DIVERSITY AND EFFICIENCY

A Hybrid Evolutionary Algorithm Combining an Island Model with a Steady-state Replacement Strategy

LIKAI WANG[1], PATRICK JANSSEN[2] and GUOHUA JI[3]
[1,3]*School of Architecture and Urban Planning, Nanjing University*
[1]*dg1436002@smail.nju.edu.cn* [3]*jgh@nju.edu.cn*
[2]*School of Design and Environment, National University of Singapore*
[2]*patrick@janssen.name*

Abstract. Standard evolutionary algorithms have limited use in practical architectural design tasks. This may be due to the poor search efficiency and the lack of diversity of the result. In order to overcome these weaknesses, this paper proposes a hybrid evolutionary algorithm combining an island model approach (parallel distributed technique) and a steady-state replacement strategy for maintaining a rich design diversity of the result while speeding up the search process. Through a demonstration, it is shown that the hybrid algorithm can effectively improve both design diversity and search efficiency.

Keywords. Hybrid evolutionary algorithms; island models; steady-state replacements; design diversity; search efficiency.

1. Introduction

Architectural design, according to Liu et al (2003), can be characterised as an iterative divergent-convergent process, where designers create a wide range of design alternatives at the divergent design exploration stage and then evaluate and modify these at the convergent exploitation stage. Compared to other types of algorithms, evolutionary algorithms are considered a potential technique to support such divergent-convergent processes with computers (Turrin et al, 2011). The reasons for this are twofold. On the one hand, for divergent design exploration, evolutionary algorithms take advantage of the population-based strategy and the stochastic search approach making such algorithms an explorer of the design space, which may discover unexpected design alternatives. On the other hand, for convergent design exploitation, the application of the tournament selection and the population replacement ensures the evolutionary search process focus on optimal solutions in the design space. However, evolutionary algorithms are still less widespread in real-world architectural design tasks as one might expect. The research identifies two key challenges that degrade the utility of evolutionary algorithms in architectural design practice. The first challenge relates to poor design diversity while the second one relates to poor search efficiency.

Intelligent & Informed, Proceedings of the 24th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2019, Volume 2, 593-602.

With regards to design diversity, the converging nature of the evolutionary process eventually makes all individuals become more and more similar (Figure 1 left). In other words, the population loses design diversity soon after the process discovers a subspace with local-optima and starts fitness exploitation. The convergent search process and the lose of design diversity undermine the divergent design exploration process. In this regards, instead of searching only for optimum design solutions, it is more relevant for designers to execute an evolutionary process that can identify a wide variety of design alternatives with acceptable performance. By comparing these design alternatives and identifying patterns and trends, designers can learn more about the design problem and come up with creative design solutions (Woodbury & Burrow, 2006).

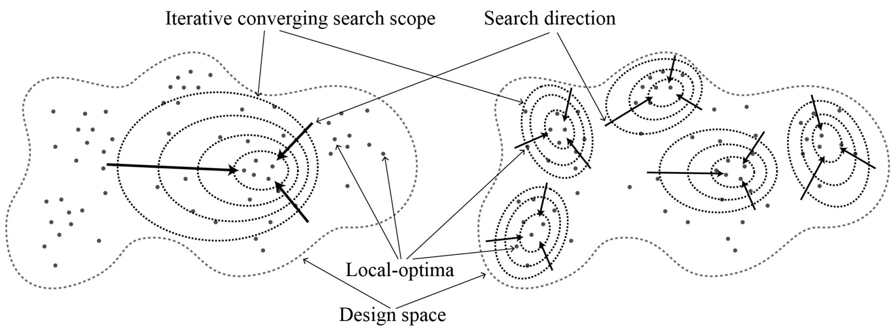


Figure 1. Diagrams of a standard (centralised population) search process (left) and a distributed parallel search process (right).

With regards to search efficiency, the generational replacement strategy applied by most evolutionary algorithms results in slow search processes because the design population can evolve only after all individuals in the current generation have been evaluated. Such a process does not proceed as the natural evolutionary process does, where elite individuals affect the population immediately and asynchronously once they come into being. For many studies or applications with a fast evaluation process, poor search efficiency may not draw much attention. However, most architectural design optimizations require evaluation processes based on detail and time-consuming performance simulations, which can prolong the overall search process. When constrained by tight design schedules, poor overall search efficiency will significantly degrade the utility of evolutionary algorithms for real-world architectural design tasks.

The two abovementioned challenges can be controlled, to some extent, by adjusting the selection pressure in order to achieve the desired intensity of design exploration or exploitation. Even so, these two challenges still act against one another. The pursuit of design exploration inevitably hinders search efficiency and vice versa. Thus, in order to reconcile the contradiction between these two challenges, the research proposes a new hybrid evolutionary algorithm integrating an island model approach (Alba & Troya, 1999) with a state-steady replacement strategy (Rasheed, 1998). The development of the algorithm is aimed at accommodating the requirements of design diversity and search efficiency concurrently. Compared with standard evolutionary algorithms, the proposed hybrid algorithm meets the need for improving both design diversity and search efficiency through a series of parallel search processes focusing on different subspaces, while each of the search processes exploits the subspace more greedily (Figure 1 right).

2. Hybrid Evolutionary Algorithm

For the proposed hybrid evolutionary algorithm, the application of an island model approach and a state-steady replacement strategy will help maintain design diversity and speed up the search process. However, the combination of these two techniques is non-trivial, as overcoming both poor design diversity and poor search efficiency at the same time is challenging. First and foremost, the design space defined by the parametric model of architectural design is commonly multimodal, which means that there are many feasible design subspaces sparse within the design space (Wang et al, 2018). Multimodal design spaces are, in fact, in favour of designers' divergent design exploration, as design exploration does not strictly require the search process focus on finding the global optimum but can instead perform multiple parallel searches. With such parallel search processes, the evolutionary process can provide multiple design solutions, helping designers to uncover patterns or trade-offs within the design problem. In addition, focusing too much on speeding up the evolutionary process by emphasising design exploitation can result in premature convergence. Such risks can be offset if design diversity is maintained by parallel search processes.

With regards to these two facets, the combination of the island model and steady-state replacement strategy allows the hybrid algorithm to search the design space both exploratively and exploitatively: the island model approach enables the search process to focus on several feasible subspaces rather than one single subspace simultaneously, while the steady-state replacement strategy speeds up the search process in exploiting each of the design subspaces (Figure 1 right). In the remainder of this section, we describe these two techniques and the implementation of the proposed hybrid algorithm.

2.1. ISLAND MODEL

For maintaining design diversity, it is important to preserve the *accessibility* of different promising subspaces during the evolutionary search process. Accessibility implies that the search process can reach all locations in the design space with reasonable effort (Woodbury & Burrow, 2006), through recombining (crossover) or random shuffling (mutation) certain parameters (genes) of the already discovered individuals. From this point of view, a weakness of the standard evolutionary algorithm is that many subspaces with desirable design solutions are often lost when a local-optima with slightly better fitness is found. Such local-optima quickly start to overwhelm the rest of the population, and as a result, individuals carrying information that could guide the search process to locate other promising subspace are typically killed and replaced. For preserving the accessibility of different subspaces, the island model can be used to reduce competition between individuals from different promising subspaces.

The island model uses a parallel distributed search approach that can overcome certain disadvantages inherent in the centralised population models, including prematurity and poor robustness (Alba & Troya, 1999). Such centralised models treat the entire population as a single breeding unit and execute evolutionary operations (crossover and mutation) on the whole population. In contrast, the island model splits the population into several 'niche' subpopulations that are then evolved relatively independently. This separation of the search process into subpopulations avoids the entrapment of local-optima at the outset of the evolutionary process.

While the initial incentive for developing the island model approach is mainly aimed at preventing prematurity, it also introduces other benefits. One such benefit is the ability to launch parallel searches from multiple points in the design space. This makes the island model a compelling approach for maintaining design diversity during design exploration. With the island model approach, the sub-optimal individuals found by each subpopulation are able to be preserved for a longer time and will not be immediately replaced when better individuals are found in other subpopulations. Thus, the accessibility of different subspaces can be preserved, which allows the evolutionary process to exploit multiple sub-optima inside each subspace.

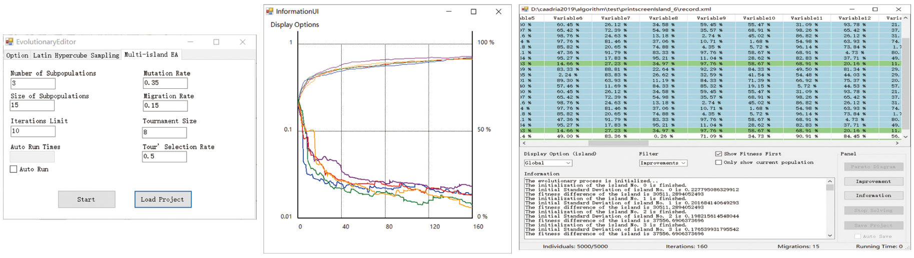
2.2. STEADY-STATE REPLACEMENT STRATEGY

In order to improve search efficiency, the steady-state replacement strategy may have certain advantages (Janssen, 2015). Compared with the generational replacement strategy, the steady-state replacement strategy evaluates only a small number of individuals and immediately replaces inferior individuals in the population. Thus, the evolutionary process is more responsive to the discovery of new improving individuals, as it can react to feedback from performance evaluation more promptly.

In the proposed hybrid algorithm, a steady-state replacement strategy is used. All other evolutionary operations are still run sequentially, as in standard evolutionary algorithms. In each iteration, a fixed number of individuals in every subpopulation are randomly selected as parents to create offspring. The parents and offspring then all compete with one another. The higher ranking individuals (either parents or offsprings) are then inserted back into the subpopulation while the rest are discarded. The selective nature of the steady-state replacement strategy renders the evolutionary process exploitative. However, as mentioned above, the risk of over-exploitation or premature convergence can be compensated by the explorative nature of the island model.

2.3. IMPLEMENTATION

In order to facilitate ease of use for designers, the hybrid algorithm is implemented in the Rhino-Grasshopper environment as a built-in component. The component has several graphical user interfaces (GUI) for providing information to assist designers to understand the search progress. The component provides visualizations at both the global population level or at the subpopulation level for enabling users to keep track of the evolutionary progress. These visualizations show progress trend lines and other basic information (Figure 2).



Setup Menu Information GUI Runtime Monitor GUI

Figure 2. GUIs of the component.

In addition to the island model approach and steady-state replacement strategy, there are two other important techniques implemented in the algorithm worth mentioning. The first technique focuses on the generation of the initial population. It is reported that the distribution of the initial population has a significant impact on the quality of the subsequent evolutionary process. An uneven initial population distribution can result in early entrapment of the evolutionary search process by poor local-optima (Maaranen et al, 2007). In this regards, we used Latin Hypercube Sampling to generate an initial population to ensure the population can spread more evenly in the design space.

The second technique relates to the exchange of individuals between subpopulations. Natural evolutionary processes often benefit from mating across the boundaries of the subgroups of different species (Chipperfield et al, 1994). Likewise, the exchange of genetic material among subpopulations is also encouraged when using the island model, and the exchange process is referred to as *migration*. The use of migration can help a subpopulation entrapped in a poor fitness subspace to escape to other subspaces by introducing genetic material from other subpopulations. The mixture of genetic material from two subpopulations allows the search process to reposition itself somewhere between the two associated subspaces.

For most algorithms based on an island model, the migration is triggered synchronously after a fixed number of iterations. When triggered, all subpopulations will simultaneously send and receive individuals with other subpopulations. However, such migration operations can be harmful if well-evolved individuals with better fitness are sent to a subpopulation still under evolution. In order to prevent this weakness, the proposed hybrid algorithm executes the migration in an asynchronous fashion. The approach is based on previous studies by Horii et al. (2002). With this approach, the exchange process will only be triggered when the evolution of a subpopulation falls into stagnation.

3. Demonstration

For demonstrating the utility of the hybrid algorithm, the paper uses a design problem from a previous study (Wang et al, 2018). The design generates high-rise buildings with an atrium and a series of vertical gardens. The objective is to search for design alternatives that optimise economic performance, taking into account various factors including rental profit, and facade and structural cost. With the economic performance evaluation, a simplified calculation is used that has the advantage of being fast to evaluate. Otherwise, running relevant simulations would be too time-consuming. We will not go into other implementation details of the design here, as they can be found in the above-mentioned study.

With regards to search efficiency, the design space defined by the parametric model is huge and multimodal. Searching such a design space, the evolutionary process can be easily misled by a local-optima, resulting in premature stagnation. In this regards, section 3.1 compares the proposed hybrid algorithm in terms of search efficiency with two other search techniques: a random search technique and a simple genetic algorithm (SGA).

With regards to the design diversity, section 3.2 presents the evolutionary result from the proposed hybrid algorithm. The result presents a rich diversity of design variation, giving designers a better understanding of the trade-off between economic performance and geometric features.

3.1. SEARCH EFFICIENCY

The three search techniques were all executed in Rhino-Grasshopper platform. In addition to the hybrid algorithm, the random search was executed by setting the parameter list to always be 100% random, while SGA was executed using the Grasshopper Galapagos genetic algorithm component (Rutten, 2013). The setup of each search technique is illustrated in Table 1. In order to reduce the impact of stochastic variation, the search process for each technique was repeated five times. During each search process, the best two solutions found at each point in time were recorded. The reason for recording the best two is that focusing only on the best solution can conceal the overall progress of the whole population. By recording the best two solutions, the improvement of the population is shown more clearly.

Table 1. The parameter setup for the three search techniques .

	Simple Genetic Algorithm	Random Search	Hybrid Algorithm
Mutation Rate	25% (*Inbreeding*)	-	35%
Selection Rate	25% (*Maintain*)	-	50%
Number of Subpopulation	-	-	5
Size of Population	100 (*Initial boost* is 2)	-	200
Size of Subpopulations	-	-	40
Stop Criteria	20 generation stagnation	5000 birds	5000 birds

For SGA, the number of stagnant generation was set to 20. In comparison, since there are no mechanisms for the hybrid algorithm and the random search to stop the search process, 5000 births were set as the limit of the search. The setting of 20 generations or 5000 births considered both the factors of search completeness and of run-time limits.

Figure 3 shows the fitness progression trend lines of the search processes for the three alternative search techniques. In term of the improvement of the fitness, SGA is inferior to the other two other techniques. The result suggests that SGA is incapable of solving design problems with such large multimodal spaces. This is most likely due to the fact that the exploitative nature of SGA can easily mislead the search process towards current sub-optimal subspaces. In such cases, the search process can often spend a long time exploiting a subspace that does not include any acceptable designs. This is the case for the bottom trend line in Figure 3, that produced 12 thousand births but only made very limited progress.

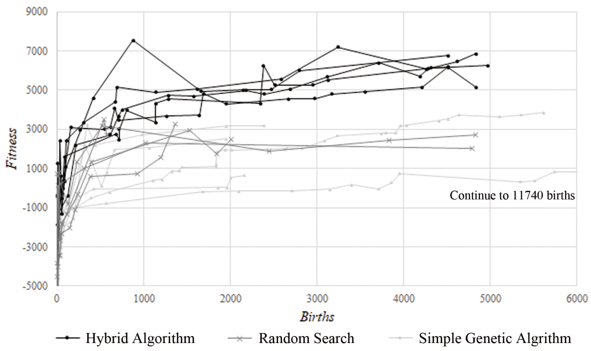


Figure 3. Fitness progress trend lines of the evolutionary runs.

Compared with SGA, the random search is little more efficient in solving the assigned design problem. Without exploitation, the search process randomly explores the design space in an unbiased way. As a result, overall search progress is generally better than SGA. However, without any search heuristic, the improvement achieved by the search process is limited.

In comparison, the proposed hybrid algorithm outperforms the other two search techniques in terms of search efficiency. The evolutionary process with the hybrid algorithm is able to identify promising subspaces faster than the other two approaches. Partly this is due to the use of Latin Hypercube Sampling, which ensures that the initial population evenly covers the design space. This gives the evolutionary process access to a wide range of design subspaces at the outset of the search. In addition, the steady-state replacement strategy enables the search process to discard inferior design variants faster. The evolutionary heuristics of the hybrid algorithm also prevent the search process from just wandering around the design space as the random search does.

In addition to the initial fast discovery of the feasible design alternatives, over the long term the hybrid algorithm is also able to find better designs than the other two techniques. This ability to deliver steady progress is due to the use of the island model. At the early search stage, the island model approach prevents the evolutionary process from being entrapped by a single local-optima. But this also divides the computational search effort into multiple parallel search processes.

However, as the search progresses, the parallel search process allows the hybrid algorithm to exploit each of the subspaces more efficiently. As a result, it is more likely to discover designs with better fitness among these subpopulations. Lastly, the migration of individuals among subpopulations can allow the search process to escape from the entrapment of poor local-optima. The island model approach contrasts with the centralised population model used by SGA, which must pay the price of putting “all the eggs in one basket”.

3.2. DESIGN DIVERSITY

As a sign of search convergence, the genetic distribution displayed by parallel coordinates (Wortmann, 2018) can also be used as an indicator of the similarity among the individuals in the design population. With SGA, the genetic distribution will typically become reduced with the search progress (Figure 4), which means that most individuals in the final population will come from one particular subspace. Figure 4 and Figure 5 respectively show the genetic distribution of one of the conducted evolutionary runs based on SGA and the hybrid algorithm. The comparison shows that the hybrid algorithm can preserve richer diverse genotypes than SGA. Even so, the genotypes preserved with the use of the hybrid algorithm all have approximately the same fitness (the right-hand two columns of Figure 5 indicates the fitness of the genotype), which means that there is no significant trade-off on obtaining richer design diversity.

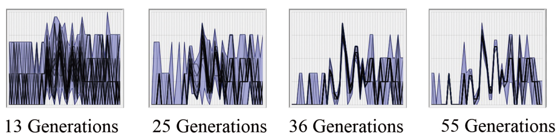


Figure 4. An example genetic distribution change process with Galapagos.

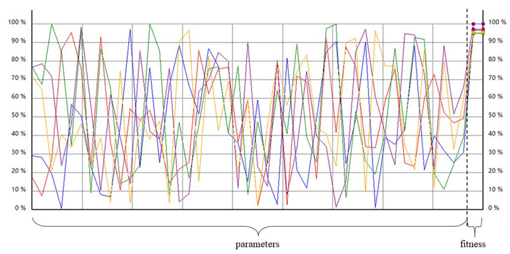
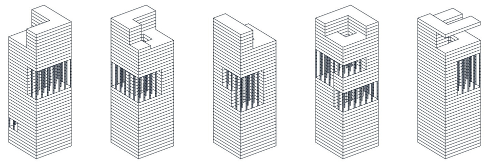


Figure 5. Genetic distribution of elite individuals from different subpopulations with the hybrid algorithm.

The diverse genetic distribution can help designers grasp the implication of design performance. It is clear from Figure 5 that these individuals share similar values for certain parameters while differing in others. The pattern of genetic distribution indicates that the parameters with similar values are significant for creating satisficing solutions. By identifying these significant parameters, designers are able to modify the design in consideration of other objectives but avoid steering the modification into infeasible design solutions. The pattern of the genetic distribution also indicates that the found designs should share a certain geometric similarity, which can reveal certain relationships between design performance and geometric features.

Figure 6 shows a set of example design alternatives evolved using the hybrid algorithm. Each of the design alternatives comes from a different sub-population. On first sight, all design alternatives have a similar geometric feature, namely that vertical gardens are inserted on middle-to-upper floors. This result is aligned with the conclusion of the previous study that reveals how the insertion of vertical gardens can cause a smaller loss in economic performance. There are also noticeable geometric differences among these design alternatives. By comparing the geometric features and the fitness value, the designer can find the trade-off relationship of the fitness with geometrical features. Among the presented designs, the design alternatives with complex interlocking vertical gardens are typically poor in economic performance because the increase in the number and the size of vertical gardens raises the facade cost while reducing the rental profit.



Fitness 4417.41 7189.30 5564.76 3349.61 5137.96

Figure 6. Optimal designs in different subpopulations.

4. Discussion

The conducted experiment compares the search efficiency of the hybrid algorithm with a random search and SGA while also investigating design diversity. The results of the experiment reveal that, by combining the island model and the steady-state replacement strategy, the hybrid algorithm can simultaneously overcome the challenges of poor search efficiency and poor design diversity. This is achieved by integrating these two techniques in a way that synergies their individual strengths. The analysis in Section 3.1 makes it clear that even if the SGA search was repeated numerous time, it is highly unlikely that it would ever reach the fitness levels achieved by the hybrid algorithm.

Compared with the other two search techniques, the use of the hybrid algorithm can also make the evolutionary design process more robust. In Figure 3, the pattern of the fitness trend lines demonstrates that the evolutionary processes based on the hybrid algorithm are more stable than the other search techniques in terms of the final fitness value and overall search progress. This will allow designers to develop greater trust in the evolutionary result. In contrast, the trend lines for SGA are unstable, rendering it untrustworthy.

The experiment also highlights the weakness of SGA in searching multimodal design spaces. The poor design diversity and search efficiency of SGA make it unsuitable for use within the divergent-convergent design process. This also confirms what Wortmann (2018) has identified - the problematic integration of SGA within architectural design. In this regards, the research provides a potential solution for applying evolutionary search in architectural design. The proposed hybrid algorithm can allow the results from evolutionary search to act as catalysts in the divergent-convergent architectural design process.

The research has successfully demonstrated the proposed hybrid algorithm for a specific design problem. However, according to the No-free-lunch theorem (Wolpert & Macready, 1997), it is not possible for an algorithm to solve all problems effectively and efficiently. Hence, further systematic examinations are needed to identify the range of problems that the proposed hybrid algorithm is able to solve, which also points out the future research direction.

Acknowledgements

This paper was supported by the National Natural Science Foundation of China (51378248) and the China Scholarship Council (201706190203).

References

Alba, E. and Troya, J.M.: 1999, A survey of parallel distributed genetic algorithms, *Complexity*, **4**(4), 31-52.

Chipperfield, A.J. and Fleming, P.: 1994, “Parallel genetic algorithms: A survey” . Available from no URL given.

Janssen, P.: 2015, Dexen: A scalable and extensible platform for experimenting with population-based design exploration algorithms, *AI EDAM*, **29**(4), 443-455.

Likai, W., Patrick, J. and Guohua, J.: 2018, EFFICIENCY VERSUS EFFECTIVENESS, *In Proceedings of the 23rd International Conference on Computer-Aided Architectural Design Research in Asia*, Beijing, 163-172.

Liu, Y.C., Chakrabarti, A. and Bligh, T.: 2003, Towards an ‘ideal’approach for concept generation, *Design Studies*, **24**(4), 341-355.

Maaranen, H., Miettinen, K. and Penttinen, A.: 2007, On initial populations of a genetic algorithm for continuous optimization problems, *Journal of Global Optimization*, **37**(3), 405.

Rasheed, K.M.: 1998, *GADO: A genetic algorithm for continuous design optimization*, Ph.D. Thesis, Ph. D. Thesis, utgers University New Brunswick, NJ.

Rutten, D.: 2013, Galapagos: On the logic and limitations of generic solvers, *Architectural Design*, **83**(2), 132-135.

Turrin, M., Von Buelow, P. and Stouffs, R.: 2011, Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms, *Advanced Engineering Informatics*, **25**(4), 656-675.

Wolpert, D.H. and Macready, W.G.: 1997, No free lunch theorems for optimization, *IEEE transactions on evolutionary computation*, **1**(1), 67-82.

Woodbury, R.F. and Burrow, A.L.: 2006, Whither design space?, *Ai Edam*, **20**(2), 63-82.

Wortmann, T.: 2018, *Efficient, Visual, and Interactive Architectural Design Optimization with Model-based Methods*, Ph.D. Thesis, Ph. D. Thesis, Singapore University of Technology and Design.