



PDF OFF-PRINTS

Author(s):

MALAKOV, I[vo] & ZAHARINOV, V[elizar]

This Publication has to be referred as:

Malakov, I. & Zaharinov, V.: Interactive Software System for Multicriteria Choosing of the Structural Variant of Complex Technical Systems, Annals of DAAAM for 2012 & Proceedings of the 23rd International DAAAM Symposium, ISBN 978-3-901509-91-9, ISSN 2304-1382, pp 0199 - 0204, Editor B[ranko] Katalinic, Published by DAAAM International, Vienna, Austria 2012

www.daaam.info

INTERACTIVE SOFTWARE SYSTEM FOR MULTICRITERIA CHOOSING OF THE STRUCTURAL VARIANT OF COMPLEX TECHNICAL SYSTEMS

MALAKOV, I[vo] & ZAHARINOV, V[elizar]

Abstract: *The choosing of a structural variant, is one of the main problems solved during the design process of complex technical systems. The real world conditions demand this choice to be made, on the basis of a set of criteria, which in the general case are conflicting. In the present paper are given the results, from the development of an interactive software system for multicriteria optimization, aiding the designers of technical products, during the decision making process of choosing of a structural variant. While developing the software system, the specifics of the problem have been taken into account. An effort has been made, for avoiding the disadvantages of similar, already known software systems. The application of the development is illustrated with a solved example.*

Keywords: *design, technical system, structural variant, multicriteria optimization, software*

1. INTRODUCTION

The analysis of today's technical products shows, that most of them are heterogeneous multicomponent systems, characterized by a great number of parameters. Their effectiveness through all phases of their lifecycle is, in a significant way, determined during the product design process. One of the most important stages of the product design process, is the stage during which the structure of the technical system (TS) is created. In this stage, the types of structural components building the TS and the connections (relations) between them are determined [2, 8, 11]. Choosing the structural variant of a system serves as a basis for the next stages of design. Making a correction in a later stage of an inadequately chosen decision, is related to a significant amount of expenditures. Results from this stage depend essentially, on finding a variant, which satisfies in the best way possible and under certain conditions, the set objectives, i.e., on finding the optimal structural variant of the TS.

Despite the importance of this problem a relatively small number of developments of software systems for optimization of the structure of complex TS are known [1, 2, 6, 7, 10]. This can be explained with the specific problems, that must be solved, and with the complex mathematical apparatus of multicriteria optimization, which is constantly in development. Also, the developments for designing and choosing of an optimal structural variant of a TS, are "know how" for the companies interested in such activities. The specialized scientific and research departments working in this area seek to sell their achieved results as special developments with high market price.

The known software systems for optimization of the structure of a TS possess some disadvantages: the methods used for optimization are inappropriate and/or not effective; the number of the analyzed structural variants is limited; the models do not take into account the possible presence of polyfunctional components, able to execute more than one partial function of the TS, or for constraints in the compatibility between them; not taking into account the presence of incomplete information, etc. An important issue is the controversy between striving for maximal universality of the software system and requirements about the knowledge of the users working with this software. Often a greater knowledge of the specialized mathematical methods and algorithms for optimization is expected from the users, and also certain level of programming skills. In many cases the interface of the software products is not "user friendly" – complex procedures for input and/or editing of the data, difficult information exchange between the user and the software, not comprehensive enough graphical representation of the results [8]. For these reasons the interest of the researchers about this topic remains constant.

The purpose of the paper is to present the results from the development of an interactive software system for multicriteria optimization of the structure of a complex technical system.

2. PROBLEM FORMULATION

2.1 Building the set of possible structural variants

The choosing of an optimal structural variant is done after developing the functional structure of the TS and filling in a morphological table, in which for every partial function of the system, are defined suitable for its execution elementary devices (structural components) and the connections (relations) between them. A main criterion for inclusion of a particular elementary device in the morphological table, is the qualitative execution of the corresponding partial function. An essential characteristic while composing the morphological table is taking into account the compatibility between the elementary devices, i.e., their ability for joint operation in a general structure.

While developing the set of possible structural variants, the terms "Partial function" and "Elementary device" are conditional and apply to the terminology of the systematic approach („Teilfunktion" [11, 12], "elementare Funktionseinheit" [12]).

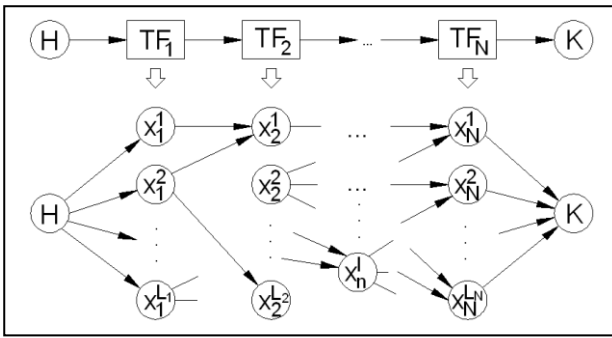


Fig. 1. Net model of the set of possible structural variants

The set of structural variants for the construction of the TS can be presented as a net model (a directed graph) [2]. Every node of the model represents a particular elementary device and every arc – a connection between two devices. On Fig. 1 is shown an example of a functional structure of a TS and the set of its possible structural variants as a net model, where N is the partial functions' count for the given system; TF_N – the N^{th} partial function of the TS; x_n^l – the l^{th} elementary device, executing the n^{th} partial function; L_N – the count of the alternative elementary devices, which execute the N^{th} partial function.

The possible combinations of the elementary devices into structures, that can execute the general function of the TS are shown with arrows. Every path linking the beginning (H) with the end (K) of the net model represents a possible structural variant.

2.2 Characteristic features of the problem for choosing optimal structural variant

The choosing of optimal structural variant of a TS is related to solving the following problem [2, 8]:

For a given functional structure of a TS and defined alternative elementary devices executing its partial functions, define such a combination of the said devices, which would be able to execute the general function of the system, and would satisfy preliminary defined requirements and conditions (constraints) related to the technological and economical properties of the system.

The analysis of the formulated problem reveals some of its characteristic features, which have to be taken in consideration while formalizing the task, choosing the suitable methods for optimization and the development of a software.

1. The formulated problem is characterized by a high level of uncertainty, which is caused by the insufficient and incomplete information in the first stages of the design process, when the set of structural variants is developed and the optimal one of them is chosen.

2. The choosing of a structural variant is done after a comprehensive (complex) evaluation of the competitive variants by a set of technical, economic, ergonomic, ecological, aesthetic and other parameters (objective functions or criteria for effectiveness) of the TS, which depending on the specific requirements and goals of the particular problem, have to take optimal (minimal or maximal) values.

$$\text{opt}F(x) = \begin{cases} \max F_1(x) = \{f_k(x); k \in K_1\}, \\ \min F_2(x) = \{f_k(x); k \in K_2\} \end{cases},$$

$$K_1 \cup K_2 = K, \quad (1)$$

where $F(x)$ is the vector criterion of optimality; $F_1(x)$ – the subset of particular (local) criteria, which have to be maximized; $F_2(x)$ – the subset of particular (local) criteria, which have to be minimized; $f_k(x)$ – the k^{th} characteristic of the system, for which an optimal value is sought, $k \in K$.

In general the defined problem is a vector (multi-criteria, multiobjective) optimization problem. Its solution is found through the use of different principles of optimality [7, 10].

3. The solving of multicriteria optimization problems is difficult, because it is related to a number of specific tasks, described in the specialized literature [6, 10], such as:

- choosing a method for normalization (normalizing) of the criteria;
- choosing of a principle of optimality;
- defining priority for each criteria;
- choosing of a suitable method for optimization and etc.

The number of these difficulties increases significantly, if uncertainty is accounted for, and predetermine the obligatory participation of a human decision maker, in the final decision making for choosing of an optimal structural variant. Only the designer of the system or the team manager which develops it, can make the final decision based on his/her experience, knowledge and intuition.

4. The great number of partial functions of today's TS and the great variety of alternative elementary devices for their execution, makes for a great amount of possible structural variants which have to be evaluated. For example the number of possible structural variants of a complex TS can reach 10^{20} [8], and in some cases can even exceed significantly this number. Therefore the choosing of optimal structural variant involves a great amount of calculation procedures.

5. The formulated problem belongs to the class of discrete programming problems. The finding of a solution is accompanied by significant difficulties. The solving of the discrete problem by replacing it with a similar continuous one, and consecutive rounding of the obtained solution to the nearest integer solution is not possible.

6. When solving the problem it is necessary to account for the existence of constraints related to the compatibility between the elementary devices. Here by the term "compatibility" it is meant the possibility for combining and simultaneous "work" of the elementary devices in an aggregate structure, while executing the general function of the system [8]. The compatibility depends on a number of requirements and constraints – geometrical, informational, functional, etc. The most simple, yet the most seldom met case, is the case in which all elementary devices are compatible. In general the compatibility between the devices is constrained. Moreover, it is possible two devices executing neighbouring partial functions to be incompatible, and also to exist incompatibility between devices, which are separated by more than one partial function in the morphological table of the system. A significant feature is the possibility some of the elementary devices to be

able to execute more than one partial function. A typical example for such polyfunctional devices are industrial robots.

7. From the practical interpretation of the problem it is evident, that the individual criteria can have different “importance” (weight, significance) while choosing the optimal variant of a TS, i. e., one or more criteria to be of grater priority than the others. The priority of the objective functions significantly influences the choice of solution, but there are no existing, scientifically well-founded and commonly accepted methods for its determination. The solving of this task is carried out by the designers of the system together with the client.

8. While formalizing the problem, one must take into account the possibility for existence of constraints over the values of some of the technological and economic characteristics of the designed TS. Therefore in most of the cases the problem has “resource” constraints. They can be functional, type “equality”, and/or regional, type “inequality”:

$$g_m(x) \leq b_m, m \in M_1, \quad (2)$$

$$g_m(x) \geq b_m, m \in M_2, \quad (3)$$

$$g_m(x) = b_m, m \in M_3, \quad (4)$$

where $|M_1| = m_1, |M_2| = m_2, |M_3| = m_3, M = m_1 + m_2 + m_3$; $g_m(x)$ is the m^{th} characteristic of the TS, over which value there is a defined constraint, $m = 1 \div M, M$ – the constraints’ count; x – the structural variant of the designed TS.

3. DESCRIPTION OF THE SOFTWARE

The interactive software system (ISS) PolyOptimizer has been developed for aiding the decision maker, when choosing the optimal structural variant of a complex TS. PolyOptimizer can solve discrete single- and multi-criteria optimization problems. The maximal number of the objective and constraint functions can be 10, and the maximal number of analyzed variants of a TS – 20²⁰. These parameters are defined after analysis of frequently met practical problems and recommendations in the specialized literature.

On Fig. 2 is shown the modular architecture of PolyOptimizer. The software is written in the object oriented scripting language Python 2.7.1 and ISO C (ISO/IEC 9899:2011).

The main module PolyOptimizer GUI is responsible for the management of the graphical user interface and the synchronization between the rest of the modules. The ISS offers an intuitive user-friendly interface, shown on Fig. 3, designed after the most frequently used products for data analysis and data table management, ensuring a smooth learning curve for the routine operations. Aside its common functions for table editing, PolyOptimizer provides more specialized functionality related to the specific problems that it solves.

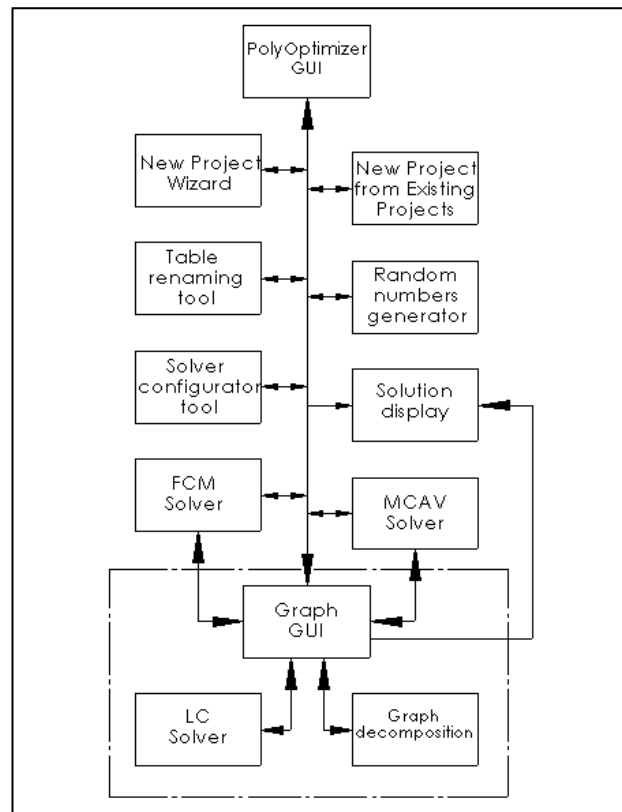


Fig. 2. A block-diagram of the modular architecture of PolyOptimizer

During creation phase of a new project through the module New Project Wizard, the user configures the parameters of the current problem. After specifying particular details, the user can begin with the data entering process.

Input of the problem’s data in the tables for objective and constraint functions is accelerated by a specialized procedure for automated input. The software is constantly monitoring the user’s input for errors and informs him/her in time, about the occurred error, through standard dialogue messages.

The module New Project from Existing Projects gives the user possibility for building a database, allowing reuse of already inputted data tables, which enhances the fast reconfiguration of a given problem and the analysis of different scenarios and combinations of problems.

Table renaming tool is a module allowing the naming and renaming of tables in one place. It makes the arrangement of the problem and work with the data tables more easy and orderly. Also naming tables with names intuitive to the user, aids for quick orientation when the tables are used for another project through New Project from Existing Projects module.

Module Random numbers generator is used for generation of pseudo random numbers. Its use is meant primarily for testing purposes, but with this tool the user can quickly setup tables with random numbers in a given range and type. The module has a level of flexibility that allows the user to choose from a number of settings, such as: data type for input in the tables, which tables to be filled with data and the range of the values inputted, which can be different for every single data table.

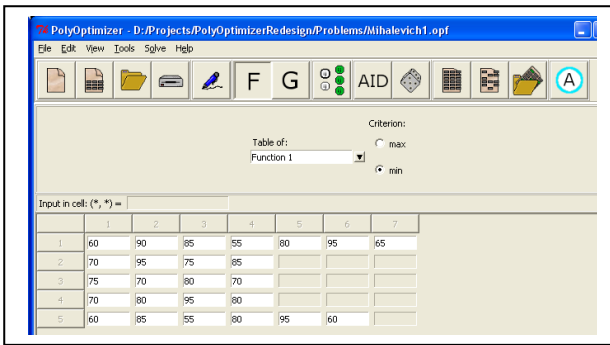


Fig. 3. Graphical user interface of PolyOptimizer

The current version of PolyOptimizer uses two methods for solving of optimization problems – Full Combinations Method (FCM) - FCM Solver module and Method of the Consecutive Analysis of Variants [10] (MCAV) - MCAV Solver module. FCM is used for problems with small count of possible structural variants. The algorithm is realized in such a way, as to take advantage of multiprocessor systems – the calculations are made parallel for each of the processors – the process is multithreaded. Despite the optimized code of the software realization of FCM algorithm, it can only be used for solving problems with up to 9^{10} structural variants. In case of a bigger number of structural variants, the time needed for solving the problem becomes impractically long. For example, a problem with $1,05 \cdot 10^{26}$ variants, would take $3,32 \cdot 10^{11}$ years [8] to solve on a personal computer with an efficiency of 10^7 variants/s, if the problem is to be solved by FCM.

For solving of problems with more than 9^{10} combinations, improved algorithms based on MCAV [8, 10] are used. MCAV is characterized by a directed search of the optimal solution (a deterministic method). After reducing the problem to under 9^{10} number of possible structural variants, FCM can be applied for finding the solution.

The module Solver configurator tool is used for configuration of various solver parameters, both for FCM and MCAV solvers. The module provides great flexibility and freedom for analysis of the problem to be solved. The user himself can define when and how the available solvers are applied, without needing knowledge in the mathematical methods or doing any additional programming. The number of combinations, to which the problem must be reduced, before applying FCM, can be controlled by the user. In the case of multicriteria optimization, the user can define solution search areas and different priority for each objective function. In that way, after initially solving the problem and finding the upper and lower bounds of the feasible set for each criterion, the decision maker can analyze the problem in context of different boundary conditions. When solving with MCAV, a detailed annotated report of the solution process is generated in a text file, which can be viewed with any text editing software.

Entering information for the set of structural variants of a TS is made via the module Graph GUI. At first the user works with a graph, comprised only of elementary devices, entering the compatibility between them. After compositing the graph, the user inputs information about polyfunctional devices via graphical user interface. The polyfunctional devices are automatically added to the

graph, and their compatibility with the rest of the elementary devices is derived from the already provided relationships. Inputting needed information for problems with compatibility constraints is a slow and tedious process. PolyOptimizer provides a set of tools which lead to reduction of the required volume of work and time at least two times. Provided are buttons, allowing the setting of compatibility between one selected device and all elementary devices in a given partial function.

The user can also edit the compatibility matrix directly. The user interface is provided with buttons, allowing transposition of the compatibility matrix relative to its main diagonal, keeping it symmetric, an obligatory condition (symmetrical editing) [9]. Through entering information related to compatibility between elementary devices the software system automatically builds the compatibility matrix. On Fig. 4 is shown graphical interpretation of the compatibility matrix in PolyOptimizer.

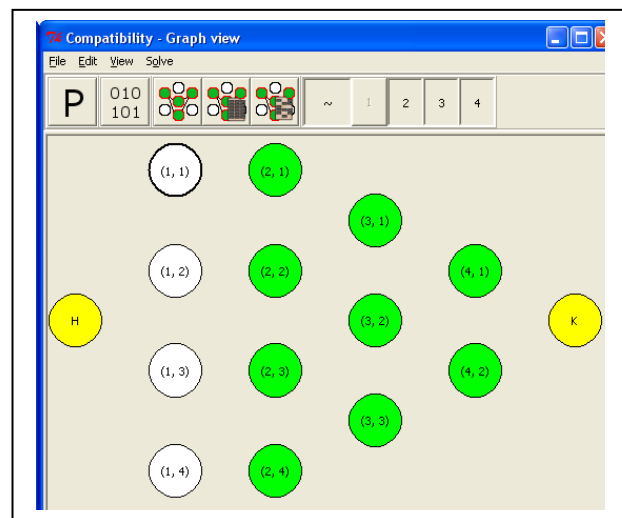


Fig. 4. Graphical interpretation of the compatibility matrix

The module Graph decomposition provides algorithms for decomposition of the compatibility matrix to sub matrices, the latter include only compatible elementary devices. This is a necessary condition for applying one of the integrated solvers. Module Graph decomposition, applies original algorithms developed by the authors [9]. After decomposing the graph, the found sub problems can be visualized by the graphical interface. This is done by marking the elementary devices comprising the sub problems, with a color scheme, over the graph.

Module LC Solver is a procedure for finding the optimal solution of problems with compatibility constraints and/or the presence of polyfunctional devices. It is an automated sequence which decomposes the compatibility graph and applies the methods for finding optimal solution for every found sub problem. Similarly to the module Solver configurator tool, this module provides the user with means for configuring parameters of the used algorithms.

4. EXAMPLE

The developed ISS PolyOptimizer has been used for solving the following problem:

For a given set of possible structural variants of an automated system for the assembly of an unit “Brush holder” (Fig. 5), find the Pareto optimal structural variant for which:

$$\min C(x), \min C_B(x), \min T_p(x), \min P(x), \quad (5)$$

satisfying the constraints:

$$S(x) \leq 8, \quad (6)$$

$$T(x) \leq 5, \quad (7)$$

where $x \in X$; $n = 1 \div 6$; $L_1 = 5$, $L_2 = 6$, $L_3 = 3$, $L_4 = 2$, $L_5 = 2$, $L_6 = 2$.

$C(x)$ – technological expenditures;

$C_B(x)$ – energy expenditures;

$T_p(x)$ – time expenditures;

$P(x)$ - price;

$S(x)$ – occupied area;

$T(x)$ – buy off period.

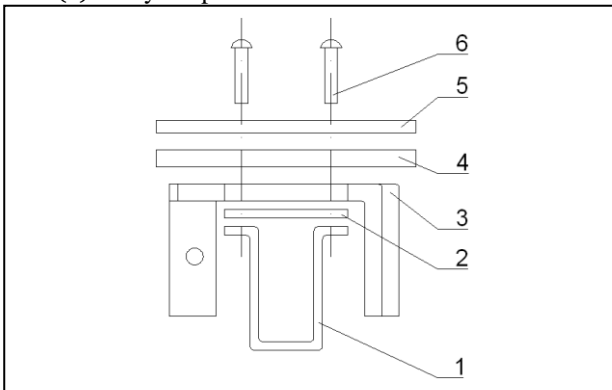


Fig. 5. Assembled unit „Brush holder”; 1 – socket; 2 – plate; 3 – holder; 4, 5 – isolator; 6 – rivet (4 pcs.)

The assembly operations are grouped and represent one partial function of the system.

The general function of the designed assembly system includes the following partial functions TF_n , $n = 1 \div 6$:

TF_1 , assembly of parts "Socket" and "Plate";

TF_2 , assembly of part "Holder";

TF_3 , assembly of part "Isolator" – 2 pcs.;

TF_4 , assembly of part "Rivet" – 4 pcs.;

TF_5 , transportation;

TF_6 , calibration and riveting.

The set of possible structural variants X , which can execute the general function of the assembly system, is presented as a directed graph on Fig. 6. The nodes of the graph model are arranged in 6 columns corresponding to each partial function of the AAS. The polyfunctional devices $x_1^3, x_1^4, x_1^5, x_2^4, x_2^5, x_2^6$ and x_3^3 are represented as nodes marked in cyan and placed in the columns corresponding to the first partial function they can execute. The possible ways of combining the elementary devices x_n^l in structures x , executing the general function of the assembly system are shown as arrows. Every path connecting the beginning (H) and the end (K) of the graph model is a possible structural variant.

The total count of the possible structural variants of the designed assembly system is 220.

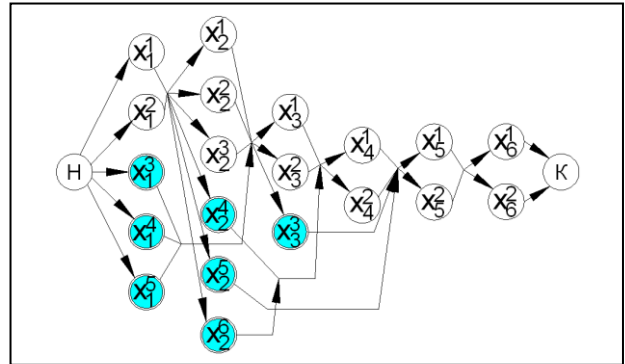


Fig. 6. Directed graph of the set of structural variants of an AAS for assembly of unit “Brush holder”

With the developed ISS, the Pareto optimal structural variants of the designed assembly system are found, for four cases of relative importance of the objective functions, defined by a priority vector. The obtained results are summarized in Table 1 and on Fig. 7.

Problem №	Priority vector	Optimal structural variant
P1	-	$\{x_1^1, x_2^3, x_3^3, x_5^1, x_6^2\}$
P2	$\bar{P}^1 = \{1; 0,25; 0,167; 0,4\}$	$\{x_1^1, x_2^1, x_3^3, x_5^1, x_6^2\}$
P3	$\bar{P}^2 = \{0,5; 0,1613; 0,2222; 1\}$	$\{x_1^1, x_2^3, x_3^1, x_4^1, x_5^1, x_6^1\}$
P4	$\bar{P}^3 = \{0,6; 1; 0,4; 0,2\}$	$\{x_1^4, x_3^3, x_5^1, x_6^2\}$

Tab. 1. Compromise solutions of the problem

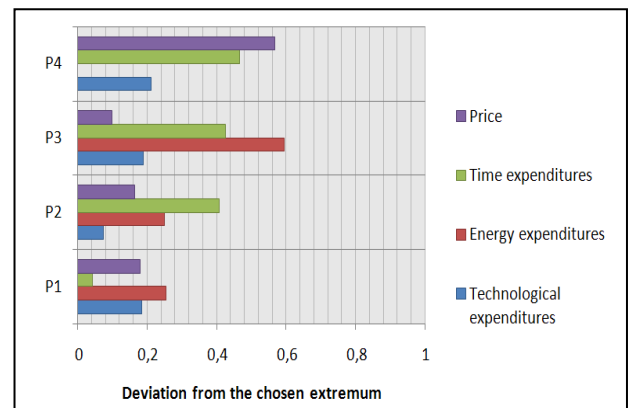


Fig. 7. Results from the analysis of the problem for choosing an optimal structural variant of an AAS for the assembly of unit “Brush holder”

5. FUTURE RESEARCH AND DEVELOPMENT OF POLYOPTIMIZER

The architecture of PolyOptimizer is modular, which makes possible the fast and easy adding of new features.

A particular interest is the integration of algorithms based on biological analogy – ant colony optimization, genetic algorithms, particle swarm optimization [3, 4]. In a stage of research and development is a modification of the ant colony optimization algorithm, which will be integrated as a module in PolyOptimizer. It will solve discrete optimization problems (single- and multicriteria) and it will be capable of decomposing the compatibility matrix simultaneously with the solution search process. Also an advantage, is the great flexibility of the method in terms of parameters, which can be changed by the user, affecting the effectiveness of the solution search

process. The ant colony optimization algorithm is a metaheuristic algorithm. The advantage of using such type of algorithm is seen while solving problems with a great number of variants, which can not be reduced with MCAV to a number of less than 9^{10} . In that case the problem can not be solved with the application of FCM in a reasonable time frame. The metaheuristic methods are characterized by a solution search process based on trial-and-error, learning and adaptation techniques. These methods do not guarantee the finding of an optimal solution, but it can be expected that an approximation of the optimal solution would be found. This is a compromise for problems with great number of variants and no possibility for reduction of their volume.

Further consideration is the integration of algorithms based on fuzzy logic [5], used for design of complex technical systems in the presence of incomplete information.

The possibility for graphical interpretation of the obtained results and comparison between different solutions of a problem, generated for different situations (scenarios) is also in a development phase. A module for plotting the set of solutions, leading to easier analysis and comparison is in consideration.

Finding of the Pareto front is a very important issue in multi-criteria optimization. By varying the weight coefficients, with the aid of PolyOptimizer, the user can study and find sufficiently good approximation of the Pareto front for a given discrete multicriteria optimization problem. In development is the automation of this procedure, by the addition of a function, searching for the Pareto front's representation. It iteratively varies the weight coefficients of the objective functions by a given step.

6. CONCLUSIONS

The problem for choosing optimal structural variant of complex technical systems has been analyzed and its characteristic features have been defined. They have been taken into consideration in the process of the problem's algorithmic and software support. An interactive software system has been developed characterized by:

- usage of elaborate algorithms of current methods for multicriteria optimization – method of the consecutive analysis of variants, metaheuristic algorithms;

- solving of big sized problems (big number of the analyzed variants – 20^{20}) with compatibility constraints and/or polyfunctional devices;

- easy to learn working environment, automated procedures for entering and checking of input data and reuse of the already entered information;

- fast study of different scenarios of the solved problem by varying the priority of the objective functions, of their variation limits, study the influence of the objective and constraint functions over the mathematical model of the problem, etc.;

- comprehensive results presentation;

- no special requirements for the user of the system, regarding the used mathematical methods for optimization;

- small software size (about 15 MB) and low system requirements.

The development will aid the designers of complex technical systems for choosing of structural variant. It can be used also for solving of similar problems in economics, transportation, management, ecology, military science, etc.

7. ACKNOWLEDGEMENTS

The scientific studies, the results from which are presented in the current paper, are financed by the Inner competition of TU-Sofia-2012.

8. REFERENCES

- [1] Batishchev, D. & Shaposhnikov, D. (1994) *Multiobjective choice with regard to individual preferences*, RAN IPF, ISBN 5-201-09297-7, Nizhny Novgorod
- [2] Boyadjiev, I.; Boyadjiev, B. & Malakov, I. (1996) Rechnerunterstuetzte systemvorgehenweise zur variantenauswahl beim konstruieren. *Proceedings of the 7th International DAAAM Symposium*, ISBN 3-901509-02-X, Katalinic, B. (Ed.), TU Vienna, Austria, 17÷19.10. 1996, pp 059-060, Published by DAAAM International, Vienna
- [3] Branke, J. & Deb, K. (2005) *Integrating User Preferences into Evolutionary Multi-Objective Optimization*. In: Jin Y (ed.) *Knowledge Incorporation in Evolution-ary Computation*, pp 461-477. Springer, ISBN 3-540-22902-7, Berlin, Heidelberg
- [4] Coello, C. A. C.; Lamont, G. B. & Veldhuizen, D. A. V. (2007) *Evolutionary algorithms for solving multi-objective problems*, Springer, ISBN 978-0-387-33254-3, New York
- [5] Dimitrov, K. & Nacheva, I. (2010) Intelligent Agents and Fuzzy logic for systems monitoring and fault diagnosis. *Proceedings of the 6th Scientific and Technical Conference "Mechanics and Machine Elements"*, Sofia, November, 2010, pp 102 - 116
- [6] Ehrgott, M. & Gandibleux, X. (2002). *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys*, Springer, ISBN: 978-1-4020-7128-7, Kluwer
- [7] Kalashnikov, A. (2004) *Dialogic system for multiobjective optimization of technological processes*. Dissertation for doctoral degree in technical science. MGISS, Moscow
- [8] Malakov, I. (2009) *Methodology for choosing of optimal structural variant of an automated assembly system*. (in Bulgarian) Research work qualifying for a "professor", TU, Sofia
- [9] Malakov, I.; Zaharinov, V. & Flemming V. (2011) Algorithmic and software development for the decomposition of graph models for the optimization of the structure of technical systems. (in Bulgarian) *Scientific news of NTA for mechanical engineering*, Year XIX, Issue 4/124, ISSN 1310–3946, 20 ISTC „ADP 2010“, Sozopol, June 2011, pp. 46÷53.
- [10] Mihalevich, V. & Volkovich V. (1982) *Computational methods for study and design of complex systems*. (in Russian) Nauka, ISBN , Moscow
- [11] Pahl, G. & Beitz, W. (2007). *Konstruktionslehre. Grundlagen erfolgreicher Produktentwicklung. Methoden und Anwendung*. Springer-Verlag, ISBN 3-540-34060-2, Berlin
- [12] Roth, K. (2000) *Konstruieren mit Konstruktionskatalogen*. Band1. Dritte Aufgabe, Springer-Verlag, ISBN 3-540-67142-0, Berlin