Energy and Buildings 127 (2016) 714–729

Contents lists available at ScienceDirect

Energy and Buildings

journal homepage: www.elsevier.com/locate/enbuild

Improving evolutionary algorithm performance for integer type multi-objective building system design optimization

Weili Xu*, Adrian Chong, Omer T. Karaguzel, Khee Poh Lam

Center for Building Performance and Diagnostics, Carnegie Mellon University, Pittsburgh, PA 15213, USA

A R T I C L E   I N F O

Article history:
Received 16 March 2016
Received in revised form 14 June 2016
Accepted 14 June 2016
Available online 21 June 2016

Keywords:
Multi-objective evolutionary optimization
Building system design
Building cost estimation
Optimization performance

A B S T R A C T

Building system design optimization is becoming popular for design decision making. State-of-the-art technique that couples evolutionary algorithms with a building simulation engine, which is time consuming and often cannot reach the "true" optimal solutions. Studies addressing these issues focus on implementing strategies such as fine tuning optimization algorithm's parameters, hybrid evolutionary algorithms with a local search algorithm or optimizing meta-models. Unlike the previous studies, this paper proposes two improvement strategies for building system design optimization. The two strategies, adaptive operators approach and adaptive meta-model approach, modify the behaviors of conventional evolutionary algorithms to improve the optimization convergency and speed performance. To demonstrate the effectiveness of these two strategies compared to conventional algorithms, a case study was conducted. The case study observed high convergency performance from both strategies with 30% and 60% time savings respectively. Furthermore, this study examines the performance comparison in respect to convergency, diversity preservation and speed between these two strategies.

1. Introduction

In building design process, decisions are usually constrained by multiple factors. Solutions that satisfy building owners' requirements are commonly infeasible to reach by performing parametric studies. Therefore, more efficient methods are needed in searching the solution space to find optimal solutions, which not only reduce the evaluation time, but also provide optimal designs to achieve building owners' investment goals. This raises the topic of building system design optimization (BSDO) that uses advanced optimization algorithms for searching for optimal design solutions. The topic has increasingly drawn attention from the academic community, as well as the architecture, engineering and construction (AEC) industry. Current studies well address the behavior of various optimization algorithms including pattern search methods and stochastic methods [1]. These studies have also extensively examined various objectives and levels of detail regarding single system performance optimization and integrated building design optimization [2]. In recent years, both academia and industry are developing tools that support the ease of implementing optimization in building design process. Tools such as MOBO [3], GenOpt [4] and jEPlus [5] provide user-friendly interface and capability of coupling building design evaluation toolset (TRNSYS, EnergyPlus, etc.) as well as allow stakeholders to explore their design options more effectively.

Although BSDO has been actively discussed among academic community for decades, it is still not a common technique used in today's typical building projects. One of the barriers is computation time. A typical BSDO process could take days to find optimal solutions. In order to reduce the computation time, many researchers are looking for computational efficient strategies which can fully utilize computational resources to boost the optimization speed [6]. Although these researches did not address the computationally expensive design evaluation process in BSDO, the strategies proposed can effectively alleviate the impact of evaluation speed.

Such strategies can be categorized into three types, namely: parallel computing, model simplification and meta-model approaches [1]. Parallel computing allows optimization algorithms distributing a number of simulation tasks into multiple process threads simultaneously, thus reducing the overall computation time. Implementation of this approach requires advance level of programming skills, nevertheless, the majority of current energy simulation software have included such features. Simplifying the complexity of problems is another popular approach that frequently appears in many studies. Such studies usually construct a simple geometry layout with small amount of design variables. However, simplification highly relies on expert knowledge and designers have to take risks for losing building system interaction

* Corresponding author.
E-mail address: weilix@andrew.cmu.edu (W. Xu).

http://dx.doi.org/10.1016/j.enbuild.2016.06.043
0378-7788/

information, which may result in sub-optimal solutions. Lastly, a meta-model approach that optimizes design parameters on a "model of model" instead of real simulations can effectively find optimal solutions in negligible time [7]. However, this method requires a pre-computed database that contains design variables and parameters for constructing the meta-model. Therefore, it demands hundreds of energy simulations upfront. Furthermore, a single meta-model is typically not general enough to adapt to different cases such as evaluating energy consumption for a building in different climates.

In addition, literature review indicates that current practice has no indication of optimization convergency, thus it is hard to examine whether the optimal solutions on pareto front curve are the "true" optimized solutions. Secondly, the expensive computational power of building energy simulation largely slows down the optimization process, thus performing optimization studies are not feasible at practice point of view. However, from these past studies, some unique characters of evolutionary optimizations in building designs are suggested:

- A number of design solutions may appear in multiple generations.
- Algorithm performance strongly depends on parameter settings of operators.
- A large amount of energy simulations are produced in every generation and they are discarded in the next iteration.
- Energy simulations in every generation are mainly used for providing search directions towards the optimal region.

Utilizing the findings, this study proposes two separate improvement strategies that change the behavior of the conventional evolutionary algorithm. These two strategies focus on reducing the number of building energy simulations at algorithm level and achieving better optimal solutions. The first strategy relates to optimize operator's parameters setting by employing adaptive strategy. This strategy transforms the algorithm's behavior by updating its operators' parameters adaptively based on the current generation performance so that the algorithm could dynamically optimize its search power. The second strategy proposes a dynamic meta-model based multi-objective optimization procedure. The idea of this procedure is to marry machine learning techniques with optimization procedure in order to enhance the algorithm's searching power. The above building optimization characteristics imply that optimization algorithms can potentially reuse all energy simulations evaluated in the previous generations for constructing a meta-model and employ this meta-model to adjust and advance its exploration direction and speed. In addition, the meta-model should be capable of self-update along with optimization process for refining its prediction power. Implementing these two strategies could reduce optimization time as well as improve the convergency of optimal solution set. The detail implementation and their performance in BSDO of these two strategies will be discussed in this paper.

2. Method

2.1. Multi-objective optimization

BSDO is a complex problem and it is typically solved by evolutionary algorithms with multiple conflicting objectives. Pareto optimality is a frequently used method for analyzing BSDO optimization results [2]. This method introduced a set of design solutions as optimal solution set. In this solution, a unique situation occurs where a single objectives adversely affects other objectives. Fig. 1 shows data plot results of a typical optimization study. It can be observed that there is no solution in this optimal solution set (purple dots), which has both lower first and operation costs than any other solutions in the same set.

A typical workflow for BSDO is summarized in Fig. 2. The study objectives are goals that clients would like their buildings to achieve through optimization. Fitness functions are the functions that perform energy simulations and post-process simulation outputs with respect to defined objectives. Design options are a limited set of building system designs that clients want to test on their properties. With defined objectives, fitness functions and design options, the optimization algorithm can be initialized. An initialization process usually involves setting the algorithm's parameters such as mutation probability and crossover probability in genetic algorithms. Results generated in the process indicate that the fundamental differences between single objective optimization and multi-objective optimization is lying in the cardinality of the optimal set. Although clients' need only one solution to their problem, this is natural behavior of multi-objective optimization with conflicting objectives because there is no single solution that performs better at every objective than any other solutions. Since a

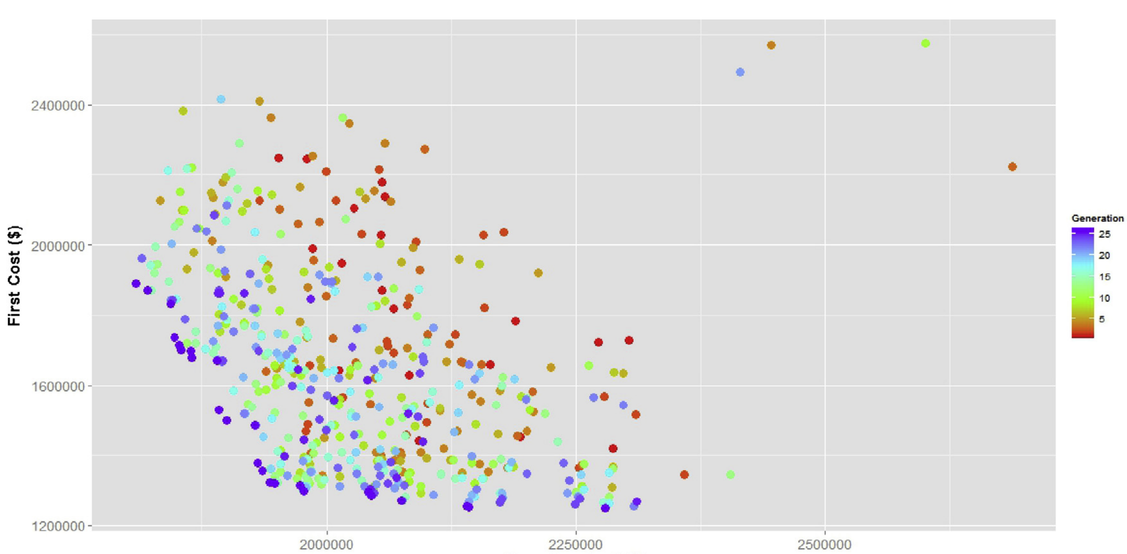

Fig. 1. A typical MOO solution plot with pareto front curve. (For interpretation of the references to color in text near the reference citation, the reader is referred to the web version of this article.)

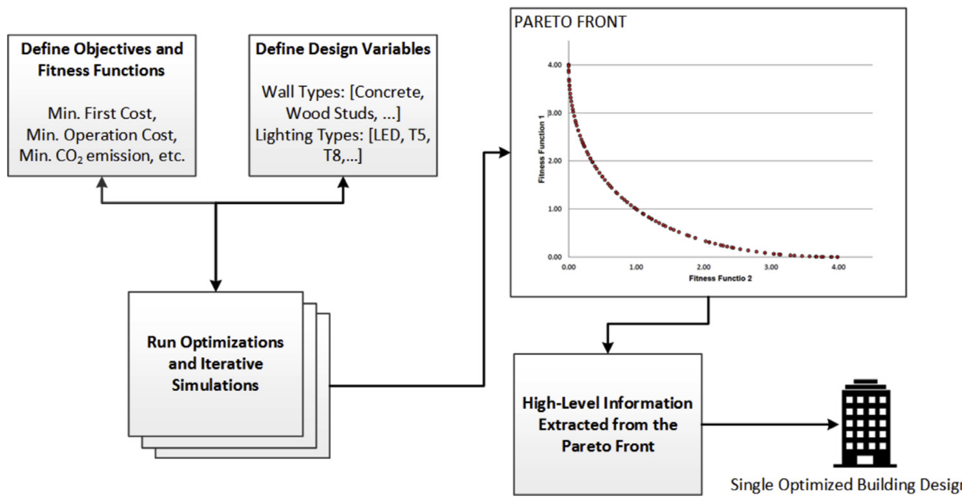

Fig. 2. Typical building design multi-objective optimization process.

number of solutions are optimal, higher-level information, which typically are non-technical and unquantifiable, should be applied in the process [8]. Methods such as listing pros and cons for each optimized solutions and checking the feasibility of the generated design combinations, can be implemented to finalize the design.

It is common that evolutionary multi-objective optimization (EMO), which searches for the optimal set based on generations of solutions, are employed in solving BSDO problems [1]. These population-based meta-heuristic algorithms including nondominanted sorting genetic algorithm II (NSGA-II), pareto-archived evolution strategy (PAES) and particle swarm optimization (PSO), have been found in a number of building optimization studies [1,9]. Although these algorithms are proven to be effective in generating a set of optimal solutions, the process usually takes a number of energy simulations at each generation, which greatly increases the optimization time. Improvement of this process includes tuning algorithm parameters and hybrid local search algorithms with meta-heuristic algorithms [10,11]. Tuning parameters are infeasible in actual case studies since one optimization process can take days to complete. The other approach, hybrid optimization algorithms, typically starts with an evolutionary search in a large space to advance the population to the optimal region. Once the search space is narrowed down sufficiently, the search process will switch to a fast and accurate gradient-based search algorithm to converge on the optimal region [12,13]. However, the time spent by EMO is still highly dependent on the number of design options and objective functions, which could make the project impossible from a practical point of view.

2.2. Algorithm

The NSGA-II is used as the optimization algorithm in this study. The genetic algorithm (GA) is developed based on the mechanics of natural selection and natural genetics. Through reproduction, crossover and mutation, its mathematic foundation shows the GA can achieve exponential search power in a space [14]. The NSGA is a GA based algorithm that implements a nondominanted sorting algorithm, which allows GA to explore multi-dimension search spaces simultaneously. NSGA-II is developed with the intent of decreasing the NSGA's computational complexity and improving the optimal solution set's diversity and intensity by introducing the elitism approach and a crowding distance calculation [15]. There are a few parameters that determine NSGA-II's performance. These parameters should be carefully addressed as part of algorithm

Table 1
NSGA-II parameter sets.

Parameter	Value
Population size	30
Number of generation	30
Probability of crossover	0.9
Probability of selection	0.5
Probability of mutation	0.7

initialization procedure. In the NSGA-II, offsprings are determined by selection, crossover, and mutation operators. In this study, a binary tournament based on dominance sorting and crowding distance is chosen as the selection operator [15]. Since the design problem is an integer type problem, single point crossover and bit flip mutation methods are implemented as the crossover and mutation operator respectively. The probability rates for these three operators are set based on various studies [10,37]. Table 1 lists the parameters' setting of the NSGA-II algorithm. In addition, strategies such as parallel simulations and recall duplicated design solutions are implemented in the optimization as recommended by [10].

2.3. Adaptive operator approach

The first approach focuses on improving the NSGA-II's performance by implementing adaptive operators (aNSGA-II), which is similar to an approach proposed in a 2009 Genetic and Evolutionary Computation Conference [16]. Mutation operator is selected as the test operator. In conventional NSGA-II, selection and crossover operators reserve good designs in generations and explore new ideas by randomly marrying two good designs from the previous generations. However, these two operators may produce premature optimal results and lose some potentially useful designs [14]. The existence of a mutation operator minimizes the chance of losing important designs by performing a random walk in the search space. The mathematic formulation of mutation operator is $O(H)p_m$, where $O(H)$ denotes a design solution and p_m represents the mutation probability, which determines the chance to change design options in a design solution. A strong mutation operator means a high mutation probability and a soft mutation operator means a low mutation probability. In the initial optimization stage, a strong mutation operator is preferred. This is because high mutation probability allows the algorithm to test out more different design options. On the other hand, a soft mutation operator is favored at later stages, which stabilize design solutions in the

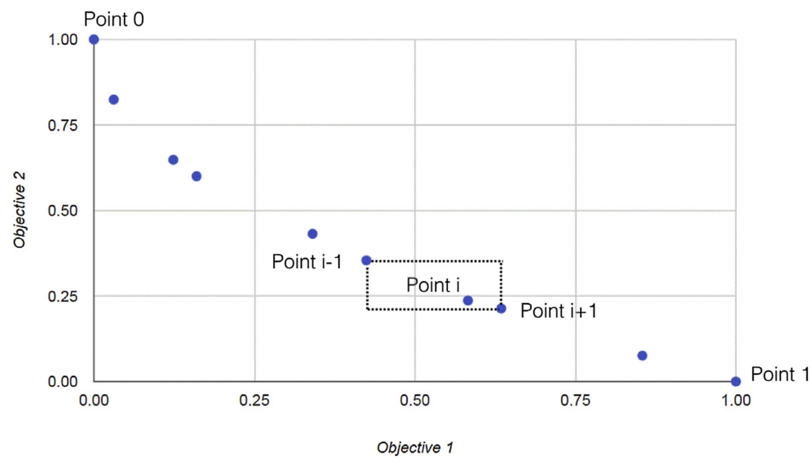

Fig. 3. Crowding distance calculation on a pareto front.

optimal set. This implies that two metrics, number of generation and crowding distance, in the proposed aNSGA-II, are critical in determining a good mutation probability value.

2.3.1. Number of generation

The efficacy of the NSGA-II algorithm ensures that the design solutions will advance towards the optimal region as the number of generation increases. This suggests a strong mutation operator may work better in the first few generations and a soft mutation operator is more suitable at a later stage.

2.3.2. Crowding distance

Crowding distance is defined as an estimation of the density of solutions surrounding a particular solution in a population [15]. It is introduced to ensure the diversity of design solutions in a generated pareto front. Fig. 3 shows the calculation process of crowding distance for a solution. In Fig. 3, point 0 and point 1's crowding distance are set to infinite. The calculation process starts from calculating the distance for point *i*, which is an estimation of the largest cuboid enclosing *i* without including any other points in a population [16]. Then crowding distance is determined by normalizing the calculated distance with the difference between highest and lowest distance (except point 0 and point 1) on the same pareto front. Based on this calculation, the difference between highest and lowest crowding distances is lifted at the initial stage but reduced towards the end of the optimization process.

2.3.3. Formulation of adaptive mutation operator

Eq. (1) represents the mathematical formulation of a adaptive mutation rate generator. G denotes the number of the generations and d indicates the crowding distance difference. C is a constant that determines the decreasing speed of the mutation rate.

$$\left(1 - \frac{1}{1 + e^{(C \times G)}}\right) \times d \qquad (1)$$

Fig. 4 shows the probability distribution of mutation rates based on Eq. (1). In this Figure, the number of generation ranges from 0 to 100 and the difference in crowding distance ranges from 0 to 1. The highest mutation rate happens at the first generation with the crowding distance difference equal to 1 and the lowest mutation rate residing at the 100th generation and 0 crowding distance difference. In aNSGA-II, the mutation rate will be refreshed at each generation based on the calculation result from Eq. (1).

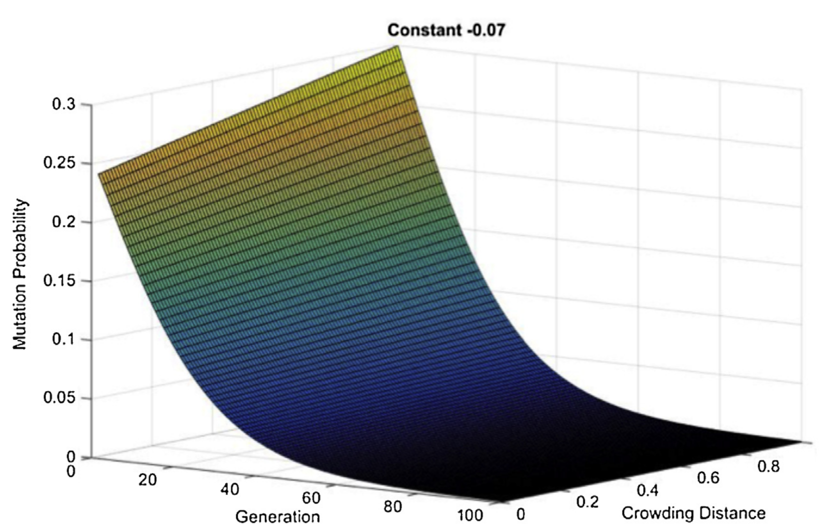

Fig. 4. Probability distribution of creating a mutation rate.

2.4. Adaptive meta-model NSGA-II algorithm

Another method is to marry machine learning techniques with the NSGA-II algorithm's framework to boost the algorithm's search power. This method is commonly referred to as meta-model assisted EMO. Currently, two different approaches are frequently used. In one approach, optimization performs on a regression model, which is generated from a database [7]. This approach greatly reduces the computational time in the optimization process since regression models can evaluate a solution in a negligible amount of time. However, it requires creating a large amount of pre-simulated data that is proven to be general enough for accurately predicting various design combinations. Furthermore, the pre-simulated data are usually case based, which means they can not be reused for another study. In the other approach, which can be referred to as adaptive meta-modeling procedure, a successive meta-modeling approach is used in optimization process. The meta-model updates itself every few generations as solutions move towards the optimum region in the search space [8]. Compare to the first approach, it requires significantly more time in the optimization process. However, the elimination of pre-simulated data and its significant improvement over conventional NSGA-II have made it one of the most popular solvers in various engineering problems. These problems include aeronautics designs [17], airfoil design [18], and building designs [19,6].

2.4.1. Adaptive meta-model NSGA-II

A proposed adaptive meta-modeling procedure based on [20] is implemented in the NSGA-II algorithm for solving BSDO. n and Q are the two new parameters introduced in this procedure. Fig. 5 depicts the overall process of this proposed adaptive meta-model NSGA-II (ammNSGA-II). At the beginning, the ammNSGA-II behaves similarly to conventional optimization. A population is raised by randomly selecting from pre-defined design options and is evaluated through energy simulations. Meanwhile, an external storage is created for storing all the design solutions at each generation. At generation n, the ammNSGA-II pauses the optimization process and then trains the meta models with data retrieved from the external storage. Once regression models are built, the ammNSGA-II replace the energy simulation in the fitness functions with newly trained meta models and restarts the optimization process until Q's generation is reached. Every Qth generation is called one optimization cycle. The cycle will continue until a maximum evaluation number is reached.

This proposed procedure requires dynamic fitness functions that can be switched between actual energy simulation and regression model prediction based on the optimization cycle. Therefore, amendments to the conventional NSGA-II algorithm framework are necessary. The major modifications are made in the main NSGA-II loop and fitness functions. The optimization framework is provided by JMetal v4.5 library [21] and the machine learning algorithms are called from Weka v3.6 [22]. Two candidate machine learning algorithms, linear regression and support vector machine, are tested.

2.4.2. Linear regression

Linear regression (LR) is a commonly used method where energy performance is estimated by building characteristics [23,24]. The Mathematical representation of LR is shown in Eq. (2). In this equation, y_i is the output vector, and x stands for the input vector. β and ε denotes parameters estimated from the data. The selection of the LR model is performed through a greedy search using Akaike-based criterion (AIC) as shown in Eq. (3) [22]. In this equation, I represents the number of instances considered in the regression model and k is the number of attributes.

$$y_i = x^T\beta + \varepsilon_i \tag{2}$$

$$\mathrm{AIC} = (I - k) + 2k \tag{3}$$

2.4.3. Support vector machine

Support vector machine (SVM) is one of the popular supervised machine learning algorithms that is often used in the building field [25]. In this study, LibSVM was used for SVM training and predicting [26]. The algorithm constructs a set of hyperplanes in high dimensional space to separate training data into different classes with a functional margin, which offers high generalization. Consider a set of training points, $((x_1, z_1), \ldots, (x_i, z_i))$ where $x_i \in R^n$ is a feature vector and $z_i \in R^1$ is the target output. Under the given parameters $C > 0$ and $\varepsilon > 0$ [26], $\varepsilon - SVR$ can be formulated as:

$$\begin{aligned}
\min_{\omega,b,\xi,\xi^*} \quad & \frac{1}{2}\omega^T\omega + C\sum_{i+1}^{l}\xi_i + C\sum_{i+1}^{l}\xi_i^* \\
\text{s.t.} \quad & \omega^T\omega(x_i) + b - z_i \le \varepsilon + \xi_i, \\
& z_i - \omega^T\varphi(x_i) - b \le \varepsilon + \xi_i^*, \\
& \xi_i\xi_i^* \ge 0, \quad i = 1, \ldots, l.
\end{aligned} \tag{4}$$

The formation of Eq. (4) represents the Lagrange Multiplier. In this equation, ω denotes the support vectors, which are the input vectors that lie within the margin's boundary. ξ_i is a slack variable that controls the sensitivity of the algorithm to outliers, and lastly ε provides a soft margin which alters hard constraints into ranges.

The SVM hyper-parameters were tuned with a set of pre-simulated data to ensure the algorithm's prediction performance. A polynomial kernel was used in this test. Different combinations of

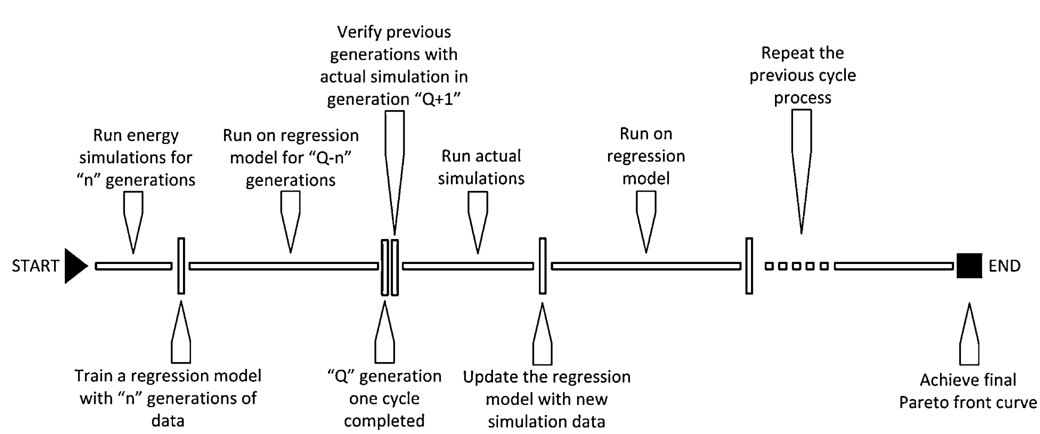

Fig. 5. Procedure of adaptive meta-model NSGA-II.

C (0.0625, 1, 16) and degree in kernel function (3, 4, 5) were tested. The results indicated no significant difference in prediction errors among these combinations. Therefore, the default value $C = 1$ and $d = 3$ were used in this study.

2.4.4. Cross validation

The selection of regression models is performed through a k-fold cross validation method according to their prediction errors. The prediction error is a function of the differences between the actual and the predicted output values. It is typically being used as an accuracy metric for evaluating regression model's generalization property, which prevents the model from under-fitting and over-fitting issues. In this study, normalized root mean square error (NRMSE) is used as the prediction error (Eq. (5)).

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{t=1}^{n}(\hat{y}_t - y)^2}{n}}$$
$$\mathrm{NRMSD} = \frac{\mathrm{RMSE}}{y_{\max} - y_{\min}} \tag{5}$$

In the regression model selection process, 10-fold cross validation method is performed on SVM and LR models with respect to each objective. The model with smallest calculated accuracy metric in one objective is selected and used for predicting this objective.

2.5. Strategies performance comparison

To compare different strategies, 9 optimizations were conducted in this study. These optimizations can be categorized into three parts: optimization using the conventional NSGA-II, optimizations using the aNSGA-II, and optimizations using ammNSGA-II. The optimization that employed the conventional NSGA-II is used as the control group in this comparison.

2.5.1. Reference optimal solution set

To test the strategies' performance, an optimal solution set should be identified as a comparison reference. This reference set is assumed to be the "true" optimal solution set in search spaces. It can be obtained from a combined pool of the 9 optimizations as suggested in [38] following procedure:

1. Merge optimal solution sets from the 9 optimizations to create an optimal solution pool.
2. Apply NSGA-II sorting and crowding distance algorithms to select 30 design solutions from the pool.

2.5.2. Algorithm performance metrics

The performance of multi-objective optimization is measured by three attributes: (i) convergence, (ii) diversity preservation, and (iii) time. The first two attributes are assessed by comparing the optimal solution sets from an optimization study with a reference optimal set. A normalized convergence metric is used for evaluating the algorithm's convergence performance [38]:

$$d_i = \min_{j=1}^{|P^*|} \sqrt{\sum_{k=1}^{M} \left(\frac{f_k(i) - f_k(j)}{f_k^{\max} - f_k^{\min}} \right)^2}.$$
$$C(P^{(t)}) = \frac{\sum_{i=1}^{|F^t|} d_i}{|F^t|}. \tag{6}$$

In Eq. (6), P^* is the reference optimal set and F is the generated optimal solution set, $f_k^{\max}$ and $f_k^{\min}$ are the maximum and the minimum function values of kth objective function in P^* [38]. $C(P^{(t)})$ is the convergence metric that averages the normalized distance d_i for all points in F^t. The normalized convergence metric is achieved in Eq. (7):

$$\overline{C}(P^{(t)}) = \frac{C(P^{(t)})}{C(P^{(\max)})} \tag{7}$$

This suggests that a smaller normalized convergence metric indicates a better convergence.

Diversity preservation prevents the algorithm from having premature convergence as well as ensuring the diversity of design solutions in search spaces. The performance metric for evaluating the diversity of the optimal solution set is through a diversity metric [15]:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \overline{d}|}{d_f + d_l + (N-1)\overline{d}} \tag{8}$$

In Eq. (8), d_f and d_l are the euclidean distances between the extreme solutions and the boundary solutions of the obtained non-dominated set. $\overline{d}$ is the average of all distance $d_i = 1, 2, \ldots, (N-1)$ [15]. A smaller diversity metric indicates a better diversity preservation.

Lastly, the speed performance is measured by the speed metric, which is the elapsed time for completing one optimization study.

3. Case study

The focus of this study is to propose improvement strategies on BSDO using the NSGA-II algorithms. An actual building is selected as a case study to demonstrate the effectiveness of the proposed strategies. The building is located at Carnegie Mellon University, Pittsburgh campus, Pennsylvania. It has four stories and a basement with more than 3700 m^2 of floor area. In total, 112 thermal zones are identified based on the location of air terminals in the provided computer aided design (CAD) drawings. DesignBuilder v4.5 is used to create the building geometry. The geometry is then exported to EnergyPlus [27]. Fig. 6 shows the actual building exterior view and its energy model in EnergyPlus. Since building system design options are usually identified

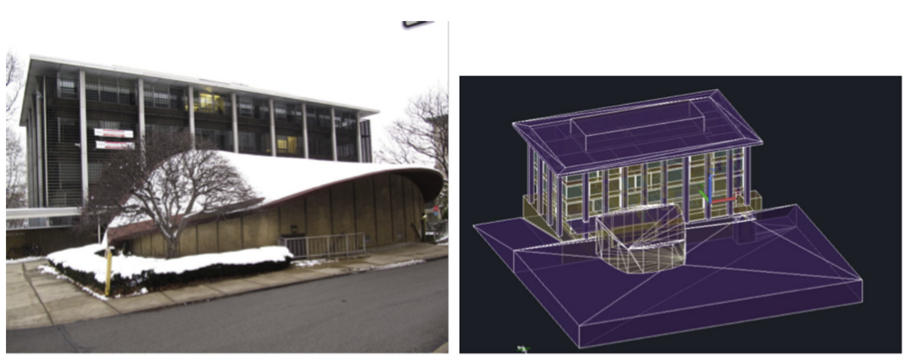

Fig. 6. Representation of building in EnergyPlus.

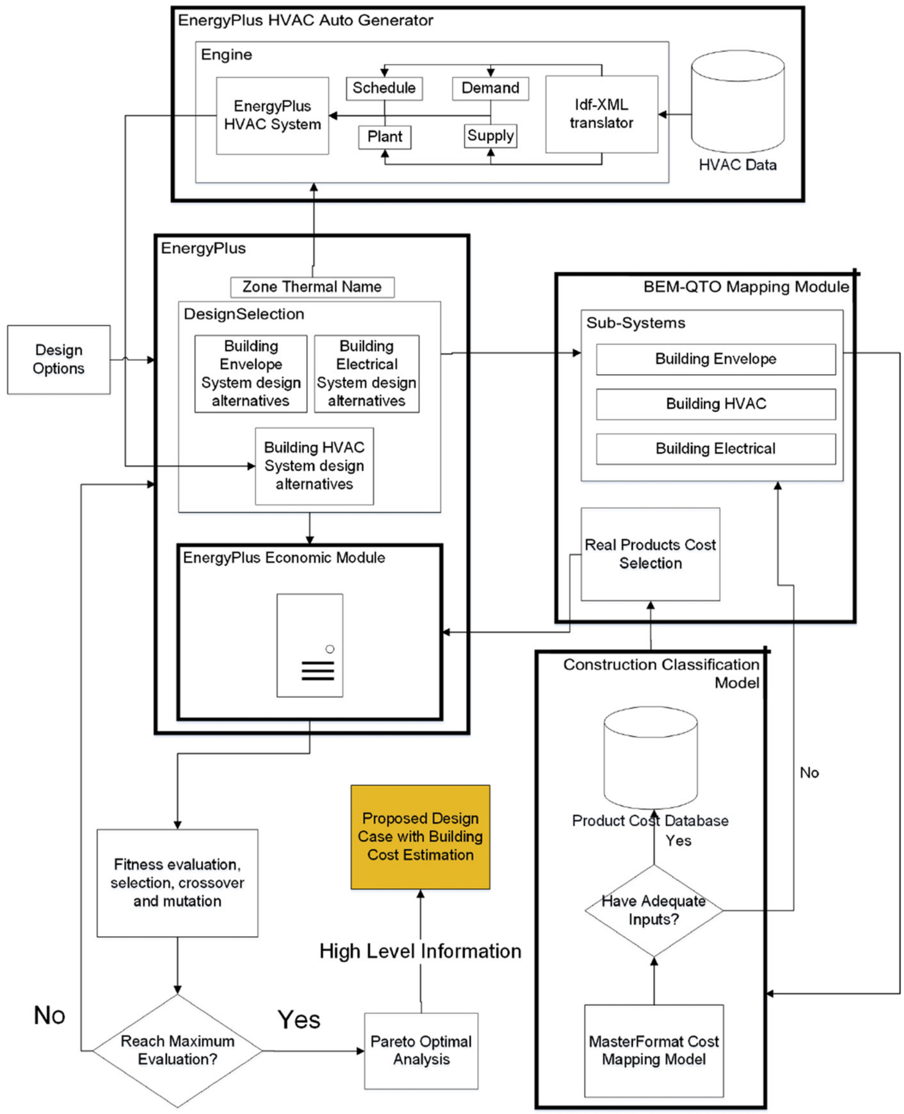

Fig. 7. Building system design optimization framework.

by the product type and name, an EnergyPlus product component database (Construction Classification Model in Fig. 7) was created which not only contained performance data of products, but also their cost information. The building products were classified according to the MasterFormat 2004 classification standard. MasterFormat is one of the leading trade-based industry classification standards developed for construction work that are specified jointly by the Construction Specifications Institute (CSI) and the Construction Specifications Canada (CSC) [28]. In this database, performance data were collected from various literatures or manufacturers' specifications and cost information were extracted from RSMeans® Building Construction Cost Data Book [29]. This data was managed in MySQL, an open source relational database management system (RDBMS). A BEM-QTO mapping module in Fig. 7 was developed to link this database with EnergyPlus components.

A Java-based application was built to transfer and process data among the tools, libraries and databases. An EnergyPlus HVAC system auto-generator was formed to assist in the creation of the HVAC systems in EnergyPlus. This application allows user to customize the design options, select up to two objectives and execute the optimization process automatically. Fig. 7 shows the framework of this application.

3.1. Optimization formulation

The first cost and operation costs were selected to be the two objectives. The design problem can be formulated as:

$$\min\{f_1(x), f_2(x)\} \tag{9}$$

where $f_1(x)$ is building operation costs, which is stated:

$$f_1(x) = x_{rep} - x_{res} + x_e + x_{om\&r} \tag{10}$$

$f_2(x)$ is the first costs (initial investment costs) and it can be formulated as:

$$f_2(x) = x_{ei} + x_{hi} + x_{li} + x_{rest} \tag{11}$$

where x_{ei} (envelope costs), x_{hi} (HVAC system costs), x_{li} (lighting system costs), x_{rest} (other costs), x_{rep} (present value of capital replacement costs), x_{res} (present value of residual value less the disposal costs), x_e (present value of energy costs), $x_{om\&r}$ (present value of non-fuel operating, maintenance and repair costs). Eq. (9) is adapted from NIST Handbook 135 with no water cost included [30]. The constraint of this optimization process is:

$$X = \{x \in \mathbb{Z}^{sc} | x^{ij},\ i \in \{1, \ldots, c\},\ j \in \{1, \ldots, s\}\} \tag{12}$$

Table 2
Design options for each design variable (external wall and roof options can be found in [31]).

Design variables	Design options	Property	Unit cost
Windows (6 options)	Double clear (DC)	U[a]-3.13, SHGC[b]-0.73, Vt[c]-0.8	$242/m^2
	Double tinted (DT)	U-2.58, SHGC-0.37, Vt-0.53	$290/m^2
	Double thick clear (DTC)	U-1.40, SHGC-0.41, Vt-0.61	$356/m^2
	Heat reflective clear (HRC)	U-1.40, SHGC-0.25, Vt-0.45	$479/m^2
	Triple glazing (TG)	U-0.81, SHGC-0.71, Vt-0.53	$480/m^2
	Quadruple (Q)	U-0.781, SHGC-0.46, Vt-0.62	$862/m^2
Lights (3 options)	T8	LPD[d]-10.2 W/m^2	$150/m^2
	T5	LPD-8 W/m^2	$164/m^2
	LED	LPD-6.5 W/m^2	$390/m^2
Control (1 option)	Daylighting sensor		$268/Each
HVAC (3 options)	VAV system	[e]	$1233/kW
	VRF system[f]	[e]	$1418/kW
	DOAS + VRF system	[e]	$1541/kW

[a] U stands for overall coefficient of thermal transmission (W/m^2 K).
[b] SHGC stands for solar heat gain coefficient.
[c] Vt stands for visible transmittance.
[d] LPD stands for lighting power density.
[e] HVAC system efficiency is based on manufacturers' system performance data. If manufacturers' data is not available, then the minimum performance data in ASHRAE 90.1 2007 performance rating method [32] is used.
[f] VRF stands for variable refrigerant flow system.

The $\mathbb{Z}^{sc}$ is the constraint set, which contains the pre-defined building system types s and their cost information c. The design parameter x is specified as discrete independent variables that can only take the values from $\mathbb{Z}^{sc}$.

3.2. Design variables

6 design variables were selected, namely external wall, roof, window, lighting fixture, daylight control and HVAC systems. These design variables covered building envelope, lighting and HVAC system. External wall and roof constructions were built according to Table 16 and 17 in Chapter 18 of ASHRAE Handbook 2009 [31]. Table 2 lists the design options for the design variables excluding external walls and roofs. Firstly, the selected design options query their performance and cost data from the database through the BEM-QTO mapping module. The mapping module will then map the fetched performance and cost data to the related EnergyPlus objects. The first costs are transferred to "ComponentCost:LineItem" object in EnergyPlus. It should be noted that due to some limitations of this object, unit cost of several components were converted to units that are compatible to EnergyPlus. For instance, the unit cost of windows ($/Each) was translated to ($/m^2). Similarly, to calculate the first cost for a HVAC system, the mapping module has to convert the sum of all the HVAC components' unit cost to $/kW. Besides first costs, MRR costs are mapped to the "LifeCycleCost:RecurringCosts" object for life cycle analysis. More information about these two EnergyPlus objects can be found in EnergyPlus Input/Output reference [33].

In total, there were 71,820 possible design combinations with 35 external wall and 19 roof constructions as well as other design variables specified in Table 2. A desktop with a configuration of i7 quad-core 3.5 GHz CPU and 16 GB RAM was used for performing the case study. One annual EnergyPlus simulation took about 8–10 min depending on the selected design options. Using a brute-force search would require more than 400 days to complete a full enumeration of the parameter space.

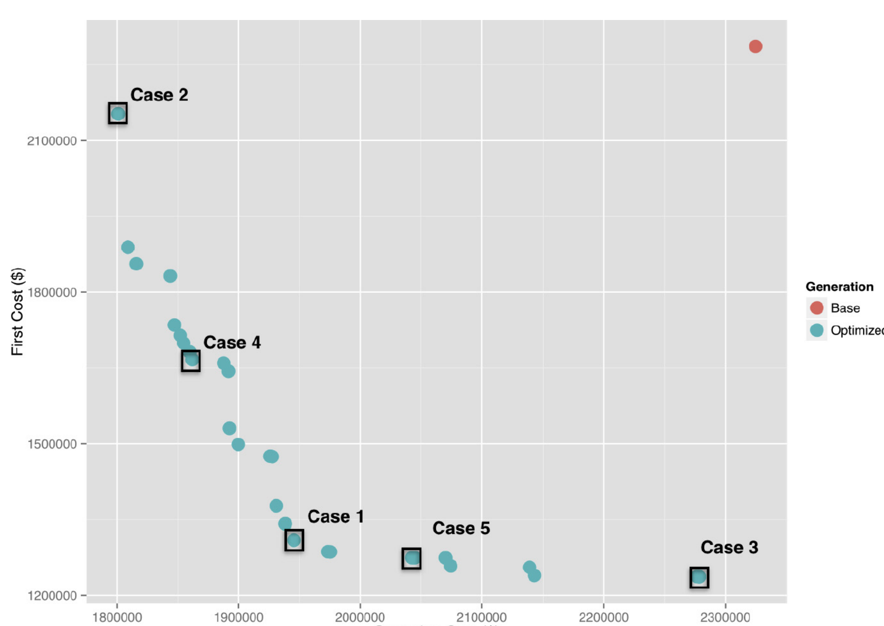

Fig. 8. Base case and optimized final pareto front curve.

3.3. Study assumptions

Designs other than selected design variables remained identical in the optimization process. System controls were set according to ASHRAE 90.1 2007 performance rating method [32]. Thermal zone's operation schedules, infiltration rate and human activities were assumed based on their correspondent type in DOE's commercial reference buildings [34]. In addition, thermal zones' ventilation rates were designed to meet ASHRAE 62.1 2007 requirements [35].

The life cycle cost parameters were defined according to the annual supplement report to NIST Handbook 135 in Table 3 [36]. As part of the life cycle cost component, a utility cost model was built based on local utility providers' published tariffs (Table 3). EnergyPlus's economics module is capable of modeling complex

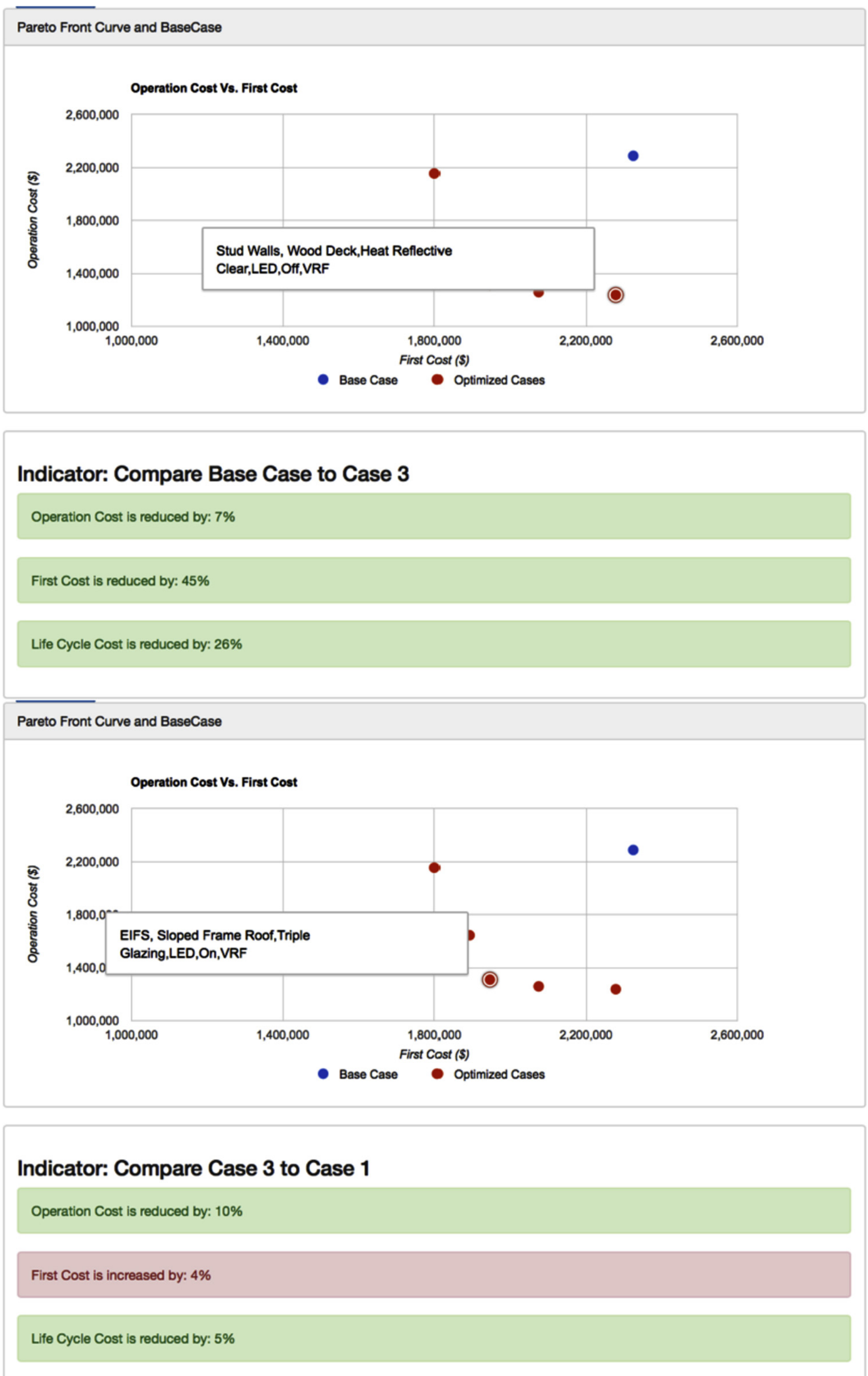

Fig. 9. Proposed visualization of design decision making scenario.

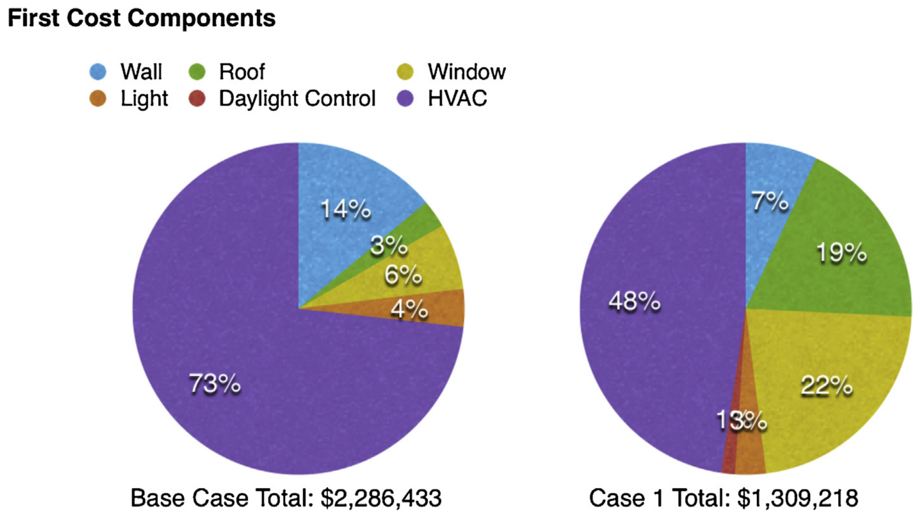

Fig. 10. Comparison of component costs.

Table 3
Life cycle cost model parameters.

Field	Input
Discounting convention	End of year
Inflation approach	Constant dollar
Real discount rate	0.03
Base date	2015 January
Service date	2017 January
Study length	25 years
Tax rate	0.06
Depreciation method	StraightLine-27 years
Price escalation	NIST Handbook 135 Table Ca-1, Census Region 1
Electricity tariff	Duquesne light 2015 rate GS
Natural gas tariff	Equitable gas 2015 rate GSS

utility cost tariffs. Unlike an average energy cost value, the well established tariff model can accurately represent the monthly energy cost based on both energy consumption and monthly peak demand. Thus, with a tariff model, the optimization process will not only minimize the building's annual energy consumption, but also reduce its peak demands.

4. Results and discussion

4.1. Optimization results analysis

A reference optimal set was created using the method described in Section 2.5.1. Fig. 8 shows the plot of optimal design solutions in this set. A base case was identified with the design options that had lowest unit prices and its results were also plotted in Fig. 8 for comparison.

Following the process in Fig. 2, a higher level of information was applied to these 30 design solutions to provide a better view on the optimized cases in a smaller set of solutions. In Table 4, "case 1" represents the lowest life cycle cost; "case 2" has the lowest operation costs; "case 3" has the lowest first cost; "case 4" is a design solution found in the middle of the upper curve and lastly, "case 5" is a design solution found in the middle of the lower curve. It can be concluded that high performance products are preferred in the final design due to the most frequent design options being triple glazing system, LED lighting fixtures and daylight controls. By comparing these optimized solutions to the base case, it demonstrates high performance design options do not cost more than cheap options if they are evaluated as an integrative system. Since these optimized design solutions yield lower first and operation costs than the base case, payback period is not a suitable evaluation criteria. Thus, a new finance metric referred to as additional break-even first cost (ABEFC), is introduced. ABEFC is a first costs based analysis, which converts all the cost savings in a project's life cycle period to a present value. It allows project stakeholders to compare optimized design solutions by evaluating how much additional first cost could be justified to break even the base case [39]. Although "case 2" shows the lowest operation costs in 25 years of operation, its ABEFC is the lowest among all other design solutions because of its high first cost. The other cases show similar ABEFC with the highest being $1,356,498. Besides ABEFC, building stakeholders can assess the trade-offs between first and operation costs among the optimized design solutions by visualizing the solutions on a pareto front curve (Fig. 9). Fig. 9 shows a proposed visualization of a design decision making scenario. In Fig. 9, comparing the base case with "case 3", the first cost is reduced by 45% and the life cycle cost is decreased by 26%. Similarly, comparing "case 1" with "case 3", "case 3" demands 4% higher first cost but yield 10% lower operation costs in 25 years.

Furthermore, Fig. 10 compares cost components between the base case and "case 1". It shows that with a high performance integrative building system strategy, the percentage of HVAC system

Table 4
Optimal solution packages.

Case	Wall[a]	Roof[a]	Window	Light	Daylight	HVAC	First cost ($)	Operation cost ($)	ABEFC ($)
Base	24	9	*DC*	*T8*	Off	VAV	2,286,433	2,324,763	0
1	9	3	*TG*	*LED*	On	VRF	1,309,218	1,945,480	1,356,498
2	34	17	*TG*	*LED*	On	DV[b]	2,152,940	1,800,880	657,376
3	7	7	*HRC*	*LED*	Off	VRF	1,237,045	2,278,456	1,095,695
4	22	16	*TG*	*T5*	On	DV[b]	1,643,671	1,891,623	1,075,902
5	9	4	*TG*	*LED*	On	VRF	1,258,314	2,074,458	1,278,423

[a] Wall and roof number is based on ASHRAE Handbook 2009 Fundamental Chapter 18 Table 16 and 17.
[b] DV denotes hybrid DOAS and VRF system.

costs can be dramatically decreased. On the other hand, it also demonstrates that building systems are interrelated with each other and they should be evaluated at an integrated level so that first cost trade-offs at the building product level can be correctly captured by the optimization algorithm.

4.2. Improvement strategies analysis

4.2.1. Adaptive mutation operator

Fig. 11 shows the algorithm performance comparison between the conventional NSGA-II and aNSGA-II. In Fig. 11 legend, "true" labels solutions from the reference set and numbers indicate the number of generation. Generation 2, 15, and 30 were selected in this comparison. At the initial stage, aNSGA-II's high mutation rate allowed the algorithm to explore larger solution space so that the design solutions could quickly advance to the optimal region. On the contrary, at the later stage, aNSGA-II slowed its searching ability down to preserve the generated optimal design solutions. Comparing the normalized convergence metric in Table 5 proves that aNSGA-II achieves better convergence performance than the NSGA-II algorithm. However, the conservative strategy at the later stage may affect the solutions' diversity, thus in Table 5, a slightly higher

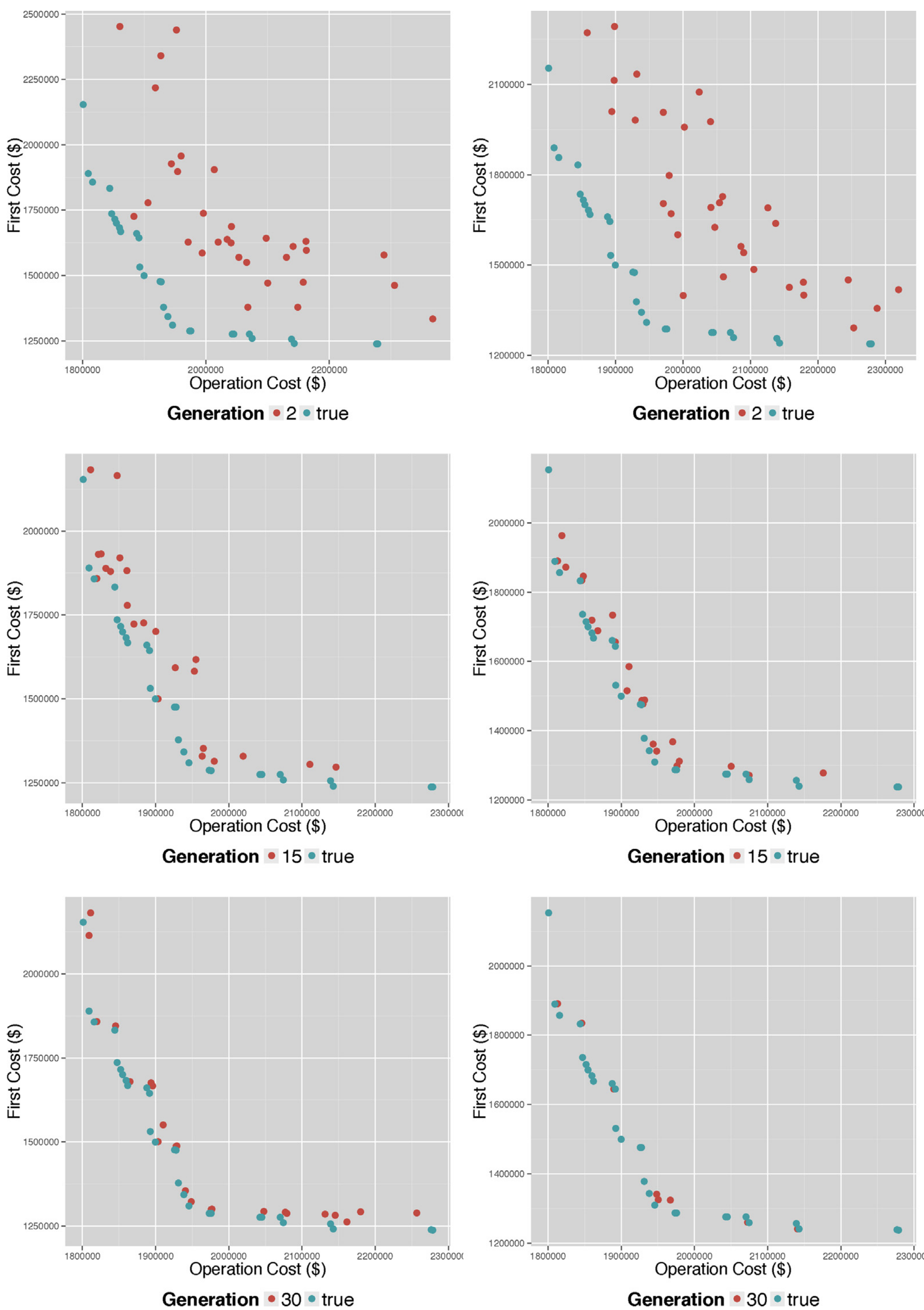

Fig. 11. Optimization comparison between NSGA-II and aNSGA-II (left: NSGA-II; right: aNSGA-II).

Table 5
Performance metrics comparison for NSGA-II, aNSGA-II with different *C*.

Model name	Generation	Time (s)	Normalized convergence metric ($\bar{C}(P^{(t)})$)	Spread metric (Δ)
NSGA-II	30	82,924	0.025	0.410
aNSGA-II (−0.1)	30	49,959	0.011	0.465
aNSGA-II (−0.07)	30	54,682	0.004	0.423
aNSGA-II (−0.04)	30	62,802	0.017	0.416

spread metric is observed in aNSGA-II. Comparing the speed metric, aNSGA-II reduce the optimization time by 34%, which proves its significant advantages over the conventional NSGA-II.

4.2.2. Sensitivity analysis on adaptive mutation operator

Eq. (1) determines the mutation probabilities in generations. In this sigmoid equation, *C* is a constant that impacts the slope of probability rate distribution. A value recommended in [16] is −0.07; however, the sensitivity of aNSGA-II performance to *C* in building system design problem should be carefully evaluated. Therefore, a slightly lower (−0.1) and a slightly higher (−0.04) were proposed for comparison. Fig. 12 shows the comparison of probability distribution with three different values. It implies that there is a mild reduction trend in mutation probability as *C* becomes smaller. Therefore, for constant −0.04, the change of mutation probability is slower compared to the other two cases. Fig. 13 shows the comparison of optimal solutions distribution at specific generations. Apparently, in this case, $C = -0.07$ shows the best convergency performance. This is also proved by their performance metrics in Table 5. For $C = -0.1$, the sharp drop of the mutation rate slowed down its advancing speed towards the optimal region by comparing the generation 15 of these three cases. On the other hand, a lower *C* value showed higher mutation probabilities at the later stage, which created difficulty for convergency. A slightly higher spread metric was found in $C = -0.1$ case; however, these three algorithms showed no obvious advantage regarding the diversity conservation performance. Comparing the speed metric, $C = -0.1$ required the least time due to a large amount of duplicate cases that were found at later stage.

4.2.3. Adaptive meta-model optimization

Fig. 14 compares the performance of the conventional NSGA-II and ammNSGA-II. 3 cycles were performed for ammNSGA-II, therefore, a total of 12 generations' solutions were evaluated on actual energy simulation. With these 12 generations, ammNSGA-II was able to product a nearly identical optimal set to the reference set with a normalized convergence metric around 0.001. This saved the optimization time by 61% (Table 6). However, its spread metric indicated that ammNSGA-II performed poorly on diversity preservation.

The prediction accuracy was also examined after the simulation. Fig. 15 shows the comparison between prediction values and simulated values in the optimization process. Table 7 shows relatively small NRMSE values for each regression model at every cycle.

4.2.4. Sensitivity analysis on adaptive meta-model optimization

Fig. 5 shows the overall procedure of adaptive meta-model evolutionary optimization. The important parameters are *n* and *Q* which determine the size of the training dataset and the search power provided by the regression model. In integer type optimization, a small *n* may cause insufficient training data, which impacts the accuracy of the regression model. On the contrary, a large *n* will increase the time for optimization. *Q* affects the search power of the regression model because the proposed procedure expects regression models to improve their accuracy and precision after each cycle. A large *Q* will certainly mislead the search to an incorrect region, which could possibly trap the optimal solutions in local optimal regions.

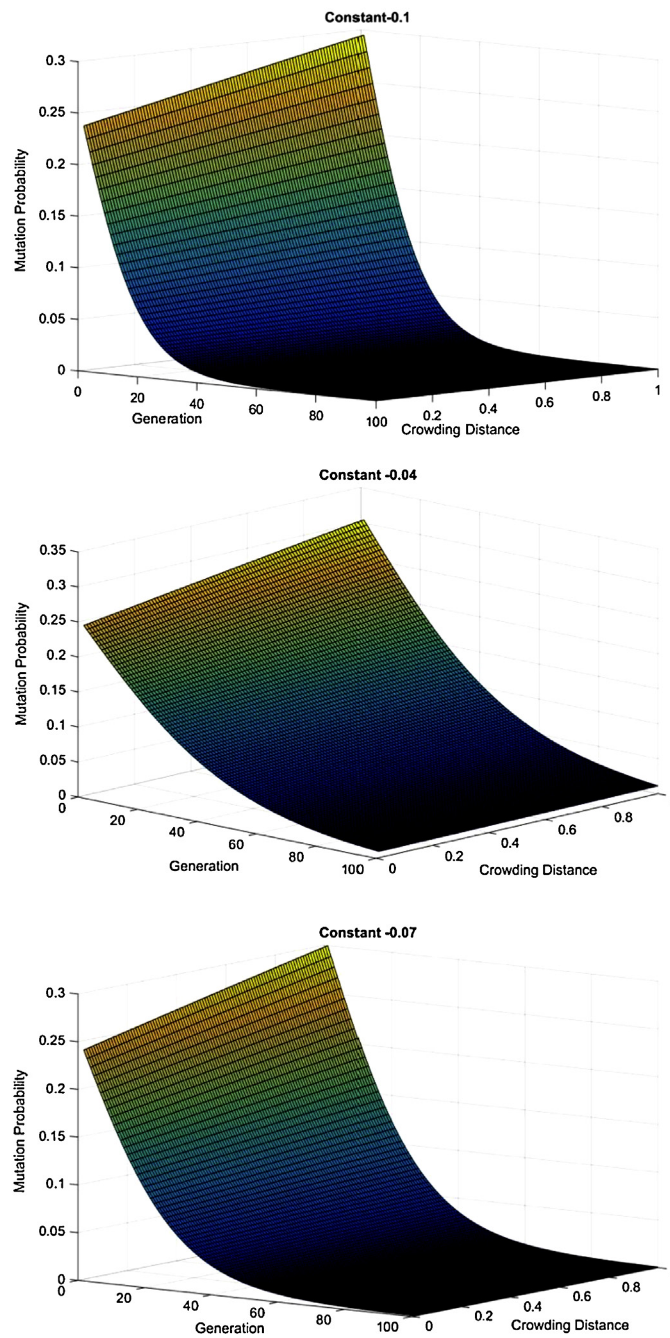

Fig. 12. Mutation probability rate distribution comparison.

In this study, 5 sets of *n* and *Q* were compared. The name of the tests were structured in a convention of "Type-n-G". Type included "I" and "B". "I" represented cumulative training data, where the database was built cumulatively after each cycle. "B" represented intermittent training data, where the database was rebuilt after each cycle. Each test performed three cycles in the optimization procedure. Table 6 lists the performance metrics for these 5 tests.

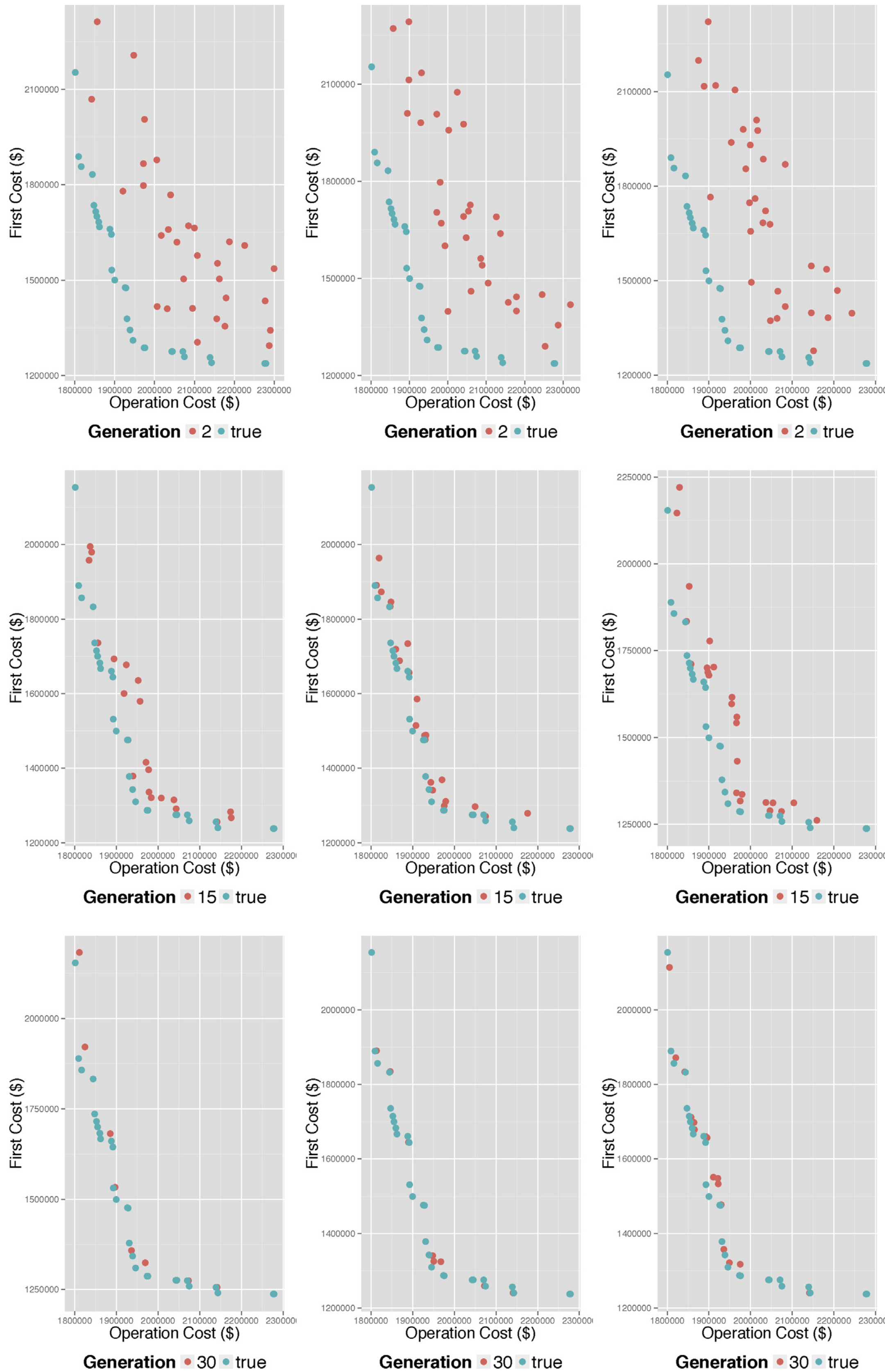

Fig. 13. Parametric study on adaptive mutation operator (left: −0.1, middle: −0.07, right: −0.04).

When comparing B-3-20, B-2-20 and B-4-20, it implies that a training dataset with less than 90 data points is insufficient for accurate prediction. B-2-20 has only 60 data points at each cycle, which means that it converges poorly at the third cycle. On the other hand, it can be concluded that $G = 20$ can provide enough search power to advance design solutions to the optimal region by comparing B-3-20 and B-3-10. I-3-20 indicates no obvious advantages on convergency performance; however, it can slightly improve

Table 6
Performance metrics comparison between the NSGA-II and different adaptive meta-model NSGA-II configurations.

Model name	Generation	Time (s)	Normalized convergence metric ($\overline{C}(P^{(t)})$)	Spread metric (Δ)
NSGA-II	30	82,924	0.025	0.410
B-3-20	12	32,070	0.001	1.022
B-3-10	12	32,008	0.019	1.158
B-4-20	16	42,543	0.003	1.002
B-2-20	8	20,432	0.099	0.810
I-3-20	12	33,432	0.002	0.870

the diversity of design solutions. All the test cases have similar spread metrics. The diversity preservation of these test cases are poorer than the conventional NSGA-II algorithm. This could be due to the regression model being confined to the search region at the later stage. Therefore, this result in a limited number of the available design solutions in the final optimal set. Thus, several duplicate design options appear in the optimal set, which increases the spread metric. Comparing the time, B-2-20 requires

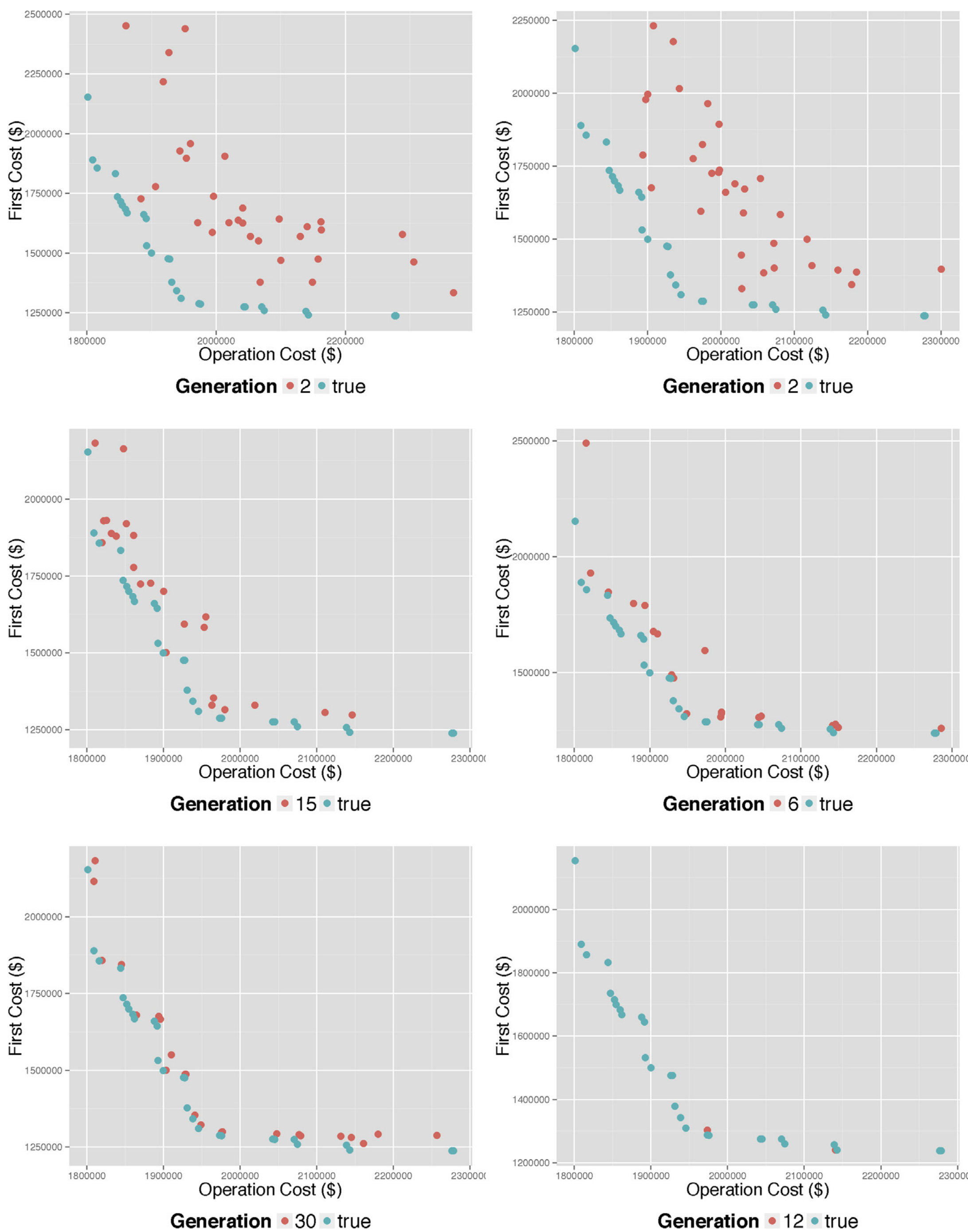

Fig. 14. Optimization comparison between NSGA-II and ammNSGA-II (left: NSGA-II; right: ammNSGA-II).

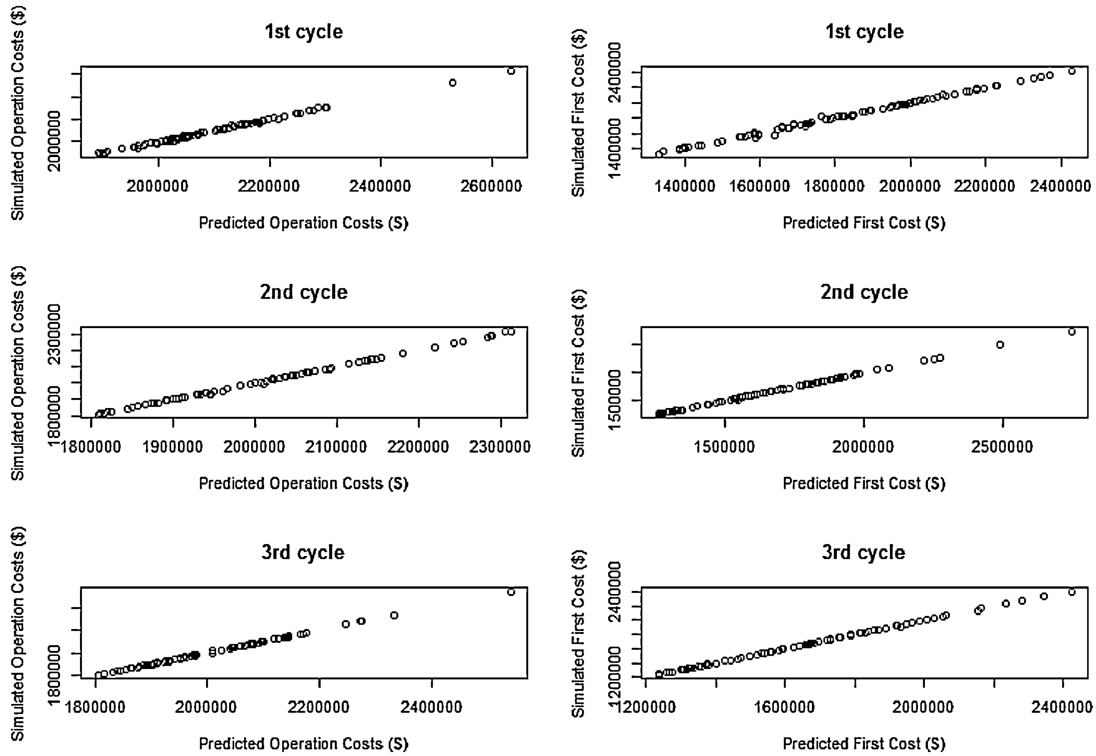

Fig. 15. Predict value vs. simulated value for three cycles in the optimization procedure.

Table 7
NRMSE for each objective prediction at each cycle.

Cycle	Objective	NRMSE
1	Operation costs	5.3
	First cost	5.0
2	Operation costs	2.7
	First cost	1.2
3	Operation costs	2.6
	First cost	1.8

the least number of simulations, hence it performs the fastest. On the contrary, B-4-20 needs the most number of simulations, thus its demands twice as much as B-2-20. The other models have a similar time performance. However, I-3-20 shows a slightly higher time metric compared to the other two models. The observation suggests a longer time spent on training regression models with a cumulative database.

4.2.5. Comparison between two strategies

In this section, the performance of aNSGA-II and ammNSGA-II are compared in Table 8. The advantage of using aNSGA-II is to achieve diverse and good design solutions. A noticeable amount of time (34%) saved by using aNSGA-II is also reported in this study. On the contrary, ammNSGA-II has the poorest performance on preserving solutions' diversity; however, it has the lowest normalized convergence metric and it can save up to 60% of computation time.

5. Conclusion

In this study, an integer type, non-constraints BSDO problem was solved by implementing an evolutionary optimization algorithm. An actual office building was used as a case study to test the performance of the algorithm. Six building systems were identified as design variables and multiple design options were defined under each variable. By evaluating these systems in an integrative workflow, the optimized design solutions demonstrated large reductions on both first cost and operations costs.

More importantly, this study proposed and tested two improvement strategies on the NSGA-II. These strategies mainly focus on improving the design solution convergence and reducing computational time in the optimization process. One of the improvement strategies, aNSGA-II, proved about a 30% time reduction and better convergence performance. The other strategy, ammNSGA-II procedure, demonstrated around a 60% time reduction with high quality optimized design options. Although both strategies show slight losses on solution diversity preservation, in the building system design field, design solutions' precision is more important than their diversity. This is because building owners could only view a limited number of design solutions instead of a whole range of solution set. The proposed improvement strategies can be adopted in various design optimization practices and research for finding high performance building system design solutions. The benefits of using these two strategies are as follows:

- Reduce optimization results' variability: The NSGA-II algorithms are heuristic algorithms, which means each round of an

Table 8
Performance metrics comparison between aNSGA-II and ammNSGA-II.

Model name	Generation	Time (s)	Normalized convergence metric ($\overline{C}(P^{(t)})$)	Spread metric (Δ)
NSGA-II	30	82,924	0.025	0.410
aNSGA-II	30	49,959	0.004	0.423
ammNSGA-II	12	32,070	0.001	1.022

optimization study could result in very different optimized design solutions if the maximum evaluation number is not sufficient. However, by implementing either one of the strategies on a limited number of evaluations, the optimized design solutions could be closer to the "true" optimal set compared to the conventional NSGA-II, thus, reducing the variability of the optimization results.
- Minimize optimization time: Although several studies have implemented various techniques combined with conventional evolutionary algorithms to reduce optimization time, the proposed improvement strategies focus on changing the behavior of conventional evolutionary algorithms to achieve time reduction, which was proven to be highly effective in the case study.

Future work will focus on testing further improvements by integrating adaptive operators in ammNSGA-II procedure. As mentioned above, ammNSGA-II runs faster and more precisely than aNSGA-II; however, the generated regression model may confine the search space during the $Q-n$ phase, which produces duplicate solutions in the optimal solution set. This could be solved by having a correct mutation probability rate that encourages the algorithm to more actively search outside of the optimal set that is generated by the regression model.

References

[1] V. Machairas, A. Tsangrassoulis, K. Axarli, Algorithms for optimization of building design: a review, Renew. Sustain. Energy Rev. 31 (2014) 101–112.
[2] R. Evins, A review of computational optimisation methods applied to sustainable building design, Renew. Sustain. Energy Rev. 22 (2013) 230–245.
[3] M. Palonen, M. Hamdy, A. Hasan, MOBO a new software for multi-objective building performance optimization, in: Proceedings of the 13th International Conference of the IBPSA, 2013, pp. 2567–2574.
[4] M. Wetter, et al., GenOpt – a generic optimization program, in: Seventh International IBPSA Conference, Rio de Janeiro, 2001, pp. 601–608.
[5] Y. Zhang, Use jEPlus as an efficient building design optimisation tool, in: CIBSE ASHRAE Technical Symposium, London, 2012, pp. 18–19.
[6] A.E. Brownlee, J.A. Wright, Constrained, mixed-integer and multi-objective optimisation of building designs by NSGA-II with fitness approximation, Appl. Soft Comput. 33 (2015) 114–126.
[7] B. Eisenhower, Z. ONeill, S. Narayanan, V.A. Fonoberov, I. Mezić, A methodology for meta-model based optimization in building energy models, Energy Build. 47 (2012) 292–301.
[8] K. Deb, Multi-objective optimisation using evolutionary algorithms: an introduction, in: Multi-objective Evolutionary Optimisation for Product Design and Manufacturing, Springer, 2011, pp. 3–34.
[9] O.T. Karaguzel, R. Zhang, K.P. Lam, Coupling of whole-building energy simulation and multi-dimensional numerical optimization for minimizing the life cycle costs of office buildings Building Simulation, vol. 7, Springer, 2014, pp. 111–121.
[10] A. Alajmi, J. Wright, Selecting the most efficient genetic algorithm sets in solving unconstrained building optimization problem, Int. J. Sustain. Built Environ. 3 (1) (2014) 18–26.
[11] A.-T. Nguyen, S. Reiter, P. Rigo, A review on simulation-based optimization methods applied to building performance analysis, Appl. Energy 113 (2014) 1043–1058.
[12] D. Greiner, B. Galván, J. Periaux, N. Gauger, K. Giannakoglou, G. Winter, Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences, Springer, 2015.
[13] Y.-K. Juan, P. Gao, J. Wang, A hybrid decision support system for sustainable office building renovation and energy performance improvement, Energy Build. 42 (3) (2010) 290–297.
[14] D.E. Goldberg, Genetic algorithms in search, optimization & machine learning, © 1989 by Addison.
[15] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[16] A.G. Carvalho, A.F. Araujo, Improving NSGA-II with an adaptive mutation operator, in: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, ACM, 2009, pp. 2697–2700.
[17] K. Giannakoglou, Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence, Prog. Aerosp. Sci. 38 (1) (2002) 43–76.
[18] M. Emmerich, B. Naujoks, Metamodel assisted multiobjective optimisation strategies and their application in airfoil design, in: Adaptive Computing in Design and Manufacture VI, Springer, 2004, pp. 249–260.
[19] S.S. Gilan, B. Dilkina, Sustainable building design: a challenge at the intersection of machine learning and design optimization, in: Workshops at the Twenty-ninth AAAI Conference on Artificial Intelligence, 2015.
[20] K. Deb, P.K. Nain, An evolutionary multi-objective adaptive meta-modeling procedure using artificial neural networks, in: Evolutionary Computation in Dynamic and Uncertain Environments, Springer, 2007, pp. 297–322.
[21] J.J. Durillo, A.J. Nebro, JMetal: a java framework for multi-objective optimization, Adv. Eng. Softw. 42 (10) (2011) 760–771.
[22] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The Weka data mining software: an update, ACM SIGKDD Explor. Newslett. 11 (1) (2009) 10–18.
[23] M. Braun, H. Altan, S. Beck, Using regression analysis to predict the future energy consumption of a supermarket in the UK, Appl. Energy 130 (2014) 305–313.
[24] H.-X. Zhao, F. Magoulès, A review on the prediction of building energy consumption, Renew. Sustain. Energy Rev. 16 (6) (2012) 3586–3592.
[25] B. Dong, C. Cao, S.E. Lee, Applying support vector machines to predict building energy consumption in tropical region, Energy Build. 37 (5) (2005) 545–553.
[26] C.-C. Chang, C.-J. Lin, LibSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. (TIST) 2 (3) (2011) 27.
[27] D. Crawley, F. Winkelmann, L. Lawrie, C. Pedersen, EnergyPlus: a new-generation building energy simulation program, in: Forum – Proceedings, American Solar Energy Soc. & The American Institute of Architects, 2001, pp. 575–580.
[28] CSI, CSC, MasterFormat™ 2004 edition numbers & tiles, Tech. rep., Construction Specifications Institute and Construction Specifications Canada, 2005.
[29] RSMeans, Building Construction Cost Data, Construction Publishers & Consultants, Norwell, MA, 2015.
[30] S.K. Fuller, S.R. Petersen, NIST Handbook 135: Life-cycle Costing Manual for the Federal Energy Management Program, National Institute of Standards and Technology, Gaithersburgh, MD, 1995.
[31] ASHRAE, ASHRAE Handbook 2009 Fundamental, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Atlanta, GA, 2009.
[32] ASHRAE, ASHRAE standard 90.1-2007 energy standard for building except low-rise residential buildings (SI edition). Tech. rep., American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., 2007.
[33] LBNL, Input output reference: the encyclopedic reference to EnergyPlus input and output, Tech. rep., Lawrence Berkeley National Laboratory, 2013.
[34] M. Deru, K. Field, D. Studer, K. Benne, B. Griffith, P. Torcellini, B. Liu, M. Halverson, D. Winiarski, M. Rosenberg, et al., US Department of Energy Commercial Reference Building Models of the National Building Stock.
[35] ASHRAE, ASHRAE standard 62.1-2007 ventilation for acceptable indoor air quality, Tech. rep., American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., 2007.
[36] A.S. Rushing, J.D. Kneifel, P. Lavappa, Energy price indices and discount factors for life cycle cost analysis 2015, Tech. rep., National Institute of Standards and Technology, 2014.
[37] R. Evins, P. Pointer, S. Burgess, Multi-objective optimisation of a modular building for different climate types, in: First Building Simulation and Optimization Conference, 2012.
[38] K. Deb, S. Jain, Running performance metrics for evolutionary multi-objective optimization, Tech. rep., 2002.
[39] S.A. Mumma, Using dedicated outdoor air systems: economics of improved environmental quality, ASHRAE J. 1 (2002) 1–10.