HYBRID APPROACH FOR BUILDING ENVELOPE OPTIMISATION USING GENETIC ALGORITHMS AND SIMULATED ANNEALING

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

.

ABSTRACT

Optimisation of building envelope for thermal performance is a complex problem involving a multitude of variables. While the combinations of variables lead to a vast dataset of discrete points forming the feasibility domain, the optimal solutions are only few and difficult to discern.
The objective is to develop a methodology using a hybrid approach involving both Genetic Algorithm (GA) and Simulated Annealing (SA) approach to optimise the building envelope with respect to thermal performance.
The GA and SA are complementary algorithms. The methodology uses the advantages of both these methodologies to improve the solution. The outcome indicates that Hybrid (GASA) approach improves the reliability of solution. While there is minor improvement in solution quality Hybrid (GASA) approach achieves the optimal frequently.

INTRODUCTION

Building Envelope Optimisation

Building Envelope Optimisation is a complex problem characterised by a plethora of decision parameters and innumerable design options that swell the feasibility domain. In addition to being vast, the feasibility domain is also discontinuous, non-convex space defined by both qualitative and quantitative decision parameters. Since evaluating the entire feasibility domain is unrealistic, optimisation algorithms offer a trade-off between solution quality and time performance.

Many researchers over the past have applied optimisation algorithms to overcome the constraints of computation power and time performance. Among many approaches utilised in the past, evolutionary algorithms have been frequently used. A study of optimisation works compiled by Evins indicates that GA has been a popular choice among researchers (Evins, 2013). Since GA approach operates on a population of points instead of a single point it is more likely to avoid local optima (Caldas, 2002). In general, GA finds optimal solutions for a variety of engineering problems (Wetter et al., 2003). However, in spite of their wider appeal and popularity, GA suffers from extreme reliance on the crossover operation, which leads to premature convergence and population stagnation. Therefore, GA is unable to improve after finding near optimal solutions. (Li et al, 2014). There have been multiple instances across different application areas that indicate hybridising the GA approach can provide reliable results. Hasan et al., performed optimisation using Hooke-Jeeves and GA to minimise life cycle cost of an electrically heated single family detached house (Hasan et al., 2008). Palonen et al. combined NSGA-II and Omni-optimizer to optimise cost for construction and operation (Palonen et al., 2009). These studies indicate improvement in reliability over *'GA only'* optimisation.

While GA is compatible with other optimisation algorithms, its combination with SA is of particular interest. GA and SA are effective optimisation methods with complementary strengths and weaknesses (Lian, et al., 2009). Whereas GA has advantage over SA in exploration, SA is known to perform better than GA near already known solutions. Hence, the hybrid approach i.e. combinations of GA and SA provides opportunity for improving solution quality.

Hybrid GA and SA algorithms have successfully performed in Computer Aided Process Planning (CAPP) (Lian et al., 2009), Signal Timing (Li et al., 2014), and many other applications. In Building Envelope application, Junghans et al used a combination of GA and SA to optimise energy costs vis-à-vis building envelope and shading geometry (Junghans et al., 2014). These studies indicate improvement in solution reliability over GA as well.

Optimisation of building performance is typically performed by combining optimisation solvers with robust simulation software. This paper uses Admittance Method instead of utilising simulation software for thermal performance evaluation. Even though present generation simulation software are robust, Admittance Method provides results comparable to Fourier methods and computes thermal loads within maximum deviation of 10% (Sodha et al., 1986). Admittance Method allows significant time advantage while evaluating solution with reasonable accuracy.

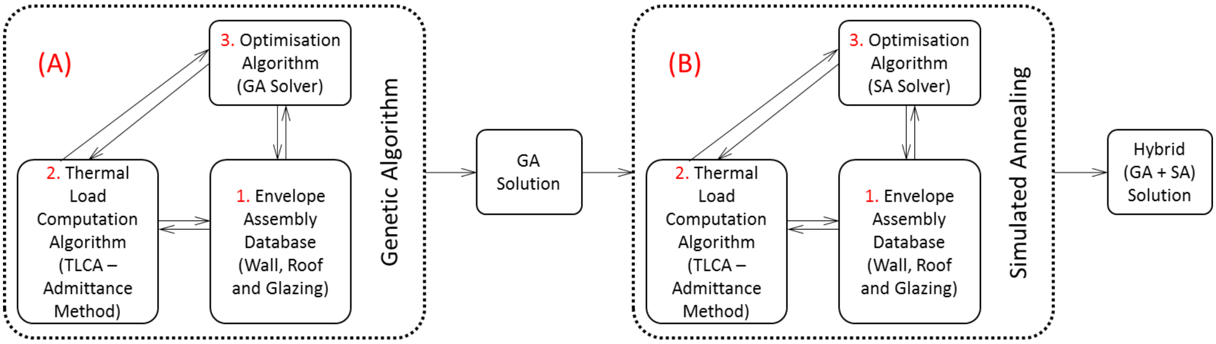

Figure 1 Model architecture: Hybrid (GASA)

This methodology is expected to improve time performance significantly. The optimisation algorithms, i.e. GA and SA solvers and Admittance based heat computation algorithms have been developed in MATLAB. Time performance is evaluated within the MATLAB using the time profiler.

This paper demonstrates a Hybrid methodology (GASA) and compares the time performance and solution quality with GA outcomes.

THE OPTIMISATION MODEL

Model Architecture

Figure 1 is a graphic representation of the model architecture, its components and their interactions. As indicated in Figure 1, this approach is a sequential process where the model finds an optimal neighbourhood using the GA approach (see (A) in Figure 1) and the SA approach refines the solution further (see (B) in Figure 1). This approach leverages the exploratory powers of GA and systematic search capability of SA near known solutions. In combination, refinement over *'GA only'* approach is expected.

Essentially, the optimisation process in (A) and (B) are identical with the only exception of the optimisation solvers. While the solver in (A) is based on GA, (B) uses an SA based solver.

The optimisation process consists of the following three modules:

1. Envelope Assembly Database (EAD)
2. Thermal Load Computation Algorithm (TLCA)
3. Optimization Algorithm (OA)

The EAD is a combination of decision parameters that represent a building envelope. The OA randomly selects one or more of these building envelopes. The TLCA evaluates these building envelopes and reports the outcome to the OA. The OA interacts with EAD again to optimise the solution iteratively. The outcome is the optimised solution. The following sections discuss these modules in detail.

Envelope Assembly Database

Materials for wall and roof assembly have been compiled from material database in eQuest. The compiled materials have been organised into conventional construction assemblies. The wall and roof construction assemblies account for thermal properties of materials including conductivity, density, specific heat and thickness. The wall construction assemblies include Brick construction, Cavity walls, Insulated cavities (Polystyrene, Polyurethane, Mineral Board and Mineral Wool) and Curtain wall systems. Collated roof construction assemblies include concrete slab, over-deck insulation (Polystyrene, Polyurethane, Mineral Board and Mineral Wool) and roof gardens over concrete decks. Glazing assemblies and corresponding specifications have been collated from various manufacturers' catalogues. Table 1 identifies the best and worst case Wall, Roof and Glazing construction assemblies.

Table 1 Best and Worst case construction assemblies used in the Envelope Assembly Database

	DESCRIPTION (OUTSIDE TO INSIDE)	**U- FACTOR (W/m^2-^{o}K).**
Wall Assembly		
Worst Case	Stucco – Plaster – Brick (230 mm) – Plaster Stucco	2.37
Best Case	Spandrel Glass – Polyurethane (75mm) – Gypsum Board - Stucco	0.29
Roof Assembly		
Worst Case	Slate Tile – Cement Mortar – Concrete Slab (150mm)– Plaster - Stucco	1.86
Best Case	Soil (300mm) – Screed – Polyurethane (100 mm) – Concrete (150mm) – Plaster - Stucco	0.16
Glazing Assembly		
Worst Case	50% Window wall ratio	5.80
Best Case	20% Window wall ratio	1.00

These assemblies are organised in a spreadsheet software and are easily editable by non-expert users as well. For this study 32 (2^5 =32) assemblies each for Roof, Wall and Glazing have been collated. Since the GA solver identifies each assembly as a binary code, the number of assemblies compiled in EAD are to the power of 2. With 32 design options for each of the three decision parameters, feasibility domain is a sample space of 32,768 combinations. Function ‘f(y)’ defines the discrete feasibility domain as given in Equation 01,

$$f(y) = \prod (a_I b_J c_K \dots n_V) \tag{1}$$

Solution Space Abstraction

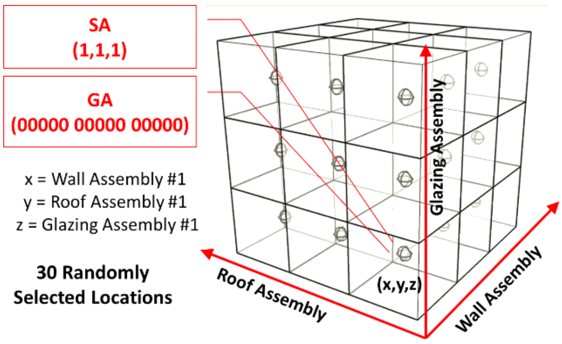

Figure 2 Solution space abstraction

Figure 2 abstracts the solution space or assembly database in a Cartesian coordinate system. Each axis of this coordinate represents a decision parameter (Roof, Wall or Glazing assembly type) and all 32 assembly types are numerically coded (1 to 32) on respective axis. Each node, characterised as a coordinate is a solution which represents a combination of the three decision parameters.

For example, node (1,1,1) would depict a combination of Wall Assembly Type 1, Roof Assembly Type 1 and Glazing Assembly Type 1. The sample space therefore extends from coordinate (1,1,1) to (32,32,32). Both GA and SA use the same assembly database. While the SA algorithm directly uses the numeric codes, GA uses binary codes. For GA, each node is a 15 - bit binary code with 5 bits each for the roof, wall and glazing assembly. For example, node (1,1,1) is coded as (00000 00000 00000). While, other coding options are available, the advantage of using Binary coding is that it allows for more crossover points and hence greater diversity.

This methodology is scalable to accommodate more dimensions. In addition to scalability the methodology can also accommodate qualitative parameters like orientation as coded inputs. Coded information indicates that the algorithms can handle discontinuous functions with ease.

Thermal Load Computation Algorithm (TLCA)

Thermal load computation is performed using Admittance Method. This procedure computes unsteady state heat transfer using frequency domain response method. The TLCA evaluates the cumulative annual plant load *(Q_p)* for a combination of decision parameters while maintaining the internal space temperature at 24ºC.

OPTIMISATION ALGORITHM

Objective Function and Fitness Assignment

Both GA and SA Optimisers have a common objective function. The objective function defined in Equation 02 is the minimization of cumulative annual plant load.

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

Fitness for each solution in a generation (GA) or neighbourhood (SA) is dictated by the cumulative annual plant load *(Q_p)* of that solution. The lower the cumulative annual plant load *(Q_p)*, the higher the fitness.

Genetic Algorithm (GA)

Influenced by Darwinian principles, GA is a process of combining generation of parents to create a generation of offspring. By natural evolution the gene pool improves after each generation resulting in fitter individuals.

In context of this paper, the gene pool refers to the decision parameters. Each decision parameter is collection of genes. Through the processes of selection, crossover and mutation, these genes are passed onto the next generation. This process is repeated iteratively until the optimal solution is achieved.

The algorithm randomly selects 30 parents and arranges them in 15 pairs. Each parent coded into 15 bits undergoes single point crossover (about a randomly generated point) with its partner to exchange genes and generate a pair of offspring. Occasionally an offspring might undergo bit inversion to increase population diversity.

Recent study indicates that lower mutation rates improved performance, while crossover probability did not contribute to statistical significance (Alajmi, et al., 2014). Therefore, the algorithm maintains a crossover probability of 1.0 and mutation probability of 0.02.

The offspring resulting from re-combination of 15 parent pairs are evaluated for fitness. The offspring supplement the existing parent pool. All the parents are ranked as per their fitness. The fittest 30 are retained. This new parent pool re-combines to produce the next generation. This 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.

Simulated Annealing (SA)

SA draws from the metallurgical process of annealing metals. The process of heating followed by slow cooling, characterises move to a lower internal energy state. Typically, these jumps from higher energy state to lower energy state tend to happen in close vicinities. Iteratively exploring close vicinities to achieve lower internal energy state lead to the optima.

In context of this paper, a combination of assemblies numerically coded as a physical location in space is surrounded by potential solutions. For example, say wall type 3, roof type 4 and glazing type 8 characterise a physical location [3,4,8]. This physical location is surrounded by 26 other possible solutions. Each of these neighbours is evaluated for fitness. In case any neighbour demonstrates fitness over the existing location, the algorithm shifts to this new location and evaluates the new set of neighbours. This process of moving to a lower internal energy state is characterised as a *'Downward Move'*.

In some cases, the current location may be the lowest internal energy state compared to its neighbours. Considering that this may be a local minimum, the algorithm moves the current location to the next higher state of internal energy. This move is characterised as *'Upward Move'*. After the 'Upward Move' the algorithm performs search for a lower internal energy state near the new current location. The 'Move' count serve as Stopping Criterion for the algorithm. The Stopping Criterion has been set as 20 'Moves'. These moves are iteratively performed until the stopping criterion is achieved. The algorithm retains memory of route traversed and associated energy states to return node defining the lowest energy state.

SIMULATION AND EXPERIMENT

A hypothetical space measuring 3.04m×3.04m×3.04m located in New Delhi (India, Asia) has been modelled in the TLCA. Considering the space as conditioned with internal temperature set point maintained at 24ºC, the TLCA evaluates the cumulative annual plant load (in Watts). Weather data for New Delhi, compiled by the IWEC has been utilised for this analysis.

From this solution space of 32,768 combinations, a population of 30 was randomly selected. The selected population underwent crossover and mutation over 20 generations to report the GA optimal. This GA optimal was fed into the SA algorithm for solution refinement over 20 iterations. The outcome was either a refined solution or the GA solution was maintained.

This procedure for 30 randomly selected solutions was performed 30 times to test the improvement in solution. Further, outcome of 30 Runs for Hybrid (GASA) and GA were compared.

Following the simulation, entire solution space was evaluated to verify the outcome. The code to evaluate the solution space was also developed in MATLAB. The MATLAB scripts were performed on a general purpose notebook with i5 processor and 2 GB RAM.

RESULTS AND DISCUSSIONS

Solution Space Evaluation and GASA Outcome

The evaluated sample space reveals the worst case cumulative annual plant load of 7.90 million watts and optimal of 2.07 million watts. This indicates a significant difference of 74% between the optimal and worst case. Figure 3 indicates the distribution of solutions in the feasibility domain and the solution refinement potential of the Hybrid (GASA) approach. Solution outcomes of 30 independent Hybrid (GASA) Runs indicate that the solutions lie in the top 1.9%. The 30 independent runs yielded 8 distinct solutions. Out of the 30 runs, Hybrid (GASA) approach yielded the optimal solution 16 times. Overall, the 8 unique solutions are comparable and within 7% of the optimal solution.

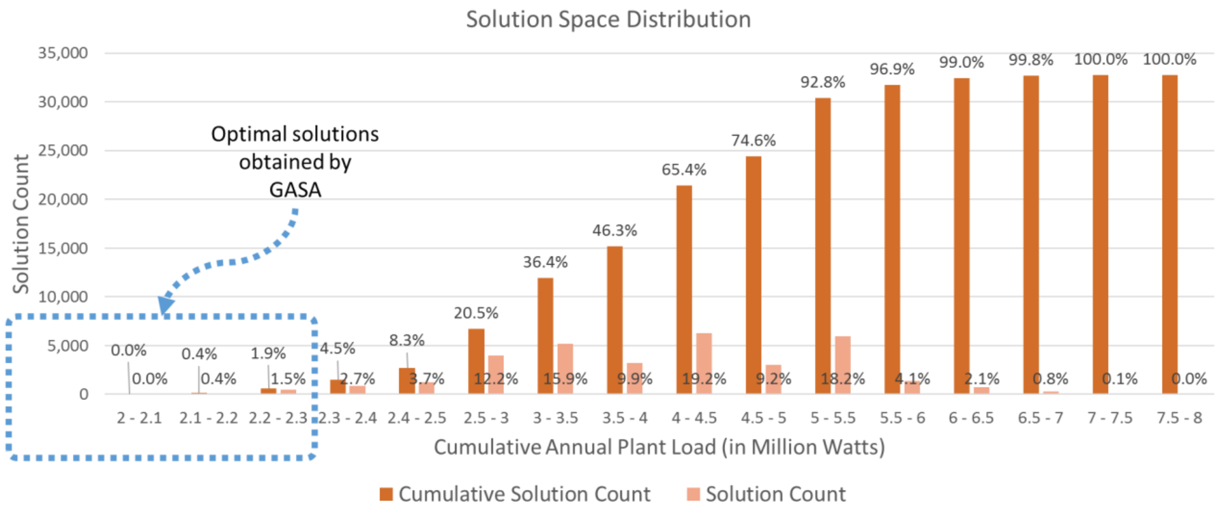

Figure 3 Histogram indicating spread of solutions in the sample space

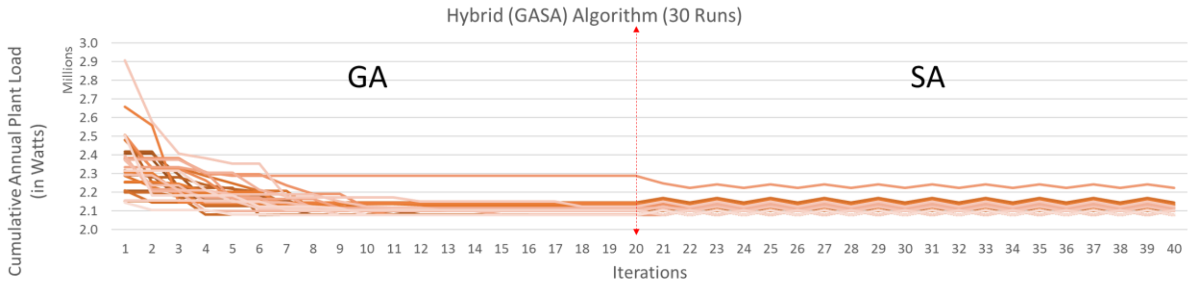

Figure 4 Outcome of Hybrid (GASA) Algorithm (30 Runs)

Table 2 Performance comparison of Hybrid (GASA) and GA approach

	HYBRID (GASA)	GA.
OUTCOMES	08	15
FREQUENCY FOR ACHIEVING OPTIMAL	16 (53.3%)	09 (30%)
MINIMUM SOLUTION (OPTIMAL) *(in Watts)*	2,078,343	2,078,343
MAXIMUM SOLUTION *(in Watts)*	2,223,505	2,288,134
AVERAGE SOLUTION *(in Watts)*	2,099,043	2,107,471
SOLUTION RANGE *(in Watts)*	145,163 (6.98% of Optimal)	209,791 (10.09% of Optimal)
STANDARD DEVIATION *(in Watts)*	30,430	39,899
AVERAGE RUN TIME PER ITERATION *(in seconds)*	~188	~143

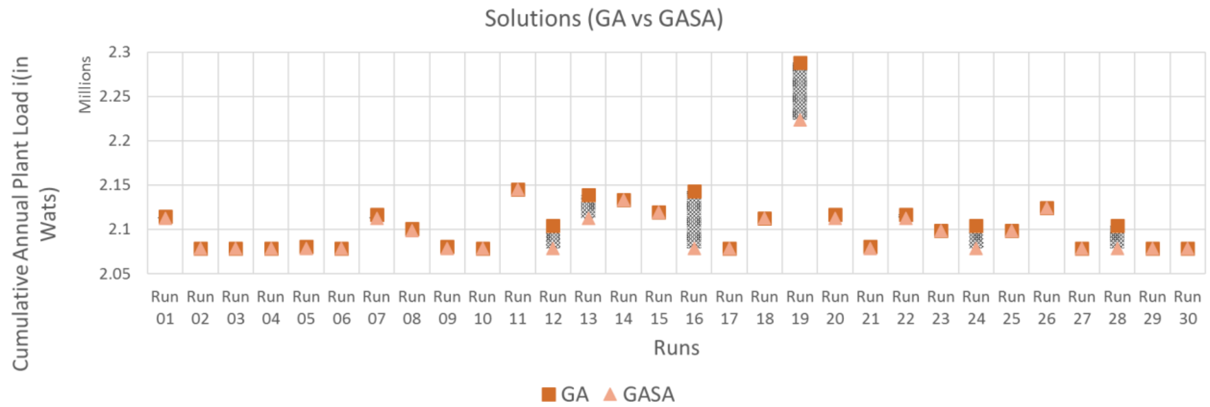

Figure 5 Solution comparison of Hybrid (GASA) and GA approach

The outcome of 30 independent runs is indicated in Figure 4. The first 20 iterations (or generations) correspond to GA optimisation while the next 20 iterations outline SA refinement over the GA outcome. The outcome indicates that while convergence for GA algorithm continues until 18 iterations for Run #26, it terminates at the 4th iteration for Runs #3, 13 and 23. Since solution convergence is seen as late as the 18th iteration, maintaining 20 iterations as Stopping Criterion is acceptable. After completing the GA cycle of 20 iterations, the SA optimiser performs optimisation for the next 20 iterations. The SA optimiser improves outcome for 16 Runs in the first iteration itself and for 5 of these Runs (# 7, 16, 19, 20 and 22), the refinement is performed in the next iteration as well. After the initial refinement, the SA solver keeps oscillating around the solution. This indicates that Stopping Criterion for the SA optimiser can be reduced to 5 iterations or less for improving time performance without compromising solution quality.

Outcome in Figure 4 indicates that while primary responsibility of solution convergence lies with GA, the SA solver improves the solution marginally. This is expected since SA is already operating in optimal neighbourhoods.

Hybrid (GASA) and GA Approach Comparison

Hybrid (GASA) approach is essentially SA solver applied over GA solution. Since GA performs independently, GA and Hybrid (GASA) approach can be compared. Table 2 indicates a performance comparison of Hybrid (GASA) and GA approach. Hybrid (GASA) and GA approach successfully achieve solutions within top 1.9% of the solution space. While Hybrid (GASA) approach achieves optimal 16 out of 30 times (53%), GA achieves the optimal only 9 out of 30 times (30%). Further, Hybrid (GASA) approach is able to improve the solution 14 out of 30 times (46.4%). of the 16 times, where no solution improvement is reported, GA is already at the optimal in 9 instances. Therefore, Hybrid (GASA)

approach is unable to contribute to solution improvement in only 7 out of 30 Runs (23.3%).
Figure 5 indicates solution comparison between Hybrid (GASA) and GA approach. The outcome indicates noticeable solution improvement in 5 of the 30 Runs. Run #16 and #19 demonstrate solution improvement of approximately 3%.
As indicated earlier, the GA approach reports 15 solutions. The location information of these 15 solutions reveals 8 neighbourhoods. The SA solver in the Hybrid (GASA) approach is able to successfully refine the 15 GA solutions to 8 solutions from 8 neighbourhoods. This indicates that the GA approach consistently visits the optimal neighbourhoods and the subsequent operations of the SA solver refine the solution to optimal or nearly optimal. Figure 6 indicates that these 8 solutions are in close vicinity to each other.

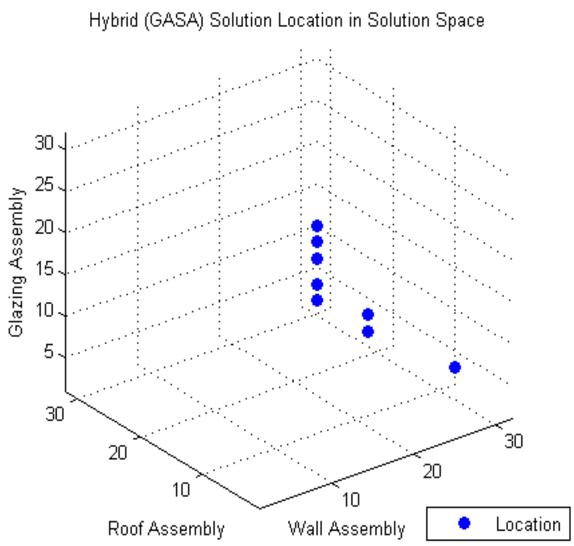

Figure 6 Location of Hybrid (GASA) solutions

Average run time for a single Hybrid (GASA) run is approximately 190 seconds, while that for GA is 140 seconds. Although Hybrid (GASA) approach utilises 35% more time, the outcome in Figure 4 indicates that significant computation time utilised by the SA solver is spent in re-evaluating already visited solutions. There is potential for improvement in time performance for the Hybrid (GASA) approach by using appropriate stopping criterion for the SA solver. The GA solver too presents an opportunity to improve time performance by limiting the number of re-evaluated solutions. It is interesting to note that significant clone populations are seen in the GA operations. For example, as early as the 10th iteration of Run#11, all parent population were clones. By 14th iteration at least 15 Runs were all clones and by 16th almost all were clone populations. Using clone populations per iteration as stopping criterion should lead to improvement in time performance. Improvement in time performance is expected to reduce the time gap between the GA and Hybrid (GASA) approach.

CONCLUSIONS

Material database used in this methodology is representative of conventional construction systems. The outcome indicates that a vast range of solutions exists for these prevalent systems. This implies that optimisation is an essential tool for design improvement.
Since the Hybrid (GASA) approach returns the optimal solution more often, the number of unique solutions are fewer and within a narrower range as compared to the GA approach, the outcome of Hybrid (GASA) approach is more reliable.
The results indicate that combination of GA and SA approach are complementary. While the exploratory nature of GA aids in exploring optimal neighbourhoods, the systematic approach of SA refines solutions in the neighbourhood.
Although this paper evaluates a simple geometry with only 3 decision parameters, this methodology is scalable to utilise complex geometries with multiple decision parameters.

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 Algorithm 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.

Hasan, A., Vuolle, M., Siren, K. 2008. Minimisation of Life Cycle Cost of Detached House Using Combined Simulation and Optimisation, Building and Environment.

Junghans, L., Darde, N. 2014. Hybrid Single Objective Genetic Algorithm Coupled With the Simulated Annealing Optimization Method for Building Optimization. Energy and Buildings.

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.

Palonen, M., Hasan, A., Siren, K. 2009. A Genetic Algorithm for Optimisation of Building Envelope and HVAC Parameters, 11th International IBPSA Conference.

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.

Wetter, M., Wright, J.A. 2003. Comparison of a Generalized Pattern Search and Genetic Algorithm Optimization Method. Eighth International IBPSA Conference