EVALUATING PERFORMANCE OF SIMULATED ANNEALING AND GENETIC ALGORITHM BASED APPROACH IN BUILDING ENVELOPE OPTIMISATION

Piyush Varma[1], Bishwajit Bhattacharjee[2]
[1]Environmental Design Solutions Pvt Ltd., India
[2]Indian Institute of Technology – Delhi, India

ABSTRACT

Among many optimisation techniques, Simulated Annealing (SA) is an approach closely related to Genetic Algorithms (GA). While SA and GA have been compared, their performance in Building Envelope Optimisation has not been explored. The objective of this research is to evaluate performance of SA and GA approach in building envelope optimisation. The outcome of study indicates that although GA and SA have comparable performance they have distinct advantages. While GA outperforms SA in initial convergence, it may take longer to reach the optimal. Results also indicate that, near already known good solutions, the SA approach performs better in comparison. There is potential of developing a hybrid optimisation technique that utilizes the strengths of both the GA and SA approaches.

INTRODUCTION

Building Envelope Optimisation techniques have been employed to achieve optimal building envelope with respect to thermal, acoustic, daylight, renewable energy generation or combination of many such considerations. These combinations lead to multiple decision parameters and furthermore variables. Combinations of these variables define a discrete feasibility domain that represents an enormous data set of possible solutions. Building envelope optimisation can then be characterised as achieving (at least) the '*nearly best'* solution without evaluating the entire feasibility domain. This exercise is often a trade-off between time performance and solution quality.

Information of 74 works compiled by Evins indicates that Genetic Algorithm (GA) is one of the most popular techniques in building design optimisation (Evins, 2013). Past research indicates that although Simulated Annealing (SA) may not necessarily outperform GA, it is a flexible approach with comparable performance. Kohonen further adds that instances in literature indicate SA as a 'quick starter' (with respect to GA), but this needs to be verified (Kohonen, 1999).

While the SA and GA approaches are comparable, and have been explored in various contexts, their performance in building envelope optimisation has not been compared. Since optimisation is sensitive to many factors, the outcome of anyone study may not be generalised.

The objective of this study is to compare thermal performance optimisation of the building envelope based on GA and SA methodologies with respect to time performance and quality of solution.

THE OPTIMISATION MODELS

While there are many optimisation methodologies based on various techniques, they are mostly limited in scope and often not comparable. These generative tools are not readily customisable to desired evaluation criteria and objective functions (Caldas et al., 2002). Generic optimisation tools are not best suited to address the complexity of dimensions, qualitative parameters (like orientation) and combination of multiple decision parameters. Additionally, evaluating the optimisation problem on distinct platforms (each for SA and GA) is not expected to produce comparable results.

Therfore instead of using existing tools/toolboxes, both the SA and GA approach were programmed in Matlab®. In addition, the Thermal Load Computation Algorithm (TLCA) based on Admittance Method was also developed in Matlab®.

Prior to this, an SA approach based on model architecture indicated in Figure 1 was developed. The results of this study have been used for comparison with the GA approach.

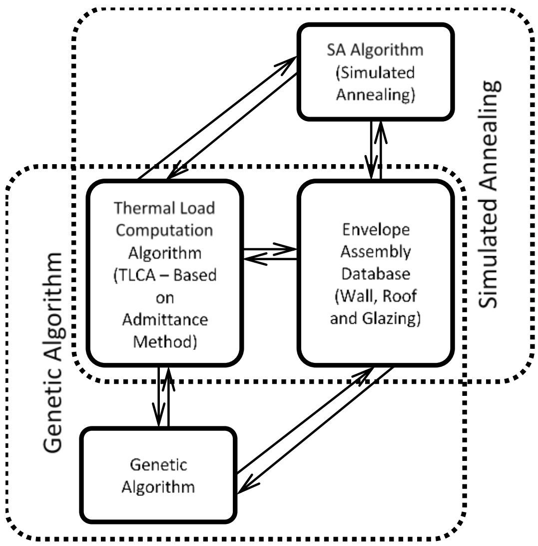

Figure 1 Model components

The Model Architecture

The SA and GA models consist of three components.

- Envelope Assembly Database (EAD)
- Thermal Load Computation Algorithm (TLCA)
- Optimisation Algorithm (OA)

While the EAD and TLCA are identical for the GA and SA approach, the OA follows respective algorithms for both the approaches. Both the models have been harmonised to minimise variability in results.

Harmonising the SA and GA Approach.

For unbiased comparison, the approach developed for GA re-used the material database and thermal load computation algorithm utilised in the SA approach. Figure 1 indicates that both the approaches utilise the EAD and TLCA for performing optimisation. While both algorithms run independent of each other, it is important to note that the objective function, stopping criterion and starting locations were identical in both the approaches. Additionally, the number of dimensions in both the approaches are identical and the applied methodologies involve comparable checks per iteration for both the SA (up to 27 checks) and GA (30 checks) approaches. The identical parameters have been maintained to minimise interference in analysis.

Envelope Assembly Database

The material properties have been referenced from eQuest™ material libraries. Eight options each for Wall and Roof construction assemblies being utilised in the Indian context have been incorporated in the database. In addition to the construction assemblies, Glazing assemblies with eight options for window wall ratios have been incorporated as well. The Roof, Wall and Glazing assemblies participate as the decision parameters in both the GA and SA approaches.

Each option of the decision parameters is mapped to a code in the database. Since all options are coded, even qualitative parameters such as orientation can be integrated into this approach. Eight options for three decision parameters provide for a sample space of 512 combinations (512=8×8×8). Function ‘f(y)’ defines the discrete feasibility domain as given in Equation 01,

$$f(y) = \prod \left(a_I b_J c_K \ldots n_V\right) \qquad (1)$$

Thermal Load Computation Algorithm (TLCA)

TLCA utilises the Admittance procedure to estimate annual cumulative plant load. Compared to robust energy simulation tools, admittance procedure is a numerically inexpensive methodology that provides reasonably accurate results. Research findings by Sodha et al. indicate that Admittance Procedure is employable on hand calculators. The research concludes that Admittance and Fourier methods match well with maximum deviation in total heat flux entering in a space limited to 10% (Sodha et al., 1986)

Therefore, admittance procedure was employed for calculating hourly heating or cooling loads in buildings with reasonable accuracy.

The cumulative annual plant load calculated by the TLCA serves as the fitness function for the OA.

Optimisation Algorithm (OA)

GA approach draws from the theory of evolution to create *‘fitter’* population generation after generation. The ‘*fitness’* of parent is defined as the thermal performance of building envelope. Fitness is essentially a function of combination of decision parameters. The GA approach defines each decision parameter as a collection of genes. Through the processes of selection, crossover and mutation, these genes are passed on from parents to offspring, creating a new generation in the process. This iterative process attempts at refining the population generation after generation to achieve the optimal solution.

OPTIMISATION ALGORITHM (OA)

Binary Encoding

The EAD is a compilation of 3 decision parameters (Wall Type, Glazing Type and Window Wall Ratio). Each of these have further 8 options. These 8 options for all three decision parameters have been encoded in bits (from 000 to 111). Therefore each discrete point in the feasibility domain is a unique collection of 9 bits. For example, a discrete node characterised by ‘Wall Type 1’, ‘Window Type 2’ and ‘Wall Window Ratio 1’ has been encoded as *000 001 000*.

Fitness Assignment

To begin the optimisation on the existing feasibility domain, 30 parents are randomly generated. This pool of 30 parents is consistent with the 30 randomly generated starting locations for the SA approach. These 30 parents form the initial parent pool. Each of these parents is evaluated using the TLCA. The TLCA evaluates the cumulative annual plant load (in kW) for each parent. The cumulative annual plant load represents the fitness value of each function. Minimising the fitness value defines the objective function. Function ‘f(y)’ defined in Equation 02 is the objective function.

$$f(x) = Min\left[\sum_{i=1}^{365}\sum_{j=1}^{24} Q_p(t)\right] \qquad (2)$$

Crossover and Mutation

The existing parent pool randomly pairs 30 parents into sets of 2. For each pair a random crossover point is generated. Both the parents, coded into 9 bits each undergo single point crossover to re-combine and form a pair of offspring. After the recombination, the pair of offspring may undergo single bit inversion for a randomly selected gene.

In a recent study, Alajmi et al. concluded that crossover probability did not contribute to statistical significance and lower mutation rates improved the performance. Therefore, for this study crossover probability of 1.0 and mutation probability of 0.02 is applied (Alajmi et al., 2014).

Ranking and Selection

The offspring pairs generated from the re-combination are evaluated for fitness. The offspring are added to the existing parent pool. Each parent in the pool is ranked as per fitness. The fittest 30 are retained for generating the next pool of offspring. The process is iteratively performed until the Stopping Criterion is achieved.

The Stopping Criterion utilized for this approach is 20 iterations. This is consistent with the Stopping Criterion applied for the SA approach.

SIMULATION AND EXPERIMENT

The simulation, models a 3.04m×3.04m×3.04m located in New Delhi (India, Asia) in the TLCA. TLCA evaluates plant load considering the space as conditioned with internal temperature set point maintained at 24°C. Weather data for New Delhi, compiled by the IWEC has been utilised for this analysis. For optimisation, OA performs optimisation on 30 randomly selected parents in a single run. The OA evaluates the space to return an optimised solution from 512 possible combinations.

RESULTS AND DISCUSSION

Analysing the GA Approach

Figure 2 indicates that within the first iteration, the algorithm achieves a nearly optimal solution. While there is no improvement for the next 6 iterations, the optimal solution is achieved in the 8th iteration. Analysing the results at a finer resolution (evaluating at iterative checks) in Figure 3 indicates high swing in annual plant loads in the first 30 iterative checks. The high swing in annual plant loads in the first 30 iterative checks indicates that the GA approach is an efficient starter. The swing in subsequent iterations damps down, which indicates the refinement in population in each passing generation. Both these observations indicate that GA approach performs efficiently.

It is interesting to note that clones started emerging in the 2nd iteration. By the 5th iteration, 7 clones represented the parent population (of 30) and by the 12th iteration all parents were clones of the optimal solution.

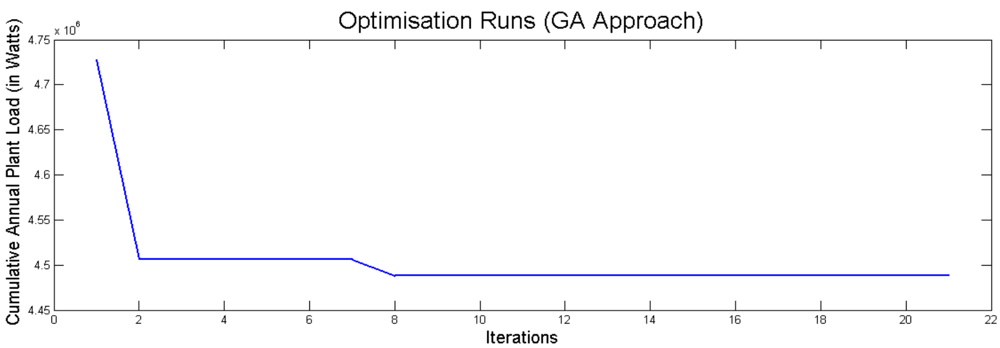

Figure 2 Iteration wise performance of GA approach

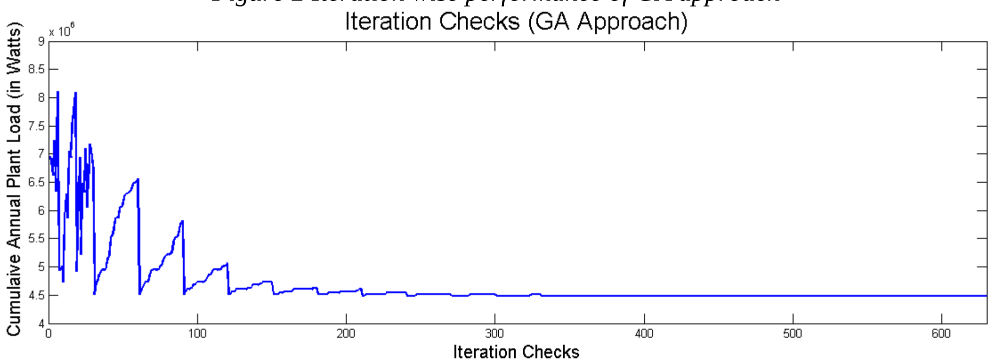

Figure 3 Iteration Checks for GA approach

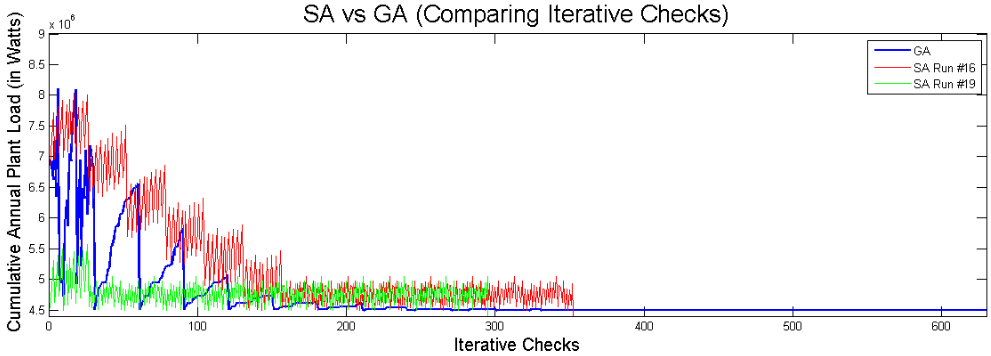

Figure 4 GA vs SA: Comparing amplitude and attenuation

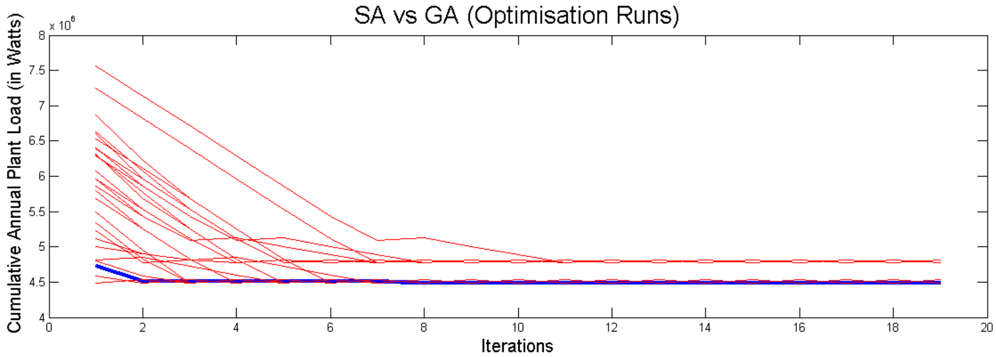

Figure 5 GA vs SA: Comparison of iteration wise performance

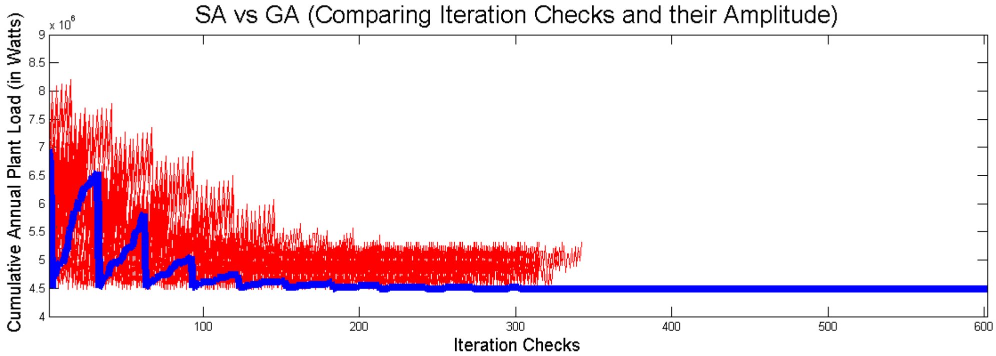

Figure 6 GA vs SA: Comparing swing between the minimum and maximum values per iteration

Comparing SA and GA

There is a significant difference between the SA and the GA approach. While the GA approach evaluates all the 30 initial solutions (or parents) simultaneously, the SA approach evaluates one initial solution (or starting location) at a time. This results in significant difference in the number of iterations performed for evaluating all 30 initial solutions. The SA approach utilises over 9,816 iterative checks, while the GA approach performs only 630. This leads to significant time difference in computation. Computed in identical environments the ratio of computation time between SA and GA is approximately 1:11 in favour of the GA approach. Further, for the 30 initial locations, the SA approach provided the optimal solution 16 times out of 30, while the GA approach reported the optimal solution in its solitary run.

The GA approach demonstrates significant advantage over SA in time performance and solution quality, however, comparison of both these approaches at a finer resolution (evaluation at iterative checks) indicates that both approaches are comparable and that SA approach may outperform GA approach if initial solution (or starting point) of reasonable quality can be pre-determined.

Comparison of Run #16 and 19 of the SA approach against the GA approach is demonstrated in Figure 4 and Figure 5. Run#16 achieves an identical solution for SA approach while starting from fitness values similar to the GA approach, i.e. both operate within the same band. Figure 4 indicates that Run #16 of SA approach and the GA approach have similar peaks (worst solution of that iteration) but different troughs (best solution of that iteration). This indicates that the GA approach achieves *'nearly optimal'* fairly quickly and operates on higher swings while the SA approach being neighbourhood explorer has smaller swings and optimises solution at a steady rate. While both achieve the optimal solution, the SA approach achieves the optimal in the 6th iteration whereas the GA approach achieved the optimal in the 8th iteration. For Run #19, Figure 4 indicates that initial solution was itself one of the nearly optimal solutions and the SA approach achieved the optimal within the 1st iteration. This indicates that near known solutions, the SA approach converges faster as compared to the GA approach.

Analysis of all 30 runs in SA approach at a finer resolution (evaluation at iterative checks) further revealed that while the GA approach achieved the optimal solution in the 211th iterative check, as many as 16 (out of 30) runs reported the optimal solution before the 211th iterative check. Interestingly, Run #7 achieved the optimal solution in the 04th iterative check itself. It is also important to note that the all of the 16 initial locations that returned the optimal solution in SA approach took fewer iterative checks to reach the optimal. Since both the GA and SA approach had the information of the nearly optimal solution, the SA approach outperformed the GA approach for 16 initial solutions. Further analysing this, Figure 6 reports all the runs performed in SA approach (in red) and the outcome of GA (in blue). It is also important to note that while SA outperformed the GA approach in 16 runs, GA approach achieved the optimal solution in single run despite starting-off with information of 14 bad solutions.

CONCLUSION

In conclusion, both SA and GA approach have their respective advantages. The SA approach may provide advantage when good solutions can be pre-determined, while GA approach may help in achieving the '*nearly optimal'* solution sooner. Therefore, while the SA approach involves a greedy algorithm that is deterministic and systematic in searching the optimal solution, the GA approach involves an exploratory algorithm that has a degree of randomness. While randomness is necessary for exploration, it may also take away from the approaching the optimal solution. Both these approaches can be combined to create a hybrid algorithm. The GA approach can be utilised in finding the nearly optimal solution, followed by which the SA approach may be utilised to achieve the optimal solution. Since the GA approach has a degree of randomness, the SA approach may be applied to scout for optimal solutions in the vicinity of solution offered by the GA approach. Therefore, the SA approach can supplement the GA approach as a validation check or as a solution improvement measure.

Study conducted by Lian et al. in the field of Computer Aided Process Planning (CAPP) indicates that the GASA (hybrid Genetic Algorithm and Simulated Annealing) approach outperformed the GA and SA approaches (Lian et al., 2009). Another study in the field of Signal Timing Optimisation outlines the solution quality and CPU time performance of SAGA over SA and GA approaches (Li et al., 2014). Several other studies indicate similar results. Therefore, the premise that hybrid approach has the potential to outperform the GA and SA approach in the Building Envelope Optimisation domain is well founded.

These results are based on evaluation of 3 dimensions (Wall Type, Roof Type and Window Wall Ratio) and binary codes with 3 bit-length only. This dataset has been used with the intent of initial evaluations. Further analysis of these approaches accounting for higher number of dimensions and bigger dataset is essential to corroborate the findings.

The GA approach indicates that clone population can also serve as an effective stopping criterion.

NOMENCLATURE

$f(y)$, sample space identifying discrete feasibility domain;

a_I, b_J ... decision parameters 'I' and 'J' with 'a' and 'b' options available for each decision parameter respectively;

$f(x)$, objective function for minimising cumulative annual plant load;

$Q_p(t)$, annual plant load at hour 't';

REFERENCES

Alajmi A., Wright J. 2014. Selecting the Most Efficient Genetic Agorithm Sets in Solving Unconstrained Optimisation Problem. International Journal of Sustainable Built Environment.

Caldas, L.G., Norford. L.K. 2002. A Design Optimization Tool Based on a Genetic Algorithm, Automation in Construction.

Evins, R. 2013. A Review of Computational Optimisation Methods Applied to Sustainable Building Design, Renewable and Sustainable Energy Reviews.

Kohonen, J. 1999. A Brief Comparison of Simulated Annealing & Genetic Algorithm Approaches, Dept. Of Computer Science, University of Helsinki.

Li, Z., Schonfeld, P. 2014. Hybrid Simulated Annealing and Genetic Algorithm for Optimizing Arterial Signal Timings Under Oversaturated Traffic Conditions, Journal of Advanced Transportation.

Lian, K., Zhang, C., Li, X., Gao, L. 2009. An Effective Hybrid Genetic Simulated Annealing Algorithm for Process Planning Problem, Fifth International Conference on Natural Computation.

Sodha M.S., Kaur B., Kumar A., Bansal N.K. 1986. A comparison of the Admittance and Fourier methods for predicting heating/cooling loads. Solar Energy.