PAPER • OPEN ACCESS

Comparison of simulation-based methods and metaheuristic optimization algorithms for optimizing window design by considering daylighting and heat transfer in a tropical region of Indonesia

To cite this article: A Budhiyanto *et al* 2021 *IOP Conf. Ser.: Earth Environ. Sci.* **907** 012016

Comparison of simulation-based methods and metaheuristic optimization algorithms for optimizing window design by considering daylighting and heat transfer in a tropical region of Indonesia

A Budhiyanto[1,2,6], A Oktavianus[3], B Tedjokusumo[4], K Harsono[4] and I T Yang[5]

[1] Department of Architecture, National Taiwan University of Science and Technology, Taipei, Taiwan
[2] Department of Architecture, Petra Christian University, Surabaya, Indonesia
[3] Department of Civil Engineering, National Taiwan University, Taipei, Taiwan
[4] Department of Architecture, National Taiwan University of Science and Technology, Taipei, Taiwan
[5] Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan

[6] Corresponding author: arisb@petra.ac.id

Abstract. This study presents evaluation and comparison of simulation-based methods and metaheuristic optimization algorithms on building design models, focussing on daylight availability maximization and energy consumption minimization. The simulation-based method was presented using Rhino/Grasshopper software supported by the Ladybug, Honeybee, and Octopus optimization plugins; while MOPSO was chosen to calculate the metaheuristic optimization algorithm. The result indicated that OTTV values of the optimum design were respectively in the range of 24.06 W/m^2 to 34.15 W/m^2 for Octopus optimization and 25.19 W/m^2 to 34.99 W/m^2 for MPSO; and the WWR value for Octopus optimization and MOPSO were in the range 15% to 23% and 15% to 26%, respectively. While both methods showed similar results, the time duration for simulating in Rhino/Grasshopper was much longer compared to calculating the algorithm using MATLAB, indicating that simulation-based was less effective.

1. Introduction

United Nation Environment Programme [1] stated that buildings and the construction sector accounted for 36% of final energy use and 39% of energy and process-related carbon dioxide (CO_2) emissions. The highest energy consuming component in a building is Heating, Ventilation, and Air Conditioning (HVAC) systems, which is affected by the indoor temperature setpoint, air infiltration, window type, window wall ratio [2, 3]. Budhiyanto [4] stated that increasing 10% window area will result in about 5.67% cooling energy usage and 0.35^{0}C temperature increase. However, reducing the window areas will affect less daylight penetration into buildings. As glazing systems influence building lighting requirements and energy usage as the solar radiation that enters the building, window design is essential for optimally calculated to reduce energy consumption in a building.

Optimizing the window design by looking at increasing the size of the window will benefit the availability of daylight but increasing the amount of energy lost through it is a challenging task [2, 5]. Window design optimization considers many factors and constraints, especially in tropical countries that have excessive sunlight. Several studies have explored the optimization of windows by calculating the thermal gain, wind pressure, solar radiation, and the shape and slope of the facade against the window. Qingsong and Fukuda [6] found the optimum window area on each different orientation wall which minimizes the energy consumption and maximises the useful daylight illuminance using simulation-based methods. The same method is also used by Lukmanto [7] to select a type of glass façade that optimizes both energy saving and construction cost, considering solar radiation and wind pressure. Cheng et al. [8] investigated the optimization of daylight and energy performance for residential buildings, with building orientation, wall and glazing properties and window areas as the parameters. A study conducted [9] in Iran which has four climates stated that the optimum window area for north, south and east and west building façade were about 20-30%, 20-50% and 20-70%, respectively, and the energy saving was around 20-100% and 16-25% based on the local climate; while in Australia [10] achieved up to 8.57% energy saving by applied 20-40% window area on building envelopes.

It was apparent that previous studies have focused on optimizing window designs based on daylight and heat gain mostly in sub-tropical countries using a simulation-based method, carried out in Revit/Dynamo and Rhino/Grasshopper simulation software. However, a simulation-based method is not the only way for solving optimization problems. Metaheuristic optimization algorithms emerged as another option for solving optimization problems [11]. The use of metaheuristics in the solution of optimization problems is impressive with very important improvements and new algorithms being proposed every day [12]. Several metaheuristic applications for optimization in buildings have been carried out in calculating window design, building energy performance, and indoor thermal [13, 14].

While the window design optimization is mostly done using simulation software or manual calculations, both the simulation-based and metaheuristic algorithm methods are used to optimizes window designs, including window-to-wall ratio (WWR) and gazing properties, such as glass U value and visual transmittance value (VT). The objective of this study is to investigate the optimum design for minimizing energy consumption as well as maximizing daylight penetration by comparing two different methods.

2. Methodology

2.1. Optimization framework

The stages carried out in this study are divided into four stages. The first stage is identifying the design variables to be examined and building a design model followed by developing the daylight and energy analysis model for simulation as well as the objective function for metaheuristic optimization. The second stage is the simulation work stage using Rhino/Grasshopper software. The third stage is the metaheuristic analysis process using Multi-Objective Particle Swarm Optimization. This stage proceeds in parallel with the second stage. The fourth stage is to compare and discuss the results obtained to obtain conclusions. Figure 1 illustrates the framework of this study according to the explanation of the four stages.

2.2. Case study

In this study, the design problem is considered a simple box shape building with 6m length x 5m width x 4m height, located in Jakarta, Indonesia. The building has one window on the south façade without a shading device (Fig 2). The wall material has U value = 1.7152 Watt/m^{2}K and α = 0.86, while the window area and glazing properties will be obtained through simulation and calculation.

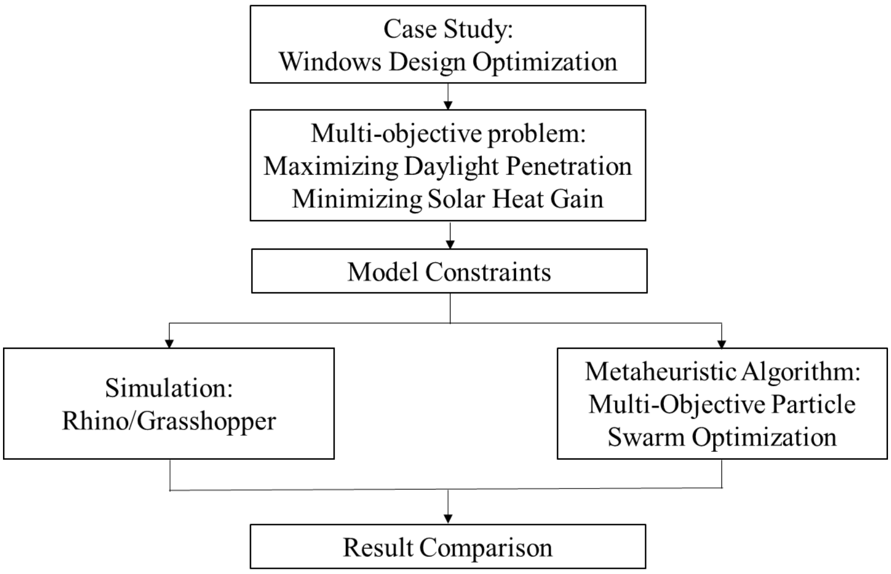

Figure 1. Optimization framework.

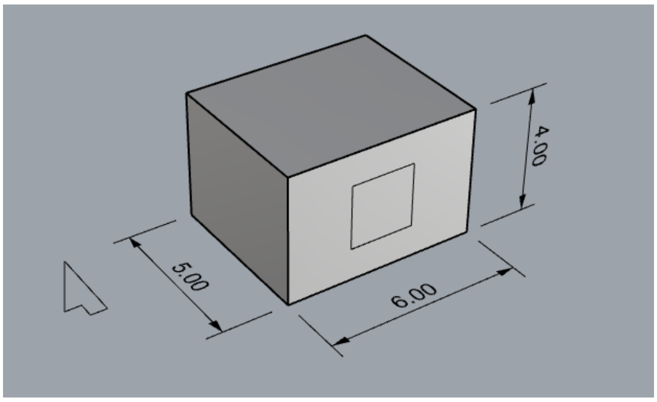

Figure 2. A simple box shape model building.

2.3. Optimization problem

Two optimization objectives, which are minimum Overall Thermal Transfer Value (OTTV) and maximum Daylight Autonomy (DA) represent the optimal energy and daylight performance in simulation-based methods. However, in a metaheuristic optimization algorithm, Effective Aperture (EA) is used to measure the daylight availability replacing DA due to the complexity of DA calculation. Based on building envelope design regulation in SNI [15], which refers to the ASHRAE standard, the OTTV equation is:

$$\mathrm{OTTV} = \alpha\ [(\mathrm{UW} * (1 - \mathrm{WWR}) * \mathrm{TDek}] + (\mathrm{Uf} * \mathrm{WWR} * \Delta\mathrm{T}) + (\mathrm{SC} * \mathrm{WWR} * \mathrm{SF}) \quad (1)$$

OTTV = overall thermal transfer value (Watt/m^2)
α = wall color absorbance
UW = wall thermal transmittance (Watt/m^2K)
WWR = window-to-wall ratio (%)
TDek = equivalent temperature difference (K)
Uf = glazing thermal transmittance (Watt/m^2K)
ΔT = highest indoor-outdoor temperature difference (K)
SC = shading coefficient

SF = solar radiation factor (W/m^2)

In this model, only the south wall façade has a window without a shading device, therefore the OTTV calculation only considers south wall heat gain. As wall thermal properties have been determined (Uw = 1.7152 Watt/m^{2}K, α = 0.86, TDek =10), the highest indoor-outdoor temperature difference in Indonesia is around 5^{0}C and south wall solar radiation factor is 97 W/m^2, so the equation of metaheuristic optimization algorithm can be simplified as:

$$\mathrm{OTTV} = 14.75 + \mathrm{WWR} * \mathrm{Uf} * 411.25 \tag{2}$$

Daylight autonomy (DA) is the percentage of occupied time when an illuminance threshold (300 lux) can be met by daylight alone under continuous overcast sky conditions throughout the year. Simulation-based or numerical-based methods are usually considered to measure daylight availability [16]. However, because the DA calculation is very complex, the metaheuristic optimization uses the Effective Aperture (EA) algorithm to calculate daylight availability. EA is defined as the product of WWR and VT (Equation 3) and is often used as an indicator of daylight availability [10]. Although DA and EA have different calculation methods, both have the same objective, which is to measure daylight availability in the buildings.

$$\mathrm{EA} = \mathrm{VT} * \mathrm{WWR} \tag{3}$$

EA = effective aperture (%)
VT = glazing visual light transmittance
WWR = window-to-wall ratio (%)

2.4. Model constraints

Three types of constraints are defined in this optimization framework. The first constraint defined the glazing material properties, which are glass U value and visual light transmittance. The second constraint defined the WWR range to prevent overglazing. The third constraint is the maximum OTTV allowed.

In this study, the properties of glazing material are determined based on Stopsol glass created by AGC Glass Europe. The glass Uf adopted can be as low as 1.25 Watt/m^{2}K and go up to 6.45 Watt/m^{2}K, while the glass VT adopted can vary from 0.15 to 0.75.

The glazed area restriction is determined based on the window-to-floor ratio (WFR) [9]. Equation 4 shows the restriction of the glazed area by setting lower and upper bounds for the WFR, which are 10% and 50%, respectively.

$$\underline{\mathrm{WFR}} \le \frac{\mathrm{WWR}*\mathrm{A}}{\mathrm{F}} \le \overline{\mathrm{WFR}} \tag{4}$$

$\underline{\mathrm{WFR}}$ = minimum window-to-floor ratio (%)
WWR = window-to-wall ratio (%)
A = wall area (m^2)
F = floor area (m^2)
$\overline{\mathrm{WFR}}$ = maximum window-to-floor ratio (%)

Based on Indonesia National Standard [15], the maximum OTTV allowed is 35 W/m^2. Therefore, the OTTV results obtained higher than 35 W/m^2 are considered violating the constraint and a penalty leading to termination of its evaluation process will be applied.

3. Simulation and metaheuristic optimization

3.1. Rhino/ Grasshopper simulation

In architectural research, optimization of sustainable design can be done through numerical-based and simulation-based methods. In terms of aperture optimization, both optimization methods can be performed. BIM software packages, such as Rhino/ Grasshopper and Revit/ Dynamo, are often used to perform this optimization simulation with the presence of plugins, such as the Octopus and Galapagos plugins in the Rhino software [17]. The simulation framework is shown in figure 3.

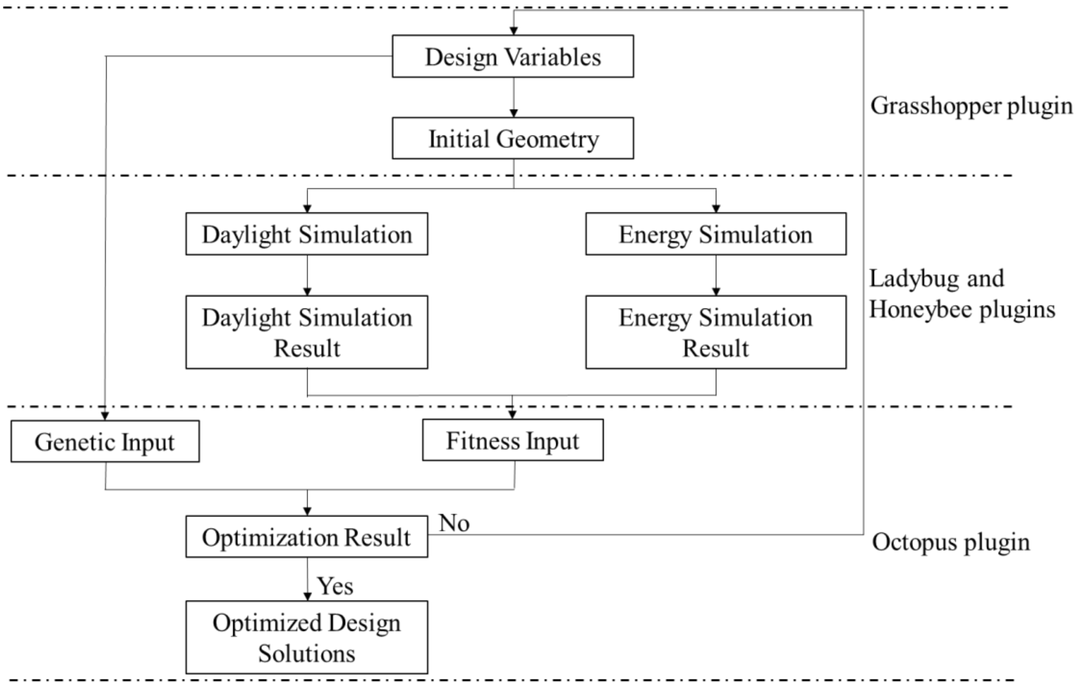

Figure 3. Simulation process and tools.

The model geometry is built in Rhinoceros (Rhino) software with Grasshopper plugin. Grasshopper, a parametric modelling plugin for Rhinoceros 3D modelling software developed by David Rutten at Robert McNeel & Associates in 2007 has the capability to quickly generate parametric shapes based on generative algorithms. Grasshopper plugins Ladybug and Honeybee are used to run daylight and energy simulation based on Radiance and EnergyPlus program, Octopus, a multi-objective optimization plugin is applied to perform the optimization. The genetic input of Octopus is connected to the design variables, while Fitness input is connected to daylight and energy optimization objectives. If the optimization result is terminated, Octopus will execute it and if it is not terminated, Octopus will automatically reset the design variables values for the next design option and continue the simulation [18].
In Octopus, the setting of population size, crossover rate, mutation rate, mutation probability and elitism are 100, 0.8, 0.09, 0.2 and 0.5, respectively. Since Octopus can only solve minimization problems, the value of DA is multiplied by -1. Jakarta, a major city in the tropical region of Indonesia is chosen for simulation. The climate condition of Jakarta is shown in Table 1.

Table 1. Average hourly global horizontal radiation, air temperature and relative humidity.

Month	**Global Horizontal Radiation (Wh/sq.m)**	**Air Temperature (^{0}C)**	**Relative Humidity (%)**
Jan	557	28	78
Feb	578	28	76
Mar	617	28	77
Apr	663	29	73
May	624	29	72
Jun	617	29	68
Jul	616	28	69
Aug	682	29	67
Sep	693	29	65
Oct	646	29	68
Nov	617	29	71
Dec	601	28	75

3.2. Metaheuristic optimization algorithm
Kennedy and Eberhart [19] first introduced Particle Swarm Optimization (PSO) as a simplified algorithm based on the movement of organisms in a flock of birds or schools of fish for optimization. The algorithm of PSO describes a possible solution as a particle that moves through the search area in a certain variable domain range and evaluates each particle to get a fitness value. Conceptually, PSO records the best individual solution of a particle as pBest and the best global solution among all particles as gBest to determine how each particle moves in the next iteration. The movement of the particles is carried out in a set number of iterations. In each iteration, the algorithm evaluates each solution and updates the pBest and gBest values if there is a better solution.
Multi-Objective Particle Swarm Optimization (MOPSO) is heuristically extended by PSO to effectively deal with multi-objective optimization problems [20, 21]. The approach of MOPSO uses the Pareto dominance concept to determine the flying direction of the particle and maintains the non-dominance vector found previously in the global repository which is then used by other particles to guide their flight, then validated using several standard test functions from the specialized literature.

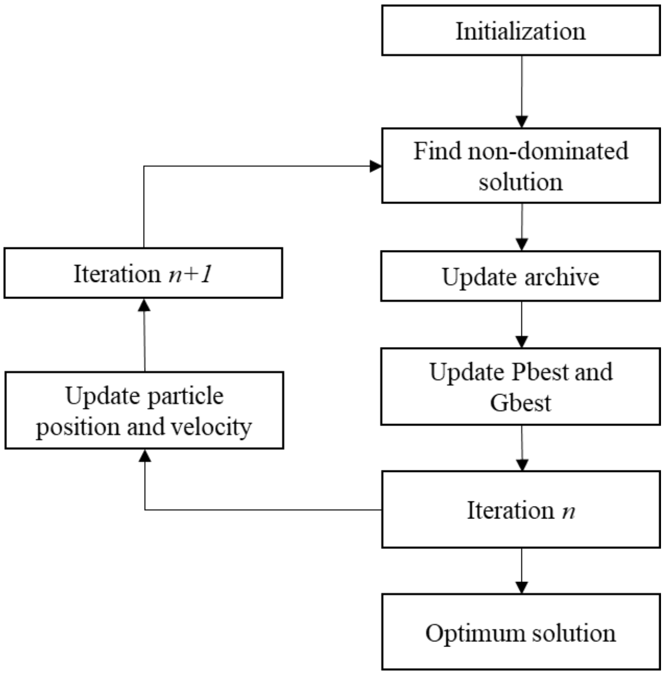

Figure 4. MOPSO process.

3.3. Setting MOPSO
From figure 4 above, it can be seen that the first step of MO-PSO is generated the initial population by randomizing from the range of the upper bound and the lower bond. In this study, a population of 100 particles was determined. The initial velocity is also randomized determined, by the range of 0-1. Each particle fitness value is calculated using the objective function, which is the minimum value of equation (2) above and also the maximum value of equation (3). The result is analysed by its position, to see whether the solution is being dominated by other solutions. The particle that is not being dominated by other particles will be stored in an archive.
In this study, the number of archives is 20 particles. The Gbest and Pbest is also selected for each iteration. 100 iteration is used to reach the optimum solution. For each iteration, the velocity is updated using the single objective PSO formula. The inertia weight is determined before which is 0,3. The solution should maintain the speed, to keep the solution from reaching global optima too quickly. Therefore, maximum velocity is needed. For this study, the maximum velocity is 0,1 of the range. The

PSO algorithm requires the solution inside the boundary, so if there any particle go beyond the boundary, adhere strategy will be implemented. After the algorithm set, each particle is evaluated for determining the Pbest, which is if the new solution dominated the old one, the new solution would replace the current Pbest. When the particle updates its velocity, a random number is generated, and if the number is bigger than the pre-determined threshold, which is 0,15, the velocity will change by a random value. Later, each particle will be evaluated according to the dominance status, if there is any dominated solution in the archive or the Pareto front, the solution will be eliminated. Finally, Gbest will be selected randomly among the Pareto front.

4. Result and discussion

4.1. Rhino/ Grasshopper simulation

The simulation is run using a computer with AMD Quad Core R5-2500U, 3.6 GHz, 8 GB memory. It takes 100 - 150 seconds to run one simulation. In this simulation, 100 generations with 100 populations in each generation are simulated, so in total the time needed for simulating all generations is around 277 - 416 hours.

The Pareto front solutions of generations 1, 25, 50, 75 and 100 are shown in Figure 5. It's obvious after generation 25, the positions of Pareto front doesn't change and it's already converged. Since Octopus optimization plugin is only able to solve minimization, the maximization DA objective shown is multiplied by -1.

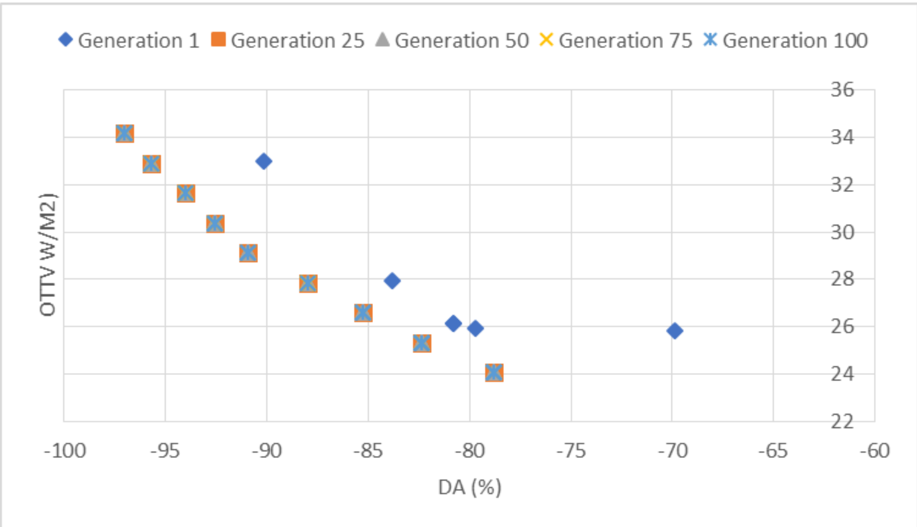

Figure 5. Pareto front solution of generation by Rhino/Grasshopper.

The minimum OTTV value is 24.06 W/m^2 obtained by presenting 15% WWR, while the maximum DA value is 97.02%, obtained by presenting 23% WWR. The balanced design option is obtained when OTTV and DA values are 29.11 W/m^2 and 90.75%, respectively, by presenting 23% WWR. Table 2 shows the design variable values and optimization objectives of the best design option.

Table 2. Design variable values and optimization objectives of best design options.

	Optimization objectives		Design variables		
	OTTV (W/m^2)	DA (%)	WWR (%)	Uf (Watt/ m^{2}K)	VT
Min OTTV	24.06	78.78	15	1.25	0.75
Max DA	34.15	97.02	23	1.25	0.75
Balanced	29.11	90.75	19	1.25	0.75

4.2. Metaheuristic optimization algorithm

Matlab software is used to run the metaheuristic algorithm. In a computer with the same specification, Matlab need 12 – 15 seconds to run the algorithm with 100 generations, each one has 100 populations. The Pareto front solutions of generations 1, 25, 50, 75 and 100 are shown in Figure 6. It's obvious after generation 25, the positions of the Pareto front doesn't change and it's already converged.
The minimum OTTV value is 25.19 W/m^2 obtained by presenting 15% WWR, while the maximum EA value is 19.26%, obtained by presenting 23% WWR. The balanced design option is obtained when OTTV and DA values are 30.02 W/m^2 and 15.20%, respectively, by presenting 26% WWR. Table 3 shows the design variable values and optimization objectives of the best design option.

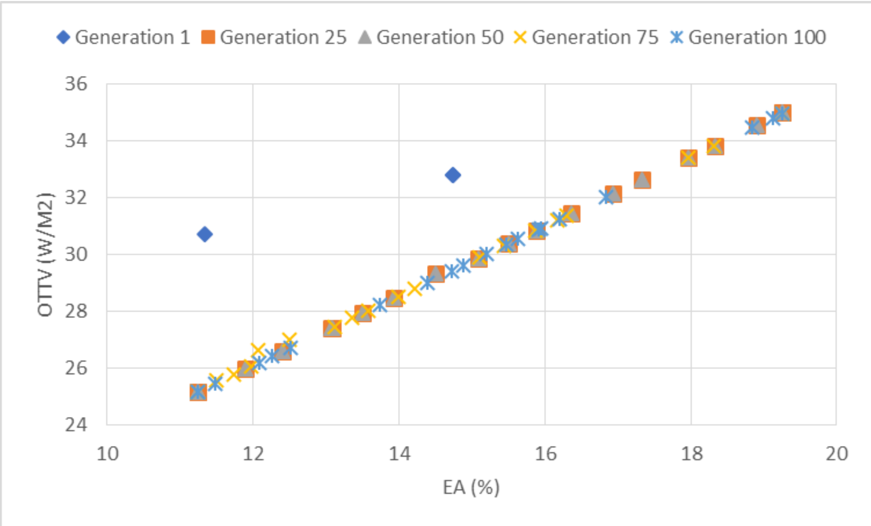

Figure 6. Pareto front solution of generation by MOPSO.

Table 3. Design variable values and optimization objectives of best design options.

	Optimization objectives		Design variables		
	OTTV (W/m^2)	DA (%)	WWR (%)	Uf (Watt/m^{2}K)	VT
Min OTTV	25.19	11.25	15	1.25	0.75
Max DA	34.99	19.26	26	1.25	0.75
Balanced	30.02	15.20	20	1.25	0.75

4.3. Metaheuristic optimization algorithm

Table 2 and 3 shows that the result of Octopus optimization and MOPSO is slightly different for OTTV and WWR values but for Uf and VT values are the same. The difference can be explained as Rhino/ Grasshopper simulating and calculating more factors than the objective function equations in MOPSO. However, both results are still in line, indicating that both methods are quite reliable. The optimum design options presented WWR in the range 15% to 26%, Uf is 1.25 (Watt/m^{2}K) and VT is 0.75. The WWR value
The design variables of best design options indicated that the WWR plays an important role to maintain daylight availability as well as energy consumption, compared to Uf and VT.
Comparing the running time of both Rhino/Grasshopper simulation and Matlab algorithm calculation, it's obvious that the simulation based method consumes much more time. Therefore metaheuristic optimization algorithm is more effective to solve an optimization problem with simple objective functions.

5. Conclusion

This study focuses on evaluating and comparing simulation-based methods and metaheuristic optimization algorithms on building design models. Daylight availability maximization and energy consumption minimization were selected as multi-objective optimization. The simulation-based method was presented using Rhino/Grasshopper software supported by the Ladybug, Honeybee and Octopus optimization plugins. MOPSO was chosen to calculate the metaheuristic optimization algorithm.

This study uses OTTV and DA to measure energy and daytime performance for Octopus optimization, while EA replaces DA for MOPSO optimization. The results of Octopus and MPSO optimization show similar values. The OTTV values are in the range of 24.06 W/m^2 to 34.15 W/m^2, respectively, for Octopus optimization and 25.19 W/m^2 to 34.99 W/m^2 for MPSO. The DA value for Octopus optimization is in the range of 78.78% to 97.02%, while the EA value for MOPSO is in the range of 11.25% to 19.26%.

As for design variables, the WWR value for Octopus optimization and MOPSO is in the range 15% to 23% and 15% to 26%, respectively. While both methods show the same value for Uf of 1.25 Watt/m^{2}K and VT of 0.75. Among the design variables, it is evident that WWR is the most important variable affecting the energy and daylight performance of the building.

Based on the running time, the simulation based method needs much more time compared to metaheuristic optimization algorithm, indicating that the latter is more effective to solve an optimization problem with simple objective functions.

References

[1] United Nations Environment Programme 2019 *Global Status Report for Buildings and Construction Sector: Towards a zero-emission, efficient and resilient buildings and construction sector* p 3
[2] Pérez-Lombard L, Ortiz J and Pout, C 2008 *Energy Build.* **40** 394
[3] Ruparathna R, Hewage K and Sadiq R 2016 *Renewable and Sustainable Energy Rev.* **53** 1032–1045.
[4] Budhiyanto A 2017 *Dimensi J. Archit. Build. Environ.* **44** 189-194
[5] Lin H-W and Hong T. 2013 *Appl. Energy* **111** 515–528.
[6] Ma Q and Fukuda H 2016 *Procedia Soc. Behav. Sci.* **216** 818–828.
[7] Lukmanto C M E 2020 *Multi-objective Optimization of Facade Glazing: Considering Solar Radiation, Wind Pressure, and Cost Estimation Analysis* (Taipei) National Taiwan University of Science and Technology
[8] Cheng C Ninic J and Tizani W 2019 Parametric Virtual Design-Based Multi-Objective Optimization for Sustainable Building Design *EG-ICE 2019 Workshop on Intelligent Computing in Engineering* (Leuven, Belgium)
[9] Shaeri J, Habibi A, Yaghoubi M and Chokhachian, A 2019 *Environ.* **6(4)** 1-16
[10] Chen Z, Hammad A W A, Kamardeen I and Haddad A 2020 *Buildings* **10(11)** 206
[11] Yang X S 2011 Metaheuristic Optimization: Algorithm Analysis and Open Problems *SEA 2011* LNCS 6630 21–32
[12] Dokeroglu T, Sevinc E, Kucukyilmaz T and Cosar A 2019 *Comput. Ind. Eng.* **137** 106040
[13] Delgarm N, Sajadi B, Kowsary F and Delgarm S 2016 *Appl. Energy* **170** 293–303
[14] Zhai Y, Wang Y, Huang Y and Meng X 2019 *Renewable Energy* **134** 1190–1199
[15] Standardization Agency of Indonesia 2011 SNI 6390:2011 Energy conservation of the building's air-conditioning system (Jakarta, Indonesia)
[16] Acosta I, Campano M A, Domínguez S and Fernández-Agüera J 2019 *LEUKOS - J. Illum. Eng. Soc. N. Am.* **15(3)** 1-19
[17] Touloupaki E and Theodosiou T 2017 *Energies* **10** 637
[18] Fang Y and Cho S 2019 *Sol. Energies* **10** 1-18
[19] Kennedy J and Eberhart R 1995 Particle swarm optimization *Proc. International Conference on Neural Networks* **4** 1942–1948

[20] Parsopoulos K E and Vrahatis M N 2002 Particle swarm optimization method in multiobjective problems *ACM symp. on Applied computing* (New York: Association for Computing Machinery) p 603–607
[21] Coello Coello C A and Lechuga M S 2002 MOPSO: a proposal for multiple objective particle swarm optimization *Proc. Evolutionary Computation IEEE Computer Society* **2** 1051–1056