requirements to get customized and optimized system for particular users and application environments. The current paper proposes a modular technical systems design approach based on discrete choice modelling technique and combinatorial optimization tasks solution. The proposed approach is described and illustrated on the example of PC configuration design. It takes into account the existing compatibility restrictions between PC main modules (MB, CPU, RAM) and can be easily extended and modified to reflect different functional and users' requirements. The developed design modelling technique is used to formulate multiobjective nonlinear discrete mixed-integer optimization tasks. Their solutions provide Pareto-optimal system configurations satisfying all of the given requirements. The practical applicability of the developed approach is tested by numerical examples based on real PC modules data. The multiobjective optimization is performed by using of the *weighted sum* method. Three optimization task formulations with three different sets of objectives' weight coefficients values reflecting different decision maker's preferences are solved.

The results of numerical experiments show the possibility for practical application of the proposed modular systems design approach. Other optimization criteria can be used for better specifying of the user preferences. Also, other methods for solving of the multiobjective optimization tasks can be investigated upon the suitability and advisability for design of modular systems. From a practical prospective, an interesting further research is to consider bigger sets of modules for testing of the corresponding large scale optimization problems.

Acknowledgment

This work has been partly supported by the project "*Application of operational research for decision making in engineering systems*".

References

[1] Andersson J. (2000) A survey of multiobjective optimization in engineering design. Tech. Report: LiTH-IKP-R-1097, Department of Mech. Eng., Linköping University, #34.

[2] Heinrich, M. and Jungst, E. (1991) W. A resource-based paradigm for the configuring of technical systems from modular components. In Proc. Artificial Intelligence Applications, pp. 57-264.

[3] Jae-Kyu, L., S. Sung-Hoon and Suhn-Baum, K. (1996) Configuration of personal computer by constraint and rule satisfaction problem approach. In Proc. First Asian Paciific DSI Conference, pp. 1-22.

[4] Kreng, V.B. and Lee, Tseng-Pin. (2004) Modular product design with grouping genetic algorithm – a case study. Computers & Industrial Engineering, vol. 46, no 3, pp. 443-460.

[5] Levin, M. Sh. (2009) Combinatorial optimization in system configuration design. Automation and Remote Control. vol. 70, no 3, pp. 519-56.

[6] Marler, R.T. and Arora J.S. (2004) Survey of multi-objective optimization methods for engineering. Structural and Multidisciplinary Optimization, vol. 26, pp. 369-395.

[7] McDermott. J. (1982) R1: A rule-based configurer of computer systems. Artificial Intelligence, vol. 19, no 1, pp. 39-88.

[8] Miettinen, K. and Makela, M.M. (2002) On scalarizing functions in multiobjective optimization. OR Spectrum, vol. 24, no 2, pp. 193-213.

[9] Qin, Y. and Wei, G. (2010) Product configuration based on modular product family modelling. Journal of Computational Information Systems, vol. 6, no 7, pp. 2321-2331.

[10] Sanchez, R. (1993) Strategic flexibility, firm organization, and managerial work in dynamic markets: a strategic options perspective. Advances in Strategic Management, vol. 9, pp. 251-291.

[11] Sóbester, A.,·Forrester, A.I.J., Toal, D. J.J., Tresidder, E. and Tucker, S. (2012) Engineering design applications of surrogate-assisted optimization techniques. Optimization and Engineering, DOI 10.1007/s11081-012-9199-x.

[12] Soininen, T., Niemela, I., Tiihonen, J. and Sulonen R. (2000) Unified configuration knowledge representation using weight constraint rules. In Proc. ECAI-2000 Workshop on Configuration, pp. 79-84.

[13] Tam,V. and Ma, K.T. (2000) Using heuristic-based optimizers to handle the personal computer configuration problems. In Proc. 12th IEEE Int. Conf. Tools with Artificial Intelligence, pp.108-111.

[14] Tam,V., Ma, K. T. (2002) Optimizing personal computer configurations with heuristic-based search methods, Artificial Intelligence Review, vol. 17, no 2, pp. 129-140.

[15] Tseng, Tzu-Liang and Huang, Chun-Che (2008) Design support systems: A case study of modular design of the set-top box from design knowledge externalization perspective. Decision Support Systems, vol. 44, pp. 909-924.

[16] Yoo, J. and Kumara, S.R.T. (2010) Implications of k-best modular product design solutions to global manufacturing CIRP Annals – Manufacturing Technology, vol. 59, pp. 481-484.