12th International Research/Expert Conference
“Trends in the Development of Machinery and Associated Technology”
TMT 2008, Istanbul, Turkey, 26-30 August 2008

THE DEVELOPMENT OF GENETIC ALGORITHM FOR SYNTHESIS OF PRODUCT COMPOSITION STRUCTURES

Eduard Napalkov
Department of Engineering Science and Transport, Riga Technical University,
Str. Ezermalas 6, Riga LV-1014, Latvia

ABSTRACT

In early stages of design, the problem of a product development can be viewed from a standpoint of composition synthesis realized in a network of interdependent design artefacts. Here, the term ‘composition synthesis’ means a gluing the product structure from suitable design artefacts through their multiple connectors (input/output ports). This paper presents the model of genetic algorithm developed with the aim of improving a quality of generated composition structures. It involves main steps of genetic algorithm, fitness function and some formulas used for evaluating intermediate transformations.

Keywords: Case Based Design, Composition Structure, Genetic Algorithm.

1. INTRODUCTION

Composition synthesis can be briefly defined as an extension of case based reasoning (CBR) design [1] with emphasis on the modularity to provide more flexible interface between interacting graphic images. The basis is a formal representiation of any product graphic image in form of so called function means pattern (FM-pattern) including one central and a finite number (e.g. no more than four) subordinate connectors (ports) [2]. Central connector is intended to identify the product construction as a whole, its behavioral law and major function. Subordinate connectors identify the most important constituent parts (design organs) and their related functions. The presence of such connectors enables the designers to analyse the different variants of connecting FM-patterns to create thereby composition modular structures of products satisfying in the best way the given functional requirements and design constraints. If graphical images (sketches, drawings, layouts, CAD models) play a role of design artefacts, then we deal with the problem of creating composition modular structures that mediate between a product real scheme and its visual representation.

Due to modularuty, the process of composition synthesis is automatized much easer than CBR design, which typically requires considerable efforts on the parametric and structural adaptation of design cases being selected from a case library. At the same time, a variety of interfaces used in automatic connecting FM-patterns leads to impossibility of applying gradient or other numerical methods for solving optimisation tasks of composition synthesis. The proposed approach is the development of genetic algorithm to achieve a desired quality of generated composition structures.

2. ENCODING COMPOSITION STRUCTURES

Genetic algorithms (GAs) are technique to model the biological evolution. Their application includes a representation of chromosomes (the candidate solutions), an initial population of chromosomes, a set of genetic operators to generate new candidates, evolution (fitness) function, and a selected method [3]. According to the proposed approach, the chromosomes are composition structures, and their gens are FM-patterns of design artefacts. Each composition structure is defined as a derivation tree to be concerned with certain collection of given functional requirements and design constraints.

In totality, composition structures should reflect design experiences and skills in certain domain of product development. Saved in a case library, they can be selected for initial population by analogy with CBR design, i.e. by retrieving the similar cases with respect to the designer query. Another way is to filter design cases through functions given in the query. A generic condition is to maintain a ratio like $\nu > n$ between a size ν of initial population *P(0)* and a maximal number *n* of FM-patterns in a designed composition structure. This condition is accomplished under a control of the designer by sequential changing functions in a filter. The result should be a selection of those composition structures, which cover in parts all functional requirements and design constraints given. In a good selection, for $n = 10$, an adequate one is a value $\nu \approx$ 12-15.

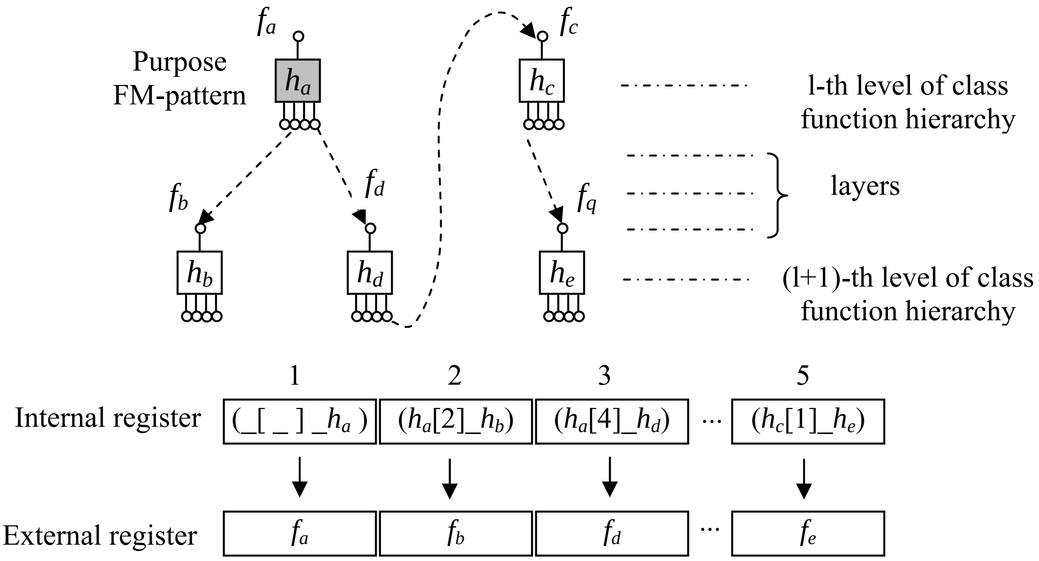

Figure 1. Example of encoding a composition structure

In classical GA, chromosomes are often encoded as bit strings of fixed length. On the contrary, in encoding composition structures, it is necessary to use explicit symbols and two variable registers concerned with each other. The first one called an internal register describes the hierarchy of relationships within a designed composition structure, and the second one called an external register contains functions corresponding to these relationships.

One of possible examples is shown on fig.1. Here, the composition structure consisting of five FM-patterns has a code represented by internal register of enumerated tree edges (from 1 to 5), each of which is recorded like (h_a[*q*]_h_b), i.e. parent h_a is linked with offspring h_b through connector *q*. The first locus of that code is always assigned for the root vertex (_ [_]_ h_a). A sequence of recording other tree edges doesn't matter. It follows an external register of central functions (f_a, f_b, …, f_e) performed by FM–patterns.

Thus, a layout of composition structure is reproduced in top-down fashion. The aim of GA is to find solutions better than in the current population *P(t)* by means of genetic modifying the selected composition structures provided that external register of each solution will contain all functions given in the designer query, where $t = 1,2,\ldots, T_{max}$ (T_{max} is maximum number of populations generated).

3. FITNESS CALCULATION

In order to generate the next population of candidate solutions, it is required to evaluate the fitness value φ_i ($i = 1,2,.., n$) for each solution in the current population. It can be calculated taking into account the evaluation of both an expected cost of a product and a complexity of designed hierarchical structure.

From a standpoint of composition synthesis technique, the most important one is the complexity that is characterized by a total number of nodes and layers in hierarchical structure including evaluations of functional, behavioral and structural compatibility of interacting nodes [4]. Each of these optimization subtasks is NP-hard. Therefore, with the aim of reducing the laboriousness of calculations and simplifying a total optimisation problem, we discuss a single objective GA, in which the objective function is:

Minimize the complexity of composition structure designed subject to given requirements of the designer query, given restrictions on a product total cost and a treshold of FM-patterns compatibility.

In our sense, the minimization of complexity means a drive for design of composition structures with a maximal coefficient of connectors' usage (*CU*). In mathematical form, a coefficient *CU* of any solution is expressed by the following way:

$$CU = 1/n\ (cu_1 + cu_2 + \ldots + cu_j + \ldots + cu_n) = (m_1 + m_2 + \ldots + m_j + \ldots + m_n)/nm\ ,$$

where cu_j is a coefficient of connectors' usage that is related to j-th FM-pattern ($j = 1,2,..,m$); m_j is a number of inputs activated in j-th FM-pattern; m is total number of inputs in one FM-pattern ($m = 4$). If the j-th FM-pattern is the leaf node, then all its inputs interact with environment, i.e. $m_j = m$ and $cu_j = 1$. For example, a value *CU* of composition structure shown on fig.1 is equal to 1/5 (0,5 +1,0 + 0,25 + 0,25 + 1.0) = 0,6. The closer a value of *CU* to one, the less a tree depth, i.e. the less a number of layers will be in designed structure. Consequently, one may expect of increasing an internal connectivity of design artefacts, and thereby an increasing a reliability of composed structures.
Fitness is expressed in terms of scalar valued objective function. Therefore, a value CU_k corresponds to evaluation of fitness φ_k with respect to reproduction capability of the solution H_k.,$k \in \{1,2,\ldots, n\}$.

4. GENETIC OPERATORS

After all solutions in *P(t)* have been assigned finess values, the generation of new population *P(t+1)* is produced by the application of genetic operators such as selection, crossover and mutation.

4.1. Selection

Since the fitness values of composition structures differ no sharply with each other, the roulette method of selection is suitable for the developed GA This is a way to calculate the selection probability p_k of solution H_k. as the proprtion to its fitness as follows:

$$p_k = CU_k / \sum_{i=1}^{n} CU_i$$

Sectors of the roulette wheel are made according to these probabilities. As a result of spinning the roulette wheel *n* times the population of the parents for a crossover is produced so that a total number of parents (without of their copies) would not be smaller than $R_{sel} \cdot n$, where R_{sel} = (0,6-0,8) is an assigned parameter of GA. Moreover, the selected parents stored in a mating pool must satisfy in their totality all requirements and constraints given in the designer query. Otherwise, the selection procedure must be repeated.

4.2. Crossover

Crossover operator selects at random the parent pairs from the mating pool to generate one or two offsprings. The feature is that the crossover selects also at random some locus of internal register from the first parent to cut off the corresponding sub-tree. By analogy, the second sub-tree belonging to other parent is isolated to produce the replacement of these sub-trees. The main strategy is to select parent pairs with an Euclidean distance $\omega \geq \omega^*$ between them, where ω^*=.(0,3-0,4).
As a result of the replacement, the two new internal registers including the related external registers are formed. Here are remained only those inherited composition structures, whose external registers will contain all given functions. This procedure is performed in accordance with the crossover rate R_{crs} = (0,4-0,6). Other feature is the necessity to evaluate a compatibility of FM-patterns in points of

their connection In case of failure the attempt is made to use other vacant inputs of a paired connection. If not, then the next parent pair is selected for crossing.

In creating paired connections of FM-patterns, a single flow of power, materials or signals must be provided. Therefore, the necessary conditions of input-output compatibility are:

- identity of entitles and entities of interacting functions;
- identity of entitles of interacting design organs;
- similarity of interacting behaviors;
- similarity of interacting graphical images.

A performance of these conditions is based on comparing functional and design parameters of FM-patterns in points of their connection including the use of similarity metrics.

4.3. Mutation

The mutation operator has a rate R_{mut} = (0,05-0,1) maintaining thereby a certain diversity in the population in order to decrease a probability of falling the best solutions into local optimums. It is used to generate new offsprings by modifying the parent compositional structures selected at random in accordance with uniform strategy. New offsprings are checked on the complitness of their external registers to remove invalid ones. After mutation the fitness value of each solution is evaluated by the objective function and the new population $P(t+1)$ is created. Simultaneously, the best solutions can be copied in the elite register used for additional application of the elitist strategy in GA [5].

5. CONVERGENCE

The global optimum value of the objective function is achieved if the stochastic sequence $\{CU\}$ converges towards CU^*. Therefore a size ν of initial population, ratio R_{sel}, R_{crs} and R_{mut} are regardes as essential variation parameters of GA. However, due to unpredictable value that CU will reach for each experiment, a convergence criterion based in the number of generations T_{max} should be used.

6. CONCLUSION

In this paper a model of CBR design based on composition synthesis of treelike function means structures with the application of a single objective GA has been described. This model considers the design component as a module represented in form of multiconnector FM-pattern. To provide an optimal interaction of FM-patterns in a designed hierarchical structure, the coefficient of connector's usage is introduced as the fitness function of GA. The main contribution is to show new opportunities for perfection of existent methods of CBR design and in making the product closer to the user requirements.

7. REFERENCES

[1] Wilson, D.C., Leake, D.B. Maintaining Case-Based Reasoners: Dimensions and Directions. Computational Intelligence, Vol.17, Issue 2, pp 196-213, 2001.

[2] Napalkov, E; Zars, V. Synthesis of Product Function/Means Structures Based on Visual Thinking. Proceedings of the 9th International Conference – "Design 2006", Dubrovnik, Croatia, pp 579-586, 2006.

[3] Reeves, C.R. and Rowe, J.E. Genetic Algorithms – Principles and Perspectives: A Guide to GA Theory. Kluwer Academic, Boston, 2003.

[4] Napalkov, E. Structural Composition of Product Patterns within Class Function Hierarchical Network of Design Artifacts, Proceedings of the International Conference "The Modern Information Technology in the Innovation Processes of the Industrial Enterprises" – MITIP2007, Florence, Italy, pp 446-451,2007.

[5] Dreo, J., et al. Metaheuristics for Hard Optimisation. Methods and Case Studies, Springer, 2003.