Optimization and Selection of Alternative Modular Structures using Genetic Algorithms

Orlando Durán
Pontificia Universidad Católica de Valparaíso, Esc. Ing. Mecánica,
Av.Los Carrera 1567, Quilpué, Chile, orlando.duran@ucv.cl
Luis Perez
Universidad Técnica Federico Santa María
Depto. Ing. Mecánica, Aula UTFSM-CIMNE. Av. España 1680, Valparaíso, Chile.
luis.perez@usm.cl
Nibaldo Rodriguez
Pontificia Universidad Católica de Valparaíso, Esc. Ing. Informática,
Valparaiso, Chile, nibaldo.rodriguez@ucv.cl

Abstract

In most transformations from dedicated to modular approaches, products are assumed to have a unique modular structure. However, it is well known that alternatives for constructing modular structures may exist in any level of abstraction. Explicit considerations of alternative structures invoke changes in the number of module instances so that lower capital investment in modules, more independency of structures and higher efficiency can be achieved. Relatively few research papers that deal with the optimization of modular structures problem with alternative assembly combinations aiming at minimization of module investments were found in the literature. A genetic algorithm (GA) was applied to solve the optimization problem of selecting and combining the alternative of modular structures to create a set of modular structures minimizing the cost involved in its implementation. Test results are presented and the performance of the proposed GA is compared with solutions obtained from total enumeration tests.
Keywords: Modularization, Genetic Algorithms, Modular structures.

Resumen

En transformaciones a estructuras modulares no siempre existe apenas una única forma de cumplir los requerimientos funcionales y estructurales, esto es, existe más de una configuración o combinación de módulos para el ensamble de un determinado producto o estructura. Al existir alternativas de configuración modular, estas tendrán probablemente diferencias en sus costos. Pocos trabajos han abordado el problema de optimización del problema de selección de alternativas de configuraciones modulares desde el punto de vista del costo. Se propone aquí la aplicación de un algoritmo genético para proveer la selección de las estructuras modulares al más bajo costo posible. Se muestran detalles del modelamiento del problema, resultados de diferentes pruebas y comparaciones con las soluciones para un conjunto de problemas cuyas soluciones han sido encontradas por búsqueda exhaustiva.
Palabras clave: Modularización, Algoritmos Genéticos, Estructuras modulares

INTRODUCTION

Market competitiveness forces many industries to evolve toward mass customization (MC), which aims at satisfying individual customer needs while keeping mass production efficiency. Modularity is a well-established strategy for attaining MC.

Modularity is one of the primary means of achieving flexibility, economies of scale, product variety and easier product maintenance and disposal. Modularity is a general concept; and many engineering problems can be generalized under the umbrella of modularity. [3] defined modularity as a concept that is applied to manage complex systems by breaking them down into parameters and tasks that are interdependent within and independent across the modules. Yet according to [4], modular units are highly interconnected in themselves, but largely independent of other units. The characteristics of modularity comprises the use of a finite set of components to meet the infinite changes of the environment; establish the module by reviewing the similarities among the components; keep as much independence of the resulting structures as possible and use different modules for different varieties of assemblies.

The motivation of modularization of a product or a structure is to meet the ever-changing demand for the needs of the product, in particular to achieve rapidly the maximum flexibility. The modularization should result in an architecture of a product or structure such that the product can be made by simple assembling pre-existing components. To realize such a modular architecture, product functions, product life cycle issues, and cost should be considered.

The replacement of dedicated structures by modular ones seems to be a trend in the manufacturing field, especially in meeting the desire for greater flexibility. Products built around modular architectures can be more easily varied without adding too much complexity to the manufacturing system, and moreover, a modular architecture makes the standardization of components more possible. At the same time this approach allows the reuse of tools, equipment and expertise, and avoids costly changeovers for personalized products. According to [13] there are three general fields where modularity could be implemented: modularity in design, modularity in use and modularity in production. Despite these clear benefits, a formal theoretical approach to modularity is still lacking and designers are often skeptical regarding the advantages of modularity. This is largely due to the inferior performance obtained by modular designs compared to their custom built optimal alternatives.

Determination of modular configuration is described in [12] as, “Given a set of candidate modules, produce a design that is composed of a subset of the candidate modules and which satisfies both a set of functional requirements and a set of constraints”. From this definition, it can be seen that here we assume that modular architectures of a particular product of structure is ready, and modular components and their interactions are predefined and available.

There are many domains where a wide variety of modular designs are available. That is, there is more than a unique form to build a given solution or a structure using a set of modular components from a given and a finite set. Depending on the specific domain at hand, many researchers have reported automated systems and methodologies for define one or more modular configurations for a given application, i.e. [1] developed an integrated method for designing modular products. To test and validate the methodology it was applied to a domestic gas detector product family. Hornby et al. [7] developed an automatic design system that produces complex robots by exploiting the principles of regularity, modularity, hierarchy, and reuse. Liu et al. [10] used genetic algorithms for intelligent design of automobile fixtures. Babu et al. [2] developed an automated fixture configuration design system to select automatically modular fixture components for prismatic parts and place them in position with satisfactory assembly relationships. Finally, Retik and Warszawski [14] developed a knowledge-based system for the detailed design of prefabricated building. [11] discussed the modularity in design of modular fixtures.

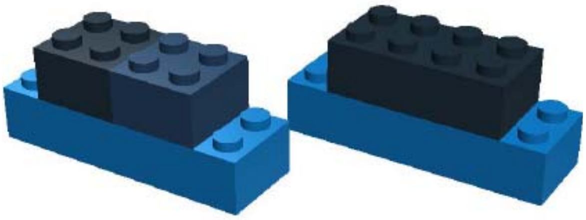
Benefits of a standardized modular design approach overtake customized approaches. Modularity attributes are influenced by the technologies used in the modular components, the work agents that perform the assembly, and their supporting infrastructure ([9], [8]).

The definition of a modular alternative requires comparative estimates of time, performance and cost among alternative of modular configurations. Recent applications have used cost models and geometric optimizations based on the physical properties (mass, volume) of candidate modules. To evaluate the economic impact of modularity, a cost approach is needed to compare alternative modular cases. The differences in costs among modular system alternatives can be used to identify preferred configurations.

The calculation of cost for a modular alternative, C1, versus another modular alternative, C2, must be computed. If there is net savings, then there is one modular alternative that indicates savings over the other alternative. For the sake of simplicity here, the following discussion assumes that the infrastructure costs and operational costs are common to both alternatives since the technologies are likely to be comparable.

PROBLEM STATEMENT

Consider a simple example, a lego-based figure, where a finite number of alternatives of combinations, using modular components, are possible for a given figure. See figure 1, where two alternatives of combinations to construct the same figure are shown. In figure 1a three components were used and in figure 1b two components were used. Each modular component has a known cost. If number of figures is needed to be assembled at the same moment, and several modules are common to some of these figures, and, if there is a limited number of each one of the modular components, the challenge here is selecting the optimum combination of modular designs for each one of the lego-like figures at a minimum cost.



(a) (b)

Figure 1. Two alternative configurations for constructing the same lego-like figure.

This research provides a systematic design method to help dedicated systems users to convert to modular system users. Modular systems are comprised of many modular elements and modules that can be stacked in a flexible way. The main purpose in using modular systems is as a substitute for more expensive, time consuming dedicated systems. However, a large-scale modular system can only serve large companies –what can a small company do, limited by financial concerns and pragmatic demand? This paper proposes a method which helps companies change their dedicate systems gradually into modular fixturing systems, no matter what the size of the company. The advantage of using modular systems over dedicated ones is that the elements and modules can be used repeatedly. The search may provide several alternatives for the designer who then may choose one of them for each one of the workpieces.

The optimization is achieved through appropriately selecting the subsets of module instances from given sets. The proposed formulation is general in the sense that products can have any number of modules. Through modularity, the number of different parts to be purchased for an assembled product set may be significantly reduced while achieving a sufficient variety by combination of different modules. In general, each module may have more than one instance and any each assembled product may have more than assembled combination of modules. The different alternative assemblies may provide the same capabilities and even functionalities that the required ones. Furthermore, current work may accommodate simultaneity constraints where the selection of a particular instance of a module necessitates the use of a particular instance of another module. The method is applied to a general problem and is found to be efficient in determining optimum subsets of each module from a given set. It is anticipated that the proposed method can be used as a systematic tool in selection of modules instances in designing and assembling modular products or transforming dedicated structures into modular ones. In the case of a custom made product the manufacturing system should be set up to produce a different product variant for each parameter set, representing a different specification, besides the large number of components that are needed to comprise all the combinations. In a modular-oriented environment, the number of required modular instances may be significantly reduced since modularity provides the desired variety of the product through different combinations of modules. Two key issues in modular product design are to determine the optimum product variety and the number of module instances required to support this variety. Without loss of generality, it is assumed that all module instances are compatible with each other (i.e. standardized interfaces across all interfacing modules) For each parameter set, the best possible product variant is selected as a result of the best combination of module instances. The problem is to select the subsets of modules instances that minimize a combined objective function that provides a trade off between the quality of the modular product and the cost of purchasing or manufacturing each one of the modules.

The approaches of transformation of dedicated systems into flexible assemblies have received attention for bringing advantages inherited from modularity. Since assembly alternatives and modules can be replaced under the constraints of using alternative assemblies and functionally, respectively.

The remaining part of the paper is organized as follows: problem specification and formulation of objective functions is presented in Section 1, working mechanism and development of Grouping Genetic Algorithm for selection and optimization of modular structures is illustrated in Section 2, computational studies are reported in Section 3 and conclusions are given in Section 4.

GENETIC ALGORITHMS

The basic algorithm of the GA is given as follows [6]:

1. Generate random population of chromosomes.
2. Evaluate the fitness of each chromosome in the population.
3. Test if the end condition is satisfied, stop, and return the best solution in current population.
4. Create a new population by repeating following steps until the new population is complete:
    - Reproduction: Select two parent chromosomes from the population according to their fitness.
    - Crossover: With a crossover probability, crossover the parents to form a new offspring (children).
    - Mutation: With a mutation probability, mutate new offspring.
5. Replace: Use new generated population for a further run of algorithm.
6. Go to step 2.

In the next section the proposed mathematical formulation is presented.

MATHEMATICAL FORMULATION

In this paper, we attempt to solve a generalized selection and optimization problem in modular construction of a set of figures. Consider a situation having ‘F’ figures and ’M’ modules. Each figure is characterized by its structure, and each structure is comprised for a given combination of modules. Parts may have S alternative structures and all parts do not have equal number of alternative structures. Under a certain modular configuration for a figure, there is associated with a given cost, in correspondence to the total number and types of modules to construct the referred structure or figure. Thus the problem is to find optimal parts configurations with their modules combinations to minimize total costs. The mathematical model for cell formation problem, in the presence of multiple modular configurations is presented below.

As it was mentioned, the problem consists in determining the minimal cost required for assembly all the figures simultaneously, selecting the appropriate combination of alternative assembly for each one of the figures. Here, all the figures are to be assembled simultaneously which it means that the need of some modules could be incremented according to the number of figures that use a specific assembly option that considers the module at hand. Let us introduce the elements of this optimisation model. The optimisation model is stated as follows. Let:

- M be the number of modules,
- F the number of figures,
- S the number of alternative assemblies or set ups,
- i the index of figures (i = 1,….,F),
- j the index of modules (j =1,…..,M),
- k the index of alternative assemblies (k=1,…..,S),
- A = [a_{ij}] the F x S binary incidence matrix,

Given an incidence matrix F= [f_{ijk}], where f_{ijk} = n ,

Where k represents the number of modules of the type j are used by the figure i in the alternative of assembly k.

Each module j has a unitary known cost cj. We selected as the objective function to be minimized the cost of a given set of set up of modules to assembly a set of figures to be constructed simultaneously.

$$Min: Z = \sum_{i}^{f} \sum_{j}^{m} \sum_{k}^{s} F_{ijk} C_j A_{ik}$$

$$F_{ijk} = \begin{cases} n & \text{if figure i uses n modules of the j type in the k assembly alternative} \\ 0 & \end{cases}$$

otherwise.

$$A_{ik} = \begin{cases} 1 & \text{if figure i will be assembled using the assembly alternative k} \\ 0 & \text{otherwise.} \end{cases}$$

Subject to

$$\sum_{k=1}^{C} A_{ik} = 1 \qquad \forall i,$$

Figure 2 shows an example of the structure of the problem.

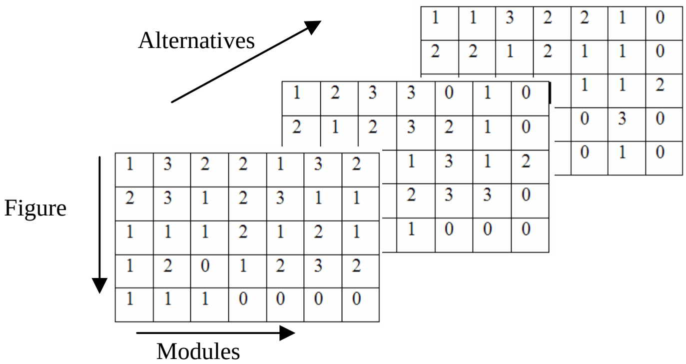


Figure 2. Generic structure of the selection and optimization problem

Optimization Model Implementation

Individual representation

In order to generate the chromosomes, the length of the chromosome is calculated first. Then random numbers in the range of {0, S} are generated to form the chromosome. Each gene represents the alternative of module subset to which each part is constructed. The chromosome representation can be represented as shown in bellow where Mj denotes the alternative to which figure j has been assigned. A chromosome encoding for the described problem is shown below:

$M_1 \quad M_2 \quad M_3 \quad \ldots\ldots \quad M_M$

For example, consider the following 3 chromosomes where figures can be comprised by 10 modular components in 4 alternative assemblies maximum.

1	2	4	2	3	3	1	2	3	2
2	1	2	1	1	2	3	3	1	1
2	2	1	2	1	2	3	3	4	1

The first chromosome indicates that figure 1 is constructed using its modular alternative number 1, as well as figure number 7. As such, figures 2,4,8 and 10 are considered as using its alternative number 2. Figures 5,6 and 9 are constructed according to alternative structure number 3. Figure number 3 is constructed according the alternative structure number 4. Note that the chromosomes are fixed in length.

Selection Process

In this work, reproduction is accomplished by copying the 10% of the best fitted individuals from one generation to the next. This approach is often called an elitist strategy. Parameterized uniform crossovers are employed. After two parents are chosen randomly from the full old population (including chromosomes copied to the next generation in the elitist pass), at each gene a biased coin is tossed to select which parent will contribute the offspring. To prevent premature convergence of the population, at each generation 1% of new members of the population are randomly generated. This scheme was based on [5]. Figure 3 depicts the used evolutionary process.

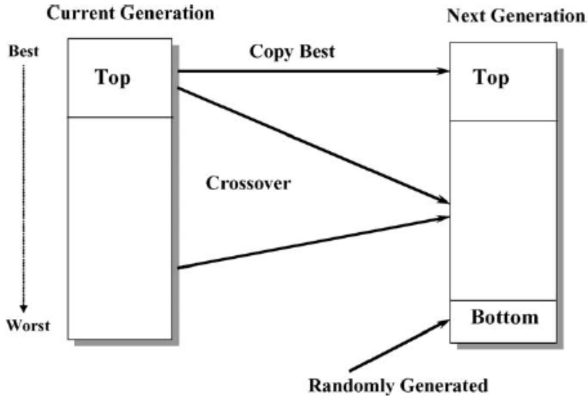


Figure 3. Evolutionary process used in this approach

NUMERICAL RESULTS

The purpose of this section is to show, using a series of numerical examples, how the proposed formulation can be used as an aid to configure a modular component system and to provide an assessment of the performance of the implementation of a genetic algorithm.

Nine test problems were used to evaluate the proposed implementation of the G.A. Each experiment considers a problem with 25 figures each one with 2 options of configuration. The module population is comprised by 20 different instances. These 9 problems were first solved through an extensive search, so that their optimal objective functions are known. Each experiment took about 16 hours on a PC with Intel Pentium 4 running at 3.6 GHz under MS windows. Table I shows the minimum cost obtained in each case. Then the same problems were solved by the proposed G.A. Two sets of experiments were run. The first set considered a population size of 30 individuals and 90 iterations. The second set of experiments considered the same population size and 150 generations. All experiments were run 10 times; the results of the first set of experiments are shown in Table II where optimal solutions are indicated in bold. The results of the second set of experiments are shown in Table III. A comparison of these two tables shows that, using the above values of the G.A. parameters it was possible to find the optimal solution for 44 (49%) of the 90 tests in the case of the first reported configuration and 82 (91%) over the 90 tests.

Table I. Results obtained through extensive search

Test number	Minimum
1	3583
2	3877
3	2622
4	3426
5	3491
6	3290
7	3714
8	3279
9	2921

Table IV shows the mean deviations (%) between the optimal solution and the best solution obtained from the 10 executions of the proposed G.A. for each test. It can be seen that the G.A. with 150 generations obtains the optimum a large number of times with small differences in the rest of occasions. The standard deviation (s) of the error of the different runs is relatively low, indicating that the performance of the algorithm is stable.

Table II. Results from the G.A. with 90 generations

	Optimal Solution	1	2	3	4	5	6	7	8	9	10
1	3583	3586	3585	**3583**	3585	**3583**	3584	3584	3590	3588	**3583**
2	3877	3885	**3877**	**3877**	**3877**	3881	3878	3878	**3877**	**3877**	**3877**
3	2622	2626	2625	2625	2623	2624	2627	**2622**	2631	2624	**2622**
4	3426	3441	3430	3430	3432	3428	**3426**	**3426**	**3426**	3428	**3426**
5	3491	3495	**3491**	**3491**	3492	3503	3495	**3491**	**3491**	**3491**	**3491**
6	3290	**3290**	**3290**	**3290**	3292	3292	3295	3302	**3290**	3291	3291
7	3714	**3714**	**3714**	**3714**	**3714**	3718	**3714**	**3714**	**3714**	**3714**	3718
8	3279	3289	3285	**3279**	3289	3285	3285	**3279**	3290	**3279**	3283
9	2921	**2921**	2930	**2921**	**2921**	**2921**	2926	**2921**	**2921**	**2921**	**2921**

Table III. Results from the G.A. with 150 generations

	Optimal Solution	1	2	3	4	5	6	7	8	9	10
1	3583	**3583**	**3583**	**3583**	**3583**	**3583**	**3583**	**3583**	**3583**	3585	**3583**
2	3877	**3877**	3880	**3877**	**3877**	3880	3880	**3877**	**3877**	**3877**	**3877**
3	2622	**2622**	**2622**	**2622**	**2622**	**2622**	**2622**	**2622**	**2622**	**2622**	2625
4	3426	**3426**	**3426**	**3426**	**3426**	**3426**	**3426**	**3426**	**3426**	**3426**	**3426**
5	3491	**3491**	**3491**	**3491**	**3491**	**3491**	**3491**	**3491**	**3491**	**3491**	**3491**
6	3290	**3290**	**3290**	**3290**	3294	**3290**	**3290**	3294	**3290**	**3290**	**3290**
7	3714	**3714**	**3714**	**3714**	**3714**	**3714**	**3714**	**3714**	**3714**	**3714**	**3714**
8	3279	**3279**	**3279**	**3279**	3285	**3279**	**3279**	**3279**	**3279**	**3279**	**3279**
9	2921	**2921**	**2921**	**2921**	**2921**	**2921**	**2921**	**2921**	**2921**	**2921**	**2921**

Table IV. Gap between PSO-GT results (using reference set) and the optimal solutions

Gener.: 90			Gener.: 150		
Mean Deviation (%)	St.Dev. of the error	Number of instances	Mean Deviation (%)	St.Dev. of the error	Number of instances
0,06%	0,07%	3	0,01%	0,02%	9
0,04%	0,07%	6	0,02%	0,04%	7
0,11%	0,10%	2	0,01%	0,04%	9
0,10%	0,13%	4	0,00%	0,00%	10
0,06%	0,11%	6	0,00%	0,00%	10
0,07%	0,11%	4	0,02%	0,05%	8
0,02%	0,05%	8	0,00%	0,00%	10
0,16%	0,13%	3	0,02%	0,06%	9
0,05%	0,11%	8	0,00%	0,00%	10

CONCLUSIONS

A genetic algorithm was presented for selection and optimization in transforming structures into modular ones. This problem is found in industries that face gradual or complete transformation of these products to modular based assemblies. The optimization problem considers the cost involved in using a set of modules to obtain a number of modular assemblies that could substitute a set of non-modular products or structures. A set of problems was generated, and their optimal solutions were obtained with exhaustive search. The GA was tested and the results show that the GA obtained the optimal solution in all the cases maintaining a low variability of the results.

REFERENCES

[1] Asan U.; Polat S.; Serdar S., An integrated method for designing modular products. Journal of Manufacturing Technology Management, Vol. 15, Number 1 (2004), pp. 29-49(21)

[2] Babu BS, Valli PM, Kumar AVVA, Rao DN. Automatic modular fixture generation in computer-aided process planning systems. Proceedings of The Institution of Mechanical Engineers Part C-Journal of Mechanical Engineering Science. Vol.219(10), pp. 1147-1152, OCT (2005).

[3] Baldwin CY, Clark KB (2000) Design rules: the power of modularity. The MIT Press, Cambridge, MA

[4] Baldwin, C.Y., Clark K.B. (1997) Managing in an Age of Modularity. Harvard Business Review. 75/5. 84-93

[5] Goncalves, J.F, M.G.C. Resende, An evolutionary algorithm for manufacturing cell formation, Computers & Industrial Engineering 47 (2004) 247–273

[6] Holland, J.H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor (1975).

[7] Hornby GS, Lipson H, Pollack JB, Generative representations for the automated design of modular physical robots. IEEE Transactions on Robotics and Automation. Vol.19 pp. 703-719. AUG (2003)

[8] Ishii, K., "Product Modularity: A Key Concept in Life-Cycle Design," Frontiers of Engineering: Reports on Leading Edge Engineering from the 1996 NAE Symposium on Frontiers of Engineering (1997), http://www.nap.edu/openbook/0309057264/html/17.html, accessed August 22, 2005.

[9] Kingston, J., "Modularity as an Enabler for a More Efficient Commercial Small Satellite Program," Proceedings of the 17th Annual AIAA/USU Conference on Small Satellites, 2003, SSC03-III-8

[10] Liu T., Chen CH, Chou JH, Intelligent design of the automobile fixtures with genetic algorithms. International Journal of Innovative Computing Information and Control. Vol.4(10), pp. 2533-2550, OCT (2008).

[11] Liu, C., 1994, "A Systematic Conceptual Design of Modular Fixtures", The International Journal of Advanced Manufacturing Technology, Vol. 9, p.217-224.

[12] O'Grady P., Wen-Yau Liang, "An internet-based search formalism for design with modules", Computers and Industrial Engineering, 35(1–2), pp. 13–16, 1998.

[13] Pandremenos, J., Paralizas, K., Salonitis, K., Chryssolouris G. (2009) Modularity concepts for the automotive industry: A critycal review. CIRP Journal of Manufacturing Science and Technology. 1 (2009) 148-152.

[14] Retik, A., Warszawski, A. (1994) Automated design of prefabricated building, Building and Environment 29 (4), pp. 421-436