A COMPARATIVE STUDY OF REPRESENTATION AND ENCODINGS FOR BUILDING SHAPE OPTIMIZATION WITH GENETIC ALGORITHMS

Weimin Wang[1], Hugues Rivard[2], and Radu Zmeureanu[3]

ABSTRACT

This paper presents a methodology to optimize the building footprint represented by a multi-sided polygon. Two geometrical representations for a polygon are considered. The first representation uses edge lengths and edge angles to define a polygon while the second representation uses edge lengths and edge bearings. These two representations are discussed with emphasis on their potential problems in binary coding for genetic algorithms: epistasis and encoding isomorphism. Epistasis implies the gene interaction when one gene pair masks or modifies the expression of other gene pairs. Encoding isomorphism means that chromosomes with different binary strings may map to the same solution in the design space. The two alternative representation methods are compared in terms of their impacts on computational effectiveness and efficiency. A problem is formulated to facilitate the comparison, where a pentagon-shaped typical floor of an office building is optimized with respect to life-cycle cost and life-cycle environmental impact. It is found that epistasis has a large impact on the performance of the multi-objective genetic algorithm while encoding isomorphism is not a problem.

KEY WORDS

building design, genetic algorithms, shape representation, optimization.

INTRODUCTION

Shape is one of the most important considerations in the conceptual stage of building design. Since the building shape determines the size and the orientation of the exterior envelope exposed to the outdoor environment, it can affect building performance in many aspects: energy efficiency, cost and aesthetics. Previous studies on building shape optimization were carried out with two approaches: the part-whole approach and the whole-part approach. The part-whole approach constructs a building from its spatial elements such as rooms and zones, as can be found in several architectural studies (Rosenman and Gero 1999; Chouchoulas 2003). This approach is capable of defining a wide range of shapes, some of which may be

[1] Dept. of Construction Engineering, ETS, 1100 Notre-Dame St. West, Montreal, H3C 1K3, Canada, weimin_wa@yahoo.com

[2] Professor, Dept. of Construction Engineering, ETS, 1100 Notre-Dame St. West, Montreal, H3C 1K3, Canada, , hugues Phone 514/396-8667, FAX 514/396-8584.rivard@etsmtl.ca

[3] Professor, Center for Building Studies, Dept. of Building, Civil, and Environmental Engineering, Concordia University, 1455 de Maisonneuve Blvd. West, Montreal, H3G 1M8, Canada, zmeur@bcee.concordia.ca

innovative solutions for a design problem. However, as noted by Caldas (2002), in the context of building design for energy performance, energy simulation programs require information about the adjacencies between spaces, and this information might be difficult to extract from a building shape generated with the part-whole approach. In contrast, the whole-part approach defines a building shape by its external boundaries and represents its internal spatial elements implicitly. Because the whole-part approach can easily describe the building geometry for energy simulation programs, it is adopted in several optimization studies on energy performance (Jedrzejuk and Marks 2002; Wang et al. 2005a). All these studies using the whole-part approach are limited to simple shapes that cannot be easily generalized to more complex ones. Moreover, since these shapes are heavily constrained, some more promising shapes may be precluded from the design space right from the start.

Following the whole-part approach, this paper proposes to use a generalized polygon representation for the optimization of building footprint. Two alternative methods for representing the building footprint and their encoding issues in genetic algorithms (GA) are discussed in the next section. Then, the platform for comparing different representations is established in section 3. The results are discussed at the end of this paper.

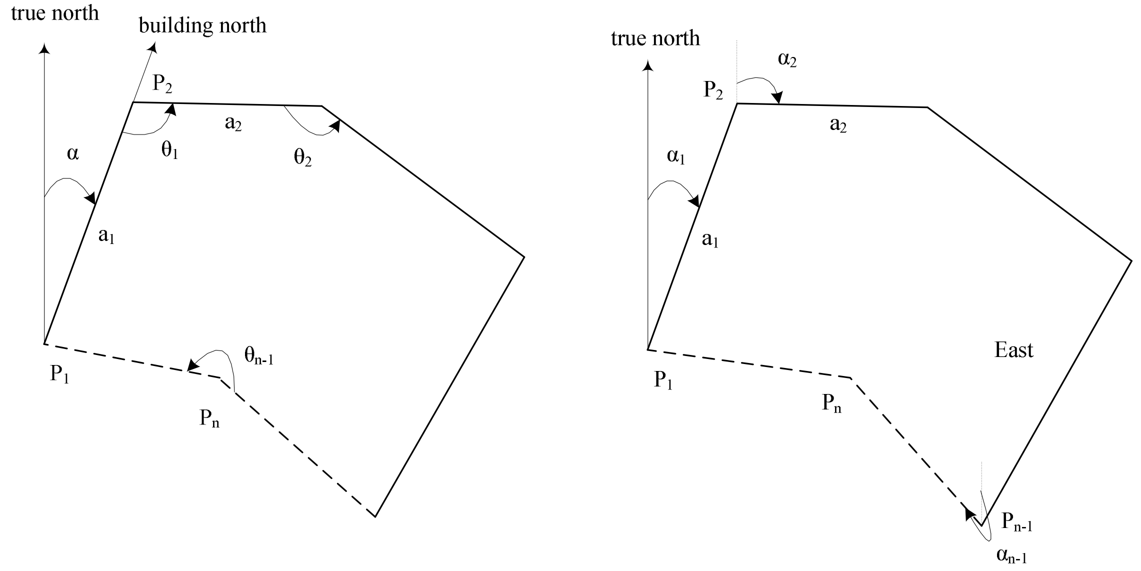
SHAPE REPRESENTATION

In this research, the building footprint is defined as a simple n-sided polygon with no intersection of non-consecutive edges. For a given area, an n-sided polygon can be determined in different ways depending on the representation methods. This paper considers two alternative representation methods: the length-angle method and the length-bearing method.

LENGTH-ANGLE REPRESENTATION

Since a polygon is made up of a number of sequential line segments (i.e., edges), it is intuitive to represent a polygon with the length of each edge and the angle between every two adjacent edges. Thus, given an area *S*, an n-sided polygon can be established with the following procedure, as illustrated in Figure 1 (a).

- From an initial point P_1, the coordinates of the endpoint P_2 of the first edge can be determined from its length a_1 and the building orientation α. The alignment of the first edge is selected here as the reference direction, called building north. This reference direction is necessary in energy simulation programs to describe the geometry of a building. Thus, the building orientation is defined as the angle α between the true north and the building north, clockwise being positive.
- The direction of the *i-th* edge is determined by the angle θ_{i-1}, measured counter clockwise from the previous edge. Note that the angle θ_{i-1} is an interior angle if the vertices are arranged in a clockwise order; otherwise, it is an exterior angle. Once the direction is known, the endpoint P_{i+1} of the *i-th* edge is determined from its length a_i. This step is repeated until the direction of the *(n-1)th* edge is known.
- The last endpoint P_n of the *(n-1)th* edge is determined through calculation to satisfy the fixed area requirement.



(a) Length-angle representation

(b) Length-bearing representation

Figure 1: Alternative representation methods

With the length-angle representation, an n-sided polygon requires a total of $2*n-3$ shape-related variables including the orientation α, the first n-2 edge lengths (a_1, a_2, …, a_{n-2}) and the first n-2 edge angles (θ_1, θ_2,…,θ_{n-2}). Since all lengths and angles except for the last two edges are explicit variables, they can be set within reasonable bounds. This is an advantage of the length-angle representation because genetic algorithms do not generate infeasible solutions due to the bounds violation for explicit variables. The length-angle representation, however, has the following two potential problems related to the binary coding for genetic algorithms (GA): epistasis and encoding isomorphism.

- Epistasis is a term used in the GA literature to describe the "gene interaction" problem, which occurs when one gene masks or modifies the expression of other genes (Davidor 1991). Usually, the higher the level of epistasis, the harder for a GA to locate the optimum. In building design, the physical expression of each facade involves its area and orientation, which depend on the length and direction of the corresponding edge. Edge lengths are modeled as independent and explicit variables, so they do not cause epistasis. On the other hand, the direction of each edge is an implicit variable derived from the building orientation and all the edge angles which have lower subscripts than the considered edge. For example, for a decagon, the direction of the fifth edge depends on the building orientation and the angles from θ_1 to θ_4. Hence, the change of an angle θ will modify the direction of all edges after it. This interaction between edge directions may cause serious epistatic problems.
- Encoding isomorphism means that chromosomes with different binary strings may map to the same solution in the design space. This leads to representational redundancy, which is not beneficial for the GA if the genetic operators cannot gain

useful information from representational variants (Ronald et al. 1995). With the length-angle representation, encoding isomorphism exists because of the possibility of using different vertices as the starting point.

The problem of epistasis can be addressed by representing the direction of each edge as an explicit variable. This approach is adopted in the second representation.

LENGTH-BEARING REPRESENTATION

The length-bearing representation has two major differences compared with the previous method. First, bearing, instead of edge angle, is used to determine the direction of each edge. The bearing is the angle between a designated north direction (e.g., the true north indicated by a compass) and an edge, clockwise being positive. Second, the bearing of the first edge replaces the building orientation in the length-angle method. After these two changes, the direction of each edge becomes an independent variable, thereby reducing gene interactions. The general procedure to establish an n-sided polygon with the length-bearing method requires the following steps (Figure 1 b):

- Starting from a point P_1, the coordinates of the endpoint P_2 of the first edge can be determined from its length a_1 and bearing α_1.
- The endpoint P_{i+1} of the *i-th* edge can be determined based on its staring point P_i, its length a_i and bearing α_i. This step is repeated until the point P_{n-1} is defined.
- Given the bearing of the *(n-1)th* edge, the position of the last point P_n can be calculated to satisfy the fixed area requirement.

With the length-bearing representation, an n-sided polygon also requires a total of $2 * \mathrm{n} - 3$ shape-related variables including α_1, a_1, α_2, a_2,…, α_{n-2}, a_{n-2}, and α_{n-1}.

Although the degree of gene interactions has been reduced significantly with the length-bearing representation method, a minor epistatic problem still exists because there are two possible orientations for a given edge direction. The applicable orientation depends on how the other edges close the polygon. In addition, the length-bearing representation still has the problem of encoding isomorphism. This can be addressed by remapping vertices.

VERTICES REMAPPING

The purpose of the remapping operation is to ensure that there is a one-to-one relationship between the genotype and the phenotype. This objective can be achieved by removing the ambiguousness in arranging vertices. In this study, we ensure the unique order of vertices with the following two rules: (1) the lower-left vertex is the first point; and (2) the vertices are arranged in a clockwise order. The remapping operation is applied whenever a solution in the genotype violates either one of the above two rules. In Figure 2, for example, the solution corresponding to the left pentagon violates both rules. After remapping, the solution has changed completely in the genotype but it stands for the same pentagon in the phenotype.

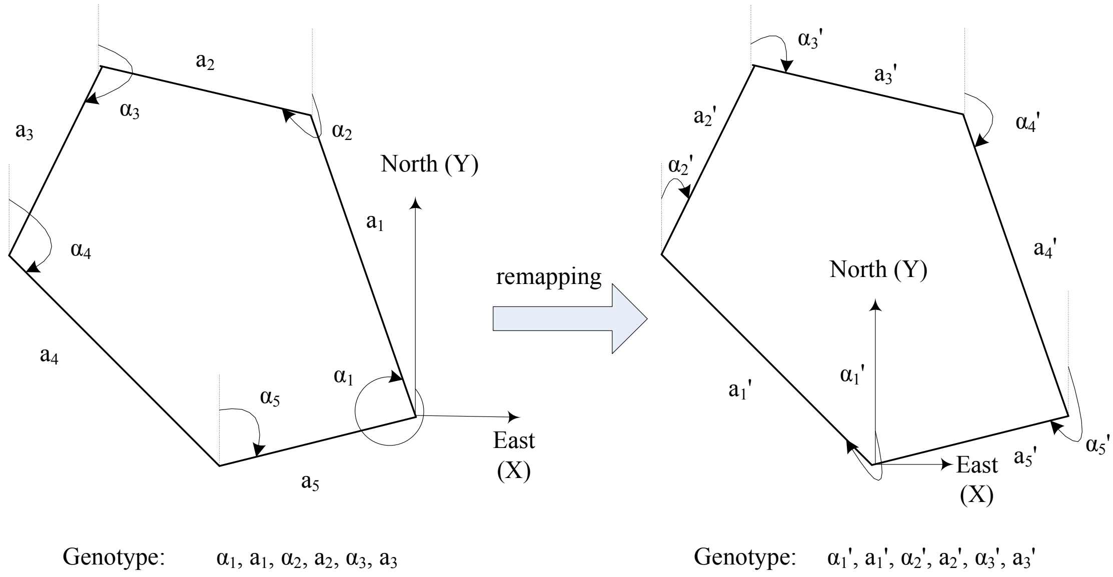


Figure 2: The remapping operation

COMPUTING EXPERIMENT SETUP

An optimization problem is formulated to facilitate the comparative study. This problem consists in the design of a multi-story office building located in Montreal, Canada. The building footprint takes the shape of a pentagon. Since this study focuses on optimizing the building shape in plan, only one typical floor with an area of 1000 m^2 is considered.

The following two objective functions must be minimized in this research: the life-cycle cost (LCC, in $) and the life-cycle environmental impact (LCEI, in MJ). The LCC consists of the initial construction cost of considered building components and the present worth of operating costs due to energy consumption. The LCEI consists of the environmental impacts due to building construction and building operation for heating, cooling and lighting. A simulation program (Wang et al. 2005b) has been developed to calculate the two objective function values.

The specific variables for the length-angle representation are orientation α and angles θ_1 to θ_3 while for the length-bearing representation they are bearings α_1 to α_4. Except for the above difference, all other variables are the same for the two alternative representations. For the pentagon floor, the interval of each edge length is set between 5 and 200 m and the interval of each edge angle is set between 15 and 345 degrees. In addition to the shape-related variables, the optimization model considers several other envelope-related design variables such as window types, window ratios, insulations and overhangs. A detailed description about these envelope-related variables can be found in (Wang et al. 2005a).

The multi-objective genetic algorithm presented in (Wang et al. 2005b) is employed here to solve the formulated optimization problems. A major advantage of multi-objective GA lies in its ability to locate multiple Pareto optimal solutions in a single run. A solution is said to be Pareto optimal if and only if it is not dominated by any other solution in the performance space. If solution $\mathbf{X_1}$ dominates another solution $\mathbf{X_2}$, it implies that $\mathbf{X_1}$ is non-inferior to $\mathbf{X_2}$

for all the considered performance criteria but it is better than $\mathbf{X_2}$ for at least one criterion. All Pareto solutions form a Pareto front in the performance space.

Since the main purpose of this paper is to study the impact of the two shape representations on the optimization results, the program is run for the following three scenarios: I) the length-angle representation method; II) the length-bearing representation method without remapping; and III) the length-bearing method with remapping. The comparison of scenarios I and II will help determine the impact of epistasis, while the comparison of scenarios II and III will help determine the impact of encoding isomorphism. In addition, due to the randomness of GA, the program is run three times for each of the three scenarios. Thus, nine program runs are needed and they are listed in Table 1. In order to lessen the influence of different initial populations on the comparison study, the same populations are used for each experimental set of scenarios II and III. This measure is not taken for the first scenario because it has different variables.

Table 1 Overview of computing experiments

Scenarios	**Experimental sets**		
	Set 1	**Set 2**	**Set 3**
I	Run I-1	Run I-2	Run I-3
II	Run II-1	Run II-2	Run II-3
III	Run III-1	Run III-2	Run III-3

Computational effectiveness and efficiency are two aspects that can be used to compare the performance of the multi-objective GA for the three different scenarios. Computational effectiveness evaluates the ability of the algorithm to find high-quality solutions. Computational efficiency evaluates the computational efforts required to solve a problem.

The computational effectiveness of multi-objective GA is measured with two simple and informative metrics: generational distance and spread (Deb 2001). The first performance metric evaluates the closeness between the obtained Pareto set and a known set of true Pareto-optimal solutions. The second metric evaluates the diversity and extension of obtained non-dominated solutions. A smaller value indicates better performance for both metrics. Both of these two metrics rely on a known set of Pareto-optimal solutions, which act as a reference in the comparison. Since the actual global Pareto solutions for the optimization problem is not known, all non-dominated solutions from the external populations in the total nine program runs are regarded to form the global Pareto optimal set. Let Q and P* denote a Pareto set and the global Pareto set, respectively, the generational distance (GD) can be calculated as (Deb 2001):

$$\mathrm{GD} = \frac{\sqrt{\sum_{i=1}^{N} d_{ij}^{2}}}{N} \qquad (1)$$

where, *N* is the total number of solutions in the Pareto set Q; d_{ij} is the distance between the *i-th* solution in the Pareto set Q and the *j-th* solution in P* , which is the closest to the *i-th* solution.

The spread can be calculated with the following formula Deb (2001):

$$\text{spread} = \frac{\sum_{m=1}^{2} d_m^e + \sum_{i=1}^{N-1} | d_{ik} - \bar{d} |}{\sum_{m=1}^{2} d_m^e + (N-1) \cdot \bar{d}} \qquad (2)$$

where, d_{ik} is the distance between the *i-th* solution and its neighboring solution *k* in the Pareto set Q; $\bar{d}$ is the average of d_{ik}; d_m^e represents the distance between the extreme solutions of P* and Q corresponding to the *m-th* objective function. Note that both d_{ij} and d_{ik} are the Euclidean distance normalized by the maximum and minimum objective function values in the global Pareto set P*. In addition, the neighbouring distance d_{ik} is computed sequentially along the sorted Pareto solutions by the objective function values; therefore, the d_{ik} does not necessarily mean that the *k-th* solution is closest to the *i-th* solution.

The computational efficiency is commonly measured with the computation time. In addition, the number of simulation calls can be employed as an equivalent measure for computational efficiency because simulation usually dominates the computation time for simulation-based optimization problems.

RESULTS AND DISCUSSION

Table 2 shows the four performance metrics used to compare the nine different program runs. All runs were done on a computer with Windows XP system (3.40 GHz Pentium-IV processor, 1 GB RAM). The following observations were made:

Table 2 Comparison of performance metrics for different program runs

scenario	program run	GD	spread	CPU time (hours)	simulation calls
I	I-1	0.029	0.70	67.7	10300
	I-2	0.028	0.84	68.2	10290
	I-3	0.040	0.71	67.4	10160
	average	0.033	0.75	67.8	10250
II	II-1	0.011	0.58	58.4	8880
	II-2	0.011	0.59	59.7	9110
	II-3	0.002	0.70	58.7	8950
	average	0.008	0.62	58.9	8980
III	III-1	0.007	0.64	59.0	8970
	III-2	0.010	0.74	58.1	8890
	III-3	0.011	0.67	60.8	9130
	average	0.010	0.68	59.3	9000

- In terms of the generational distance (GD), its results for the program runs in each experimental set (see Table 2) are ordered as follows: $GD_{I-1} > GD_{II-1} > GD_{III-1}$, $GD_{I-2} > GD_{II-2} > GD_{III-2}$, and $GD_{I-3} > GD_{III-3} > GD_{II-3}$. Since the smaller the value of GD the better the convergence to the global Pareto front, the above performance order indicates that of all three scenarios, scenario I using the length-angle representation performs worst for all three experimental sets. Scenario III performs best for two out of three experimental sets; however, scenario II has the lowest average value of GD. The GD difference between the first scenario and the other two scenarios is significant, which demonstrates that the high level of epistasis has posed an obstacle to the convergence of the genetic algorithm. In contrast, the difference is small for the last two scenarios, which indicates that the encoding isomorphism has little impact on the GA convergence for this shape optimization problem.

- In terms of the spread, a smaller value indicates a better distribution of solutions along the Pareto front. The average value is employed here for comparison in order to eliminate the impact of different ways in grouping program runs. In average, scenario II performs best while scenario I performs worst. As shown in Equation 2, the value of spread is subject to two factors: the spacing and the extent of Pareto solutions, evaluated respectively by $\left| d_i - \bar{d} \right|$ and d_m^e in that equation. Different scenarios do not affect the spacing because the same niche sharing strategy is used in the multi-objective GA to achieve a good distribution. Thus, the difference of spread for different scenarios is mainly due to the extent of Pareto solutions, which is calculated as the distance between the corresponding extreme solutions of two solution sets. Because scenario I had the worst convergence, its extreme solutions are far from the global Pareto front, consequently leading to a large value of d_m^e and spread. In average, Scenario II is slightly better than III, thus probably indicating that the former is better in locating the extreme solutions of the Pareto front.

- In terms of computational efficiency, scenario I takes the most CPU time and has the most number of simulation calls during the total 300 generations, while scenarios II and III have similar CPU time and simulation calls. The average CPU time per simulation call is about 24 seconds for all three scenarios, which demonstrates that simulation dominates the computation time and that the time on extra operations such as remapping is negligible. The reason why scenarios II and III have less simulation calls than the first scenario is due to the fact that the length-bearing representation produces more infeasible solutions than the length-angle representation, as explained in Section 2. We investigated the output file containing detailed information of individuals for every twenty generations. The investigation showed that in average, scenarios II and III produced about four infeasible solutions per generation, accounting for 10% of the population size, while scenario I produced only several infeasible solutions throughout the whole evolution. Infeasible solutions do not generate call to the simulation program, so the number of simulation calls and the computation time are less for the last two scenarios.

The performance metrics GD and spread in Table 2 are calculated for the final Pareto solutions obtained at the end of the evolution process. If GD and spread are calculated during the evolution process to show their dynamic changes, interesting findings can be found. Hence, for each scenario, GD and spread are calculated for the external population at every 50 generations, using the same Pareto set reference as before. The results from three experimental sets are averaged and shown in Figure 3. This Figure demonstrates the following two points:

- The value of GD becomes smaller during the evolution process and finally converges towards a minimum value (Figure 3 left). Scenario I consistently takes the largest GD value among the three scenarios during the evolution process. Scenarios II and III have similar GD values after the 100th generation. Moreover, they achieved smaller GD values at the end of the 100th generation than what the first scenario achieved at the end of the evolution process. All these observations indicate that the length-bearing representation has a better convergence. For each scenario, because the GA is good at finding optimal regions but weak at local refinement, it converges quickly towards the global Pareto front in the first 150 generations, while the convergence rate becomes slower in the second 150 generations.
- In contrast with GD, the spread does not monotonically decrease during the evolution process (Figure 3 right). None of the three scenarios has the smallest or largest value of spread for all considered generations. For each scenario, the general trend is to decrease the spread value, but spread may increase for some generations to concede to the requirement of convergence. This is reasonable because non-dominated solutions are usually given the highest fitness for the multi-objective GA. Thus, spread would possibly increase whenever a non-dominated solution in the new population dominates some well-spaced ones in the previous population.

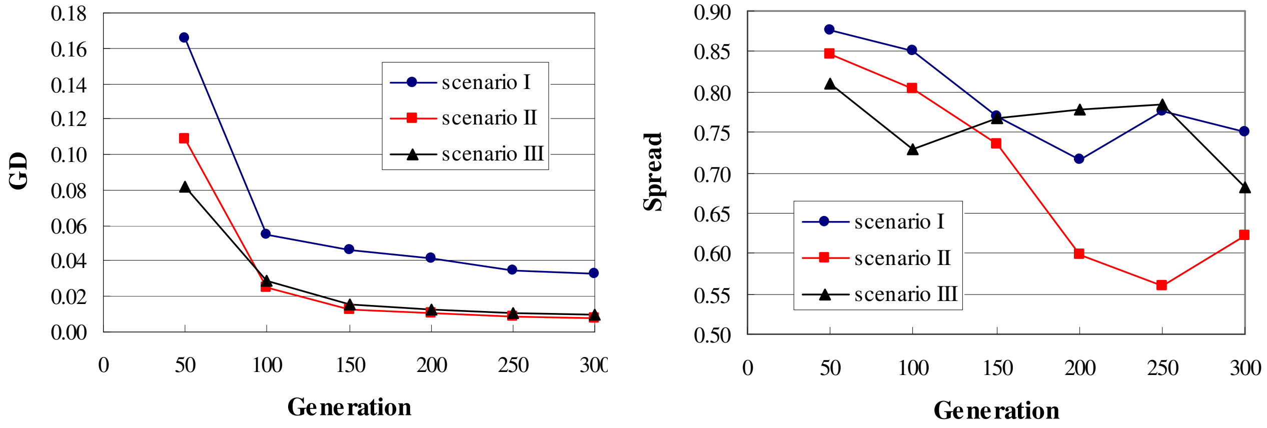


Figure 3: Evolutionary changes of GD and spread for the three scenarios

CONCLUSIONS

In shape optimization, the geometrical representation is a fundamental issue to be considered. The representation method used can have significant impacts on the ease of implementation and the performance of optimization. This comparative study leads to the following conclusions:

- The length-bearing method performs better than the length-angle method with respect to the performance of the multi-objective GA, evaluated based on its convergence to the global Pareto front and the spread of Pareto solutions. This indicates that the representation method with a high-level epistasis should be avoided because of its impact on convergence.

- The remapping of polygon vertices to arrange them in a clockwise order starting from the lower-left one has no noticeable impacts on the performance of GA. This indicates that encoding isomorphism is not a problem for shape optimization with GA.

REFERENCES

Caldas, L. (2002). Evolving three-dimensional architecture form: An application to low-energy design. In: Gero JS, (editor). *Artificial Intelligence in Design'02*, Dordrecht, The Netherlands: Kluwer Publishers, pp. 351-370.

Chouchoulas, O. (2003). *Shape evolution: An algorithmic method for conceptual architectural design combining shape grammars and genetic algorithms*. Ph.D. Thesis, Department of Architecture and Civil Engineering, University of Bath, UK.

Davidor, Y. (1991). Epistasis variance: A viewpoint on GA hardness. In: Rawlins G, (editor). *Foundations of Genetic Algorithms*, San Mateo: Morgan Kaufmann, pp. 23-35.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: John Wiley & Sons.

Jedrzejuk, H. and Marks, W (2002). Optimization of shape and functional structure of buildings as well as heat source utilization: Partial problems solution. *Building and Environment* 37 (11) 1037-1043.

Ronald, R., Asenstorfer, J., and Vincent, M. (1995). Representational redundancy in evolutionary algorithms. In: Fogel D, (editor). *Proceedings of the 1995 IEEE International Conference on Evolutionary Computation*, New York: IEEE press, pp. 631-37.

Rosenman, MA. and Gero, JS. (1999). Evolving designs by generating useful complex gene structures. In: Bentley PJ, (editor). *Evolutionary Design by Computers*, San Francisco: Morgan Kaufmann, pp. 345-64.

Wang, W., Rivard, H., and Zmeureanu, R. (2005a). An object-oriented framework for simulation-based green building design optimization with genetic algorithms. *Advanced Engineering Informatics*,19 (1) 5-23.

Wang, W., Zmeureanu, R., and Rivard, H. (2005b) Applying multi-objective genetic algorithms in green building design optimization. *Building and Environment*, 40 (11): 1512-1525.