

1 **ACCELERATING THE MULTI-OBJECTIVE OPTIMISATION OF A SEMI-**  
2 **TRANSPARENT BUILDING INTEGRATED PHOTOVOLTAIC FACADE**  
3 **THROUGH THE USE OF ANT COLONY ALGORITHM**

4  
5 John MacModeller<sup>1</sup>, Jane MacSimulator<sup>2</sup>, and Another MacAuthor<sup>2</sup>

6 <sup>1</sup>BS0-2014 Secretariat, UCL, London, UK

7 <sup>2</sup>Another Institution, University of Origin, Some City, Some Country

8 The names and affiliations SHOULD NOT be included in the draft submitted for review.

9 The header consists of 10 lines with exactly 14 points spacing.

10 The line numbers are for information only. The last line below should be left blank.  
11

*(The note below is for Reviewers. It will not be part of the paper)*

*Dear Reviewers,*

*Pls note that the title has been modified to reflect more closely to the content of the abstract submitted on 20 September 2013 and this full paper. Sorry for any inconvenience.*

*Best Regards, Authors of paper submission 136*

## ABSTRACT

Evolutionary algorithms have popularly been used for the past ten years in building performance optimisation (Attia 2013). The long runtime of a performance-based multi-objective optimisation can be reduced by using faster proxy simulations (Choo et al 2013). Beside using proxy simulations, the optimisation process can be improved by selecting other types of multi-objective algorithms. This paper will present the use of multi-objective ant colony algorithm as a possible alternative to multi-objective evolutionary algorithm. The multi-objective optimisation of a semi-transparent building integrated photovoltaic (BIPV) facade is used for the proof of concept. The design of semi-transparent BIPV facades has an impact on a wider range of factors, including solar heat gain and daylight penetration into the rooms of the building. Results from the experiments conducted show that multi-objective ant colony algorithm can speed up the multi-objective optimisation process but does not perform as well as the multi-objective evolutionary algorithm.

## INTRODUCTION

Long runtimes are common in performance-based multi-objective optimisations of building parameters. This is because detailed building performance simulation tools, which have relatively long runtimes, are typically coupled with a multi-objective optimisation evolutionary algorithm. Choo et al (2013) has shown that the runtime of a performance-based multi-objective optimisation can be reduced by using faster proxy simulations.

Commonly used multi-objective algorithm like evolutionary algorithm (EA) have been widely used in building related multiobjective optimisation (Caldas 2008, Charron and Athienitis 2006, Wang et

al 2005). Attia et al (2013) highlighted that EA has been very popular for the past ten years in building performance optimisation. They have even been integrated into software packages for the designers' convenience. An example of such integration is Galapagos, an optimisation component in Grasshopper (Rutten 2011), a visual data modelling system that allows designers who are not trained in scripting to quickly generate parametric models.

Designers usually use the existing optimisation tool provided in the software packages and do not try and implement other types of multi-objective optimisation algorithms. Such algorithms are not restricted to just multi-objective evolutionary algorithm (MOEA). There are also swarm intelligence-based optimisation algorithms like multi-objective ant colony (MOAC) which are much newer than EA and are therefore not commonly used for building related applications.

Hence, this paper will present the use of MOAC for improving the speed and performance of multi-objective optimisation. A comparison between MOAC and MOEA will show if MOAC can be a better alternative.

The multi-objective optimisation of a semi-transparent building integrated photovoltaic (BIPV) facade is used for the proof of concept. Unlike typical roof-mounted photovoltaic systems, where performance is predominantly focused on the amount of electricity generated, the design of semi-transparent BIPV facades has an impact on a wider range of factors, including solar heat gain and daylight penetration into the rooms of the building. Hence, a semi-transparent BIPV facade is used here because its conflicting performance criteria presents a good design scenario for multi-objective optimisation.

The paper is structured as follows: in the next section we will give a detailed description of parametric model and facade performance metrics used in the experiments. It is followed by an overview of both MOAC and MOEA. Thereafter we will present the methodology used to conduct the experiments. Lastly, the results will be analysed and discussed.

## EXPERIMENTS

To compare the performance of both MOAC (multi-objective ant colony) and MOEA (multi-objective evolutionary algorithm), a design problem which involves the multi-objective performance optimisation of a semi-transparent BIPV facade is first defined.

### Parametric Model.

A parametric model of a typical office space with a semi-transparent BIPV facade is created with Houdini, a procedural modelling software from Sidefx (2013). A typical north oriented office space for an occupancy of one person with 4 m (width) x 4 m (depth) x 3 m (height) is modelled for the experiment, as shown in Figure 1 (top). The facade is separated into three panels: BIPV panels 1, 2, 3, each of them is independent from one another. For each panel, three design variables, cell width, cell height, and cell spacing are defined in the parametric model as shown in Figure 1 (bottom). These design variables  $x_{ij}$ , define the PV cell pattern for each independent panel, where subscript  $i = 1, 2, 3$ , represents the panels and subscript  $j = 1, 2, 3$  represents the three design variables (cell width, cell height and cell spacing) in each facade panels. Cell width and height vary from 5 – 15.5 cm at 0.5 cm steps but are independent from each other. Cell spacing varies from 0.5 – 5 cm at 0.5 cm steps. All the cells of the semi-transparent BIPV facades will be similar in shape. The pattern occupies a facade with a height of 3 m and width of 4 m.

### Façade Performance Metrics

The multi-objective optimisation of a semi-transparent BIPV façade will involve maximising electricity generation, minimising the ETTV (envelope thermal transfer value) and maximising the working plane area, 0.85m height from the floor, that has a minimum illuminance of 300lx. These three performance metrics are used as fitness functions for the optimisation because of their relatively fast speed of computation. The competing goals of each metric presents a good setup for the multi-objective optimisation algorithms to balance the trade-offs.

### Electricity Generation

The fitness function for annual electricity generation is based on the following mathematical equation:

$$E_{pv} = A_{pv} \cdot F_{pv} \cdot G_t \cdot \eta_{mod} \cdot \eta_{inv} \quad (1)$$

where  $E_{pv}$  is the electrical energy produced by the photovoltaic system ( $\text{kWh} \cdot \text{a}^{-1}$ ),  $A_{pv}$  is the gross area of the semi-transparent BIPV facade ( $\text{m}^2$ ),  $F_{pv}$  is the fraction of surface area with active solar cells,  $G_t$  is the total annual solar radiation energy incident on the BIPV façade (which is computed at  $561 \text{ kWh} \cdot \text{m}^{-2} \cdot \text{a}^{-1}$  by Radiance),  $\eta_{mod}$  is the semi-transparent BIPV module efficiency (which is set at 12%) and  $\eta_{inv}$  is the average inverter efficiency (which is set at 90%).

### Envelope Thermal Transfer Value

ETTV is used as the second fitness function for the optimisation. ETTV is an easy-to-use mathematical equation to calculate the heat transfer through the façade (BCA 2004). It was developed as a measure of the thermal performance of the building envelope.

The equation of ETTV is shown below:

$$ETTV = 12(1 - \omega) \cdot U_w + 3.4\omega \cdot U_f + 211\omega \cdot C \cdot S \quad (2)$$

where  $\omega$  is the window-to-wall ratio,  $U_w$  is the thermal transmittance of an opaque wall which is  $0 \text{ W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}$  because the modelled facade is a full height curtain wall,  $U_f$  is the thermal transmittance of the fenestration, which is assumed to be  $5.8 \text{ W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}$  in the experiments,  $C$  is the correction factor for solar heat gain through the fenestration and  $S$  is the shading coefficient of the fenestration.

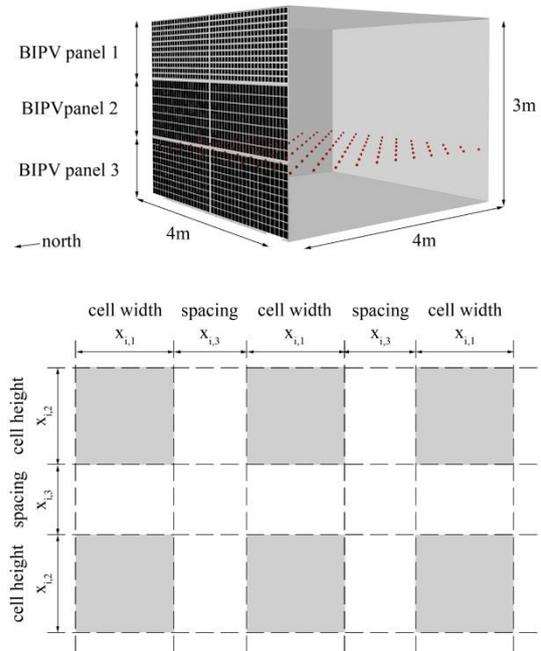


Figure 1. Top: Typical office used in the simulation model with illuminance sensor points (in red) on a  $10 \times 10$  grid. Bottom: Schematic of cell arrangement for the modelled semi-transparent BIPV façade.  $i = 1, 2, 3$ , where  $i$  is the BIPV panel number.

### Floor Area with Minimum Illuminance

The last fitness function is the working plane area, 0.85m height from the floor, which has a minimum illuminance for the office of 300 lx. The illuminance level is calculated using the Radiance software (Ward and Shakespeare 1998). Sensor points on a  $10 \times 10$  grid at 0.85m height from the floor are used for the illuminance simulation (refer to Figure 1, top). The illuminance level is calculated with an overcast sky for the June solstice at 1200h where the sun faces the north façade of the typical office space. An overcast

sky is assumed here for the worst daylight scenario. The following settings were used in Radiance:  $ab = 2$ ,  $ad = 1000$ ,  $as = 20$ ,  $ar = 300$  and  $aa = 0.1$ , where  $ab$  is ambient bounce,  $ad$  is ambient resolution,  $ar$  is ambient resolution and  $aa$  is ambient accuracy. The detailed explanation of the settings is beyond this paper. They can be referred to in the Radiance manual. (Ward and Shakespeare 1998)

## OVERVIEW OF ALGORITHMS

### Algorithm for MOAC Optimisation

Dorigo (1992) was the first to introduce ant colony optimisation. Its concept mimics the food foraging behaviours of ant colonies to find the shortest path between the food sources and their nests. Ants randomly explore their surrounding and leave a pheromone trail on the ground. Ants probabilistically choose the paths marked by strong concentration of pheromone levels. When an ant finds a food source, it will evaluate the quantity and quality of the food. The quantity of pheromone that an ant leaves on the ground depends on the quantity and quality of the food, which is the behaviour that the ant colony optimisation model replicates mathematically.

Applying MOAC to a multi-objective optimisation problem, the graphical representation in Figure 2 describes how the problem is represented in an ant colony optimisation algorithm.

In Figure 2, each layer is a design variable  $x_{ij}$  in Figure 1. Nodes  $d_{rs}$  are the values of each design variable (layer), where  $r$  is the layer number and  $s$  is the design variable number. There are a total of 9 layers. Each layer shown in Figure 2 has 22 nodes. The nodes are connected through links with pheromone value,  $\tau_{uv}^{rs}$ . The subscript  $u$  is the next layer where  $u = r + 1$  and  $v$  is the design variable number of this next layer. An example of a possible “path” or design variant is shown as a bold (red) line.

The ant colony optimisation algorithm contains the following components:

- Number of artificial ants in a colony,  $m$  – each “ant” represents a possible design variant.
- Probability of selection,  $p_{uv}$  – this is the probability of selecting a link from node  $rs$  to  $uv$  as shown in Figure 2.
- Initial pheromone trail,  $p_{init}$  – all links in the solution space of an ant colony optimisation algorithm is first initialised with a single value:

$$p_{init} = \frac{1}{\bar{a} + \bar{b} + \bar{c}} \quad (3)$$

Where  $\bar{a} + \bar{b} + \bar{c}$  are the averaged normalised fitness functions of all the design variants in the first iteration.

- Pheromone value of each link:

$$\tau_{uv}^{rs} \leftarrow (1 - p_{ev})\tau_{uv}^{rs} \text{ for all } r, s, u \text{ and } v \quad (4)$$

where  $p_{ev}$  is the pheromone evaporation rate.

- Pheromone evaporation rate,  $p_{ev}$ , is applied to the existing pheromone value when an artificial ant,  $k$ , has taken a path to the food source.
- Pheromone deposit, is the pheromone value after an artificial ant,  $k$ , has taken the same path back to the nest. Pheromone deposit is applied to the links on the path after  $a_k, b_k, c_k$  are calculated:

$$\tau_{uv}^{rs} \leftarrow \tau_{uv}^{rs} + \frac{1}{a_k + b_k + c_k} \quad (5)$$

where  $a_k, b_k, c_k$  are fitness values of fitness functions of objective  $a, b, c$  for an artificial ant,  $k$ .

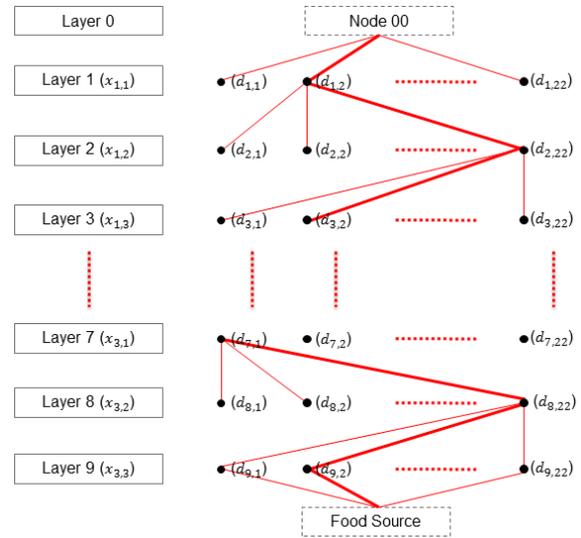


Figure 2. Graphical representation of a discrete variable problem with 9 design variables as described in the text.

The following describes the multi-objective ant colony optimisation algorithm:

*randomly select paths (feasible design variants) for  $m$  number of ants,*

*calculate values of fitness functions,*

*initialise all links with pheromone,  $p_{init}$ ,*

*update pheromone value,  $\tau_{uv}^{rs}$ , on each link of the selected path as described in equation (4),*

### Repeat

*calculate probability of each link  $p_{uv}$ ,*

*select each link to form a path (feasible design variant) for  $m$  number of ants based on probability  $p_{uv}$ ,*

*for each of the selected path,*

*evaporate pheromone value with evaporation rate  $p_{ev}$ ,*

calculate the performance metrics,  $a$ ,  $b$  and  $c$ ,  
 update pheromone value,  $\tau_{uv}^{rs}$ , on each link of  
 the selected path as described in equation (5),

**until** maximum number of iterations,  $w$ .

For a more detailed description on ant colony optimisation, Dorigo and Stutzle's book on Ant Colony Optimization (Dorigo and Stutzle 2004) provides thorough and in-depth explanations.

### Algorithm for MOEA

MOEA is an optimisation algorithm based on natural evolution (Eiben 2007). The algorithm contains the following components:

- Representation – a set of design variables,  $x_{ij}$ , which are typically called the genotypes, are used to define possible design variations, typically called phenotypes or individuals,  $y_k$ . Individual,  $y_k$ , represents the  $k^{th}$  design variant in a solution space.
- Population – a population is a set of individuals in the evolution process of the algorithm.
- Parent selection – this involves randomly selecting a set of individuals for recombination and mutation.
- Crossover and mutation – this is a process where selected parent individuals are recombined to create new “off-springs” or variants which inherit the better “genes” from their “parent individuals”. This is shown in Figure 3 (top and middle). To prevent premature convergence in the optimisation, mutation is used to randomly change the “gene” sequence of a design variant. It is based on a probability rate called the mutation rate which is shown in Figure 3 (bottom).
- Fitness function – this is used to evaluate the individuals in the population. It evaluates and selects for each generation, the individuals for recombination and mutation.

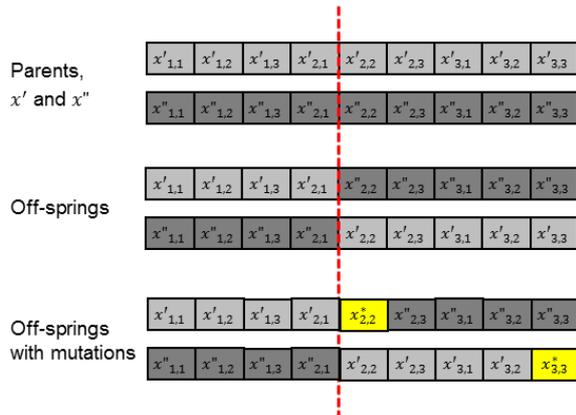


Figure 3. Graphical representation of crossovers.

Top and middle: where parent  $x'$  and  $x''$  are recombined to create two offsprings. Bottom:  $x'_{2,2}$  and  $x''_{3,3}$  are “genes” that are mutated.

The following evolutionary algorithm code was used in the experiments. It demonstrates how the components mentioned above are correlated:

randomly generate first  $n$  individuals,

calculate the fitness values of fitness function  $a$ ,  $b$ ,  $c$ ,

**Repeat**

**for** each generation

randomly select  $k$  number parents,

rank all selected  $k$  parents,

select first  $\frac{k}{2}$  of parents to recombine  $k$  off-springs and eliminate  $k$  selected parents,

randomly change genotype of off-spring with mutation rate,  $m$ ,

calculate the performance metrics for the new  $k$  off-springs,

**until** maximum number of generations,  $g$ .

## METHODOLOGY

### Performance Measures to Compare Algorithms

To compare the performance of MOAC with MOEA, a performance metric called the C measure, defined by Zitzler (1999) is used. C measure is the ratio of design solutions on one Pareto front that dominates another. Given two sets of solution  $X'$  and  $X''$  from the multi-objective optimisations, the C measure is defined as:

$$C(X', X'') = \frac{|\{a'' \in X''; \exists a' \in X': a' \geq a''\}|}{|X''|} \quad (6)$$

where  $C(X', X'') \in [0,1]$ . If the value of  $C(X', X'') = 1$ , it means that all points in  $X''$  are dominated by or equal to points in  $X'$ . Whereas, if  $C(X', X'') = 0$ , it means no points in  $X''$  are covered by  $X'$ . Both  $C(X', X'')$  and  $C(X'', X')$  are considered because  $C(X', X'')$  may not be equal to  $C(X'', X')$ .

Three variants of MOAC and MOEA as shown in Table 1 and 2 were created (for derivation see “Settings for Multi-objective Optimisation” below) and the best of each type were selected after using the C measure to analyse their performance. 10 optimisation runs were conducted for each variant of MOAC and MOEA. The C measures for ten sets of runs are later shown in Table 3 and 4.

To compare the speed of MOEA with MOAC, the best MOAC and MOEA, selected using the C measure as stated above, are ran on a computer with

a Dual-Core CPU of 3GHz and 4GB RAM. The average time of the optimisation runs for MOEA and MOAC are used for comparison. The results of the runtime are later shown in Table 5.

To compare the average time taken to complete each optimisation run, the total cumulative number of ants, for all iterations, used in each MOAC run and the total cumulative individuals, for all generations, used in each MOEA run are kept at 5,000.

### Settings for Multi-objective Optimisation

For MOAC, Arora (2011) recommended the number of artificial ants or design variants to be  $5n_v$  to  $10n_v$ , where  $n_v$  is the number of design variables or layers as shown in Figure 2. Arora (2011) also recommended a pheromone evaporation rate,  $p$ , between 0.4 and 0.8. These values were considered in the settings for the design variants.

Given an optimization run with a total of 5,000 individuals in total, MOAC 0 is first given the following settings where  $m = 100$ ,  $w = 50$  and  $p = 0.4$ . To find out the effects of  $p$ , MOAC 1 is given a value of  $m = 100$ ,  $w = 50$  and  $p = 0.8$ . To find out the effects of  $m$ , MOAC 2 is given the following settings where  $m = 50$ ,  $w = 100$  and  $p = 0.8$ . Table 1 summaries these three MOAC variants.

Table 1: Variants of MOAC

Variants	$m$	$w$	$p$
MOAC 0	100	50	0.4
MOAC 1	100	50	0.8
MOAC 2	50	100	0.8

Table 2: Variants of MOEA

Variants	$g$	$n$	$k$	$m$
MOEA 0	99	100	0.50	0.01
MOEA 1	62	100	0.80	0.01
MOEA 2	31	200	0.80	0.01

For MOEA, both Zitzler (1999) and Deb (2001) have used a crossover of  $k=0.8$  and a mutation rate of 0.01. Hence, MOEA 0 is given the following settings where  $g = 62$ ,  $n = 100$ ,  $k = 0.8$  and  $m = 0.01$ . To find out the effects of the crossover,  $k$ , MOEA 1 is given the following settings where  $g = 99$ ,  $n = 100$ ,  $k = 0.5$  and  $m=0.01$ . To find out the effects of the initial population size,  $n$ , MOEA 2 is given the following settings where  $g = 31$ ,  $n = 200$ ,  $k = 0.8$  and  $m = 0.01$ . Table 2 summaries these three MOEA variants.

All comparisons between variants of MOAC and MOEA were done using the C measure (Zitzler 1999). After comparing variants of MOAC and MOEA individually as shown in Table 3, the best performing MOAC and MOEA are compared using the C measures (see Table 4).

Both variants of MOAC and MOEA are applied to minimise ETTV, maximise electricity generation and maximise area with minimum daylight of 300 lx.

## RESULTS AND DISCUSSION

Comparing the performance of the three variants: MOAC 0, MOAC 1 and MOAC 2, Table 3 shows that MOAC 1 is the best-performing variant followed by MOAC 0. From this comparison, we can see that for MOAC, a bigger number of artificial ants improves the results of the Pareto front. In addition, a higher pheromone evaporation rate of 0.8 also improves the results of the Pareto front.

Table 3: Comparison of all variants of MOAC and MOEA. The min (minimum), avg (average) and max (maximum) of the C measure for the set of 10 runs for the MOAC and MOEA variants are presented.

Comparison of variants	C Measure		
	min	avg	max
MOAC 0, MOAC 1	0.066	0.267	0.467
MOAC 1, MOAC 0	0.226	0.421	0.566
MOAC 0, MOAC 2	0.469	0.875	1.000
MOAC 2, MOAC 0	0.000	0.152	0.583
MOAC 1, MOAC 2	0.773	0.489	0.981
MOAC 2, MOAC 1	0.000	0.049	0.131
MOEA 0, MOEA 1	0.189	0.304	0.436
MOEA 1, MOEA 0	0.327	0.469	0.613
MOEA 0, MOEA 2	0.084	0.229	0.338
MOEA 2, MOEA 0	0.360	0.506	0.762
MOEA 1, MOEA 2	0.223	0.313	0.464
MOEA 2, MOEA 1	0.307	0.391	0.432

Table 4: Comparison of best-performing MOAC and MOEA.

Comparison of variants	C Measure		
	min	avg	max
MOAC 1, MOEA 2	0.0167	0.131	0.266
MOEA 2, MOAC 1	0.226	0.557	0.867

Comparing the performance of the three variants of MOEA, MOEA 0, MOEA 1 and MOEA 2, Table 3 shows that MOEA 2 is the best-performing variant followed very closely by MOEA 1. With reference to MOEA 2, we can see that increasing the crossover

value of a MOEA improves the results of the Pareto front. However, a two-fold increase in the number of the initial population size from 100 to 200 does not yield a significant improvement in the Pareto front. Comparing the best-performing MOAC variant, MOAC 1 and best-performing MOEA variant, MOEA 2, Table 4 shows that MOEA 2 dominates MOAC 1 by an average of 0.557 whereas MOAC 1 dominates MOEA 2 by an average of 0.131. Figure 4 shows a three-dimensional plot with the Pareto fronts for an instance of MOAC 1 and MOEA 2. MOAC 1 dominates MOEA 2 with a few design variants (shown in blue) where MOEA 2 dominates MOAC 1.

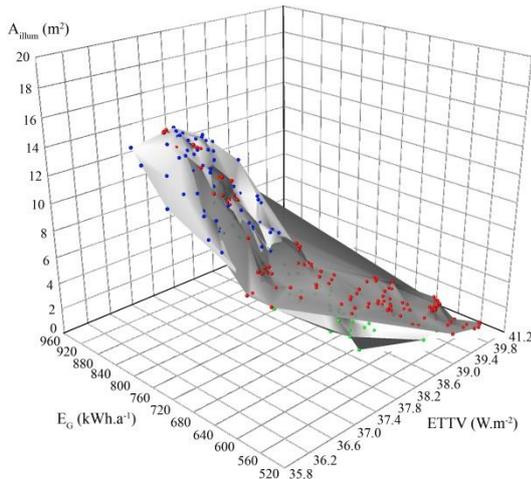


Figure 4. Pareto fronts of MOAC 1 and MOEA 2 with only a few design variants from MOAC 1 dominating MOEA 2, shown in blue. Design variants of MOAC 1 are shown in green and design variants of MOEA 2 are shown in red.

For multi-objective optimisation, the quality of the solutions along the Pareto front is determined by how close they get to the theoretical optimum where  $ETTV \rightarrow 0$ ,  $E_{pv} \rightarrow \infty$  and the area of minimum illuminance of 300 lx  $\rightarrow 16 m^2$ . Hence, if the average C measure for MOAC 1 is 0.131, it means that an average of 13.1% of the design variants on the Pareto front of MOAC 1 are closer to the theoretical optimum than those using MOEA 2. In addition, MOEA 2 has an average C measure of 0.557 where 57.7% of design variants on the Pareto front of MOEA 2 are closer to the theoretical optimum than MOAC 1.

The parallel plot for MOAC 1 in Figure 5 (top), has design variants with relative high ETTV (approximately 41.00 - 39.60  $W \cdot m^{-2}$ ). These design variants have a relatively high working plane area of minimum of 300 lx (16.0 - 9.5  $m^2$ ) and a relatively low electricity generation of (approximately 764 - 700  $kWh \cdot a^{-1}$ ). In addition, the parallel plot for MOAC 1 has design variants with relatively low ETTV (approximately 37.00 - 39.00  $W \cdot m^{-2}$ ), relatively low working plane area of minimum 300 lx (approximately 0.0 - 3.2  $m^2$ ) and a relatively high

electricity generation (approximately 800 - 880  $kWh \cdot m^{-2} \cdot a^{-1}$ ). This behaviour shows that a design with more active cell area, generates more electricity and reduces ETTV and daylighting because the PV cells are shading the sunlight which results in lower heat gain and reduced visible light transmittance.

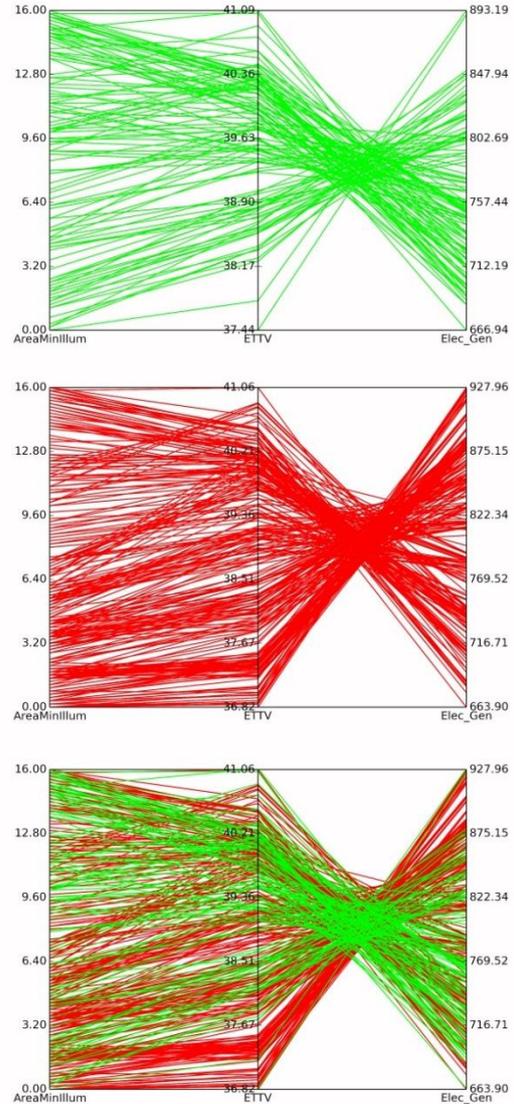


Figure 5 Top: Parallel plot of MOAC 1 shown in green. Middle: Parallel plot of MOEA 2 shown in red. Bottom: superimposed parallel plots of MOAC 1 and MOEA 2

Comparing the parallel plots of both Pareto fronts of MOAC 1 and MOEA 2, we can see that the design variants of both MOAC 1 and MOEA 2 occupy high and low regions of all the three fitness functions. This is seen in Figure 5, where both MOAC 1 and MOEA 2 have managed:

- to minimise ETTV which results in the increase in electricity generation and reduction in the area of minimum illuminance of 300 lx.
- to maximise the electricity generation which results in the reduction in ETTV and

reduction in the area of minimum illuminance of 300 lx.

- to maximise the area of minimum illuminance of 300 lx which results in the increase in ETTV and reduction in electricity generation.

However, from Figure 5 (bottom), we can observe that the design variants of MOEA 2 occupies a wider range for ETTV and electricity generation on the parallel plots when compared with MOAC 1. This means that MOEA 2 has managed to optimise the two individual objectives, ETTV and electricity generation, better than MOAC 1.

*Table 5: Comparison of the runtimes of all variants of MOAC and MOEA. The min (minimum), avg (average) and max (maximum) for the set of 10 runs for the MOAC and MOEA variants are shown.*

Variants	Runtime		
	min	avg	max
MOAC 1	36:28	36:42	37:00
MOEA 2	40:14	40:50	41:47

Table 5 shows that the averaged normalised time taken to run MOAC 1 is 36:42 hrs and 40:50 hrs for MOEA 2. This shows an improvement of 10.2% in the run when MOAC 1 is used.

We can conclude from the results discussed above that MOAC can be a good alternative, if the speed of running a multi-objective optimisation is more critical than the quality of the Pareto front which is determined by the C measure.

Although MOAC 1 dominates MOEA 2, the trend of both parallel plots for MOAC 1 and MOEA 2 are similar. However, MOEA 2 has been shown to find better design solutions, with 57.7% of its design solutions dominating those of MOAC 1.

## CONCLUSION

This paper has introduced the use of multi-objective ant colony (MOAC) in the multi-objective optimisation space of a semi-transparent BIPV façade. The experiment demonstrated that MOAC could be an alternative for multi-objective optimisations. It has shown reasonable improvements in speeding up the multi-objective optimisation process for BIPV façade design. However, future research needs to be conducted, to compare other variants of MOAC and MOEA to give more insights into the use of ant colony optimisation for multi-objective optimisation for building design optimisation.

## REFERENCES

Arora, J. 2011. Introduction to Optimum Design, Academic Press, Boston, MA.

Attia, S., Hamdy, M., O'Brien, W. and Carlucci, S. 2013, Assessing Gaps and Needs for Integrating

Building Performance Optimization Tools in Net Zero Energy Building Design, Energy and Buildings 60, 110-124.

BCA, 2004. Guidelines on Envelope Thermal Transfer Value for Buildings, Commissioner of Building Control, Singapore.

Caldas, L. 2008. Generation of Energy-efficient Architecture Solutions Applying GENE\_ARCH: An evolution based generative design system, Advanced Engineering Informatics 22:1, 59-70.

Charron, R. and Athientitis, A.K. 2006. The Use of Genetic Algorithms for a Net-zero Energy Solar Home Design Optimisation Tool, PLEA 2006 Conference, Geneva.

Choo, T.S Ouyang, J., and Janssen, P. 2013. Multi-objective Optimisation of Semi-transparent Building Integrated Photovoltaic Façade, Proceedings for Sustainable Building Conference 2013.

Deb, K. 2001. Multi-objective Optimization Using Evolutionary Algorithms, John Wiley & Sons Ltd, England.

Dorigo, M. 1992. Optimization, learning and natural algorithms. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy.

Dorigo, M. and Stutzle, T. 2004. Ant Colony Optimization, MIT Press, Massachusetts.

Eiben, A. E. and Smith, J.E. 2007. Introduction to Evolutionary Computing, Springer, Heidelberg.

Fung, T.Y.Y. and Yang, H. 2008. Study on Thermal Performance of Semi-transparent Building-Integrated Photovoltaic Glazings, Energy and Buildings 40:3, 341-350.

Robinson, L.E. and Athienitis, A.K. 2009. Design methodology for optimisation of electricity generation and daylight utilization for facades with semi-transparent photovoltaics, Proceedings Building Simulation 2009.

Rutten, D., 2011, Evolutionary Principles Applied to Problem Solving, available from <http://ieatbugsforbreakfast.wordpress.com/2011/03/04/epatps01/>, accessed on 1<sup>st</sup> December 2013.

Sidefx 2013. Houdini 3D Animation Tools, available from <http://www.sidefx.com/>, accessed on 1<sup>st</sup> December 2013.

Wang, W., Zmeireanu, R. and Rivard, H. 2005. Applying Multi-objective Genetic Algorithms in Green Building Design Optimisation, Building and Environment 40:11, 1512-1525.

Ward, G. L. and Shakespeare, R. A. 1998. Rendering with Radiance – The Art and Science of Lighting Visualization, Booksurge LLC, Charleston.

Zitzler, E. 1999. Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. PhD Dissertation, ETH Zurich

