Multiple-choice decision making by multicriteria combinatorial optimization

Ivan Mustakerov, Daniela Borissova and Evgeny Bantutov

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

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

Abstract: The paper concerns a problem associated with constrained discrete multiple-choice decision making. An flexible combinatorial optimization modeling approach is developed for choice of more than one devices complying with different decision maker requirements. The described approach is based on formulation of multicriteria linear mixed integer optimization tasks. The solutions of the tasks define Pareto-optimal selections of given number of objects. The case study numerical testing on the example of laptops choice show the practical applicability for solving of constrained discrete multiple-choice problems.

Keywords: Decision making; constrained multiple-choice; multicriteria combinatorial optimization.

1. Introduction

The choice problems naturally extend to decision making by means of proper mathematical model developing. The main problem of constrained choice is the trade-off in decision because of existing constraints and choice preferences and in general, the consideration of all of them leads to NP discrete combinatorial problems [Rao 2008; Chunga & Demange, 2008]. In many cases optimization approaches are used to determine the best choice among a set of discrete alternatives. In modern economies the service sector demands for scientific approach to explore the theoretical and practical aspects underlying the science of service systems including combinatorial discrete choice modeling [Cook et al, 1997; Korte & Vygen, 2010]. The complex decision making problems encountered in business, engineering, and other areas of human activity can be approached by multiple criteria decision making. Two complementary areas of multiple criteria decision making are multiple objective programming and decision maker-driven multiple criteria decision analysis [Collette & Siarry, 2004]. The recent advances in mathematical optimization, software development, and computer technology are motivation to develop more effective and flexible decision making techniques.

Constrained discrete multiple-choice problems arise when a given number of objects have to be selected from a finite set of objects with different parameters. For example, if an organization has to make a large scale order for some objects, (laptops, cars, etc.) it has to make a reasoned decision about the choice of particular object type or model. The most time and money consuming approach is to test each of the available types of objects and to decide which one best fits to some given requirements. The more effective approach is to test only a part of devices that best comply with the requirements and to choose among them. To realize that approach some mathematically reasoned method for choice of a subset of objects is needed. For the goal our paper proposes a combinatorial optimization based approach for devices multiple-choice. The multiple-choice case differs from single-choice case in providing more than one objects selection as a result of single task solution. It is described by an easy to understand example of portable computer systems (laptops) choice. The computer system choice problem is addressed mostly indirectly. One of the earliest and well known

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

Grosch's law, formulated for computer systems price and performance relation [Shoval & Lugasi, 1987] is debated now. Three models (additive-weight model, the Eigenvector model and multi-attribute utility model) for computer system evaluation and selection with identical results are described in [Shoval & Lugasi, 1988]. Other publications include: cost-benefit approach for evaluating and selecting computer systems [Ein-Dor, 1985]; modeling of computer characteristics that influence significantly the price [Harris & Dave, 1994]; logic scoring of preference quantitative decision method for evaluation, comparison, and selection of complex hardware and software systems [Dujmovic, 1996]; three-phase regression analysis approach to forecast the selling price of a notebook computer as a function of its constituent features [Derek & Wilbert, 1999].

In the current paper laptop choice example is used to propose a different approach directly reflecting the user requirements. It is based on discrete combinatorial modeling technique that leads to multicriteria optimization. The developed combinatorial optimization approach is suitable for multiple-choice of Pareto-optimal alternatives. Multicriteria linear mixed integer optimization tasks are formulated taking into consideration the object key features and different user requirements. The multicriteria formulation is chosen because of the fact that it reflects more closely the real world choice problems and could be used to guide decision maker (DM) in a manner beneficial to pre- and post-phase decision making. Taking into account the availability of powerful solvers for mathematical programming models the described multicriteria combinatorial optimization approach to constrained multiple-choice decision making can be effectively applied.

2. The choice problem specifics

An essential and obligatory step for any proper model definition is to understand the particular modeling object specifics. The proposed in the current paper idea is described on the example of laptop choice problem. As many other real life choice problems this example is characterized by the existence of a large diversity of laptops models available on the market. The major laptops manufacturers (Acer, Asus, Dell, Gateway, Hewlett-Packard, Lenovo, Sony, Toshiba, etc.) offer plenty of models options and in practice most of the different laptops features combinations are be taken into account. That transforms the laptop choice problem in a complex discrete combinatorial problem. To demonstrate multiple-choice modeling complexities some laptop’s key features are shortly described.

- *Battery life:* Because of the fact that the mobility is one of the major laptop characteristics, the battery life could be the first in list of the important features. It varies amongst the different models and the common rule is that larger battery life is better.
- *Weight:* Another of the main laptop characteristics is the portability and it essentially depends on the laptop weight. It is usually connected with the laptop size and what is best depends on the desired level of portability and functionality relation.
- *Screen size:* The screen size could be the next of important laptop choice criteria. The smaller screens mean lighter and more portable laptops but a bigger screen is more convenient to work with. The choice of screen size is also a compromise between working convenience and mobility.
- *RAM:* Random Access Memory (RAM) as computer main working memory is essential for computer system performance. Choosing of laptop with as much RAM as it is possible to afford is a good strategy. On the other hand, it is better to consider the real RAM needs and not to rely on future upgrades which could be expensive and not well justified taking into account the constant laptops technologies progress.

- *Processors:* The central processing unit (CPU) popularly is described as one of the most important component in the computer. There is a lot of competition between manufacturers of processors but the relation between processor power and battery consumption is what really matters for the laptops choice. The best choice would be powerful processor (i.e. with higher speed, multi-core capabilities) that uses less power and runs cooler (a plus for the laptops).
- *Permanent storage (HDD, SSD):* The permanent storage is for saving and loading of files and programs. It was the rule of thumb the choice of a laptop with larger permanent memory (meaning HDD) capacity. The HDD type and size however have essential influence on the weight and power consumption. When the portability and mobility are of top priority the smaller HDD or SSD (solid state device) storage instead could be preferable.
- *Optical drives:* Optical storage drives (CD/DVD devices) used to be standard on almost all laptops. Similarly to HDD they also add weight and shorten the operational battery time. Currently the lightest laptops do not have optical storage. Depending on the user preferences, the presence of optical storage could be another option to consider.
- *External ports and slots:* Laptops come with a number of external ports and slots for connecting of external peripheral devices. Having more of them is definitely an advantage. For example, recently the most peripheral devices are connected through USB port and the number of these ports could be another choice criterion.
- *Warranty:* The reliability is an important feature. Because of the laptop specifics (packing components in small space) the chances of having failures are greater than with desktop models. The manufacturers supply different warranty periods and the longer that period is the better is the choice.
- *Price:* This could be last in the list but definitely not least laptop characteristic that always is worth to be considered.

Other laptops features as *graphics card, wireless capabilities, operating system*, etc. can also be considered but the listed above are adequate for illustration of the proposed in the paper combinatorial optimization idea.

The multiple-choice decision making problem is formulated as choice of n Pareto-optimal laptops from a range of different laptops, given the decision maker (DM) requirements and criteria for the parameters of the device.

3. Multiple-choice model definition

A common way to approach the choice problems is by using of binary integer variables x_i as decision variables [Mustakerov & Borissova, 2007]. They define if a particular object i has been chosen ($x_i = 1$) or not ($x_i = 0$) from set of L different objects. These variables are usually used for single object choice and are defined as:

(1) $$\sum_{i=1}^{L} x_i = 1,\ \forall x_i \in \{0,1\}.$$

The multiple-choice decision making arises when the DM would like to have a number of several alternatives to choose from. The multiple-choice can also be modeled by means of the decision variables x_i but they have to be introduced in a non traditional way:

(2) $$\sum_{i=1}^{L} x_i = n,\ \forall x_i \in \{0,1\},\ 1 \le n < L.$$

This definition of the decision variables is more general and contains as a special case the single-choice case for n = 1. Using x_i as decision variables the choice process considering different object features can be modeled by constraints of type:

$$feature^j = \sum_{i=1}^{L} feature_i^j x_i \text{ , } j \in \{1,2,\ldots,J\} \tag{3}$$

where i, j are indexes of objects and their features, $feature_i^j$ is constant representing numerical value of the j^{th} feature for the i^{th} object, L and J are objects and features numbers.

It is common for the DM to have some special requirements about the parameters of the object to be selected. These requirements can be generalized as:

$$feature_{min}^j \le feature^j \le feature_{max}^j , j \in \{1,2,\ldots,J\} \tag{4}$$

The multiple-choice decision making algorithm based on multicriteria combinatorial optimization can be structured as shown on the Fig. 1.

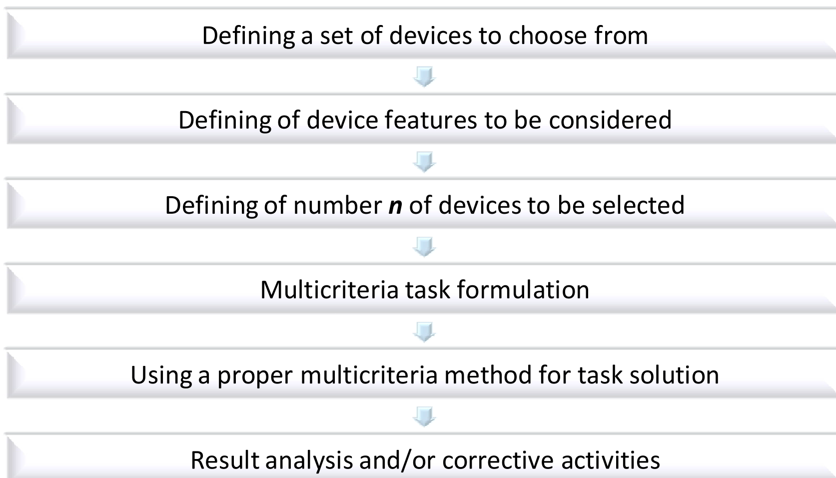

Figure 1. Multiple-choice decision making algorithm by multicriteria optimization

For laptop choice example some laptops' features (*BatteryLife, RAM, Screen, CPUclock, CPUcore, HDD, Warranty, Weight* and *Price*) are considered to be nonzero discrete variables. The laptops' features could also be treated as utility criteria to be optimized. For example, *battery life*, *screen size*, *RAM capacity*, *CPU clock* and *core number*, *HDD capacity, installed OS* are good candidates for maximization while *weight* and *price* are minimization candidates. If the data of each i^{th} laptop features are known constants then a multicriteria optimization model for multiple-choice can be defined:

$$\begin{aligned} &max\,\{BatteryLife,\ RAM,\ Screen,\ CPUclock,\ CPUcore,\ HDD\} \\ &min\,\{Weight,\ Price\} \end{aligned} \tag{5}$$

subject to

$$BatteryLife = \sum_{i}^{L} BatteryLife_i x_i \tag{6}$$

$$RAM = \sum_{i}^{L} RAM_i x_i \tag{7}$$

$$Screen = \sum_{i}^{L} Screen_i x_i \tag{8}$$

$$CPUclock = \sum_{i}^{L} CPUclock_i\, x_i \tag{9}$$

(10) $$CPUcore = \sum_{i}^{L} CPUcore_i x_i$$

(11) $$HDD = \sum_{i}^{L} HDD_i x_i$$

(12) $$Weight = \sum_{i}^{L} Weight_i x_i$$

(13) $$Price = \sum_{i}^{L} Price_i x_i$$

(14) $$OS = \sum_{i}^{L} OS_i x_i$$

(15) $$\sum_{i=1}^{L} x_i = n, \forall x_i \in \{0,1\}, 1 \le n < L.$$

To reflect the DM preferences about particular laptop features some additional constraints can be introduced. For example, the requirements for *RAM* capacity to be not less then RAM^{min} value and *Price* to be less then $Price^{max}$ value can be taken into account as:

(16) $$RAM_i x_i \ge RAM^{min}, \forall i \in \{1,...,L\}$$

(17) $$Price_i x_i \le Price^{max}, \forall i \in \{1,...,L\}$$

Also if different laptops have different operating system installed and the available operating systems are coded with numbers OS_i then the requirement for choosing of laptop with particular OS_k is:

(18) $$OS_i x_i = OS_k, \forall i \in \{1,...,L\}$$

Similar restrictions for other device features can be added to consider different DM requirements. The solution of the multiobjective optimization problem (5) – (18) will define *n* Pareto-optimal laptops conforming to the DM preferences.

4. Numerical illustration

There exist many developed methods that can be used for solution of multicriteria optimization tasks. The choice of particular method depends on when and how the decision maker expresses the preferences on the different objectives. The most commonly used methods for multicriteria task solution substitute several objective functions by a single one to get the advantages of the available single objective optimization solvers. One of the most popular among them is *weighted sum* method and it is used for numerical experiments in the paper. It is preferred when a direct specification of the objectives importance is used and is quite adequate for many practical applications.

The proposed multicriteria combinatorial optimization approach to multiple-choice decision making problems is illustrated by a case study based on real laptops' features data (Table 1). The *weighted sum* method is based on *a priori* information about the user's preferences for different objectives specified by weight coefficients w_i, $\sum_i w_i = 1.0$ [Marler & Arora, 2004].

The weight coefficients are applied to the *normalized* objective functions to define a single scalar objective function. The linear normalization technique used 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.

Table 1. Laptops features data

#	Laptop model	Battery duration [hours]	Weight [kg]	Screen [inches]	RAM [GB]	CPU clock [GHz]	CPU core [number]	HDD [GB]	Optical storage	OS	Price [$]
1	Asus EEE PC 4G SURF	4	0.92	7	0.5	0.9	1	4	0 (no)	1 (Linux)	420
2	Asus EEE PC 900HA-BLK005L	6	1.12	8.9	1	1.6	1	160	0 (no)	1 (Linux)	480
3	HP Compaq 2133 FU342EA	3	1.20	8.9	2	1.6	1	120	0 (no)	1 (Linux)	588
4	Toshiba NB 100-10Y	2.2	1.05	8.9	1	1.60	1	120	0 (no)	2 (WinXP)	799
5	Asus EEE PC 1000H	7	1.40	10	1	1.6	1	160	0 (no)	1 (Linux)	600
6	Lenovo IdeaPad S10e NS84JBM	6	1.30	10.1	1	1.6	1	160	0 (no)	1 (Linux)	518.4
7	Lenovo IdeaPad S10e NS8RFBM	5.20	1.30	10.1	2	1.6	1	160	0 (no)	1 (Linux)	614.4
8	Asus EEE PC S101H-BLK045X	3	1.10	10.2	1	1.66	1	160	0 (no)	1 (Linux)	816
9	Acer ASPIRE AS1410-743G25n	3	1.40	11.6	3	1.30	1	250	0 (no)	1 (Linux)	798
10	Asus EEE PC 1101HA	7	1.38	11.6	1	1.33	1	160	0 (no)	2 (WinXP)	822
11	Acer Aspire Timeline AS1810TZ-13G32i	9	1.35	11.6	3	1.30	2	320	0 (no)	3 (Vista)	1098
12	Toshiba Satellite T110-10X / T110-10Z	9.4	1.58	11.6	3	1.30	1	320	0 (no)	4 (Win 7)	1260
13	Acer AS1810TZ-13G32I	8	1.45	11.6	3	1.33	2	320	0 (no)	3 (Vista)	1099
14	Dell Inspiron Mini 12	3	1.23	12.1	1	1.6	1	80	0 (no)	2 (WinXP)	940.8
15	HP Compaq 2230s3NA876ES	3	1.80	12.1	3	2.16	2	320	1 (yes)	0 (no)	1080
16	Acer AS2930-583G25Mn LX.ART0X.119	2.3	1.99	12.1	3	2.00	2	250	1 (yes)	3 (Vista)	1146
17	Acer Aspire Timeline AS3410-723G32n	6	1.60	13.3	3	1.20	1	320	0 (no)	1 (Linux)	894
18	Acer Aspire TimeLine AS3810TZ-413G32n	9	1.60	13.3	3	1.30	2	320	0 (no)	3 (Vista)	1230
19	Acer AS4810T-354G50MN	8	1.9	14.0	4	1.40	1	500	1 (yes)	1 (Linux)	1239
20	Asus X80LE-4P151	3	2.80	14.1	2	2.0	1	160	1 (yes)	0 (no)	756
21	IBM Lenovo 3000 N200 572D752	3.5	2.74	15.4	1	1.86	2	160	1 (yes)	0 (no)	669.6
22	Dell 500 2GB/ 320GB	2.2	2.70	15.4	2	1.8	2	320	1 (yes)	1 (Linux)	858
23	Fujitsu-Siemens Amilo Pi 3525 CCE: SEE-110139-001	3.5	2.90	15.4	3	2.26	2	250	1 (yes)	3 (Vista)	1059.6
24	HP Compaq Presario CQ61-205EQ NZ891EA	3	2.68	15.6	2	1.8	2	250	1 (yes)	0 (no)	864
25	Dell Inspiron 1545 Core™ 2 Duo T5800	3	2.64	15.6	3	2.00	2	250	1 (yes)	3 (Vista)	1068
26	Asus PRO72Q-7S008	3.2	3.30	17	2	2.00	2	250	1 (yes)	0 (no)	1020
27	Acer eMachine eMG725-433G50Mi	2.2	3.00	17.3	3	2.10	2	500	1 (yes)	1 (Linux)	1038

Using the normalized objectives and corresponding weight coefficients w_i, the multiple objectives are aggregated into single objective:

$$max\left(w_1BatteryLife^* + w_2Weight^* + w_3Screen^* + w_4RAM^* + w_5CPUclock^* + w_6CPUcore^* + w_7HDD^* + w_8Price^*\right) \quad (19)$$

Two types of tasks are formulated - without and with additional user requirements and each of them is solved twice – for multiple-choice and for single-choice:

Task 1: (19) s.t. (6) – (15), for $n = 3$;

Task 1a: (19) s.t. (6) – (15), for $n = 1$;

Task 2: (19) s.t. (6) – (18), for $n = 3$, $RAM^{min} = 3$, $Price^{max} = 1100$ and $OS_k = 1$.

Task 2a: (19) s.t. (6) – (18), for $n = 1$, $RAM^{min} = 3$, $Price^{max} = 1100$ and $OS_k = 1$.

The weighted sum method is used for each task with three sets of weight coefficients values representing three different DM type of preferences about the objectives. The formulated optimization tasks have been solved on PC with CPU Intel Celeron 2.67 GHz and 2 GB RAM using LINGO v. 12 optimization solver [Lindo Systems, 2012]. The examples solutions times are about few seconds. These times depend mainly on the size and structure of the formulated tasks. The laptops data shown in Table 1 are taken from real laptops catalogues and are used as illustrative data for the numerical examples. Any other number and/or types of devices with different features can be used appropriately to the particular DM preferences. The same is valid for the defined optimization objectives (utility criteria).

5. Results analysis and discussion

The solutions results corresponding to different Tasks and weight coefficients values are shown in Table 2.

Table 2. *Weight coefficients and corresponding solutions*

Problem		w_1 (Battery Life)	w_2 (Weight)	w_3 (Screen)	w_4 (RAM)	w_5 (CPU core)	w_6 (CPU clock)	w_7 (HDD)	w_8 (Price)	***Solution*** (*laptop* # from Table 1)
without additional DM requirements										
Task 1	1st	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	# 11, #18, #27
	2nd	0.300	0.100	0.200	0.080	0.080	0.080	0.080	0.080	# 18, #11, #19
	3rd	0.060	0.040	0.040	0.040	0.040	0.040	0.040	0.700	# 2, #1, #6
Task 1a	1st	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	# 11
	2nd	0.300	0.100	0.200	0.080	0.080	0.080	0.080	0.080	# 18
	3rd	0.060	0.040	0.040	0.040	0.040	0.040	0.040	0.700	# 2
with DM requirements about *RAM ≥ 3 GB, Price ≤ 1100* and *Linux operating system*										
Task 2	1st	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	# 27, #17, #9
	2nd	0.300	0.100	0.200	0.080	0.080	0.080	0.080	0.080	# 11, #13, #17
	3rd	0.060	0.040	0.040	0.040	0.040	0.040	0.040	0.700	# 9, #17, #27
Task 2a	1st	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	# 27
	2nd	0.300	0.100	0.200	0.080	0.080	0.080	0.080	0.080	# 11
	3rd	0.060	0.040	0.040	0.040	0.040	0.040	0.040	0.700	# 9

As it can be seen from Table 2 the DM preferences expressed by weight coefficients influence on the devices choices. The 1st set of weight coefficients values represents equivalent importance of all objectives while the 2nd set emphasizes on *battery life, screen size* and *weight* and the 3rd set focuses exclusively on *weight* objective.

The solutions of the first type of tasks (Task 1 and Task 1a) reflect only DM preferences given by the weight coefficients values. The result of Task 1 solution is Pareto-optimal multiple-choice of 3 devices. The different sets of weighted coefficients result in different multiple-choices. The multiple-choice of preliminary given (by the DM) number of devices supplies more than one alternative to assist DM in making his/her final decision. The Task 1a solutions for n = 1 demonstrate the single-choice as a special case of multiple-choice case.

Task 2 illustrates the possibility to impose additional DM requirements by means of additional constraints for some object features (as the example of laptop RAM capacity, price and installed operating system). These types of tasks are solved for same as for Task 1 sets of weight coefficients but imposing of additional constraints define different choices.

It can be seen that some device choices coincide in some solutions. That is more obvious for Task 2 examples where additional constraints limit the set of possible choices. Increasing of the number of devices and their diversity will increase the variety of choices but will also increase the tasks sizes and their computational complexity.

In general, these types of problems are NP-hard, but the numerical experimentation show acceptable solution times. The branch and bound algorithms implemented in LINGO solver proved to be quite effective for those kinds of sparse restrictions matrixes that are typical for the formulated optimization problems. Further investigations with large scale problems are needed to define the computational complexity of the proposed approach.

6. Conclusion

Constrained discrete multiple-choice problems arise when a given number of objects have to be selected from a finite set of objects with different parameters. The paper proposes an flexible combinatorial optimization modeling approach for discrete multiple-choice. The developed modeling technique is used for formulation of multicriteria linear mixed integer optimization tasks. The tasks formulations can take into consideration different devices key features. The specific user requirements about the device parameters are introduced into optimization tasks formulation as additional constraints.

The described approach is tested numerically by case study example of laptops choice. The *weighted sum* method is used as one of the most commonly used multicriteria methods with direct specification of the DM preferences for objectives importance. The optimization tasks solutions show that different parameters requirements and objectives preferences define different choices. The results of numerical testing show the practical applicability of the developed multicriteria optimization modeling approach to constrained discrete multiple-choice problems. The solutions space of that kind of problems depends on the cardinality of the set of devices to choose from. Increasing the number of devices will presumably give more precise choices better conforming to different requirements but that increases also the computational complexity of the combinatorial optimization problems.

The proposed multiple-choice decision making by multicriteria combinatorial optimization can be used for other real life multiple-choice problems. Different multicriteria solution methods can be used accordingly to the DM preferences.

References:

[1] Rao, M. (2008) Solving some NP-complete problems using split decomposition. Discrete Applied Mathematics. 156(14), pp. 2768-2780.

[2] Chunga, Y., M. Demange. (2008) The 0–1 inverse maximum stable set problem. Discrete Applied Mathematics. 156(13), pp. 2501-2516.

[3] Cook, W.J., W. H. Cunningham, W. R. Pulleyblank, Al. Schrijver. (1997) Combinatorial Optimization, Jon Willey & Sons.

[4] Korte, B., J. Vygen. (2010) Combinatorial Optimization: Theory and Algorithms (Algorithms and Combinatorics), Springer.

[5] Collette, Y., P. Siarry. (2004) Multiobjective optimization: principles and case studies, ISBN 3-540-40182-2, Springer-Verlag Berlin Heidelberg New-York.

[6] Shoval, P., Y. Lugasi. (1987) Models for computer system evaluation and selection. Information & Management. 12(3), pp. 117-129.

[7] Shoval, P., Y. Lugasi. (1988) Computer systems selection: The graphical cost-benefit approach. Information & Management. 15(3), pp. 163-172.

[8] Ein-Dor, Ph. (1985) Grosch's law re-revisited: CPU power and the cost of computation. Communications of the ACM archive. 28(2), pp. 142-151.

[9] Harris, A. L., D. S. Dave. (1994) Effects of computer system components on the price of notebook computers. Information & Management, 27(3), pp. 151-160.

[10] Dujmovic, J. (1996) A method for evaluation and selection of complex hardware and software systems. In: CMG 96 Proceedings

[11] Derek, P., E. Wilbert. (1999) Forecasting notebook computer price as a function of constituent features. Computers & Industrial Engineering, 37, pp. 823-845.

[12] Mustakerov, I., D. Borissova. (2007) Technical systems design by combinatorial optimization choice of elements on the example of night vision devices design. Comptes rendus de l'Academie bulgare des Sciences, 60(4), pp. 373-380.

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

[14] Lindo Systems ver. 12, http://www.lindo.com, (accessed May 2012)