Pareto Front Upconvert on Multi-objective Building Facility Control Optimization

Naru Okumura
n.okumura@uec.ac.jp
The University of Electro-Communications
Chofu, Tokyo, Japan

Tomoaki Takagi
tomtkg@uec.ac.jp
The University of Electro-Communications
Chofu, Tokyo, Japan

Yoshihiro Ohta
Ota.Yoshihiro@dw.MitsubishiElectric.co.jp
Mitsubishi Electric Corporation
Kamakura, Kanagawa, Japan

Hiroyuki Sato
h.sato@uec.ac.jp
The University of Electro-Communications
Chofu, Tokyo, Japan

ABSTRACT

This paper verified the effects of a supervised multi-objective optimization algorithm (SMOA) efficiently upconverting the Pareto front representation by utilizing known solutions on a real-world multi-objective building facility control optimization problem. Also, several sampling methods for evaluating promising candidate solutions in SMOA were proposed and compared. Evolutionary variations, such as crossover and mutation involving randomness, are not preferred in practical scenarios, particularly when the objective functions are computationally expensive. In order to suppress obtaining inferior solutions, SMOA constructs the Pareto front and Pareto set estimation models using known solutions, samples promising candidate solutions, and evaluates them. It was reported that SMOA could efficiently generate well-distributed solutions that upconvert the Pareto front representation compared to evolutionary variations with limited solution evaluations in artificial test problems. This paper focuses on the real-world building facility control problem with 15 known solutions, and results show that SMOA can efficiently improve the Pareto front representation compared to evolutionary variations. Also, results show that crowding distance-based one-time sampling considering the distribution of the known solutions achieved the best Pareto front approximation performance in the sampling methods compared in this paper.

CCS CONCEPTS

• **Applied computing → Engineering**; • **Theory of computation → Continuous optimization**.

KEYWORDS

building facility control, multi-objective optimization, evolutionary optimization, data-driven optimization, supervised multi-objective optimization, Pareto front estimation, Pareto set estimation

GECCO '23 Companion, July 15–19, 2023, Lisbon, Portugal

ACM ISBN 979-8-4007-0120-7/23/07.
https://doi.org/10.1145/3583133.3596339

ACM Reference Format:
Naru Okumura, Tomoaki Takagi, Yoshihiro Ohta, and Hiroyuki Sato. 2023. Pareto Front Upconvert on Multi-objective Building Facility Control Optimization. In *Genetic and Evolutionary Computation Conference Companion (GECCO '23 Companion), July 15–19, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3583133.3596339

1 INTRODUCTION

Carbon neutral has attracted attention, and its activity has been growing rapidly. New sustainable schemes to reduce CO_2 emissions and electricity consumption are required. We have been focused on office buildings as a partially controllable environment where office workers spend a lot of time together as a small society [5, 7, 8]. Our main concern is reducing the electricity consumption of the building facilities, such as air conditioning, lighting, and ventilation. However, an excessive reduction in electricity consumption deteriorates the comfort of the office workers in the building, and it harms productivity. Therefore, we have aimed to reduce electricity consumption while maintaining or improving the comfort of office workers. So far, we have addressed the optimization of temperature settings in an air conditioning system [7] and the optimization of air conditioning as well as lighting and ventilation systems [5, 8]. These two cases employed evolutionary algorithms because they have multiple objectives, and each objective function involves a complex building simulation that must be treated as a black box.

The goal of multi-objective optimization is to acquire Pareto optimal solutions, known as the Pareto set, representing the Pareto front, which is the optimal trade-off among objectives in the objective space. The Pareto set may involve infinite solutions depending on the problem. Generally, the larger the number of solutions obtained, the higher the Pareto front approximation. However, most real-world optimization problems require repetitive executions of computationally expensive objective functions, and it is hard to obtain a large number of evaluated solutions. In addition, since evolutionary variations such as crossover and mutation involve randomness, it is not rare to be disappointed that evaluated solutions taking a long evaluation time are inferior. It is desirable to avoid evaluating inferior solutions as much as possible and efficiently improve the quality of the Pareto front approximation.

In order to efficiently improve the Pareto front approximation quality by suppressing to obtain inferior solutions and encouraging to obtain non-inferior solutions, a supervised multi-objective optimization algorithm (SMOA) was proposed [12]. SMOA is a data-driven optimization method. SMOA assumes that several good solutions are known in advance and utilizes them as training data. SMOA estimates the Pareto front by smoothly complementing the known solutions in the objective space using the Kriging interpolation [10]. SMOA also estimates the Pareto set by smoothly complementing the known solutions in the variable space using the Kriging interpolation [10]. SMOA maintains every relationship between each point in the estimated Pareto front and its corresponding point in the Pareto set. SMOA samples well-distributed points upconverting the Pareto front approximation from the estimated Pareto front and executes the objective functions to their corresponding points in the variable space. Thus, SMOA attempts to obtain promising solutions to improve the quality of the Pareto front approximation without depending on evolutionary variations involving randomness. It was reported that SMOA could efficiently improve the Pareto front quality compared to evolutionary variations in the DTLZ test problems [4] with a limited number of solution generations [12]. However, the effects of SMOA in real-world scenarios still need to be clarified. In addition, the method to sample points from the estimated Pareto front to be evaluated has an impact on the Pareto front upconvert. Although the quality of generated solutions is affected by the accuracies of the estimated Pareto front and estimated Pareto set, it is important to sample points from the estimated Pareto front appropriately.

In this work, we focus on the multi-objective building facility control problem as a real-world problem and verify the effects of SMOA on it. In addition, we propose and compare four sampling methods to select points from the estimated Pareto front to be evaluated. First, we apply a multi-objective evolutionary algorithm to the multi-objective building facility control problem and obtain a set of solutions. We use them as known solutions and execute SMOA. To sample promising solutions from the estimated Pareto front, we propose four methods: Method 1 is a confidence interval-based iterative sampling, Method 2 is a crowding distance-based one-time sampling, Method 3 is a crowding distance-based iterative sampling, and Method 4 is a truncation method based one-time sampling. We compare the Pareto front approximation qualities by the above four SMOAs with Methods 1–4 and an evolutionary solution generation using crossover and mutation.

2 MULTI-OBJECTIVE OPTIMIZATION PROBLEM WITH CONSTRAINTS

For given variable space $\mathcal{X}$ and d dimensional variable vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_d) \in \mathcal{X}$, a multi-objective optimization problem with m objective functions f_i $(i = 1, 2, \ldots, m)$ and k constraint functions g_j $(j = 1, 2, \ldots, k)$ is given by

$$\begin{cases} \text{Minimize} & f_i(\boldsymbol{x}) \ (i = 1, 2, \ldots, m) \\ \text{Subject to} & g_j(\boldsymbol{x}) \leq 0 \ (j = 1, 2, \ldots, k). \end{cases} \tag{1}$$

For each constraint j, $\omega_j(\boldsymbol{x}) = \max\{0, g_j(\boldsymbol{x})\}$ is considered as a constraint violation. Solution $\boldsymbol{x}$ satisfying $\omega_j(\boldsymbol{x}) = 0$ for all $i \in \{1, 2, \ldots, k\}$ is said to be *feasible* and *infeasible* otherwise. For two feasible solutions $\boldsymbol{x}$ and $\boldsymbol{y}$, $\boldsymbol{x}$ dominates $\boldsymbol{y}$ $(\boldsymbol{x} \preceq \boldsymbol{y})$ if $\forall i \in \{1, 2, \ldots, m\} : f_i(\boldsymbol{x}) < f_i(\boldsymbol{y})$ and $\exists j \in \{1, 2, \ldots, m\} : f_j(\boldsymbol{x}) \leq f_i(\boldsymbol{y})$. In a feasible solution set $X \subseteq \mathcal{X}$, non-dominated solutions are given by $\{\boldsymbol{x} \in X \mid \nexists \boldsymbol{y} \in X : \boldsymbol{x} \preceq \boldsymbol{y}\}$. The task is to find the set of Pareto optimal solutions, the *Pareto set*, $\mathcal{X}^* = \{\boldsymbol{x} \in \mathcal{X} \mid \nexists \boldsymbol{y} \in \mathcal{X} : \boldsymbol{x} \preceq \boldsymbol{y}\}$, which is the set of the non-dominated solutions in the whole variable space $\mathcal{X}$. The *Pareto front* is given by $\mathcal{F}^* = \{\boldsymbol{f}(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{X}^*\}$, which is the set of the objective vectors of the Pareto set $\mathcal{X}^*$ and represents the optimal trade-off between the objectives.

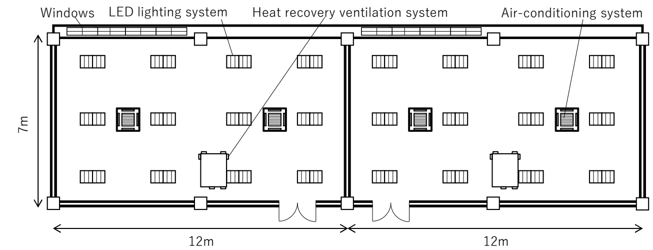


Figure 1: Target two rooms for the simulation [5, 8]

3 MULTI-OBJECTIVE BUILDING CONTROL OPTIMIZATION PROBLEM

In this study, we focus on the multi-objective building control optimization problem defined in [5, 8], which is a real-world problem with multiple objectives and constraints.

Fig. 1 shows the target two rooms in the building. The area of each room is 84 [m^2]. Each room has three facilities: air conditioning, lighting, and ventilation. This problem uses the building simulator EnergyPlus [6] for a digital twin simulation that precisely copies the target rooms in real space into digital space. The input to the simulator is a set of facility control parameters, which are the variable vector described next. EnergyPlus simulates the room environment, such as temperature, humidity, and illuminance, based on the given facility control parameters with weather conditions, and the number of office workers in the rooms. Consequently, the simulator outputs multiple quantitative values used in the objective and constraint function values.

3.1 Variables

The facility control parameters of each room are represented by an eight-dimensional real-value vector normalized in the range $[0, 1]^{d=8}$. Each value represents the start and end times of the air conditioning system, start and end times of the lighting system, temperature settings of cooling and heating, dimming rate, and ventilation level. For the two rooms, the design variable vector is represented by $\boldsymbol{x} \in \mathcal{X} = [0, 1]^{d=16}$.

3.2 Objectives

The previous study [8] addressed seven objectives. This study addresses two of them to visually observe the optimization effects in the two-dimensional objective space. The first is the annual electricity consumption f_1 and the second is environmental satisfaction $f_2(\boldsymbol{x})$. Both the objective functions should be minimized. The details can be found in [8].

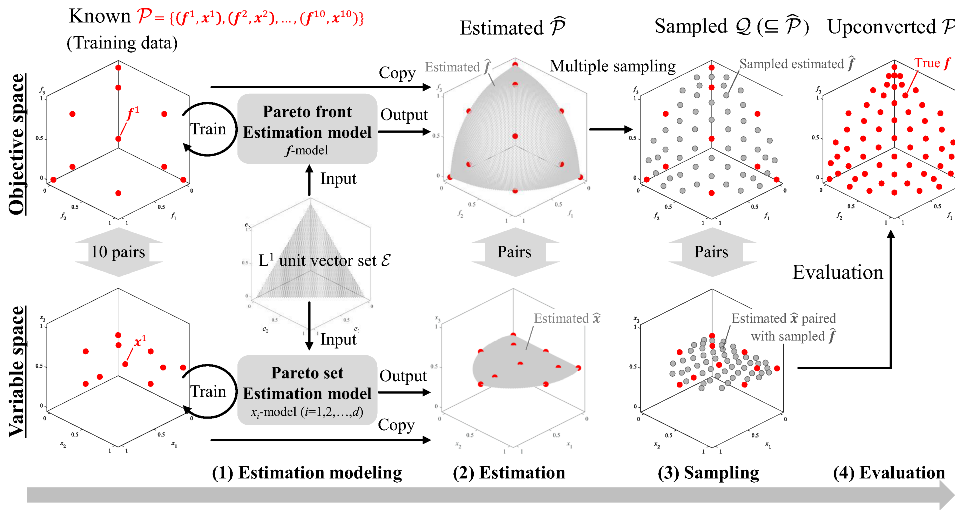


Figure 2: One-time supervised multi-objective optimization algorithm (O-SMOA) [13]

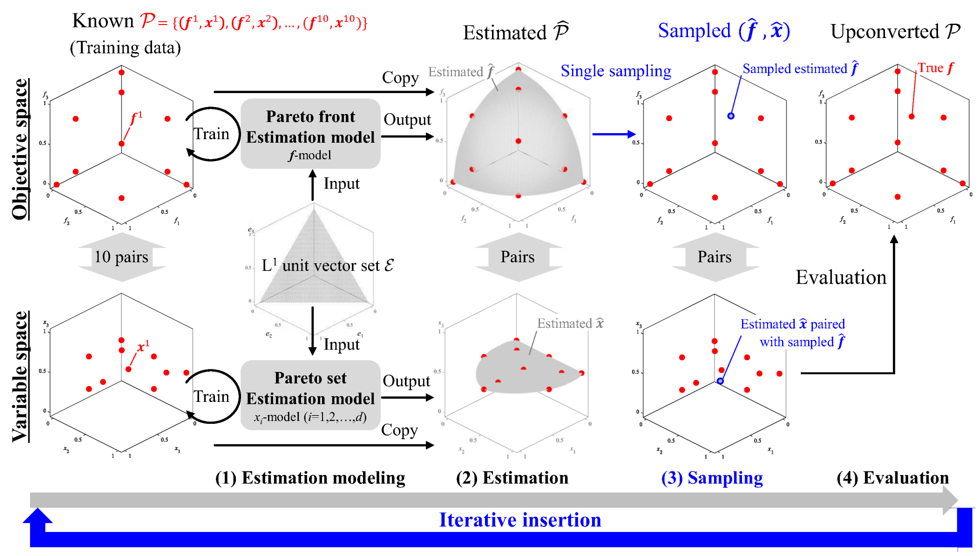


Figure 3: Iterative supervised multi-objective optimization algorithm (I-SMOA) [14]

3.3 Constraints

There are six constraints [8]. They are the upper limit of annual electricity consumption $g_1(\boldsymbol{x})$, upper and lower limits of temperature $g_2(\boldsymbol{x})$, upper and lower limits of humidity $g_3(\boldsymbol{x})$, upper limit of CO_2 concentration $g_4(\boldsymbol{x})$, lower limit of illuminance $g_5(\boldsymbol{x})$, and lighting time $g_6(\boldsymbol{x})$. The details can be found in [8].

4 PARETO FRONT AND SET ESTIMATION

4.1 Summary

SMOA uses Pareto front estimation and Pareto set estimation using several known solutions $\mathcal{P}$ [11]. Known solutions $\mathcal{P}$ are mutually non-dominated feasible ones that have already been evaluated and exhibit good objective values. Each member of $\mathcal{P}$ is represented as $(\boldsymbol{f}, \boldsymbol{x})$, which is a pair of an objective vector $\boldsymbol{f}$ and a variable vector $\boldsymbol{x}$. The known solution set is represented as $\mathcal{P} = \{(\boldsymbol{f}^1, \boldsymbol{x}^1), (\boldsymbol{f}^2, \boldsymbol{x}^2), \ldots\}$.

4.2 Pareto Front Estimation

Pareto front estimation first converts the objective vector $\boldsymbol{f}$ of each known solution $(\boldsymbol{f}, \boldsymbol{x}) \in \mathcal{P}$ into an L^1 norm $n = \sum_{i=1}^{m} f_i$ and an L^1 unit vector $\boldsymbol{e} = \boldsymbol{f}/n$. $\boldsymbol{e}$ is the direction of the objective vector $\boldsymbol{f}$ in the objective space. n is the distance to the objective vector $\boldsymbol{f}$ in direction $\boldsymbol{e}$. The Pareto front estimation builds $\boldsymbol{f}$-model based on the Kriging method [10] using $\boldsymbol{e}$ as the input and n as the output in the known solution set $\mathcal{P}$. For any L^1 unit vector $\boldsymbol{e}$, $\boldsymbol{f}$-model can output the estimated L^1 norm $\hat{n}$, and it can be converted to the estimated objective vector $\hat{\boldsymbol{f}}$ ($= \hat{n} \cdot \boldsymbol{e}$). We input a large L^1 unit

Algorithm 1 O-SMOA based on One-time Sampling [13]

Require: Known candidate solutions (training data) $\mathcal{P} = \{(\boldsymbol{f}^1, \boldsymbol{x}^1), (\boldsymbol{f}^2, \boldsymbol{x}^2), \dots\}$, L^1 unit vector set $\mathcal{E} = \{\boldsymbol{e}^1, \boldsymbol{e}^2, \dots\}$, size of upconvert solution set N
Ensure: Upscaled solution set $\mathcal{P} \cup \mathcal{P}'$

$Q \leftarrow \emptyset$ ▷ Sampled estimated solutions
$\mathcal{P}' \leftarrow \emptyset$ ▷ Newly evaluated solutions

(1) Estimation Modeling
$\boldsymbol{f}$-model $\leftarrow$ Train Pareto front estimation model ($\mathcal{P}$)
for $i \leftarrow 1, 2, \dots, d$ **do**
  x_i-model $\leftarrow$ Train Pareto set estimation model ($\mathcal{P}$)
end for
(2) Estimation
$\hat{\mathcal{P}} \leftarrow \emptyset$
for all $\boldsymbol{e} \in \mathcal{E}$ **do**
  $\hat{\boldsymbol{f}} \leftarrow \boldsymbol{f}$-model ($\boldsymbol{e}$)
  for $i \leftarrow 1, 2, \dots, d$ **do**
    $\hat{x_i} \leftarrow x_i$-model ($\boldsymbol{e}$)
  end for
  $\hat{\mathcal{P}} \leftarrow \hat{\mathcal{P}} \cup \{(\hat{\boldsymbol{f}}, \hat{\boldsymbol{x}} = (\hat{x_1}, \hat{x_2}, \dots, \hat{x_d}))\}$
end for
(3) Sampling
$Q \leftarrow$ Multiple sampling$(\mathcal{P}, \hat{\mathcal{P}}, N)$
(4) Evaluation
for all $(\hat{\boldsymbol{f}}, \hat{\boldsymbol{x}}) \in Q$ **do**
  $\boldsymbol{f} \leftarrow$ Evaluate ($\hat{\boldsymbol{x}}$)
  $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{(\boldsymbol{f}, \hat{\boldsymbol{x}})\}$
end for
return $\mathcal{P} \cup \mathcal{P}'$

vector set $\mathcal{E} = \{\boldsymbol{e}^1, \boldsymbol{e}^2, \dots\}$ into the $\boldsymbol{f}$-model, obtain the estimated L^1 norm set $\{\hat{n^1}, \hat{n^2}, \dots\}$, and convert it to the estimated objective vector set $\{\hat{\boldsymbol{f}}^1 (= \hat{n^1} \cdot \boldsymbol{e}^1), \hat{\boldsymbol{f}}^2 (= \hat{n}^2 \cdot \boldsymbol{e}^2), \dots\}$, which is the estimated Pareto front.

4.3 Pareto Set Estimation

Pareto set estimation focuses on each of variable elements x_i ($i = 1, 2, \dots, d$) in $\boldsymbol{x}$. We build x_i-model based on the Kriging method using $\boldsymbol{e}$ as the input and x_i as the output in the known solution set $\mathcal{P}$. For any L^1 unit vector $\boldsymbol{e}$, x_i-model can output the estimated variable value $\hat{x_i}$. d models, x_i-model ($i = 1, 2, \dots, d$), can output the estimated variable vector $\hat{\boldsymbol{x}} = (\hat{x_1}, \hat{x_2}, \dots, \hat{x_d})$ for any direction $\boldsymbol{e}$ in the objective space. We input a large L^1 unit vector set $\mathcal{E} = \{\boldsymbol{e}^1, \boldsymbol{e}^2, \dots\}$ into the x_i-model ($i = 1, 2, \dots, d$), and obtain the estimated variable vector set $\{\hat{\boldsymbol{x}^1}, \hat{\boldsymbol{x}^2}, \dots\}$, which is the estimated Pareto set.

4.4 Estimated Solution Set

For the known solution set $\mathcal{P}$ and a large L^1 unit vector set $\mathcal{E} = \{\boldsymbol{e}^1, \boldsymbol{e}^2, \dots\}$, the Pareto front estimation outputs the estimated objective vector set $\{\hat{\boldsymbol{f}}^1, \hat{\boldsymbol{f}}^2, \dots\}$, and the Pareto set estimation outputs the estimated variable vector set $\{\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \dots\}$. The estimated solution set is represented by $\hat{\mathcal{P}} = \{(\hat{\boldsymbol{f}}^1, \hat{\boldsymbol{x}}^1), (\hat{\boldsymbol{f}}^2, \hat{\boldsymbol{x}}^2), \dots\}$.

Algorithm 2 I-SMOA based on Iterative Sampling [14]

Require: Known candidate solutions (training data) $\mathcal{P} = \{(\boldsymbol{f}^1, \boldsymbol{x}^1), (\boldsymbol{f}^2, \boldsymbol{x}^2), \dots\}$, L^1 unit vector set $\mathcal{E} = \{\boldsymbol{e}^1, \boldsymbol{e}^2, \dots\}$, size of upconvert solution set N
Ensure: Upscaled solution set $\mathcal{P} \cup \mathcal{P}'$

$Q \leftarrow \emptyset$ ▷ Sampled estimated solutions
$\mathcal{P}' \leftarrow \emptyset$ ▷ Newly evaluated solutions
for each $1, 2, \dots, N$ **do**
  (1) Estimation Modeling
  $\boldsymbol{f}$-model $\leftarrow$ Train Pareto front estimation model ($\mathcal{P} \cup \mathcal{P}'$)
  for $i \leftarrow 1, 2, \dots, d$ **do**
    x_i-model $\leftarrow$ Train Pareto set estimation model ($\mathcal{P} \cup \mathcal{P}'$)
  end for
  (2) Estimation
  $\hat{\mathcal{P}} \leftarrow \emptyset$
  for all $\boldsymbol{e} \in \mathcal{E}$ **do**
    $\hat{\boldsymbol{f}} \leftarrow \boldsymbol{f}$-model ($\boldsymbol{e}$)
    for $i \leftarrow 1, 2, \dots, d$ **do**
      $\hat{x_i} \leftarrow x_i$-model ($\boldsymbol{e}$)
    end for
    $\hat{\mathcal{P}} \leftarrow \hat{\mathcal{P}} \cup \{(\hat{\boldsymbol{f}}, \hat{\boldsymbol{x}} = (\hat{x_1}, \hat{x_2}, \dots, \hat{x_d}))\}$
  end for
  (3) Sampling
  $Q \leftarrow Q \cup (\hat{\boldsymbol{f}}, \hat{\boldsymbol{x}}) \leftarrow$ Single sampling$(\mathcal{P}, \hat{\mathcal{P}}, \mathcal{P}')$
  (4) Evaluation
  $\boldsymbol{f} \leftarrow$ Evaluate ($\hat{\boldsymbol{x}}$)
  $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{(\boldsymbol{f}, \hat{\boldsymbol{x}})\}$
end for
return $\mathcal{P} \cup \mathcal{P}'$

5 SUPERVISED MULTI-OBJECTIVE OPTIMIZATION ALGORITHM (SMOA)

We describe two SMOAs: the one-time SMOA (O-SMOA) [13] and the iterative SMOA (I-SMOA) [14].

5.1 One-time SMOA (O-SMOA)

O-SMOA estimates the Pareto front and set, samples N estimated solutions at a time, and evaluates them [13]. **Fig. 2** shows a conceptual figure with $m = 3$ objectives and $d = 3$ variables. **Algorithm 1** shows the pseudo code. O-SMOA has four processes: (1) estimation modeling, (2) estimation, (3) sampling, and (4) evaluation.

The input is a known solution set $\mathcal{P} = \{(\boldsymbol{f}^1, \boldsymbol{x}^1), (\boldsymbol{f}^2, \boldsymbol{x}^2), \dots\}$.

In (1) estimation modeling, we generate the Pareto front estimation model $\boldsymbol{f}$-model and the Pareto set estimation model x_i-model ($i = 1, 2, \dots, d$) using the known solution set $\mathcal{P}$ according to the procedure described in the previous section. In **Fig. 2**, $|\mathcal{P}| = 10$ known solutions are shown in red.

In (2) estimation, we input a large L^1 unit vector set $\mathcal{E} = \{\boldsymbol{e}^1, \boldsymbol{e}^2, \dots\}$ into $\boldsymbol{f}$-model and x_i-model ($i = 1, 2, \dots, d$) and obtain the estimated solution set $\hat{\mathcal{P}} = \{(\hat{\boldsymbol{f}}^1, \hat{\boldsymbol{x}}^1), (\hat{\boldsymbol{f}}^2, \hat{\boldsymbol{x}}^2), \dots\}$. In **Fig. 2**, each estimated objective vector $\hat{\boldsymbol{f}}$ and estimated variable vector $\hat{\boldsymbol{x}}$ is shown in gray.

In (3) sampling, we sample N estimated solutions from the estimated solution set $\hat{\mathcal{P}}$ as the sampled estimated solution set Q, i.e.,

$Q \subseteq \hat{\mathcal{P}}$. Note that the sampling method of Q from $\hat{\mathcal{P}}$ affects the quality of the Pareto front approximation.

In (4) evaluation, we execute the objective functions f_i ($i = 1, 2, \ldots, m$) to the estimated variable vector $\hat{x}$ of each sampled estimated solution $(\hat{f}, \hat{x}) \in Q$ and add a pair of obtained objective and variable vectors $(f, \hat{x})$ to the newly generated solution set $\mathcal{P}'$.

5.2 Iterative SMOA (I-SMOA)

Aforementioned O-SMOA samples N estimated solutions from the estimated solution set $\hat{\mathcal{P}}$ at a time and evaluates them. On the other hand, I-SMOA described here samples a single estimated solution from the estimated solution set $\hat{\mathcal{P}}$ at a time and evaluates it. I-SMOA then re-constructs the estimation models using the newly generated solution. I-SMOA repeats this process N times. **Fig. 3** shows the conceptual figure. **Algorithm 2** shows the pseudo code.

There are three differences between **Algorithm 1** of aforementioned O-SMOA and **Algorithm 2** of I-SMOA described here. First, I-SMOA repeats processes (1)–(4) N times, whereas O-SMOA executes processes (1)–(4) once. Second, in (1) estimation modeling, I-SMOA builds estimation models using the known solution set $\mathcal{P}$ and generated solutions $\mathcal{P}'$ whereas O-SMOA builds estimation models using the known solutions $\mathcal{P}$ only. Third, in (3) sampling, I-SMOA samples a single solution, whereas O-SMOA samples N solutions.

Since I-SMOA builds estimation models N times, its computational cost is higher than that of O-SMOA. However, the accuracy of the estimation models of I-SMOA can be expected to increase gradually during the repetition, and the difference between the estimated and true objective vectors of the sampled solution can be expected to decrease. We anticipate this contributes to improving the accuracy of the newly generated solution set $\mathcal{P}'$.

6 PROPOSAL: SAMPLING METHODS OF ESTIMATED SOLUTIONS

6.1 Summary

In this work, we propose and compare four sampling methods of estimated solutions Q from the estimated solution set $\hat{\mathcal{P}}$ in (3) sampling process of two SMOAs of **Algorithm 1** and **Algorithm 2**, respectively. Four methods are summarized in **Table 1**. Each method is based on O-SMOA of **Algorithm 1** or I-SMOA of **Algorithm 2** and calls its own sampling method at (3) sampling process. Method 1 is a confidence interval-based iterative sampling, Method 2 is a crowding distance-based one-time sampling, Method 3 is a crowding distance-based iterative sampling, and Method 4 is a truncation method-based one-time sampling. Details are described below.

6.2 Method 1: Confidence Interval-Based Iterative Sampling

The Kriging-based estimation model can output the estimated value and its confidence interval [10]. Method 1 takes an iterative sampling that samples a single estimated solution with the largest confidence interval, evaluates it, and updates the estimation models to reduce the confidence interval of the estimation model.

Table 1: Four SMOA Variants

	SMOA		
	One-time **Algorithm 1**	Iterative **Algorithm 2**	(3) sampling
Method 1	-	✓	**Algorithm 3**
Method 2	✓	-	**Algorithm 4**
Method 3	-	✓	**Algorithm 5**
Method 4	✓	-	**Algorithm 6**

Method 1 is based on **Algorithm 2** of I-SMOA. In (3) sampling process in blue, Method 1 calls **Algorithm 3**. In **Algorithm 3**, Method 1 finds the estimated solution $(\hat{f}, \hat{x})$ with the maximum confidence interval of L^1 norm $\hat{n}$ of $\hat{f}$ in the estimated solution set $\hat{\mathcal{P}}$ and returns it as the sampled result.

6.3 Method 2: Crowding Distance-Based One-Time Sampling

Method 2 takes a one-time sampling that samples N estimated solutions at a time based on the crowding distance used in NSGA-II [3], which is a diversity maintenance criterion for the solutions in the objective space.

Method 2 is based on **Algorithm 1** of O-SMOA. In (3) sampling process in red, Method 2 calls **Algorithm 4**. In **Algorithm 4**, the estimated solution set $\hat{\mathcal{P}}$ is copied to the sampled estimated solution set Q; note $|\hat{\mathcal{P}}| = |Q| \gg N$ at this time. Method 2 finds the estimated solution $(\hat{f}, \hat{x}) \in Q$ with the shortest crowding distance in the combined set of the known solution set $\mathcal{P}$ and Q and deletes it from Q, i.e., $Q = Q \backslash (\hat{f}, \hat{x})$. The purge of a single estimated solution from Q is repeated until its size $|Q|$ gets N. Method 2 returns Q of size N as the sample result.

6.4 Method 3: Crowding Distance-Based Iterative Sampling

Method 3 also uses the crowding distance but takes an iterative sampling that samples a single estimated solution at a time.

Method 3 is based on **Algorithm 2** of I-SMOA. In (3) sampling process in blue, Method 3 calls **Algorithm 5**. In **Algorithm 5**, Method 3 finds and returns the estimated solution $(\hat{f}, \hat{x}) \in \hat{\mathcal{P}}$ with the longest crowding distance in the combination set of the known solution set $\mathcal{P}$, the newly evaluated solution set $\mathcal{P}'$, and $(\hat{f}, \hat{x})$ as the sample result.

6.5 Method 4: Truncation Method-Based One-Time Sampling

Method 4 takes a one-time sampling that samples N estimated solutions at a time based on the truncation method used in SPEA2 [17], which is an alternative diversity maintenance criterion.

Method 4 is based on **Algorithm 1** of O-SMOA. In (3) sampling process in red, Method 4 calls **Algorithm 6**. In **Algorithm 6**, the estimated solution set $\hat{\mathcal{P}}$ is copied to the sampled estimated solution set Q; note $|\hat{\mathcal{P}}| = |Q| \gg N$ at this time. Method 4 finds the estimated solution $(\hat{f}, \hat{x}) \in Q$ to be truncated from Q and deletes it from Q, i.e., $Q = Q \backslash (\hat{f}, \hat{x})$. In the truncation method [17], we first find two

Algorithm 3 Method 1: Confidence Interval Based Iterative Sampling

Require: Known candidate solutions (training data) $\mathcal{P} = \{(f^1, x^1), (f^2, x^2), \dots\}$, Estimated solution set $\hat{\mathcal{P}} = \{(\hat{f}^1, \hat{x}^1), (\hat{f}^2, \hat{x}^2), \dots\}$, Newly evaluated solution set $\mathcal{P}' = \{(f^{1\prime}, x^{1\prime}), (f^{2\prime}, x^{2\prime}), \dots\}$
Ensure: Single sampled estimated solution $(\hat{f}, \hat{x})$

```
function SINGLE SAMPLING(𝒫, 𝒫̂, 𝒫′)
    return (f̂, x̂) ← argmax_{(f̂,x̂)∈𝒫̂} Confidence interval of f̂
end function
```

Algorithm 4 Method 2: Crowding Distance Based One-Time Sampling

Require: Known candidate solutions (training data) $\mathcal{P} = \{(f^1, x^1), (f^2, x^2), \dots\}$, Estimated solution set $\hat{\mathcal{P}} = \{(\hat{f}^1, \hat{x}^1), (\hat{f}^2, \hat{x}^2), \dots\}$, Sample size N
Ensure: Sampled estimated solutions Q

```
function MULTIPLE SAMPLING(𝒫, 𝒫̂, N)
    Q ← 𝒫̂
    while |Q| > N do
        (f̂, x̂) ← argmin_{(f̂,x̂)∈Q} CD of f̂ in 𝒫 ∪ Q
        Q ← Q\{(f̂, x̂)}
    end while
    return Q
end function
```

solutions with the shortest Euclidean distance between them in the objective space from Q. For each of the two found solutions, we compute the distance to the second nearest solution and take one with a shorter distance than another to be truncated. If two distances to their second nearests are the same, we compute the distances to their third nearests and so on. The purge of a single solution from Q is repeated until its size $|Q|$ gets N. Method 4 returns Q with size N as the sample result.

7 EXPERIMENTAL SETTINGS

7.1 Problem

We used the multi-objective building facility control problem with two objectives: annual electricity consumption $f_1(x)$ and environmental satisfaction $f_2(x)$.

As the known solution set $\mathcal{P}$, we used $|\mathcal{P}| = 15$ non-dominated solutions obtained by executing the indicator-based evolutionary algorithm (IBEA) [8, 16] to solve the problem. Note that we cannot avoid executing evolutionary optimization to obtain non-dominated solutions $\mathcal{P}$ used as the input of SMOA since the multi-objective building facility control problem is a black box problem. Note that the role of SMOA is to efficiently upconvert the Pareto front representation of the known non-dominated solutions $\mathcal{P}$.

7.2 Algorithms

We compared five solution generation methods.

Algorithm 5 Method 3: Crowding Distance Based Iterative Sampling

Require: Known candidate solutions (training data) $\mathcal{P} = \{(f^1, x^1), (f^2, x^2), \dots\}$, Estimated solution set $\hat{\mathcal{P}} = \{(\hat{f}^1, \hat{x}^1), (\hat{f}^2, \hat{x}^2), \dots\}$, Newly evaluated solution set $\mathcal{P}' = \{(f^{1\prime}, x^{1\prime}), (f^{2\prime}, x^{2\prime}), \dots\}$
Ensure: Single sampled estimated solution $(\hat{f}, \hat{x})$

```
function SINGLE SAMPLING(𝒫, 𝒫̂, 𝒫′)
    return (f̂, x̂) ← argmax_{(f̂,x̂)∈𝒫̂} CD of (f̂, x̂) in 𝒫 ∪ 𝒫′ ∪ (f̂, x̂)
end function
```

Algorithm 6 Method 4: Truncation Method Based One-Time Sampling

Require: Known candidate solutions (training data) $\mathcal{P} = \{(f^1, x^1), (f^2, x^2), \dots\}$, Estimated solution set $\hat{\mathcal{P}} = \{(\hat{f}^1, \hat{x}^1), (\hat{f}^2, \hat{x}^2), \dots\}$, Sample size N
Ensure: Sampled estimated solutions Q

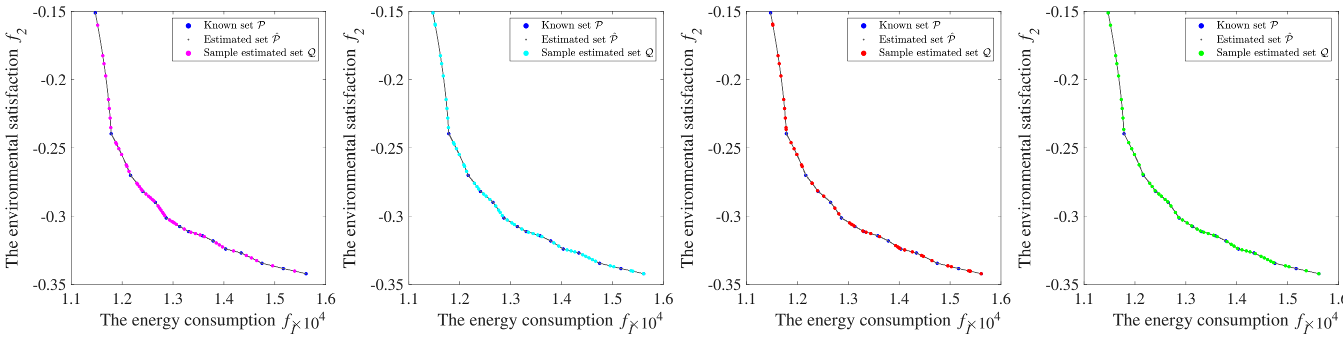
```
function MULTIPLE SAMPLING(𝒫, 𝒫̂, N)
    Q ← 𝒫̂
    while |Q| > N do
        (f̂, x̂) ← Truncation(Q)
        Q ← Q\{(f̂, x̂)}
    end while
    return Q
end function
```

The first method is based on evolutionary variations. We computed the IBEA-based fitness [16] for each known solution in $\mathcal{P}$. We selected two parents based on a binary tournament selection using fitness and applied the simulated binary crossover (crossover ratio 1.0, distribution index 20) and the polynomial mutation (mutation ratio $1/d$=0.0625, distribution index 20) [2]. We repeated it until obtaining $N = 50$ of newly generated solutions $\mathcal{P}'$. It should be noted that this method involves randomness.

The remaining four methods are SMOA variants, Methods 1–4. Each method obtained $N = 50$ of newly generated solutions $\mathcal{P}'$. As a large L^1 unit vector set $\mathcal{E}$ for the Pareto front and Pareto set estimations, we employed $|\mathcal{E}| = 10,001$ of L^1 unit vectors generated by the simplex lattice design [1] with division parameter $H = 10,000$. That is, the upconverted Pareto front and Pareto set were respectively represented by $|\mathcal{E}| = 10,001$ points. We utilized the MATLAB Kriging Toolbox [9] and PlatEMO [15] for the implementation.

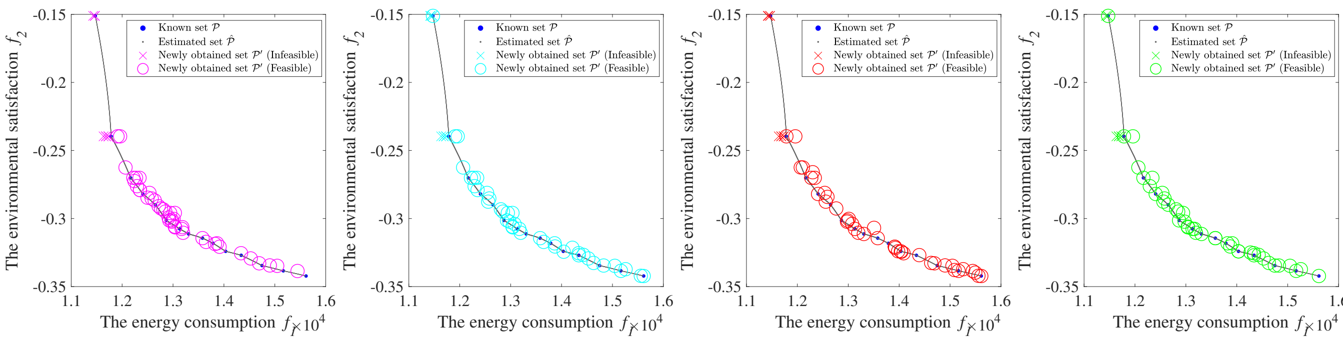
7.3 Metric

Hypervolume (HV) [18] was used to assess the quality of the Pareto front approximation. HV treating $m = 2$ objectives is an area enclosed by solutions for the Pareto front approximation and a dominated reference point $\boldsymbol{r}$ in the objective space. The larger the HV, the better the Pareto front approximation. In general, the convergence, diversity, uniformity, and number of solutions affect HV. In particular, in this study, the last three aspects strongly affect HV because the convergence is almost satisfied by the known solution set $\mathcal{P}$. In this study, we normalized the objective values into



(a) Method 1 **(b) Method 2** **(c) Method 3** **(d) Method 4**

Figure 4: Distribution of sampled estimated solutions $\mathcal{Q}$ by SMOA variants



(a) Method 1 **(b) Method 2** **(c) Method 3** **(d) Method 4**

Figure 5: Distribution of the newly generated solutions $\mathcal{P}'$ by SMOA variants

$[0,1]^{m=2}$ involving all the obtained solutions and calculated HV with $\boldsymbol{r} = (1.0, 1.0)$.

8 RESULTS AND DISCUSSION

8.1 Distribution of Sampled Estimated Solutions

In **Figs. 4 (a)**–**4 (d)**, the known solution set $\mathcal{P}$ is in blue, the estimated Pareto front $\hat{\mathcal{P}}$ is in gray, the sampled solution set $\mathcal{Q}$ of each of Methods 1–4 is respectively in different colors.

Fig. 4 (a) shows that most of the sampled estimated solutions $\mathcal{Q}$ by Method 1 are distributed in the central area of the Pareto front. Confidence levels in the central area of the convex Pareto front tend to be high [11] since the Pareto front is estimated using L^1 unit vectors. This would affect the result in this work.

Figs. 4 (b) and **4 (c)** show that the sampled estimated solutions $\mathcal{Q}$ by Methods 2 and 3 are distributed in the less-crowded areas of the known solution set $\mathcal{P}$. Methods 2 and 3 sample the estimated solutions $\mathcal{Q}$ from the estimated solution set $\hat{\mathcal{P}}$ by considering the distribution of the known solution set $\mathcal{P}$. However, we see that Method 3 intensively samples estimated solutions around $(f_1, f_2) = (1.3, -0.31)$, $(f_1, f_2) = (1.4, -0.325)$ in the objective space. Method 3 iteratively samples a single estimated solution while updating the estimation models with its evaluated one. If the sampled estimated objective vector and its true objective vector are distanced due to the accuracy of the estimation models, these intensive samplings arise.

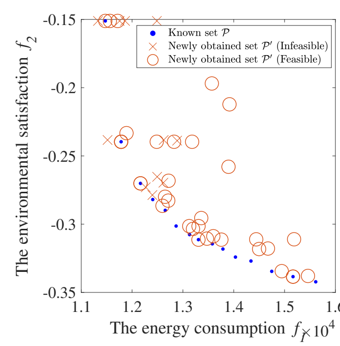


Figure 6: Distribution of the newly generated solutions $\mathcal{P}'$ by evolutionary variations

Fig. 4 (d) shows that the sampled estimated solutions $\mathcal{Q}$ by Method 4 are distributed with a high uniformly for the known solution set $\mathcal{P}$ in the objective space. This is because Method 4 uniformly samples the estimated solutions $\mathcal{Q}$ from the estimated solution set $\hat{\mathcal{P}}$ without considering the distribution of the known solution set $\mathcal{P}$.

Table 2: HV of newly generated set $\mathcal{P}'$

Method 1	Method 2	Method 3	Method 4	Evolutionary variations
0.71864	0.76663	0.73466	**0.76839**	0.75681

Table 3: HV of known and newly generated sets $\mathcal{P} \cup \mathcal{P}'$

Method 1	Method 2	Method 3	Method 4	Evolutionary variations
0.76848	**0.76852**	0.76838	0.76841	0.76806

8.2 Distribution of Newly Generated Solutions

Figs. 5 (a)–**5 (d)** shows $N = 50$ of newly generated solutions $\mathcal{P}'$ by SMOA variants respectively with Methods 1–4. **Fig. 6** shows $N = 50$ of newly generated solutions $\mathcal{P}'$ by evolutionary variations of crossover and mutation. ◯ are feasible solutions and × are infeasible solutions.

Fig. 6 shows that the evolutionary variations generate many solutions dominated by the known solution set $\mathcal{P}$. This is because crossover and mutation have randomness, even parents in the known solution set $\mathcal{P}$ have good objective values.

Figs. 5 (a)–**5 (d)** show that SMOAs generate dominated solutions less than those of the evolutionary variations. SMOAs generate many non-dominated solutions that are distributed around the known solution set $\mathcal{P}$. These results reveal that SMOAs can generate well-distributed solutions upconverting the Pareto front representation with a limited number of solution generations compared to evolutionary variations. However, even SMOA generates infeasible solutions. This suggests that the multi-objective building facility control problem has promising areas in terms of objective values around the border of feasible and infeasible areas in the variable space. Another reason is that SMOAs do not have any special mechanism to obtain feasible solutions.

From **Figs. 4 (a)**–**4 (d)** and **Figs. 5 (a)**–**5 (d)**, we see that distributions between the sampled estimated solutions and their evaluated solutions are distanced. Especially in the range of $f_2 = [-0.24, -0.15]$, we sampled estimated solutions Q as shown in **Figs. 4 (a)**–**4 (d)**. However, their evaluated solutions are not there actually as shown in **Figs. 5 (a)**–**5 (d)**. Thus, the accuracy of the estimation models depends on each part of the Pareto front and the Pareto set.

8.3 Quantitative Comparison Using HV

Table 2 shows HV of newly generated solution set $\mathcal{P}'$ without the known solution set $\mathcal{P}$. **Table 3** shows HV of the combined set $\mathcal{P} \cup \mathcal{P}'$ of the known solution set $\mathcal{P}$ and the newly generated solution set $\mathcal{P}'$. Only for evolutionary variations, the tables show the average values of 31 runs.

From **Table 2**, we can see that Method 4 using the SPEA2-based truncation mechanism achieves the highest HV among the five methods when we only assess the newly generated solution set $\mathcal{P}'$. This is because Method 4 samples the estimated solutions without considering the distribution of the known solution set $\mathcal{P}$, which is distributed biasedly in the objective space. Methods 1 and 3 show lower HV values than that of evolutionary variations since the newly generated solution with the minimum f_1 was infeasible as shown in **Figs. 5 (a)** and **5 (c)**, respectively. Although effects of Methods 1 and 3 may be under-evaluated from their HV values due to the infeasible solution with the minimum f_1, we see their newly generated solutions are well-distributed around the rage $f_2 = [-0.24, -0.35]$ respectively shown in **Figs. 5 (a)** and **5 (c)** compared to those of evolutionary variations shown in **Fig. 6**.

From **Table 3**, we can see that HV values of Methods 1–4 based on SMOA are higher than those of the evolutionary variations. This result quantitatively reveals that SMOA-based solution generations can efficiently generate better solutions than evolutionary variations involving randomness in the multi-objective building facility control problem with the known solution set $\mathcal{P}$. Also, we can see that Method 2 achieves the highest HV among the five methods when we assess the combined set $\mathcal{P} \cup \mathcal{P}'$ of the known solution set $\mathcal{P}$ and the newly generated solution set $\mathcal{P}'$. Methods 2, which considers the distribution of the known solution set $\mathcal{P}$, achieves a higher HV than Method 4 without considering this. Method 2 samples estimated solutions Q all at once, and Method 3 iteratively samples a single estimated solution and updates the estimation models. Although Method 3 is expected to be better in terms of algorithmic aspects, Method 2 is superior to Method 3 quantitatively in this case. The number of known solutions and the accuracy of the estimation models would affect the results. In Method 3, if the true objective vector of the sampled estimated objective vector is distanced, it would result in this case.

9 CONCLUSIONS

In this work, we aimed to verify the effects of the supervised multi-objective optimization algorithm (SMOA) in the multi-objective building facility problem as a real-world problem. Also, we proposed and compared four methods to sample promising solutions to be evaluated from the estimated Pareto front. The experimental results showed that SMOAs could generate better solutions in terms of objective value than evolutionary variations using crossover and mutation frequently used in evolutionary algorithms, and efficiently improve the quality of the Pareto front approximation with a limited number of solution generations. When we required the Pareto front approximation quality by newly generated solutions only, Method 4, based on truncation method in SPEA2 without considering the distribution of the known solution set, achieved the best. When we required the Pareto front approximation quality by the newly generated solutions and the known ones, Method 2, based on crowding distance-based one-time sampling by considering the distribution of the known solution set, achieved the best.

In future work, we will verify the effects of SMOA by varying the number of known solutions and accuracy of their estimation models. In addition, we will observe the accuracy transition of the estimation models in the iterative SMOA over time. Furthermore, we will study a method to estimate feasible regions in the variable space.

REFERENCES

[1] Indraneel Das and John E Dennis. 1998. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization* 8, 3 (1998), 631–657.

[2] Kalyanmoy Deb and Mayank Goyal. 1996. A combined genetic adaptive search (GeneAS) for engineering design. *Computer Science and informatics* 26 (1996), 30–45.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197. https://doi.org/10.1109/4235.996017

[4] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. 2005. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization*. Springer, 105–145.

[5] Kohei Fukuhara, Ryo Kumagai, Fukawa Yuta, Tanabe Shin-ichi, Hiroki Kawano, Yoshihiro Ohta, and Hiroyuki Sato. 2022. Digital Twin Based Evolutionary Building Facility Control Optimization. In *2022 IEEE Congress on Evolutionary Computation (CEC)*. 1–8. https://doi.org/10.1109/CEC55065.2022.9870207

[6] National Renewable Energy Laboratory (NREL). 2022. EnergyPuls. Retrieved Jan. 7th, 2022 from https://energyplus.net/

[7] Yoshihiro Ohta, Takafumi Sasakawa, and Hiroyuki Sato. 2020. Evolutionary Air-Conditioning optimization Using an LSTM-Based Surrogate Evaluator. In *2020 IEEE Congress on Evolutionary Computation (CEC)*. 1–8. https://doi.org/10.1109/CEC48606.2020.9185769

[8] Naru Okumura, Yoshihiro Ohta, and Hiroyuki Sato. 2022. Solution Archive and Its Re-use in Evolutionary Many-objective Facility Control Optimization. In *2022 Joint 12th International Conference on Soft Computing and Intelligent Systems and 23rd International Symposium on Advanced Intelligent Systems (SCIS&ISIS)*. 1–7. https://doi.org/10.1109/SCISISIS55246.2022.10002109

[9] Søren N. Lophaven, Hans Bruun Nielsen, and Jacob Søndergaard. 2022. DACE - A Matlab kriging toolbox. Retrieved Jan. 7th, 2022 from https://www.omicron.dk/dace.html

[10] Michael L Stein. 2012. *Interpolation of spatial data: Some theory for kriging*. Springer Science & Business Media.

[11] Tomoaki Takagi, Keiki Takadama, and Hiroyuki Sato. 2021. Pareto Front Estimation Using Unit Hyperplane. In *Evolutionary Multi-Criterion Optimization: 11th International Conference, EMO 2021, Shenzhen, China, March 28–31, 2021, Proceedings* (Shenzhen, China). Springer-Verlag, Berlin, Heidelberg, 126–138. https://doi.org/10.1007/978-3-030-72062-9_11

[12] Tomoaki Takagi, Keiki Takadama, and Hiroyuki Sato. 2022. Supervised Multi-Objective Optimization Algorithm Using Estimation. In *2022 IEEE Congress on Evolutionary Computation (CEC)*. 1–8. https://doi.org/10.1109/CEC55065.2022.9870375

[13] Tomoaki Takagi, Keiki Takadama, and Hiroyuki Sato. 2022. Supervised multi-objective optimization algorithm using estimation. In *2022 IEEE Congress on Evolutionary Computation (CEC'22)*. 1–8.

[14] Tomoaki Takagi, Keiki Takadama, and Hiroyuki Sato. 2023. Pareto Front Upconvert by Iterative Estimation Modeling and Solution Sampling. In *12th Edition of International Conference on Evolutionary Multi-Criterion Optimization (EMO'23), Lecture Notes in Computer Science (LNCS)*, Vol. 13970. 218–230.

[15] Ye Tian, Ran Cheng, Xingyi Zhang, and Yaochu Jin. 2017. PlatEMO: A MATLAB platform for evolutionary multi-objective optimization. *IEEE Computational Intelligence Magazine* 12, 4 (2017), 73–87.

[16] Eckart Zitzler and Simon Künzli. 2004. Indicator-Based Selection in Multiobjective Search. In *Parallel Problem Solving from Nature - PPSN VIII*. Springer, 832–842.

[17] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. 2001-05. *SPEA2: Improving the strength pareto evolutionary algorithm.* Report. Zurich. https://doi.org/10.3929/ethz-a-004284029

[18] Eckart Zitzler and Lothar Thiele. 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. 3, 4 (1999), 257–271.