

A Discrete Choice Modeling Approach to Modular Systems Design

Ivan C. Mustakerov, and Daniela I. Borissova

Abstract—The paper proposes an approach for design of modular systems based on original technique for modeling and formulation of combinatorial optimization problems. The proposed approach is described on the example of personal computer configuration design. It takes into account the existing compatibility restrictions between the modules and can be extended and modified to reflect different functional and users' requirements. The developed design modeling technique is used to formulate single objective nonlinear mixed-integer optimization tasks. The practical applicability of the developed approach is numerically tested on the basis of real modules data. Solutions of the formulated optimization tasks define the optimal configuration of the system that satisfies all compatibility restrictions and user requirements.

Keywords—Constrained discrete combinatorial choice, modular systems design, optimization problem, PC configuration.

I. INTRODUCTION

THE purpose of any system design is to create a technical solution that satisfies the functional and user requirements for the system. One of the most challenging aspects of the design process is to reduce the prospect of costly errors that will be obstructive to mass production. A form of production strategic flexibility is using of modular design by selecting and integration of modular components. The modularity has some essential advantages – decreasing of production costs and order lead-time, flexible designs to respond to the changing markets and technologies but also has some disadvantages – lack of performance optimization and excessive product similarity [1] – [5].

The modular design does not involve the development of new component types but require knowledge of the modules characteristics and their compatibility with other modules. It is of particular interest to apply modeling approaches in an optimization context [6], [7]. Using of optimization methods to solve engineering design problems incorporates a number of disciplines and includes the selection of proper design and decision variables, constraints, objectives and models. One of the prospective direction in system design is using of combinatorial optimization [8]. In that direction optimization is often based on using of 0-1 integer variables and heuristic methods [9].

The current work proposes a discrete choice modeling

approach for modular system design. The proposed approach provides exact optimal solution of formulated single objective nonlinear mixed-integer problem. It is examined through the prism of an easy to understand example of personal computer (PC) system configuration. The users of designed system can have different requirements in respect to the system functionality. For computer system example, business users, home users, scientists, students, gamers and others will have specific requirements toward the system performance. The main problem of modular system design is the existence of large diversity of modules with specific compatibility restrictions. In that case the choice of proper modules is far from trivial combinatorial problem. For the PC example, well-known motherboards (MB) manufacturers as ASUS, ABIT, MSI, Intel and Gigabyte, offer many different MB types. Each motherboard has specific compatibility requirements for central processing unit (CPU), for random access memory modules (RAM) and for other modules it can support. The PC configuration design demonstrates many of the problems inherent to modular system design and in itself is an widely used application example [10]. As a real world problem, the computer system configuration has been approached by different scientific methods – heuristic-based methods [11], [12]; constraint and rule satisfaction problem [13]; weight constraint rule language [14]; rule-based [15], grouping genetic algorithm [16].

The proposed in the paper approach to modular system design is based on original design process modeling technique. It is used to formulate optimization tasks adjustable to different functional and user requirements.

The rest of the paper is organized as follows: the problem specifics are described in Section II; the numerical experimentation and discussions are in Section III and IV and the paper is closed by the conclusions in Section V.

II. THE MODULAR SYSTEM DESIGN SPECIFICS

The formalization of the design process starts with understanding of the designed system specifics. The modeling ultimate goal here is the determination of optimal design configuration taking into account all of the given system specifications, restrictions and requirements. The modular systems design approach proposed in the paper can be structured as steps of the following algorithm:

A. Defining of Modules to be Considered in the Design

The configuration design of an engineering system includes the choice of modules to provide required system functionality. For the example of personal computer system

I. C. Mustakerov is with the Institute of Information and Communication Technology at the Bulgarian Academy of Sciences, Sofia – 1113, Bulgaria (phone: 3952 9793241; e-mail: mustakerov@iit.bas.bg).

D. I. Borissova is with the Institute of Information and Communication Technology at the Bulgarian Academy of Sciences, Sofia – 1113, Bulgaria (phone: 3952 9792055; e-mail: dborissova@iit.bas.bg).

configuration problem, MB, CPU and RAM modules can be considered as a good illustration of basic modules that define most of the system functionality (Fig. 1).

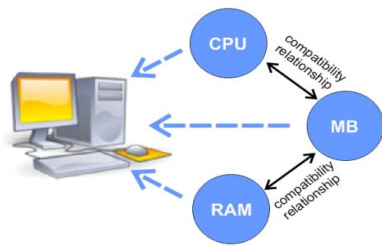


Fig. 1 PC main modules

B. The Modules Compatibility Relationship

In general, there exists a compatibility relationship between modules that should be taken into account when choosing a particular module. The specific of this relationship between MB, CPU and RAM is described as follows:

- MB (motherboard) is usually the first basic module to start the PC configuration design process. It has specific requirements about the rest of the modules. For example, each MB has a particular type of CPU socket that determines what type of processor can be used, specific chipset that connects the CPU to other modules of the computer and a number of slots and ports (PCI, PCI-Express, AGP, IDE, SATA, USB, FireWire, RAM slots, etc.) defining what and how many peripheral devices can be connected. The MB memory specifications determine type and size of RAM that can be supported. Some MBs also incorporate newer technological advances as RAID, on-board sound, networking, video etc., that also could be considered in the design process.
- CPU (*processor*) is another important module that strongly influences the most important computer capability – computational performance. The two of the leading producers of CPU's – Intel® and AMD® each have different CPU designs compatible with different CPU sockets found on the motherboards. For example, Socket 478 for the Pentium IV and Celeron processors; Socket 775 for the Pentium, Celeron, Core™ 2 Duo, Quad and Extreme processors; Socket 1156 for Core™ i7 and i5 processors and Socket 1366 for the Core™ i7 product line. The AMD sockets include Socket A for the Athlon™ and Duron™ CPU's, Socket AM2 for the Athlon™ 64 and Sempron™ processors, Socket AM2+ for the Phenom™ and Athlon™ 64 product lines, and Socket F for the Opteron™ and Athlon™ 64 7x processors and so on.
- RAM (*memory*) is other main component that has a significant effect on PC performance. The motherboard's specifications define what type and how much memory could be supported. For example, some of RAM types as DDR, DDR2, DDR3, DDR4 have different characteristics (working frequency, modules capacity, etc.) that are important for the computational capabilities of the computer.

Other PC modules as hard disk storage, video controller, RAID controller, optical storage devices, case and power supply, keyboard, etc., could also be considered in problem formulation but the choice of the above basic modules is adequate to illustrate the proposed approach to modular systems design.

C. The Design Objectives

The optimization of any system design is connected with defining of design objective (optimization criterion). The cost of any system design is always worth to consider and is used here as a design objective to illustrate the applicability of the proposed approach.

It is important to design systems optimized for particular user and application environment. The user requirements describe the user-level aspects of the system and are the basis for the acceptance of the system. For the PC configuration example, including of some user requirements about CPU, RAM and RAID option will be demonstrated when modeling the choice process.

D. Mathematical Modeling and Optimization Task Formulation

Defining of an adequate mathematical model of design process is a better alternative to intuitive decision making. The design process basically consists of two problems: the generation of design alternatives and decision making about the choice of the best alternative. Mathematical model proposed in the paper optimizes the modular systems design by providing optimal modules combination. Three considerations that could be related to any system design – costs, compatibility issues and user requirements are illustrated in the paper. The developed mathematical modeling approach can be described as follows:

1) Defining of the Nomenclature

Using of optimization methods to solve engineering design problems involves as a key stage the selection of design and decision variables, constraints, objectives, etc. The nomenclature of indexes, sets, variables and constants used for the example of PC configuration design is shown in Table I.

2) Modeling of the Choice Process

The modeling of modular system design process is in fact modeling of a combinatorial choice problem. The modules choice process can be formalized by using of binary integer variables taken as *decision variables*. The modules parameters that are to be considered in the choice process are treated as *design variables*. The decision and design variables are used to formalize the selection process of the modules. The following relations specify the choice of MB and CPU:

$$MB^{cost} = \sum_{i \in I} MB_i^{cost} x_i^{mb}, x_i^{mb} \in \{0, 1\}, \sum_{i \in I} x_i^{mb} = 1 \quad (1)$$

$$CPU^{cost} = \sum_{j \in J} CPU_j^{cost} x_j^{cpu}, x_j^{cpu} \in \{0, 1\}, \sum_{j \in J} x_j^{cpu} = 1 \quad (2)$$

$$CPU^{clock} = \sum_{j \in J} CPU_j^{clock} x_j^{cpu} \quad (3)$$

$$CPU^{core} = \sum_{j \in J} CPU_j^{core} x_j^{cpu} \quad (4)$$

$$MB^{RAID} = \sum_{i \in I} MB_i^{RAID} x_i^{mb}, MB^{RAIDmin} = 1 \quad (5)$$

TABLE I
INDEXES, SETS, VARIABLES AND CONSTANTS

Notation	Description
indexes	
i	index of MB, $i \in I, I \subset N^*$
j	index of CPU, $j \in J, J \subset N^*$
k	index of RAM modules, $k \in K, K \subset N^*$
sets	
I	set of indexes for different types of MBs
J	set of indexes for different types of CPUs
J_i	$J_i \subseteq J$ subset of indexes of the CPUs compatible with motherboard of type i .
K	set of indexes for different types of RAM modules
K'	$K' \subseteq K$, subset of indexes for RAM modules compatible with motherboard of type i
K''	$K'' \subseteq K$, subset of indexes for RAM modules incompatible with motherboard of type i
N^*	set of positive integers
N	set of nonnegative integers
decision variables	
x_i^{mb}	binary integer variables for MB choice, $x_i^{mb} \in \{0,1\}$
x_j^{cpu}	binary integer variables for CPU choice, $x_j^{cpu} \in \{0,1\}$
x_k^{ram}	integer variables for RAM modules choice, $x_k^{ram} \in N$
design variables	
MB^{cost}	MB cost
MB^{RAID}	MB RAID support
CPU^{cost}	CPU cost
CPU^{clock}	clock frequency of the CPU
CPU^{core}	core number of the CPU
RAM^{cost}	RAM cost
RAM^{size}	RAM size
modules parameters	
$MB_i^{CPUsocket}$	CPU socket type of MB _{<i>i</i>}
$MB_i^{CPUtype}$	CPU type supported by MB _{<i>i</i>}
$MB_i^{RAMtype}$	RAM type supported by MB _{<i>i</i>}
$MB_i^{RAMslots}$	number of available RAM slots for MB _{<i>i</i>}
MB_i^{RAMmax}	maximal RAM size supported by MB _{<i>i</i>}
$MB_i^{RAMfreq}$	RAM frequency supported by MB _{<i>i</i>}
MB_i^{RAID}	RAID option of MB _{<i>i</i>} (0 - No, 1 - Yes)
MB_i^{cost}	MB _{<i>i</i>} cost (price)
CPU_j^{socket}	CPU _{<i>j</i>} socket type
CPU_j^{clock}	CPU _{<i>j</i>} clock frequency
CPU_j^{core}	CPU _{<i>j</i>} core number
CPU_j^{cost}	CPU _{<i>j</i>} cost (price)
RAM_k^{size}	size of k -type RAM module
RAM_k^{freq}	work frequency of k -type RAM module
RAM_k^{cost}	cost (price) of k -type RAM module
$RAM^{size min}$	minimal RAM size
$CPU^{clock min}$	minimal CPU clock frequency
$CPU^{core min}$	minimal CPU core number
$MB^{RAID min}$	MB RAID support (no - 0 or yes - 1)

Formalizing the choice of RAM modules is more complicated because multiple types and number of RAM modules could be chosen simultaneously. For example, if maximal memory slots are 3 and the required RAM size should be 4 GB then memory could be combined by two modules of 2 GB (one empty slot) or by two modules of 1 GB plus one 2 GB module. This choice is modeled by using of nonnegative integer decision variables x_k^{ram} designating the number of RAM modules of type k as:

$$RAM^{cost} = \sum_{k \in K} RAM_k^{cost} x_k^{ram}, x_k^{ram} \in N \quad (6)$$

$$RAM^{size} = \sum_{k \in K} RAM_k^{size} x_k^{ram} \quad (7)$$

Similarly, other parameters of the designed system can be considered in the modeling of the choice process.

3) Compatibility Issue

It is very important to take into account the functional compatibility relationship between modules to get a working system. For the goal of mathematical modeling subsets of compatible modules are defined. Then mathematical expression of the compatibility restrictions is formulated as:

- MB and CPU compatibility:

$$\forall i \in I: x_i^{mb} \leq \sum_{j \in J_i} x_j^{cpu}, \quad (8)$$

where $J_i \subseteq J$ are indexes of the CPUs compatible with motherboard of type i . The numerical experiments show that restrictions (8) are sufficient to get choice of compatible with each other MB and CPU.

The RAM modules also have to be compatible with chosen MB. The numerical experiments show the necessity of two types of constraints:

- for MB and RAM modules compatibility:

$$\forall i \in I: x_i^{mb} \sum_{k \in K_i'} x_k^{ram} \geq 0, K_i' \subseteq K \quad (9)$$

- for MB and RAM modules incompatibility:

$$\forall i \in I: x_i^{mb} \sum_{k \in K_i''} x_k^{ram} \leq 0, K_i'' \subseteq K \quad (10)$$

The maximal RAM size and RAM modules number are limited by the MB capabilities but there should be at least one RAM module to get functional design. These facts are taken into account by constraints:

- for maximal RAM size supported by the MB:

$$RAM^{size} \leq \sum_{i \in I} MB_i^{RAMmax} x_i^{mb} \quad (11)$$

- for RAM slots supported by the MB:

$$\sum_{k \in K} x_k^{ram} \leq MB_i^{RAMslots} \quad (12)$$

- for functional design

$$\sum_{k \in K} x_k^{ram} \geq 1 \quad (13)$$

4) User Requirements

The user requirements about the designed system parameters are expressed as constraints for the design variables. For example:

- for RAM size:

$$RAM^{size} \geq RAM^{size\ min} \quad (14)$$

- for CPU clock frequency:

$$CPU^{clock} \geq CPU^{clock\ min} \quad (15)$$

- for CPU core number:

$$CPU^{core} \geq CPU^{core\ min} \quad (16)$$

- for availability of RAID option:

$$MB^{RAID} = MB^{RAID\ min} \quad (17)$$

Following that approach other user requirements about the designed system parameters can be introduced.

5) Optimization Tasks Formulation

The described modeling technique is used for formulation of proper optimization tasks. If cost is considered as design objective then optimal PC configuration is defined by solution of the following mixed integer nonlinear optimization task:

$$\min (MB^{cost} + CPU^{cost} + RAM^{cost}) \quad (18)$$

subject to (1) – (17).

The solution of this optimization task will define optimal choice of modules satisfying the compatibility restrictions and given user requirements.

III. NUMERICAL EXPERIMENTATION

The proposed approach to discrete choice modeling of modular system design is numerically tested on real PC modules data. The modules parameters are shown in Table II, Table III, and Table IV. The compatibility relationship between MB, CPU and RAM modules are summarized in Table V. Three single objective optimization tasks for three different combinations of user requirements are formulated:

- *Task 1a:* (18) s.t. (1) – (14) – with user requirement for $RAM^{size\ min} = 2$ GB

- *Task 1b:* (18) s.t. (1) – (16) – with user requirements for $RAM^{size\ min} = 3$ GB, $CPU^{clock\ min} = 2.5$ GHz and $CPU^{core\ min} = 2$

- *Task 1c:* (18) s.t. (1) – (17) – with user requirements for $RAM^{size\ min} = 4$ GB, $CPU^{clock\ min} = 2.5$ GHz, $CPU^{core\ min} = 2$ and additional requirement for the availability of RAID support, i.e. $MB^{RAID\ min} = 1$.

TABLE II
MOTHERBOARDS DATA

<i>i</i>	MB TYPE	CPU		RAM			MB^{RAID}	MB^{cost} BGL	
		$MB_i^{CPUsocket}$	$MB_i^{CPUtype}$	$MB_i^{RAMtype}$	$MB_i^{RAMslots}$	MB_i^{RAMmax} GB			$MB_i^{RAMfreq}$ MHz
1	ASROCK P45XE-WIFIN/P45	LGA775	Intel® Core™ 2E Core™ 2 Quad Core™ 2 Duo Pentium® Dual Core Celeron® Dual Core Celeron®	DDR2	4	16	1200 1066 800 667	1 (Yes)	178.50
2	ASROCK G31M-S/G31	LGA775	Intel® Core™ 2E Core™ 2 Quad Core™ 2 Duo Pentium® Dual Core Celeron® Dual Core Celeron®	DDR2	2	8	800 667	0 (No)	69.00
3	Gigabyte G31M-ES2C/G31	LGA775	Intel® Core™ 2E Core™ 2 Quad Core™ 2 Duo Pentium® EE Intel® Pentium® D Intel® Pentium® 4 E Intel® Pentium® 4 Intel® Celeron® Intel® Core 2 Quad Core 2 Extreme Core 2 Duo	DDR2	2	4	1066 800 667	0 (No)	74.50
4	Gigabyte X48-DQ6 /X48	LGA775	Pentium EE Pentium D Pentium 4 Celeron	DDR2	4	8	1200 1066 800 667 533	1 (Yes)	227.50
5	ASUS P5S800-VM /SIS661FX	LGA775	Intel® Pentium4 Celeron	DDR	2	2GB	400 333 266	1 (Yes)	36.00
6	ASUS P7P55D/ PRO/P55	LGA1156	Intel® Core i5, i7	DDR3	4	16GB	1600 1333 1066	1 (Yes)	315.50
7	INTEL DX58SO/X58/BOX	LGA1366	Intel® Core i7	DDR3	4	8GB	1600 1333 1066	1 (Yes)	421.50

TABLE III
CPU DATA

<i>j</i>	CPU type	CPU_j^{socket}	CPU_j^{clock}	CPU_j^{core}	CPU_j^{cost}
1	CELERON-D 347	LGA775	3.06 GHz	1	59.50
2	Core DUO E5200 /800/2M BOX	LGA775	2.50 GHz	2	109.00
3	Core2 DUO E7600 /1066/3M BOX	LGA775	3.06 GHz	2	251.00
4	Core2 QUAD Q8200S/1333/BOX	LGA775	2.33 GHz	4	381.00
6	C i5-750 /8M/BOX/	LGA1156	2.66 GHz	4	371.00
5	C i7-860 /8M/BOX/	LGA1156	2.80 GHz	4	531.00
7	C i7-940 /8M/BOX/	LGA1366	2.93 GHz	4	979.50

TABLE IV
RAM MODULES DATA

<i>k</i>	RAM type	RAM_k^{size} GB	RAM_k^{freq} MHz	RAM_k^{cost} BGL
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

TABLE V
COMPATIBILITY RELATIONSHIP BETWEEN MODULES

MB	Compatible CPU	Compatible RAM	Incompatible RAM
MB_1	$CPU_1, CPU_2, CPU_3, CPU_4$	RAM_2, RAM_3, RAM_5	$RAM_1, RAM_4, RAM_6, RAM_7$
MB_2	$CPU_1, CPU_2, CPU_3, CPU_4$	RAM_2, RAM_5	$RAM_1, RAM_3, RAM_4, RAM_6, RAM_7$
MB_3	$CPU_1, CPU_2, CPU_3, CPU_4$	RAM_2, RAM_3, RAM_5	$RAM_1, RAM_4, RAM_6, RAM_7$
MB_4	$CPU_1, CPU_2, CPU_3, CPU_4$	RAM_2, RAM_3, RAM_5	$RAM_1, RAM_4, RAM_6, RAM_7$
MB_5	CPU_1	RAM_1	$RAM_2, RAM_3, RAM_4, RAM_6, RAM_7$
MB_6	CPU_5, CPU_6	RAM_4, RAM_6, RAM_7	$RAM_1, RAM_2, RAM_3, RAM_5$
MB_7	CPU_7	RAM_4, RAM_6, RAM_7	$RAM_1, RAM_2, RAM_3, RAM_5$

The formulated optimization tasks can be solved by means of any of available optimization solvers. In the paper LINGO solver [17] is used for solution of the numerical examples. The optimal solution results of the tasks are shown in Table VI.

TABLE VI
THE RESULTS OF TASKS SOLUTIONS

Task	User requirements				Solution results,			Cost (BGL)
	RAM	CPU clock	CPU core	RAID MB	CPU	RAM		
1a	≥ 2 GB	--	--	--	MB5: ASUS- P5S800- VM /SIS661F X RAID	CPU1: CELER OND 347, 3.06 GHz	RAM1: 2 x 1 GB DDR A-DATA 400 MHz	205.5
					MB2: ASROCK G31M- S/G31 no RAID	CPU2: Core DUO E5200 /800/2M	RAM2: 1 GB DDR2 667 MHz, RAM5: 2 GB DDR2 800 MHz	
1b	≥ 3 GB	≥ 2.5GHz	≥ 2	--	MB1: ASROCK - P45XE- WIFIN/P 45 RAID	CPU3: Core2 DUO E7600 /1066/3 M BOX, 3.06 GHz	RAM5: 2 x 2 GB DDR2 800 MHz	297.0
1c	≥ 4 GB	≥ 3 GHz	≥ 2	1 (yes)				584.5

IV. DISCUSSION OF THE NUMERICAL EXPERIMENTATION

The results of all optimization tasks solutions define optimal combinations of MB, CPU and RAM modules satisfying compatibility and user requirements. Thus, the solutions demonstrate the adequacy of the used modeling technique and the applicability of the proposed approach to modular systems design. Different formulations of the tasks show the easiness of introducing of user requirements about the parameters of the designed system.

Other objective functions and optimization tasks formulations could be used and investigated upon the suitability of the described approach to modular systems design. It should be pointed out that the solved optimization tasks are examples of real problems and are used to illustrate numerically the proposed approach. The tasks solution times are of order of seconds on a desktop PC with Intel® Celeron® 2.80 GHz CPU and 1.96 GB of RAM under MS® Windows XP operating system. It could be anticipated that other real life examples of modular systems designs would have large scale problems demanding for proper mathematical methods to solve them. For example, a middle size company for computer modules could offer a list of hundreds of different modules.

Generally speaking, 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. In general, the class of nonlinear mixed-integer problems would be NP-hard to solve but the specifics of the proposed model formulation lead to a sparse constraints coefficients matrix and is a prerequisite for relaxation of the computational difficulties. Taking into

account the constantly growing computational power of the modern computers, it could be expected the large scale problems would not be obstacle for the practical applicability of the proposed approach. Future studies of other real-world applications will help to provide a methodology for modular system design based on the proposed approach.

V. CONCLUSION

A discrete choice modeling technique is developed for modular system design via optimal configuration of modules. The described approach is based on single objective nonlinear discrete mixed-integer optimization task formulation. The task solution defines optimal system configuration satisfying both compatibility and functional restrictions and given user requirements.

The proposed approach to modular system design is numerically tested on real example of PC configuration. The test results demonstrate its practical applicability. The described approach can be modified and extended to be used for other real life problems of modular systems design.

ACKNOWLEDGMENT

The research work reported in the paper is partly supported by the project "AComIn – Advanced Computing for Innovation", grant 316087, funded by the FP7 Capacity Programme (Research Potential of Convergence Regions).

REFERENCES

- [1] P. T. Kidd. *Agile Manufacturing: Forging New Frontiers*, Addison Wesley, New York, 1994.
- [2] J. K. Gershenson, G. J. Prasad. Modularity in product design for manufacturability. *Int. J. of Agile Manufacturing*, vol. 1, no 1, pp. 1-11, 1997.
- [3] Tzu-Liang (Bill) Tseng, Chun-Che Huang. Design support systems: A case study of modular design of the set-top box from design knowledge externalization perspective, *Decision Support Systems*, vol. 44, no 4, pp. 909-924, 2008.
- [4] J. T. Dorsey, T. J. Collins, W. R. Doggett, R. V. Moe. Framework for defining and assessing benefits of a modular assembly design approach for exploration systems. in *Proc. Space Technology and Applications International Forum – STAIF 2006*, vol. 813, pp. 969-981.
- [5] A.K. Kamrani, E.A. Nasr. *Collaborative Engineering*. Springer, 2008, ch. 10.
- [6] A. Söbester, A. I.J. Forrester, D. J.J. Toal, E. Tresidder, S. Tucker, Engineering design applications of surrogate-assisted optimization techniques, *Optimization and Engineering*, DOI 10.1007/s11081-012-9199-x, 2012.
- [7] K. Fujita. Product variety optimization under modular architecture. *Computer-Aided Design*, vol. 34, no. 12, pp. 953-965, 2002.
- [8] M. S. Levin. Combinatorial Optimization in System Configuration Design, *Automation and Remote Control*, vol. 70, no. 3, pp. 519-561, 2009.
- [9] K. Fujita, H. Sakaguchi, S. Akagi. Product variety deployment and its optimization under modular architecture and module communalization. *Proc. of the ASME Design Engineering Technical Conferences*, 1999, Las Vegas, Nevada, DETC99/DFM-8923.
- [10] Re-Designing The Computer: The Birth of the Modular Computer". Xi3 Corporation. http://xi3.com/white_paper.pdf.
- [11] V. Tam, K. T. Ma. Using heuristic-based optimizers to handle the personal computer configuration problems, in *Proc. 12th IEEE Int. Conf. on Tools with Artificial Intelligence*, 2000, pp. 108-111.
- [12] V. Tam, K. T. Ma. Optimizing personal computer configurations with heuristic-based search methods, *Artificial Intelligence Review*, vol. 17, no 2, pp. 129-140, 2002.

- [13] L. Jae-Kyu, S. Sung-Hoon, K. Suhn-Baum. Configuration of personal computer by constraint and rule satisfaction problem approach. in *Proc. First Asian Pacific DSI Conference*, Hongkong, 1996, pp. 1-22.
- [14] T. Soininen, I. Niemela, J. Tiihonen, R. Sulonen. Unified configuration knowledge representation using weight constraint rules, in *Proc. ECAI-2000 Workshop on Configuration*, pp. 79-84, Berlin, 2000.
- [15] J. McDermott. R1: A Rule-based Configurer of Computer Systems, *Artificial Intelligence*, vol. 19, no. 1, pp. 39-88, 1982.
- [16] V. B. Kreng, Tseng-Pin Lee., Modular product design with grouping genetic algorithm – a case study. *Computers & Industrial Engineering*, vol. 46, no. 3, pp. 443-460, 2004.
- [17] Lindo Systems, <http://www.lindo.com>.