INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

APPLICATION OF SOFT COMPUTING TECHNIQUES IN THE DESIGN OF ROBOT GRIPPERS

Vassilis Moulianitis [1,2], Kostas Saridakis [1], Stavros Papageorgiou [1], Vasileios Syrimpeis [1], Argiris Dentsoras[1], Nikos Aspragathos[1],

[1] Dept. of Mechanical Engineering and Aeronautics, University of Patras, Greece
[2] Dept. of Product and Systems Design Engineering, University of Aegean, Greece

ABSTRACT
The present paper presents a system used for performing design by making extended use of soft computing techniques. Throughout the paper the design of robotic grippers for flexible materials is used as a reference design case in both its conceptual and detailed design phases. The system contains three modules. The first module focuses on the conceptual design phase and is used for: a. the generation of a set of feasible solutions (concepts) of grippers for a set of pre-specified requirements which express the handling tasks that the gripper must accomplish and b. for the evaluation of the feasible solutions in order to find the best one according to the designers' experience and some pre-specified criteria. The evaluation process is based on a design index that includes several interacting criteria represented as relationships among fuzzy entities.
The concept is then submitted to the second module called Case-DeSC (Case-based Design with Soft Computing). Case-DeSC is dedicated to the development of a detailed parametric design of the gripper concept by utilizing formal design case representation, fuzzy preferences on the design parameters and by searching for optimal solutions by deploying an evolutionary algorithm. The optimal solution passes over to a third module, capable of presenting a 3D visual representation of the parametric gripper concept via Virtual Reality Modelling Language (VRML). The proposed system has been developed in MATLAB.

Keywords: conceptual design, parametric design, fuzzy, genetic algorithm, neural network, VRML

INTRODUCTION
Design has been a human activity for thousands of years. During the last decades it has been extensively studied in both its creative and routine forms for the establishment of general purpose and domain-independent scientific rules and methodologies. Finger et al. [1], in a review paper, survey the issues of design theory, design methodologies and design models. Another survey on the same topic is conducted by Evbuomwan et al. [2]. The two aforementioned articles summarize and review the developed design models and methodologies, investigate the nature and the characteristics of the design process and classify the design models into categories.
Based on the existing literature, three general design models are identified: the descriptive, the prescriptive and the computer-based model [2]. The descriptive models tend to capture the processes, strategies and methods that designers use in order to address certain design problems. These models usually initialize the design process through the generation of a single solution concept, which is then submitted to further analysis, refinement, development and evaluation. Although descriptive models incorporate past experience, they still meet difficulties in capturing the collaborative activities among designers, especially in vague conceptual phases. On the other hand, the prescriptive models are well addressed in preliminary design stages and are used by the designers as valuable tools since they provide an intuitive sense to the design process. The underlying prescription may be deployed either for providing guidelines for the design process or for the determination of the attributes of the designed artefact. The prescriptive models are mainly algorithmic-based and this fact facilitates the deployment of several analytical activities before the generation of concept solutions. However, they present the following weaknesses: a. inefficiency in mapping the design requirements to the attributes of the artefact, and b. possible incompatibility with computer-based design methods if integration is

considered. Finally, the computer-based models enhance features from the two aforementioned categories and are capable of performing many different design activities. They may address decision-making processes or analyze the design knowledge in order to provide a better understanding of the considered design problem. The computer-based models, however, perform optimally only for specific, well-defined classes of design problems.
The above short reference to the three categories of design models reveals the necessity for unified computer-based design models that would include and could manipulate both quantitative and qualitative knowledge.
The applicability of a design model (or design methodology) highly depends on the type of the problem under consideration. In the literature, three basic categories of design problems are identified: a. routine design problems characterized by the existence of predefined, explicit and well-structured design knowledge shared among identical or similar problems, b. redesign problems characterized by the need for adaptation of an existing problem representation and/or solution in order to meet some newly defined requirements, and c. non-routine or original design problems further classified in innovative and creative. In innovative design problems, although new variables, parameters or features are introduced, the resemblance with an existing design problem is preserved and the new solution is delivered through a synthesis process. On the other hand, in creative design, the newly introduced design entities differentiate the current design problem. The extracted solutions are not delivered via a predefined plan and they may be characterized as unique.
Most design tasks and activities involve intense decision-making regarding issues such as function, structure, configuration, material and geometry of the designed artefact. The quantity and the quality of the available information and data in order to perform proper decision-making vary along the different design phases. At the preliminary or conceptual design phase, very little information is available. As design advances, more information is produced and the design knowledge is continuously enriched. During the very first phases, design may be enhanced by the collaboration of multiple designers or design teams on the basis of a communication and common decision-making framework. In a relevant paper, Wang et al. [3] discuss issues of collaborative conceptual design and they review the literature for relative approaches and applications. From this survey, it becomes evident that the ability to handle different types of uncertainty and vagueness – inherently characterizing the design process - is very important, thus making it necessary that systematic methodologies be established in order to model and manipulate them.

Developments in the computer hardware in the last two decades have made it easier for the artificial intelligence techniques to grow into more complex and efficient frameworks. Moreover, it has been proven that several artificial intelligence techniques may be used as tools in problems where conventional approaches fail or perform poorly. An excellent example for demonstrating this potentiality is the field of engineering design due to its specific characteristics and requirements. The relevant research activity has been directed towards the development of architectures and ontologies for applying artificial intelligence in design, in order to assist the design activity and to apply automation to the more complex, conceptual and decision-making tasks [4]. It is also a fact that there is a parallelism among engineering design and artificial intelligence [5]. However, it is widely accepted that AI is still not widely used in CAD/CAM systems, which, nevertheless, incorporate developments from the domain of applied mathematics and information technology [6]. The reasons for this latter ascertainment are: a. the difficulty of deploying several AI techniques in a concurrent or collaborative framework, b. the increased demands in computational power and c. the need for addressing the issues of design knowledge conceptualization, visualization and representation.
In the early 1990s, the concept of Soft Computing (SC) was introduced. Soft Computing is an evolving collection of artificial intelligence methodologies aiming to exploit the tolerance for imprecision and uncertainty that is inherent in human thinking and in real life problems, to deliver robust, efficient and optimal solutions and to further explore and capture the available design knowledge. Fuzzy Logic (FL), Artificial Neural Networks (ANN), and Genetic Algorithms (GAs) are the core methodologies of soft computing. SC yields rich knowledge representation (symbol and pattern), flexible knowledge acquisition (machine learning), and flexible knowledge processing (inference by interfacing between symbolic and pattern knowledge). Additionally, SC techniques may either be deployed as separate tools or be integrated in unified and hybrid architectures. The fusion of SC techniques causes a paradigm shift (breakthrough) in engineering and science fields (including

engineering design) by solving problems, which could not be solved with the conventional or stand-alone computational tools.

Research has been deployed in the direction of applying SC to engineering design in the context of replacing existing analytical models with approximated models or meta-models. Simpson et al. [7] investigated the potentiality of soft computing techniques by comparing them to the statistical techniques in meta-modelling and they provided some recommendations about their appropriate use. Except for meta-modelling, SC techniques may be combined with expert and knowledge-based systems. The common path of expert systems (ES) and SC techniques was surveyed by Liao [8]. In another literature review, Rao et al. [9] change their focus to the new product development (NPD) activities. In their article, they categorize the applications in five discrete areas, in NPD stages and NPD core elements. The penetration of SC in activities that are strongly related with engineering design is also evident in the research work of Hsu et al. [10]. NPD is also the main research direction in this article, but this time in the context of deploying two SC techniques, namely ANN and FL, in an integrated framework. A general neuro-fuzzy model is suggested, while different formulations of neuro-fuzzy networking are discussed.

The wide acceptance of SC techniques by the scientific community is also based on the potentiality of fusing SC with conventional hard computing techniques. Kamiya et al. [11] investigate this fusion and remark the possibilities of its application in large-scale plants. The advances of successfully applying SC in demanding domains like engineering design, has also promoted the use of SC techniques to a wide spectrum of industrial applications. Dote et al. [12] review a significant number of such SC-based applications.

The consideration that the art of designing still remains a special human activity, which is in most times an entertainment for the designer, is remarked by Cross [13]. Therefore, future advances in engineering design should perhaps be limited to the cognition of this human natural intelligence. Having confirmed the efficiency of deploying SC either in terms of stand-alone techniques or as supportive tools to other conventional design methodologies, many researchers have been investigating the possibility of enhancing the existing design methodologies by simulating the design process through SC techniques. Shakeri et al. [14] simulate the design process using a multi-agent system, with some of the agents being based on SC. These agents mimic the behaviour of a design team and a set of design methodologies is constructed by using learning techniques.

Through the review of the existing literature, the following three different issues concerning the design process become distinguishable: a) the design knowledge representation (modelling), b) the search for optimal solutions and c) the retrieval of pre-existing design knowledge and the learning of new knowledge. These issues are related to design phases with different procedural steps and requirements, such as the conceptual and the detailed design phase. Research has been performed in utilizing knowledge-based systems in conceptual design by Moulianitis et al. [15] and several architectures of soft-computing has been proposed in detailed parametric design by Saridakis et al. [16]. Although there is a classification between conceptual and detailed parametric design, most of the real-life engineering problems integrate both concept development and detailed parametric solutions, in terms of a two-phases process. Considering that the designer requires the maximum support during these phases, this paper introduces a computational framework that provides tools in order to organize and manage the existing design knowledge, to extract the best conceptual and detailed solutions, while roughly visualizing the final design solutions. This integrated approach utilizes powerful characteristics of computational techniques originated from SC domain, such as fuzzy logic, genetic algorithms and artificial neural networks. The proposed architecture is applied to the design of robotic grippers for flexible materials in order to evaluate the underlying efficiency regarding the optimality of extracted solutions, the friendliness to the designer and its potential of addressing many different types of design problems. The following paragraphs describe three modules, which the proposed framework is based on, and further clarification on each focused subject is provided through the example case of electrostatic robotic grippers for handling fabrics.

CONCEPTUAL DESIGN

In this paper, design is conceived as a search for a suitable concept. According to Renner and Ekart [17] a search problem consists of a desired state, search space and a search process. In this work the desired state is a simple gripper concept or a synthesis of gripper concepts while the search space is a combinatorial synthesis of them. The search process is implemented using genetic algorithms.

According to Goldberg [18] the GA operators jives with some of the literature of invention and can be used in the conceptual design phase [19], [20].
The first module's user interface is shown in Figure 1. The user interface is separated in three panels:

1. The upper left panel is used for input of design specifications. In this context as design specifications are considered: a. the handling tasks that the gripper concept must accomplish and b. the characteristics of both environment and material. The knowledge base that maps the inputs to the problem as well as the gripper concept has been presented in Moulianitis et al.[15]
2. The lower left panel is used for the construction of the evaluation mechanism for the genetic process.
3. The upper right panel is used for the GA parameters.

In the following paragraphs the main components of the GA adopted in the current study are presented.

Figure 1. The user interface for the conceptual design module.

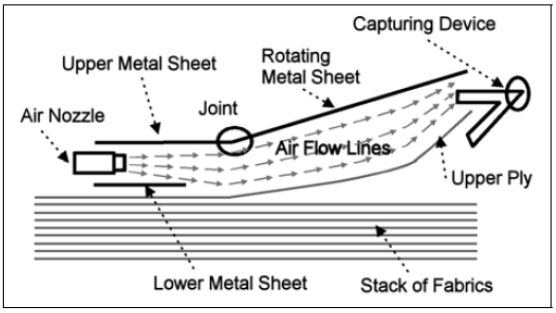
Representation mechanism

The handling tasks that the gripper must accomplish may be one of the following or a combination of them:

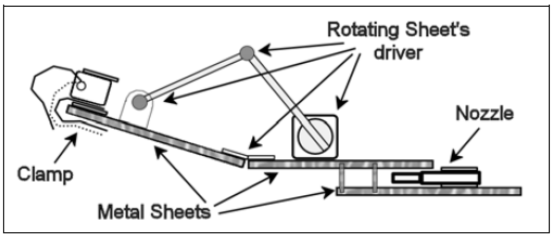
- Separation (see Figure 2.(a)),
- Picking,
- Placing,
- Applying tension,
- Assembling (for example the superposition of two panels).

For example, a sewing process includes an assembly task and an applying tension task. In addition, the fabric's characteristics and the space where the gripper will operate, constraint the design space. The gripper concept, which is the design goal of the conceptual design phase, is a simple one or a synthesis of the following [15]:

- Pinching Grippers,
- Clamping Grippers,
- Air-Jets (see Figure 2.(b)),
- Pin Grippers,
- Brush Grippers (Velcro).
- Vacuum/Pneumatic Grippers,
- Electrostatic/Magnetic Grippers,
- Adhesive Grippers,
- Freezing Grippers.



(a)



(b)

Figure 2. (a) Separation of a single ply of fabric from a stack. (b). An air jet gripper.

In the genetic algorithm, each gene represents a gripper concept and there are nine different concepts in the system. The length of the string depends on the number of tasks that the gripper concept must accomplish. In the worst case, the maximum length of the string is equal to the number of tasks.

Evaluation mechanism

Fitness function reflects the ability of an individual to survive and reproduce its structure in the population of the next generation. Fitness corresponds to the objective function to be optimized. The aim of this study is to maximize the objective function, so that the fitness function and objective function are the same.

There are seven criteria that are used to find the best gripper concept. Flexibility, intelligence and complexity are the components of the mechatronics index and their calculation is presented in [21] while the calculation of the repeatability, reliability and cycle time is presented in [15]. The degree of confidence is calculated using fuzzy rules that map the degree of confidence of the grippers with the handling tasks and the material characteristics. There are over 100 fuzzy rules of the form:

> *If handling task is … and material characteristic is … then the degree of confidence of gripper concept is…*

For example:

> *IF handling task is separation and porosity of material is high THEN the degree of confidence of concept vacuum is low.*

The objective function is constructed in two ways:

- Based on the mechatronics index with interacting criteria [22]. The objective function is calculated using the discrete Choquet integral [23]:

$$C_u \; x^i \; := \sum_{j=1}^{n} x_j^i \left[u \; A \; C_j \;\; - u \; A \; C_{j+1} \;\; \right] \tag{1}$$

where, $C = \; C_1, C_2, ..., C_n$ *is the set of criteria for constructing the objective function,*

$x^i = x_1^i, x_2^i, \ldots, x_n^i \in \Re^n$ is the profile of a feasible solution, $u \in F_C$ are the fuzzy measures, F_C denotes the set of all fuzzy measures on C, $\cdot$ indicates a permutation of C such that $x_j \leq \ldots \leq x_n$, $A\ C_j = C_j, \ldots, C_n$, $A\ C_{n+1} = \varnothing$.

- Custom objective function using a number of criteria as it is shown in Figure 3 [21]. An example of the objective function I shown in Figure 3 is:

$$I = C_1 * \min(C_2, C_3) \tag{2}$$

where C_1 is the degree of confidence of the gripper concept for accomplishes the tasks, C_2 is the degree of the gripper intelligence and C_3 is its reliability.

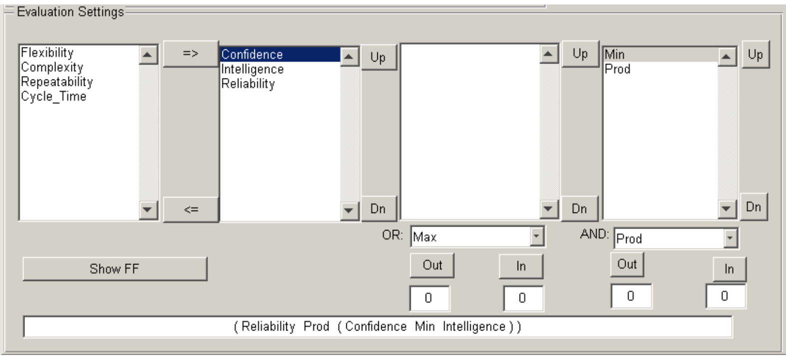


Figure 3. Construction of custom objective functions.

Genetic operators
The following genetic operators that used are a one-point crossover operator and a mutation operator (shift mutation).

- One-point crossover works as follows: a crossing point is randomly selected along the length of the first parent chromosome. The sub-section of the parent chromosome before this crossing point is copied into the offspring. The remaining places of the offspring are filled up by taking in order each legitimate gene from the second parent.
- Shift mutation: A single job is selected randomly and inserted in a random position.

Selection mechanism
Selection of parent chromosomes is done based on a binary tournament strategy: a pair of chromosomes is randomly selected and the 'stronger', i.e., that with the highest fitness is selected for reproduction.

Control parameters
The determination of the suitable settings for the control parameters of any GA is a very difficult task. The variety of the control parameters included in the components of a GA as well as the many alternatives make the determination of the 'perfect' settings almost impossible. The setting of the control parameters used is a fixed population size of 100 chromosomes, a fixed crossover rate equal to 0.8 and a fixed mutation probability of 0.01. The GA stops when a maximum number of 100 generations has been surpassed.

DETAILED PARAMETRIC DESIGN: CASE-DESC
For each of the feasible concepts that are extracted from the module used for conceptual design, a specific case representation must be provided in order to proceed to detailed parametric design, which in the context of the current approach, is addressed by Case-DeSC module. This module has been developed within the Matlab platform [24] and integrates the potential capabilities of case-based design (CBD) with an approach that utilizes fuzzy preferences on the design parameters and searches

for optimal solutions by using a genetic algorithm (GA) [16]. Despite its many advantages, Case-DeSC cannot be deployed (or can hardly be deployed) in design domains where the design knowledge (design requirements, design parameters and their associative relations) cannot be explicitly expressed. Therefore, Case-DeSC has been augmented with case-based reasoning techniques so as to become capable of addressing design problems by retrieving past solutions from a case-base through using a competitive neural network and k-means clustering [24], [25], [26]. Several other functionalities are provided in the proposed system where case-based design is combined with design based on soft computing in a variety of different architectures. For each of them, Case-DeSC provides a module that supports the underlying process. The module to be used depends on the design problem under consideration and may be chosen by the designer.

Design parameters and performance variables

In the following paragraphs, the case representation formalism utilized by Case-DeSC and used for the detailed design phase for the robot grippers is presented. All the design entities that are incorporated in a case are explicitly defined. As it is shown in Figure 4, the proposed case representation for the robotic gripper problem is based on twenty-nine (29) Design Parameters (DPs) (The numbered nodes correspond to DPs as they appear in the numbered list at the top left part of the figure). These are design entities that formally represent physical, functional or behavioural attributes of the designed artefact (robot gripper) and the different combination of their values distinguish the different (alternative) designs. Each design problem is stated in the context of DPs and associative relationships among them. In case that a design parameter is critical for the efficiency of the final outcome and its variation has to be closely examined then this design parameter is characterized as Performance Variable (PV) [27]. Usually the definition of the PVs depends on the designer's preferences.

A further classification of the design parameters leads to two basic categories, namely dependent and primary (independent) parameters (see Figure 4 for the different symbols used for the corresponding DPs for the present design case). Furthermore, the primary design parameters of the design problem may be characterized as either fixed or variable. The fixed primary design parameters represent independent design parameters used as inputs for the design problems whose values stay invariable during the design cycles, while the values of the variable design parameters change independently during design process.

The values of the dependent design parameters are fully determined by their children DPs through their associative relations. The solution search is deployed in the design space and is performed through the variation of variable primary DPs. The terms 'dependent' and 'primary' used for the classification of the design parameters are also retained for the PVs.

Some of the aforementioned design entities (DPs and PVs) are more important than others. In order to model this relative importance, Case-DeSC uses weighting factors (denoted by the symbol w). Through their values, weighting factors determine the more important and critical design parameters (performance variables), as far as the design performance evaluation is concerned. In other words, weighting factors are used in order to determine quantitatively and subjectively the most important and critical performance variables according to the designer's beliefs.

The hierarchical levels in Figure 4 that represent the design case for the robotic gripper can be obtained if the design parameters and their dependencies are represented and manipulated in a design structure matrix (DSM) [16]. DSM may be considered as a binary square-matrix representation of the digraphs formed by DPs and their interrelations. After structuring the design knowledge, the fuzzy preferences must be applied by the designers on the design parameters that have been previously identified as performance variables (PVs). The fuzzy sets so-formed, represent the preferences of designers for concepts concerning the design goals that are mainly modelled in terms of specific values or value ranges. Specifying preferences on PVs relies on the designers' knowledge and experience. An example of the application of fuzzy preferences on a performance variable is shown in Figure 5. Apart from the fuzzy preferences, weighting factors are also assigned to the performance variables and model their relative importance. This implies that every node of the hierarchical DP tree that is defined as PV is associated with fuzzy preferences and weighting factors.

Fuzzy preferences

The fuzzy preferences constitute an alternative way of modelling the objective criteria of design solution's optimality through providing the possibility to the designer of expressing his/her direct preferences values of specific design parameters in a way that depends on how loosely the objectives should be set. The fuzzy preference on a performance variable is a function taking as input the set the value set of the performance variable and resulting output a value within the set of (0,1]. Therefore, it becomes obvious that through fuzzy preferences, the modelling of design constraints can be performed, as well as intervals of values with underlying imprecision/vagueness (e.g. tolerances) and qualitative expressions of design objectives through fuzzy representations.

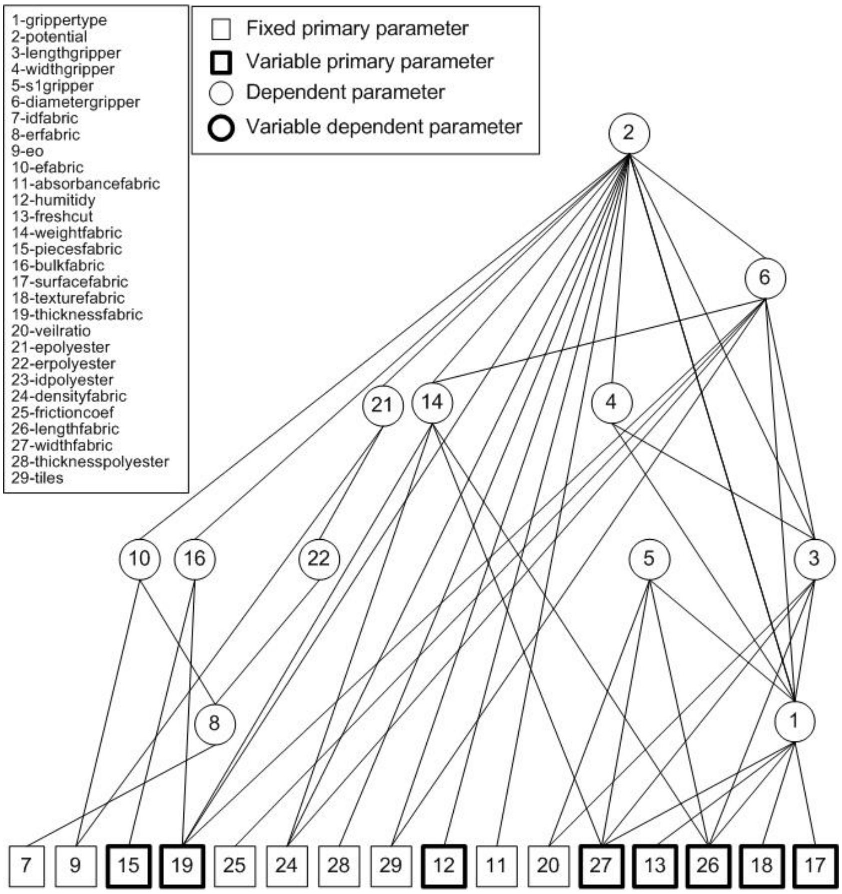


Figure 4. Hierarchical tree of design parameters for the example of electrostatic robotic gripper.

The designer's fuzzy preferences μ(PV) are deployed on sets of values of the PVs. These preferences must be aggregated by a specific aggregation function that reflects a strategy about how competing attributes should be traded-off. The function P attempts to formalize the balance of conflicting design goals and constraints by preserving the design rationality. The evaluation of the design solutions is performed after the various individual preferences have been combined or aggregated. The overall preference $\mu_o(\overrightarrow{PV})$ is a function that combines all the preferences on the performance variables for a particular design case:

$$\mu_o(\overrightarrow{PV}) = P\ \mu(PV_1), \mu(PV_2), ..., \mu(PV_j) \quad (3)$$

In the above expression, the aggregation function P reflects the strategy that indicates how the competing attributes of the designs should be traded-off. The aggregation functions should be extracted from a list of available aggregation functions, judged as suitable for use in design problems. It is possible for the designer to select deliberatively different aggregation strategies for groups of performance variables. The selection of the trade-off strategy (min, weighted-means, etc) requires adequate knowledge of the design problem under consideration and specific knowledge on the relation between the aggregated parameters [27].

The optimal solution

Having completed the assignment of values, fuzzy preferences and weights and determined the aggregation strategy, the next step is the extraction of the optimal solution according to the criterion of the maximum overall aggregated fuzzy preference. Case-DeSC provides several different options in order to search for solutions' optimality. The simplest option is based on the deployment of a genetic algorithm, which through the variation of the values for some pre-defined design parameters (usually the input design parameters and the performance variables) results in the optimal solution, as it is shown in Figure 6 (the optimal solution is shown in the table in the lower part of the figure).
Through the deployment of the genetic algorithm many sub-optimal solutions that are extracted through the genetic optimization can be recorded and stored in a case-base suitable for the specific design problem. Further capabilities are provided by Case-DeSC by enhancing the genetic algorithm with other optimization methods, or by using a case-based retrieval mechanism for restoring a number of past design cases that can be used in a variety of ways [16]. This retrieval mechanism is based on a competitive neural network that utilizes the assigned preferences and the case base records, in order to find the most similar cases (old solutions) to the current one.

VISUALIZATION AND VRML MODULE

The 3D model of the gripper carries all the parametric details that have been formed through the aforementioned two modules such as type, dimensions, etc. The solution that is produced by the second module is processed into the VRML module and a 3D model is produced. The 3D VRML models of the grippers that are produced may be viewed locally or over the Internet through any VRML capable Internet browser [28]. This capability gives the opportunity to present and exchange 3D gripper representation via the Internet among multiple users allowing each one of them to examine and evaluate virtually the gripper model. In Figure 7, the 3D model of a SCARA robot and a cylindrical tool is illustrated. This model is viewed into the MS IE browser and can be examined, studied, rotated by anyone that has an Internet connection without the need of any kind of expensive 3D CAD software.

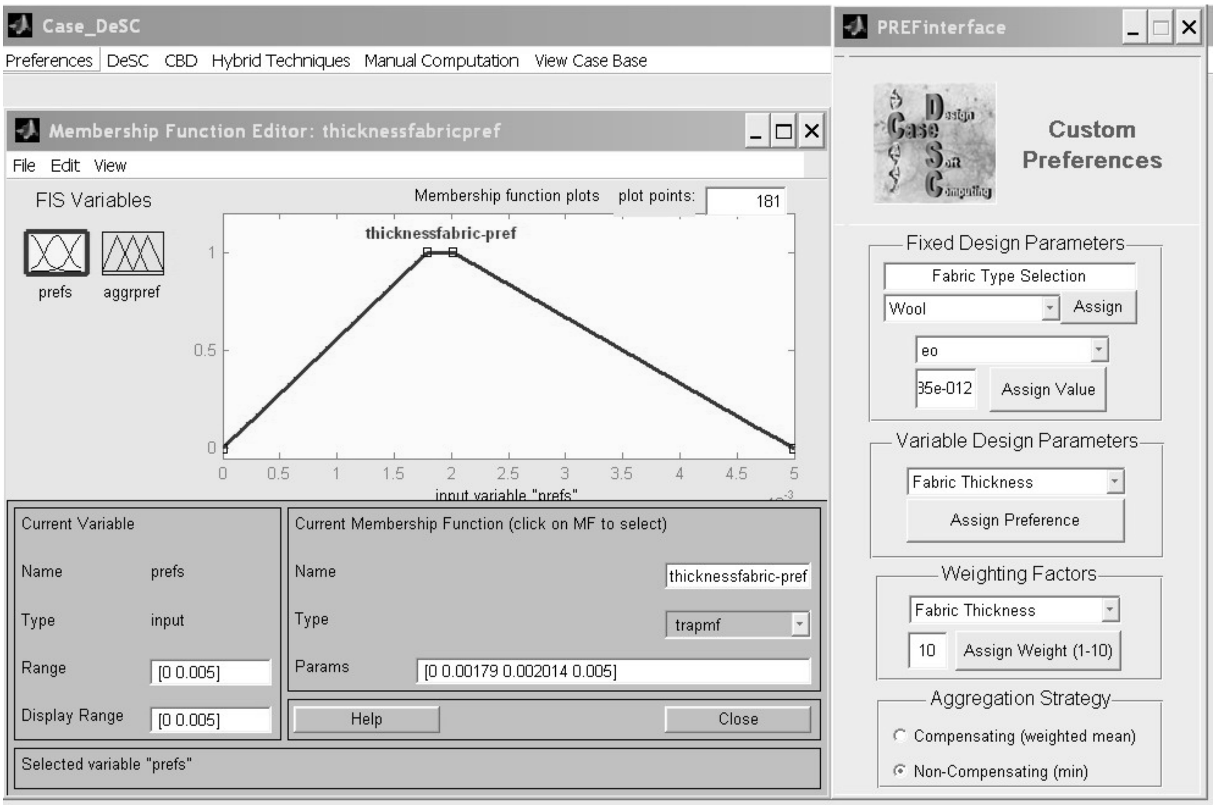


Figure 5. Assignment of values, fuzzy preferences, weights and aggregation strategy.

CONCLUSIONS

In the present paper, the design process of a electrostatic gripper for handling fabrics is presented as a three-step process, namely concept generation and evaluation, extraction of detailed parametric solutions and draft visualization of the final solution. In order to realize this process, a system consisting of different modules was developed in Matlab, in order to provide the designer support in all the aforementioned steps. The system utilizes fuzzy logic for the design case and concepts representation and evaluation, genetic algorithms for the extraction of optimal solutions, artificial neural network for the retrieval of past design solutions and is supplemented by a VRML module for the visualization of the final solutions.

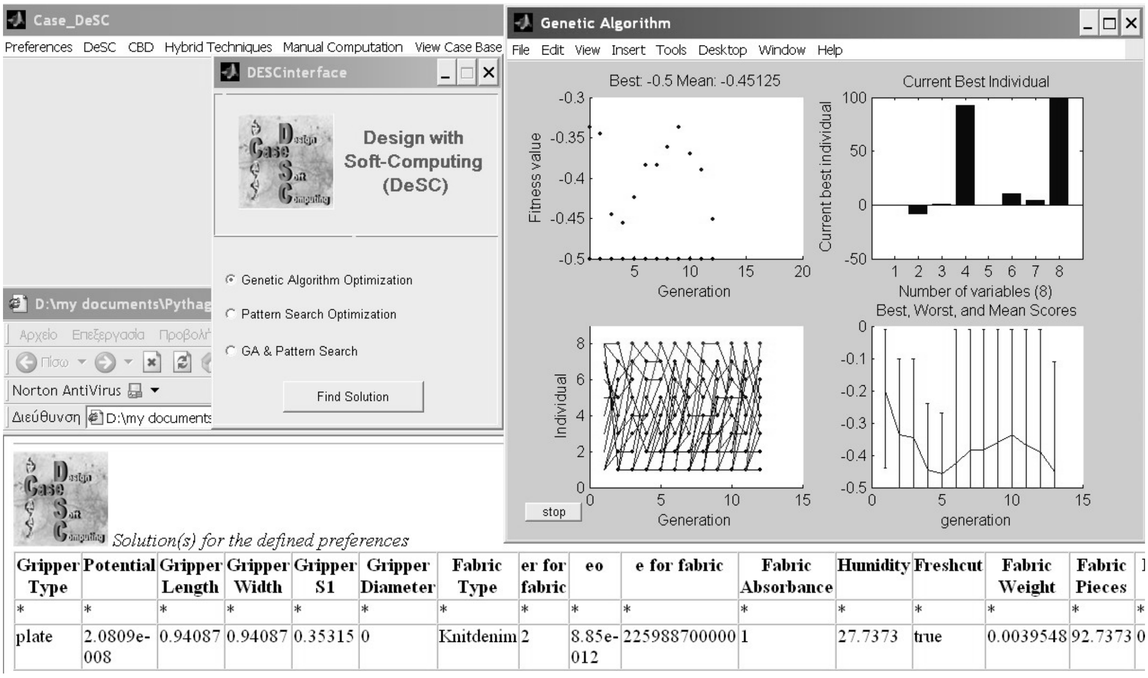


Figure 6. Deployment of genetic algorithm for the extraction of optimal solution.

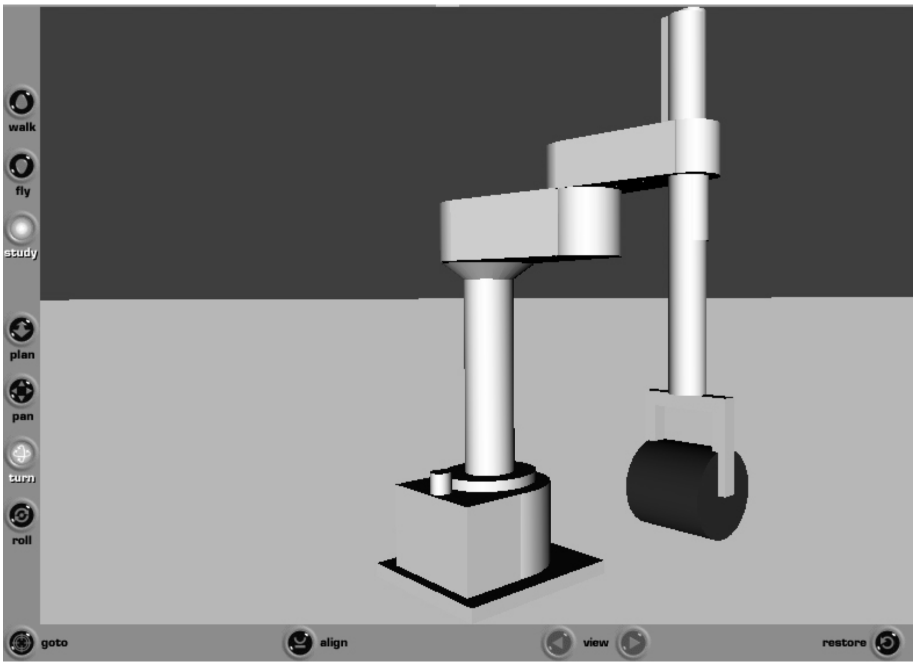


***Figure 7:** VRML 3D model of a SCARA robot attached with the produced cylindrical tool*

The system was tested on an application of the design of robotic grippers for flexible materials and succeeded in providing: a. a single concept – among nine different available concepts - that satisfies the posed design requirements, b. an optimal detailed solution based on that concept that complies absolutely with the preferences set by the designers and c. a graphical representation of the gripper that is available via the Internet to multiple users allowing each one of them to examine and evaluate virtually the gripper model.
The proposed system provides both the generality for addressing a variety of design problems and the extendibility for integrating additional modules and other hybrid architectures. Future research work should focus on a. improving the efficiency and the interface of the existing modules b. on providing collaboration basis among multiple designers on each of the described modules, c. to enhance visualization in conceptual module and d. to create a web-based platform on which all the developed modules will be deployed, thus enhancing communication and knowledge availability. In addition, the development, test and comparison of different gripper designs will contribute towards the validation of the proposed system.

AKNOWLEDGMENTS

The present work has been funded by the European Social Fund (ESF), the national Operational Program for Educational and Vocational Training II (EPEAEK II) and the Program PYTHAGORAS II.

REFERENCES

[1] S.Finger, J.R. Dixon,. A review of research in mechanical engineering design. Part I: descriptive, prescriptive, and computer-based models of design processes, Part II: representations, analysis and design for the life cycle, *Research in Engineering Design*, 1989, (1), 51-67, 121-137.
[2] N.F.O. Evbuowman, S. Sivaloganathan, A. Jebb. A survey of design philosophies, models, methods and systems, *Proc Insts Mech Engrs Part B, Journal of Engineering Manufacture*, 1996, 210, 301-320.
[3] L. Wang, W. Shen, H. Xie, J. Neelamkavil, A. Pardasani. Collaborative conceptual design-state of the art and future trends, *Computer-Aided Design*, 2002, 34, 981-996.
[4] J.R. Dixon, M.K. Simons. An architecture for application of artificial intelligence to design, 21st Design Automation Conference IEEE, 1984, Paper 40.2, pp.634-640.
[5] W.P.S. Dias. Reflective practice, artificial intelligence, and engineering design: Common trends and interrelationships, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 2002, (16), 261–271.
[6] A. Papavasileiou, E. Gegov, K. Gavros. Application of the artificial intelligence methods in CAD/CAM/CIM systems, *Annual of the University of Mining and Geology "St. Ivan Rilski", Mechanization, electrification and automation in mines*, 2002, Vol. 44-45, part III, Sofia, pp. 75-79.
[7] T.W. Simpson, J.D. Peplinski, P.N. Koch, J.K. Allen. Metamodels for computer-based engineering design: survey and recommendations, *Engineering with Computers*, 2001, 17, 129–150.
[8] S.H. Liao. Expert system methodologies and applications—a decade review from 1995 to 2004, *Expert Systems with Applications*, 2005, 28, 93–103.
[9] S.S. Rao, A. Nahm, Z. Shi, X. Deng and A. Syamil. Artificial intelligence and expert systems applications in new product development-a survey, *Journal of Intelligent Manufacturing*, 1999, 10, 231-244.
[10] C.H. Hsu, B.C. Jiang, E.S. Lee. Fuzzy neural network modeling for product development, *Mathematical and Computer Modeling*, 1999, 29, 71-81.
[11] A. Kamiya, S.J. Ovaskab, R. Roy, S. Kobayashi. Fusion of soft computing and hard computing for large-scale plants: a general model, *Applied Soft Computing*, 2005, 5, 265–279.
[12] Y. Dote, S.J. Ovaska. Industrial applications of soft computing: a review, *Proceedings of the IEEE*, 2001, 1243–1265.
[13] N. Cross. Natural intelligence in design, *Design Studies*, 1999, 20, 25–39.
[14] C. Shakeri, D.C. Brown. Constructing design methodologies using multiagent systems, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 2004, 18, 115–134.

[15] Moulianitis V.C., Dentsoras A. J. and Aspragathos N. A. A knowledge-based system for the conceptual design of grippers for handling fabrics. *Artificial Intelligence in Engineering, Design, Analysis and Manufacturing*, 1999, 13, 13-25.

[16] Saridakis K.M. and Dentsoras A.J. Case-Desc: A system for Case-Based Design with Soft Computing Techniques. *Expert Systems with Applications*, 2007, 32, pp. 641–657.

[17] Rener G, Ekart A. Genetic Algorithms in computer aided design, *Computer Aided Design*, 2003, 35, 709-726.

[18] Goldberg, D.E. Genetic algorithms as a computational theory of conceptual design. *Applications of Artificial Intelligence in Engineering*, 1991, 6, pp, 3-16.

[19] Bentley PJ, Wakefield JP. Conceptual Evolutionary Design by a Genetic Algorithm. *Engineering Design & Automation*, 1997, 3, 119-131.

[20] Lin J-J. Constructing an intelligent conceptual design system using genetic algorithm. *Journal of Materials Processing Technology*, 2003, 140, 95-99.

[21] V.C. Moulianitis, N. A. Aspragathos and A. J. Dentsoras . A model for concept evaluation in design- An application to mechatronics design of robot grippers. *Mechatronics*, 2004, 14, 599–622.

[22] Moulianitis V. C. and Aspragathos N. A. Design evaluation with mechatronics index using the discrete Choquet integral. *Mechatronics 2006, 4th IFAC-Symposium on Mechatronic Systems*, Heidelberg, September 2006.

[23] Grabisch M., The application of fuzzy integrals in multicriteria decision-making. *European Journal of Operation Research*, 1996, 89, 445-456.

[24] MATLAB © The Mathworks Inc, version 7.0.1.24704, User's Guide.

[25] Pal, K.S. & Dillon, S.T. & Yeung, S.D. *Soft Computing in Case Based Reasoning*, 2001, (Springer Verlag, London).

[26] Kasabov, K.N. *Foundation of Neural Networks, Fuzzy Systems and Knowledge Engineering*, 1996, (MIT Press).

[27] Saridakis K.M. and Dentsoras A.J. Collaborative Parametric Design with Soft Computing Techniques. *Advanced Engineering Informatics*, 2006, 32, 379–399.

[28] http://www.parallelgraphics.com/products/cortona/

Contact: A.J. Dentsoras, Assoc. Prof.
University of Patras
Department of Mechanical Engineering and Aeronautics
Patras, 26500
Greece
+30 2610 997217
+30 2610 997217
dentsora@mech.upatras.gr
http://www.mech.upatras.gr/~dentsora.