A novel Pareto Front Symbiotic Organism Search (PF-SOS) combined with metaheuristic-optimized machine learning for optimal recycled aggregate concrete mixtures

Gunawan, Hanna Chintya Febriani; Thedy, John; Setiadji, Bagus Hario; Han, Ay Lie; Ottele, Marc

DOI
10.1016/j.jobe.2025.112991

Publication date
2025

Document Version
Final published version

Published in
Journal of Building Engineering

Citation (APA)
Gunawan, H. C. F., Thedy, J., Setiadji, B. H., Han, A. L., & Ottele, M. (2025). A novel Pareto Front Symbiotic Organism Search (PF-SOS) combined with metaheuristic-optimized machine learning for optimal recycled aggregate concrete mixtures. *Journal of Building Engineering*, *108*, Article 112991. https://doi.org/10.1016/j.jobe.2025.112991

Journal of Building Engineering 108 (2025) 112991

Contents lists available at ScienceDirect

Journal of Building Engineering

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

A novel Pareto Front Symbiotic Organism Search (PF-SOS) combined with metaheuristic-optimized machine learning for optimal recycled aggregate concrete mixtures

Hanna Chintya Febriani Gunawan [a], John Thedy [b], Bagus Hario Setiadji [a], Ay Lie Han [a,*], Marc Ottele [c]

[a] Departement of Civil Engineering, Diponegoro University, Semarang, 50275, Indonesia
[b] Departement of Bioenvironmental System Engineering, National Taiwan University, Taipei, 106319, Taiwan
[c] Microlab, Faculty of Civil Engineering and Geosciences, Delft University of Technology, 2628 CN, Delft, the Netherlands

ARTICLE INFO

Keywords:
Recycle aggregate concrete
Machine learning
Pareto front
Multi-objective optimization

ABSTRACT

Recycled Aggregate Concrete (RAC) represents a significant innovation aimed at reducing the carbon footprint in the construction industry. Over the past few decades, numerous investigations and experiments have confirmed the viability of RAC as a construction material when the optimal combination of recycled and natural aggregates is used. This study seeks to further enhance the application of RAC by providing a robust framework for determining the optimal RAC mixture. To achieve this, machine learning is developed to predict the compressive strength of RAC by considering various mixture properties. To improve the accuracy of these predictions, the Symbiotic Organism Search (SOS) metaheuristic algorithm is employed, not only to fine-tune the machine learning hyperparameters but also to select the most suitable model. In this study, the SOS algorithm is tasked with choosing between Artificial Neural Networks (ANN), Support Vector Machines (SVM), or Random Forests (RF), based on predefined upper and lower bounds for their hyperparameters. The resulting machine learning model is then integrated with the novel Pareto Front Symbiotic Organism Search (PF-SOS) to generate a Pareto front of optimal mixtures, with compressive strength and production cost as the objectives. To validate the efficiency of the proposed method, the PF-SOS results are compared with those from other well-known multi-objective optimization algorithms. The findings demonstrate that PF-SOS offers faster convergence and a broader range of mixture options within the same function evaluation limit. The supporting source codes are available at https://github.com/johnthedy/MLSOS and https://github.com/johnthedy/PFSOS.

1. Introduction

In recent decades, global efforts to reduce the carbon footprint have intensified. In the construction sector, the use of concrete as a structural material accounts for approximately 8 % of global carbon emissions [1]. Significant research and initiatives have been undertaken to reduce the environmental impact of concrete production [2,3]. One of the most promising solutions is the utilization of

* Corresponding author.
E-mail address: hanaylie@live.undip.ac.id (A.L. Han).

https://doi.org/10.1016/j.jobe.2025.112991
Received 26 November 2024; Received in revised form 17 April 2025; Accepted 23 May 2025
Available online 23 May 2025

recycled concrete materials, particularly in the form of Recycled Aggregate Concrete (RAC). This process involves crushing waste concrete from demolition or construction activities to produce aggregate material for new concrete. The use of demolished concrete as aggregate has been increasingly promoted in many countries, as it not only reduces carbon emissions but also minimizes the need for gravel extraction. In 2020, EU countries reported recycling construction and demolition waste at a rate of 89 % [4]. In Asia, Japan leads the way, with a recycling rate of 88.2 % for construction waste [5]. Although some studies [4] indicate that recycling construction waste often results in downgrading the material's value—such as converting concrete into filler aggregate—the practice still plays a crucial role in reducing the carbon footprint.

Due to its significant role in reducing carbon emissions, RAC has been extensively studied over the past few decades [6–11]. From a structural perspective, concrete can be regarded as a composite material consisting of three key components that contribute to its strength: aggregates, mortar paste (comprising cement and fine aggregates), and interfacial transition zones (ITZ) that connect the mortar and aggregates. Several studies [6,7] have reported that the ITZ often governs the failure limit state, making it the weakest link in the concrete composite system. The bond strength in the ITZ is largely influenced by the absorption level of the aggregate material, where higher permeability or absorption leads to reduced concrete strength [8–10]. In addition to strength, the permeability of aggregates significantly affects concrete durability, with research [10,11] indicating that higher permeability results in reduced durability. Guo et al. [9] identified four major factors affecting the permeability of RAC, as outlined below.

- **Recycled Aggregate Size:** The use of larger aggregate particles reduces the surface area, thereby decreasing concrete permeability. However, larger particles may also increase the occurrence of flaws within the material, meaning that aggregate size does not have a straightforward positive or negative correlation with compressive strength. Several studies [10,12] even indicated that smaller recycled aggregates tend to retain more old mortar, which affects the absorption properties of the aggregate.
- **Recycled Aggregate Source and Crushing Process:** Recycled aggregate derived from higher-strength concrete exhibits lower permeability compared to aggregate sourced from lower-strength concrete. Additionally, contamination with clay can increase the aggregate's permeability. The method of crushing also plays a role in aggregate quality [13]; for instance, longer crushing durations can reduce the amount of adhered old mortar, increasing aggregate density and reducing permeability.
- **Curing Treatment:** The curing process has a significant impact on the permeability of RAC. Extended curing times enhance concrete density, resulting in lower permeability. Moreover, increased curing duration reduces gas permeability, which improves resistance to chloride penetration, thereby increasing the concrete's durability.

In addition to permeability, the mixing process can also influence the compressive strength of RAC. Tam et al. [6] investigated the mixing procedure for RAC to achieve optimal concrete strength. The study found that RAC exhibits higher porosity, leading to greater absorption compared to conventional aggregates. To address this issue, Tam et al. proposed a two-stage mixing approach, where the first stage aims to fill the pores of the recycled aggregate. This method was shown to significantly improve the compressive strength of RAC compared to standard mixing procedures. In a subsequent study [7], Tam et al. recommended a mix composition of 30 % recycled aggregate with normal aggregate to achieve the best performance of RAC. This recommendation aligns with findings by Etxeberria et al. [14], who proposed an optimal recycled aggregate proportion of 25 %.

As previously discussed, RAC exhibits complex and highly variable mechanical properties, which can significantly influence the final quality of concrete. Unlike conventional concrete, which has a long history and well-established formulas, RAC involves greater complexity and nonlinearity that are difficult to define explicitly. Variations in the source of recycled material, crushing processes, and even mixing procedures can lead to significant differences in the final product. With advancements in technology, several researchers have employed machine learning to address the complexities of concrete mixture design [15–21]. In the context of RAC strength prediction, research innovations typically focus on new machine learning frameworks [15–19] or novel approaches to feature selection and predicted properties [20,21]. To enhance prediction robustness and accuracy, some researchers utilize metaheuristic algorithms for hyperparameter tuning [22–24]. For instance, Sun et al. [22] applied a Back Propagation Neural Network to predict concrete strength and used the Beetle Antennae Search algorithm to optimize the network architecture. Similarly, Chou et al. [23] employed metaheuristic optimization to fine-tune the hyperparameters of an Extreme Gradient Boosting (XGBoost) model for optimal performance, concluding that XGBoost combined with Symbiotic Organism Search (SOS) yields better accuracy in predicting concrete shear capacity compared to other machine learning models. Peng et al. used Particle Swarm Optimization (PSO) to optimize Support Vector Regression (SVR), resulting in improved concrete strength predictions. Furthermore, Falah et al. [25] developed a hybrid machine learning approach by combining multiple standard machine learning models with XGBoost to improve prediction accuracy. In this study, XGBoost was used to determine feature importance and identify the optimal data-splitting ratio.

The aforementioned research on utilizing machine learning primarily focuses on developing robust and accurate surrogate models for predicting the mechanical properties of concrete. While these surrogate models effectively predict concrete strength, they do not directly provide the optimal mixture composition. To generate an optimal concrete mixture based on a target strength, an additional optimization algorithm is typically paired with the trained machine learning model. Yeh et al. [26], for instance, approached concrete mixture optimization by developing a neural network surrogate model as a concrete strength predictor. The concrete mixture design was formulated as an optimization problem, where both strength and cost were set as optimization objectives, and the upper and lower bounds of the concrete composition and mechanical properties were imposed as constraints. A nonlinear programming algorithm was then used to solve this optimization problem. Similarly, Cheng et al. [27] proposed a method to determine the optimal mixture for high-performance concrete by combining machine learning with a metaheuristic approach. Cheng et al. introduced an enhanced version of the Genetic Algorithm, known as the K-Clustering Genetic Algorithm (KCGA), for optimizing the mixture, while Strength Prediction within the KCGA employed Support Vector Regression (SVR) with Genetic Algorithm (GA) for hyperparameter tuning.

Zhang et al. [28] also developed a framework to determine the optimal concrete mixture by integrating machine learning with metaheuristic algorithms. In their approach, Artificial Neural Networks (ANN) and Random Forest (RF) models, optimized using Particle Swarm Optimization (PSO), were used to predict compressive strength and concrete slump, respectively. The optimal mixture was then determined using Multi-Objective Particle Swarm Optimization (MOPSO), with compressive strength and slump requirements as dual objectives.

As observed from the references, the majority of studies aimed at developing optimal mixture algorithms rely on a combination of optimization algorithms and trained machine learning models. The performance of these algorithms largely depends on the accuracy of the machine learning model, which is determined by the type of machine learning used and the selection of hyperparameters. For optimization algorithms, the efficiency can be improved by employing more advanced techniques. However, this study identifies a gap in providing a practical approach for selecting the best machine learning model. In most cases, users must manually compare the performance of different machine learning models across a range of hyperparameters, which is not a convenient process. The need to perform extensive data training for various machine learning types, along with the broad variation of hyperparameters, adds to the complexity. Additionally, there is a limited selection of algorithms available for performing multi-objective optimization (MOO). While numerous MOO algorithms have been proposed, the most commonly adopted methods remain the Non-Dominated Sorting Genetic Algorithm (NSGA-II) and Multi-Objective Particle Swarm Optimization (MOPSO). This study aims to address these challenges by proposing a framework to produce an optimal mixture for RAC. To achieve this, a novel MOO algorithm, termed Pareto Front Symbiotic Organism Search (PF-SOS), is introduced to identify the optimal RAC mixture. Furthermore, the machine learning model is constructed using the Machine Learning Symbiotic Organism Search (ML-SOS) metaheuristic algorithm, which not only tunes the hyperparameters but also selects the most suitable machine learning type. Unlike existing methods that focus solely on hyperparameter tuning, this study includes the machine learning type as an additional optimized parameter, offering a more practical and efficient approach. To demonstrate the effectiveness of the proposed algorithms, PF-SOS is compared with NSGA-II and MOPSO, a widely used and well-regarded MOO algorithm. The results indicate that PF-SOS achieves faster convergence and provides a broader range of solutions within the same number of function evaluations. Additionally, the use of ML-SOS for selecting the optimal machine learning type and its parameters offers a more convenient and objective solution for users.

To enhance comprehensiveness and readability, the significance of this research will be emphasized in Section 2. Section 3 will provide a detailed explanation of the methodology behind the proposed PF-SOS and ML-SOS algorithms. In Section 4, the applicability of these algorithms to RAC will be demonstrated, along with a comparison to the widely recognized NSGA-II and MOPSO method. Finally, Section 5 will briefly summarize the conclusions of the current study.

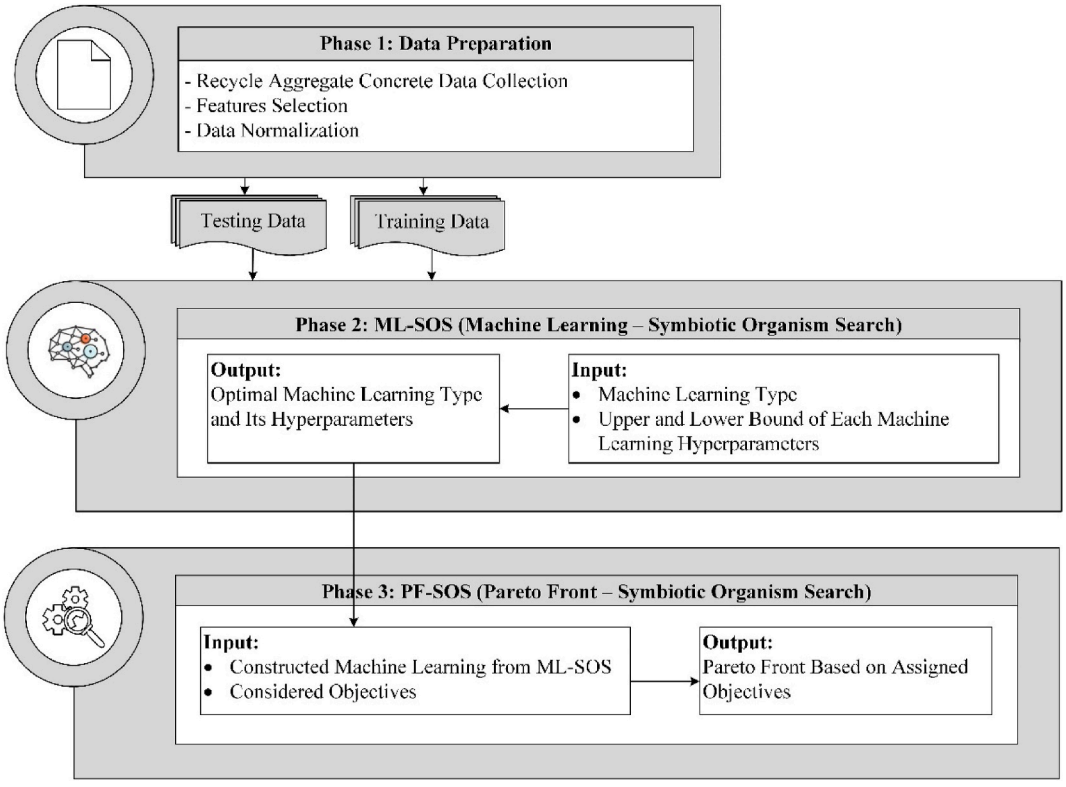


Fig. 1. General scheme of proposed determination of optimal RAC mixture framework.

2. Research significance

This section highlights the significant contributions and novel aspects of this study. Two major innovations are introduced, which will be detailed as follows.

1. **Novel Multi-Objective Optimization Framework**: This study proposes a robust framework for determining the optimal mixture of RAC using a newly developed multi-objective optimization algorithm termed PF-SOS. The proposed algorithm is a modification of the single-objective SOS algorithm. Results in Section 4 show that the PF-SOS algorithm demonstrates superior convergence compared to the well-established NSGA-II in determining the optimal mixture composition of RAC.
2. **Automated Machine Learning Selection with ML-SOS**: In the PF-SOS framework, machine learning is employed to predict the compressive strength of RAC. To enhance accuracy, this study introduces a novel ML-SOS metaheuristic algorithm, which aims to identify the most suitable machine learning model along with its optimal hyperparameters. The input to the algorithm includes various machine learning models and their respective upper and lower bounds for hyperparameters. ML-SOS automatically selects the best machine learning model and tunes its hyperparameters. Based on the author's literature review, most studies focus solely on hyperparameter tuning, while the selection of the machine learning model is typically done manually by comparing error indicators across different models.

3. Methodology

As discussed in Sections 1 and 2, this study primarily focuses on proposing a novel algorithm to generate the optimal mixture of RAC, taking into account multiple objective functions. Fig. 1 outlines the flowchart of the proposed framework. The entire process consists of three phases, as illustrated in Fig. 1. The method begins with the data preparation phase, where RAC mixture compositions and compressive strength test results are collected and normalized. This collected data is then divided into training and testing datasets, which will be utilized for machine learning construction in Phase 2. In Phase 2, the ML-SOS (Machine Learning Symbiotic Organism Search) algorithm is developed to create a suitable machine learning model, including its hyperparameters. Unlike most algorithms that focus solely on hyperparameter tuning, ML-SOS also incorporates the type of machine learning model as an optimized parameter. The user inputs various machine learning options along with the upper and lower bounds for their hyperparameters. The adopted metaheuristic ML-SOS algorithm will search for the most suitable machine learning model and its corresponding hyperparameters, providing convenience to the user without the need for manual comparison of each model's error indicators.

Once the optimal machine learning model is constructed, Phase 3 is initiated to generate the optimal RAC mixture. In this phase, an algorithm termed PF-SOS (Pareto Front Symbiotic Organism Search) is developed. PF-SOS is a modified version of the SOS algorithm designed to address optimization problems with multiple objective functions. The output of the PF-SOS algorithm is a set of non-dominated solutions based on the assigned objectives. This set of solutions, known as the Pareto Front, provides users with options to select solutions that align with their priorities. In this study, two objectives are considered in designing the RAC mixture: concrete compressive strength and the cost of RAC. For the compressive strength objective, machine learning is employed to predict the compressive strength of the RAC mixture. Further details regarding Phase 1 (Data Preparation) are provided in Section 3.1, while the development of ML-SOS in Phase 2 and PF-SOS in Phase 3 are explained in Sections 3.2 and 3.3, respectively.

3.1. Phase 1: data preparation

This study utilizes Recycled Aggregate Concrete (RAC) data collected by Yuan et al. [29], who gathered 28-day-old RAC data from 68 different references. As emphasized in Section 1, absorption is a significant factor influencing the compressive strength of RAC, so only data with complete information on aggregate water absorption are included in this study. From the database provided by Yuan et al., a total of 528 data points are utilized. Seven mechanical properties are employed as inputs for the machine learning model: effective water-cement ratio (w/c), aggregate-cement ratio (a/c), recycled aggregate replacement percentage (RA%), nominal maximum recycled aggregate size (RA_s), nominal maximum normal aggregate size (NA_s), water absorption of recycled aggregate (RA_w), and water absorption of normal aggregate (NA_w). These inputs are used to predict the compressive strength of RAC (f_c'). The input feature selection is based on its impact on production cost and compressive strength, its controllability during the production stage, and the availability of large datasets. One of the main challenges in this study is the limited availability of sufficiently large

Table 1
RAC input and output data properties.

Type	Description	Notation	Unit	Mean	COV	Min	Max
Input	Water cement ratio	w/c	–	0.49	0.22	0.19	0.87
	Aggregate cement ratio	a/c	–	2.97	0.27	1.50	6.50
	Recycle aggregate replacement percentage	RA%	%	51.05	0.77	0.00	100.00
	Nominal maximum recycles aggregate size	RA_s	mm	21.43	0.26	8.00	32.00
	Nominal maximum normal aggregate size	NA_s	mm	22.07	0.24	10.00	38.00
	Water absorption of recycle aggregate	RA_w	%	4.22	0.65	0.00	11.90
	Water absorption of normal aggregate	NA_w	%	0.74	1.01	0.00	3.00
Output	Concrete Compressive Strength	f_c'	MPa	43.79	0.34	13.40	108.50

datasets for training machine learning models. Although certain parameters, such as the quality of the parent concrete used for aggregates or the age of the recycled aggregate, are worth investigating, their data availability remains highly limited. Table 1 summarizes the properties of the input and output data utilized in this study, listing the mean, coefficient of variation (*COV*), minimum, and maximum for each parameter. The correlations among the data are presented in Table 2, which indicates that the water-cement ratio has the strongest correlation with compressive strength, followed by aggregate size. Further, significance testing of each input feature's impact on compressive strength was performed using Student's t-distribution, with the corresponding p-values presented in Table 3. The results indicate that all input features exhibit a high significance level in relation to compressive strength, except for RA_w data, which has a p-value of 6.1 %. Table 3 is arranged from left to right in descending order of significance level. To provide a visual representation of the utilized data, histograms for each input and output are presented in Fig. 2. In this study, 85 % of the prepared data will be used as training data, while the remaining 15 % will serve as testing data. The prepared training data will be utilized to construct a high-accuracy machine learning model in Phase 2 using the ML-SOS algorithm. Section 3.2 will provide a detailed explanation of the ML-SOS algorithm for constructing the machine learning model.

3.2. Phase 2: ML-SOS (Machine Learning Symbiotic Organism Search)

Phase 2 aims to construct a high-accuracy machine learning predictor using the prepared training data from Phase 1. To achieve this goal, the selection of the machine learning type and its parameters is critical to ensure that the generated model is suitable for the investigated case. As described in Section 1, efforts to enhance the accuracy of machine learning have been undertaken in several studies [22–24], typically by optimizing the hyperparameters of the models through metaheuristic algorithms. This study also employs a metaheuristic algorithm, specifically the Symbiotic Organism Search (SOS), to optimize the chosen machine learning models. Unlike most proposed frameworks, this study includes the machine learning type as an optimized variable.

SOS is a metaheuristic algorithm inspired by the interactions between organisms in nature. Its superiority in finding optimal solutions compared to other notable optimization algorithms has been demonstrated by Cheng et al. [30]. Similar to other swarm-based metaheuristic algorithms, SOS employs a swarm optimization approach, where the swarm is referred to as organisms. The algorithm begins with parameter determination, as illustrated in the flowchart in Fig. 3. To enhance clarity and comprehensibility for the reader, the pseudocode of the proposed ML-SOS is provided in Algorithm 1, while the fitness evaluation process is detailed in Algorithm 2 at the end of Section 3.2. As shown, the SOS algorithm is initialized by determining the number of iterations and the size of the organism population. The number of iterations serves as the stopping criterion, while the organism count determines the swarm size for finding solutions. Each organism contains optimized parameters for the machine learning model, as formulated in Equation (1). In this study, each organism is represented as a vector $\boldsymbol{O}$ containing x_0, x_1, x_2, …, x_n. Here, x_0 represent the normalized hyperparameters of the machine learning model, where n is the number of hyperparameters. Each normalized hyperparameter, x_1, x_2, …, x_n is assigned a value between 0 and 1. This value will subsequently be transformed into hyperparameters based on the value of x_0 during fitness evaluation. The transformation is accomplished using the pre-determined upper (u_b) and lower (l_b) bounds of each machine learning hyperparameter, as shown in Equation (2).

$$\mathbf{O} = [\,x_0 \quad x_1 \quad x_2 \quad \ldots \quad x_n\,] \qquad x_0 = [\,1 \quad 2 \quad \ldots \quad n_{ML}\,] \tag{1}$$

$$hyperparameters = (u_b - l_b)^*(\,x_1 \quad x_2 \quad \ldots \quad x_n\,) + l_b \tag{2}$$

$$Performance_i = RMSE + \left(1 - R^2\right) \tag{3}$$

In the proposed framework, x_0 can represent various machine learning types depending on the available algorithms suitable for the investigated case. This study does not impose a limit on the number of n_{ML} as long as the algorithms and their hyperparameter boundaries are defined. For instance, the current study considers $n_{ML} = 3$, with the SOS algorithm tasked with selecting among Artificial Neural Network (ANN), Support Vector Regression (SVR), and Random Forest (RF), as illustrated in Fig. 4. In this context, x_0 can be assigned integer values of 1, 2, or 3, corresponding to ANN, SVR, and RF, respectively.

Fig. 4 illustrate the SOS algorithm tasked with selecting among Artificial Neural Network (ANN), Support Vector Regression (SVR), and Random Forest (RF). In this context, x_0 can be assigned integer values of 1, 2, or 3, corresponding to ANN, SVR, and RF,

Table 2
Correlation between each data properties.

Data	*w/c*	*a/c*	*RA%*	RA_s	NA_s	RA_w	NA_w	f_c'
w/c	1.00	0.49	−0.09	−0.08	−0.12	−0.05	0.08	−0.44
a/c	0.49	1.00	−0.03	0.17	−0.12	0.03	0.04	−0.28
RA%	−0.09	−0.03	1.00	0.14	0.05	0.54	−0.60	−0.16
RA_s	−0.08	0.17	0.14	1.00	0.24	−0.11	−0.32	−0.37
NA_s	−0.12	−0.12	0.05	0.24	1.00	0.03	0.01	−0.23
RA_w	−0.05	0.03	0.54	−0.11	0.03	1.00	−0.13	−0.09
NA_w	0.08	0.04	−0.60	−0.32	0.01	−0.13	1.00	0.17
f_c'	−0.44	−0.28	−0.16	−0.37	−0.23	−0.09	0.17	1.00

Table 3
Input Feature Significance Toward f_c'.

Input Feature	w/c	a/c	RA_s	NA_s	NA_w	$RA\%$	RA_w
p-value (%)	0.00	0.00	0.00	0.00	0.02	0.03	6.11

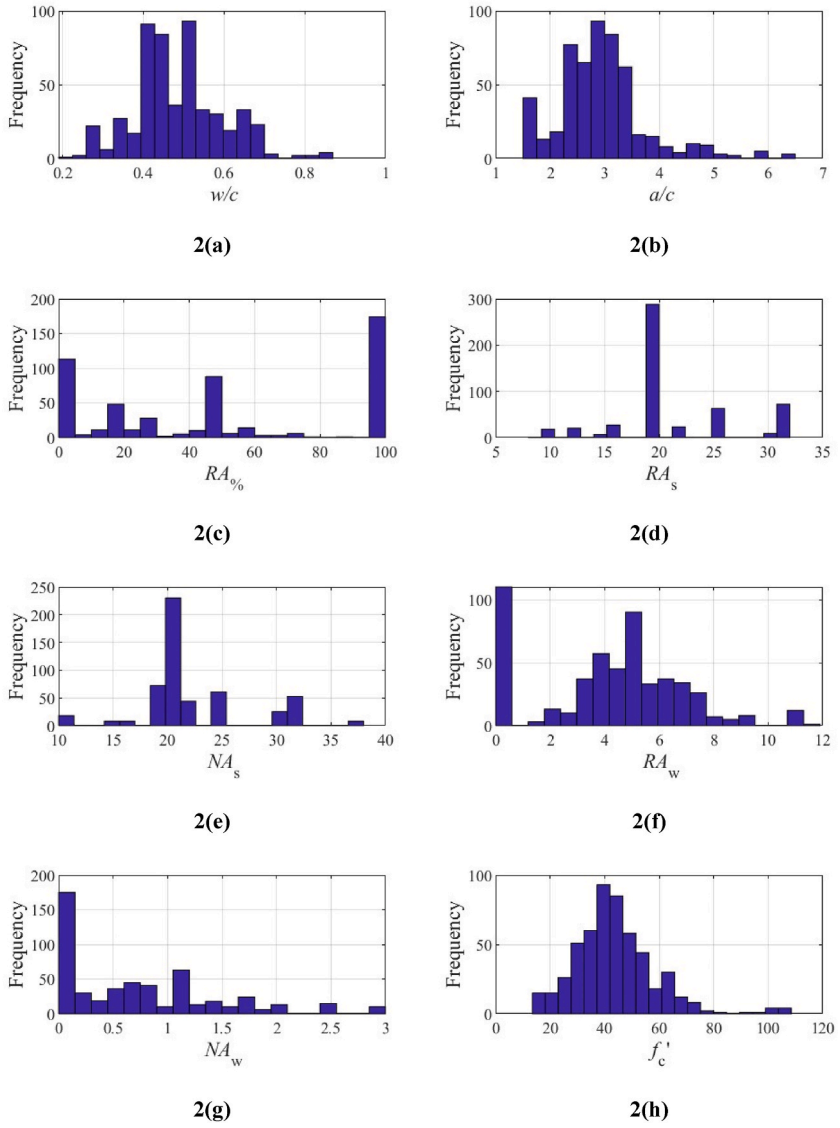


Fig. 2. Histogram of each input and output RAC data.

respectively. If each machine learning model (ANN, SVR, and RF) has two hyperparameters, the additional optimized parameters x_1 and x_2 will be introduced as shown in Fig. 4. During fitness evaluation, the normalized hyperparameters x_1 and x_2 will be transformed into actual hyperparameters based on the value of x_0 using Equation (2). In cases where ANN has two hyperparameters while SVR and RF each have three, n will be set to 3, with the additional hyperparameters for ANN treated as dummy variables. This dummy variable is not utilized during fitness evaluation but is still required to maintain uniformity in the number of optimized variables across the machine learning options. Utilizing the framework illustrated in Fig. 4, the SOS algorithm aims to determine the optimal values for x_0, x_1 and x_2.

For fitness function calculation, this study employs the K-fold cross-validation method to evaluate machine learning error. As illustrated in Fig. 5, the training data is divided into five fragments, with each fragment serving as validation data in turn. For each validation iteration, the performance function is computed, and the average is used as the fitness function for the SOS algorithm. The performance function utilized in this study is formulated in Equation (3), incorporating both the Root Mean Square Error (*RMSE*) and

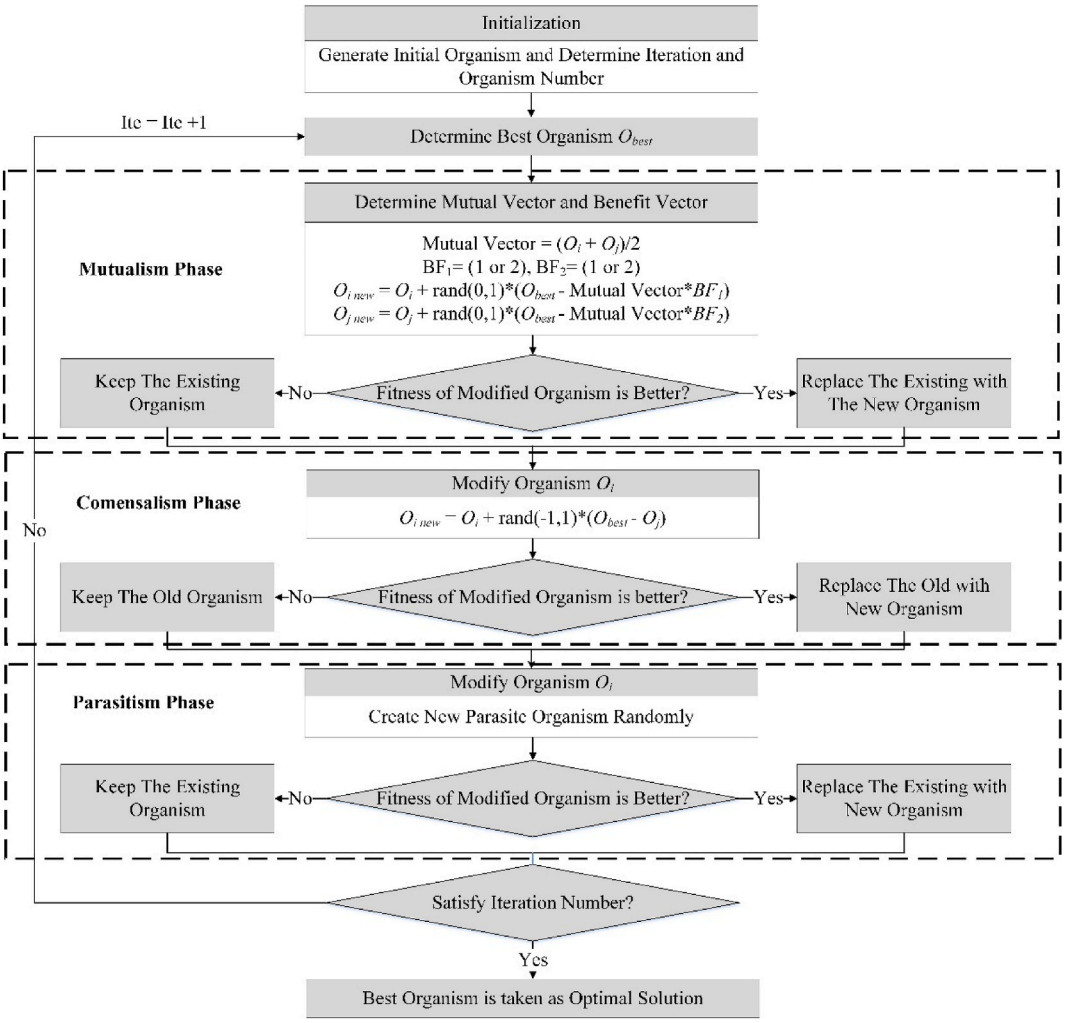


Figure 3. Flowchart of symbiotic organism search.

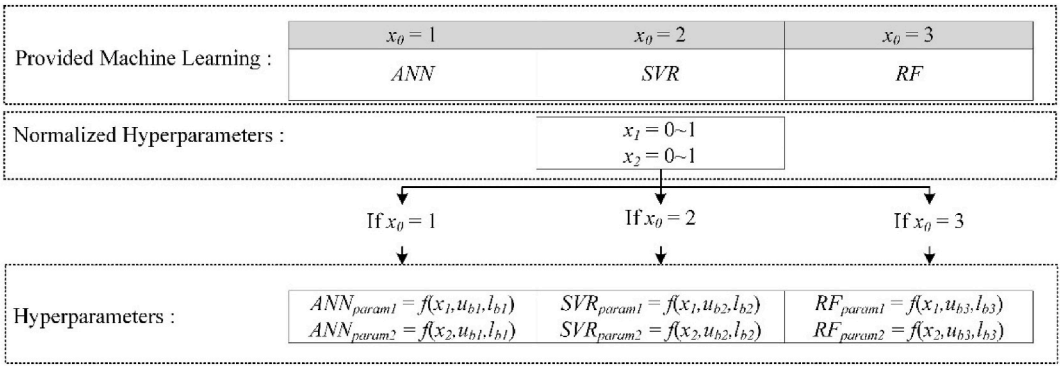


Fig. 4. Illustration of optimized parameters.

the Coefficient of Determination (R^2). This study employs RMSE due to its ability to impose a higher penalty on larger errors. The square root feature of RMSE amplifies the impact of higher prediction errors, making it a suitable metric for assessing accuracy. Additionally, R^2 is incorporated to measure the correlation between predictions and test data, ensuring the model captures the underlying relationship effectively. However, error indicators can be modified based on specific preferences and requirements of the

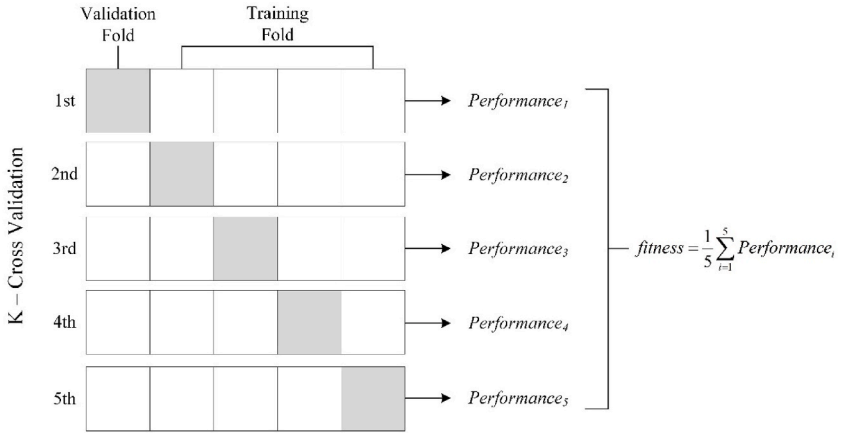


Fig. 5. Cross validation as fitness function.

investigated case. In this study, RMSE and R^2 are chosen due to their suitability for evaluating RAC compressive strength prediction. Since the SOS algorithm operates under a minimization task, lower *RMSE* values and higher R^2 values are preferred. The results of applying the ML-SOS algorithm to the RAC data can be found in Section 4.1. For a detailed algorithm description, the pseudocode of ML-SOS is provided in Algorithm 1, while the fitness evaluation process is described separately in Algorithm 2.

Algorithm 1: Machine Learning Optimization ML-SOS

01Initialize:
02Number of Organism (n_O) and Iteration (n_{ite}).
03Number of machine learning option (n_{ML})
04Upper and lower bound of each machine learning option hyperparameters (u_b and l_b)
05
06Randomly Generate $O_i = [x_0 \quad x_1 \quad x_2 \quad \dots \quad x_n]$ according to Eq. (1) with $i = 1$ to n_O
07
08**For** $i = 1$ to n_O
09Evaluate fitness/objective (O_i) using Algorithm 2:
10
11**For** $h = 1$ to n_{ite}
12**For** $i = 1$ to n_O
13Determine O_{best} (Best Organism)
14//Mutualism Phase
15Randomly select other Organisms O_j where $j \neq i$
16Mutual Vector = $(O_i + O_j)/2$
17BF_1= (1 or 2), BF_2= (1 or 2)
18Oi new = Oi + rand(0,1) × (Obest - Mutual Vector*BF1)
19Oj new = Oj + rand(0,1) × (Obest - Mutual Vector*BF2)
20Evaluate fitness of $O_{i\ new}$ and $O_{j\ new}$ using Algorithm 2
21**If** fitness($O_{i\ new}$) < fitness(O_i)
22Replace original Organism-*i*
23**If** fitness($O_{j\ new}$) < fitness(O_j)
24Replace original Organism-*j*
25//Commensalism Phase
26Randomly select other Organisms O_j where $j \neq i$
27$O_{i\ new} = O_i$ + rand(-1,1) × (O_{best} - O_j)
28Evaluate fitness of $O_{i\ new}$ using Algorithm 2
29**If** fitness ($O_{i\ new}$) < fitness(O_i)
30Replace original Organism-*i*
31//Parasitism Phase
32Create New Parasite Organism Randomly ($O_{i\ new}$)
33Evaluate fitness of $O_{i\ new}$ using Algorithm 2
34**If** fitness ($O_{i\ new}$) < fitness(O_i)
35Replace original Organism-*i*
36**End of Algorithm 1**

Algorithm 2: Fitness Evaluation in ML-SOS

01**Input**:
02$O_i = [x_0 \quad x_1 \quad x_2 \quad \dots \quad x_n]$
03Number of machine learning option (n_{ML})
04Upper and lower bound of each machine learning option hyperparameters (u_b and l_b)

(*continued on next page*)

(*continued*)

Algorithm 2: Fitness Evaluation in ML-SOS

05Number of K-fold cross validation
06
07**Output:** Fitness of O_i
08
09**If** round $(x_0) = 1$
10Use Type 1 Machine Learning
11Translate $x_1, x_2, .. x_n$: *hyperparameters* $= (u_b - l_b)^*(x_1 \quad x_2 \quad \dots \quad x_n) + l_b$
12Partition training data into K-fold training data
13**For** $h = 1$ to Number of K-fold
14Perform Training Type 1 Machine learning using training data – h
15Perform Testing Type 1 Machine learning using testing data – h
16Calculate Error Using Eq. 3
17Fitness = Calculate Average Error Among from h = 1 to Number of K-fold
18.
19
20**Else If** round $(x_0) = n_{ML}$
21Use Type n_{ML} Machine Learning
22Translate $x_1, x_2, .. x_n$: *hyperparameters* $= (u_b - l_b)^*(x_1 \quad x_2 \quad \dots \quad x_n) + l_b$
23Partition training data into K-fold training data
24**For** $h = 1$ to Number of K-fold
25Perform Training Type n_{ML} Machine learning using training data – h
26Perform Testing Type n_{ML} Machine learning using testing data – h
27Calculate Error Using Eq. 3
28Fitness = Calculate Average Error Among from h = 1 to Number of K-fold
29
30**End of Algorithm 2**

3.3. Phase 3: PF-SOS (Pareto Front Symbiotic Organism Search)

As described in Section 1, constructing machine learning alone is insufficient for generating an optimal mixture of recycled aggregate concrete (RAC). The machine learning model developed in Section 3.2 needs to be integrated into a Multi-Objective Optimization (MOO) algorithm to generate RAC compositions based on predetermined objectives computed using machine learning. To achieve this, this study introduces a novel MOO algorithm by modifying the original Symbiotic Organism Search (SOS) algorithm. This modified algorithm will be referred to as Pareto Front Symbiotic Organism Search (PF-SOS). Originally, SOS is a single-objective metaheuristic algorithm developed by Cheng et al. [30], designed to mimic interactions among animals in nature. The original SOS produces a single optimal solution with one predefined objective, whereas PF-SOS outputs a set of non-dominated solutions for multiple objectives, known as the Pareto Front. Fig. 6 illustrates the non-dominated solutions generated by PF-SOS, where the colored dots represent non-dominated solutions and the non-colored dots are categorized as dominated solutions. In this study, the final solution comprises the Pareto Front set shown in Fig. 6, where each dot contains a specific RAC mixture composition and the corresponding objectives. This set of solutions allows for further decision-making processes to determine the optimal preferences for the user. However, this study limits its investigation to the Pareto Front as the final output.

Several modifications have been introduced to adapt the original Symbiotic Organism Search (SOS) into the Pareto Front Symbiotic Organism Search (PF-SOS). Fig. 7 illustrates the flowchart of the PF-SOS procedure. For clarity, the pseudocode of the proposed PF-SOS is presented in Algorithm 3 at the end of Section 3.3. In general, PF-SOS follows almost a similar process to that of the SOS algorithm, starting from initialization, finding the best organism, performing organisms interaction, and then repeating these steps until the

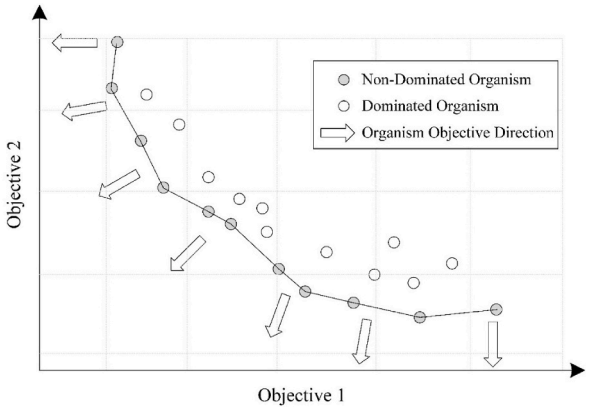


Fig. 6. Illustration of PF-SOS constructing pareto front.

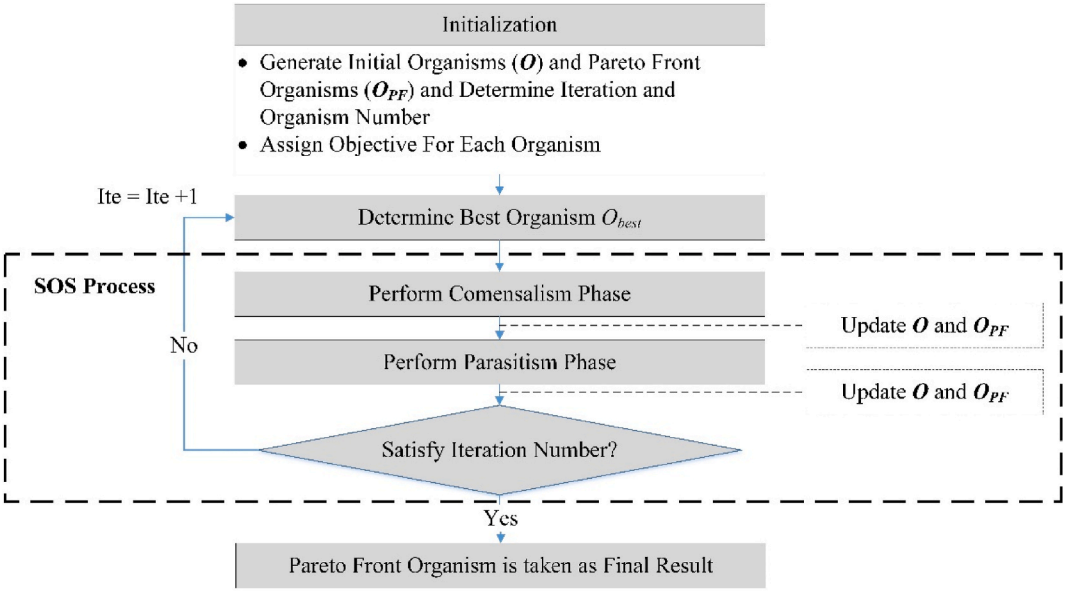


Fig. 7. Flowchart of PF-SOS.

predetermined iteration number is satisfied. The detailed step-by-step procedure is as follows.

1. **Initialization Phase**: The algorithm begins with the initialization phase, where the user must determine four algorithm parameters: the number of iterations (n_{ite}), number of organism (n_{org}), and the vectors $\boldsymbol{O}$ and $\boldsymbol{O_{PF}}$ that consist of the set of optimized variables. In PF-SOS, the organisms stored in vector $\boldsymbol{O}$ maintain a similar physical meaning to those in the original SOS, representing the set of optimized variables that will be refined during iterations based on the predetermined objective functions. Initially, the $\boldsymbol{O_{PF}}$ vector will contain the non-dominated organisms of $\boldsymbol{O}$**.** At final iteration, $\boldsymbol{O_{PF}}$ will output the Pareto Front solution. The distinction and purpose of creating both $\boldsymbol{O}$ and $\boldsymbol{O_{PF}}$ will be clarified in subsequent steps.
2. **Defining Objective Function:** Current modification adopts weighting method or secularization method in determining objective function which has been done on several MOO study [31,32]. In modified algorithm, each organism within $\boldsymbol{O}$ has a distinct objective function derived from main objectives (f_{obj1}, f_{obj2}, f_{obj3}, etc.). Unlike the original SOS, which employs a single objective function for all organisms, PF-SOS assigns different objective functions to each organism based on a weighted combination of the main objectives, as formulated in Equation (4). Equation (4) illustrates how the objectives assigned to each organism are derived using the weighted main objectives f_{obj1} and f_{obj2}. The weight is calculated based on the index i and the total number of organisms n_{org}. For clearer visualization, Equation (5) demonstrates the application of Equation (4) with $n_{org} = 5$ and two main objectives. By assigning designated objectives to each organism, PF-SOS ensures that the organisms explore a wide range of non-dominated solution areas, as indicated by the arrows in Fig. 6. In this study, PF-SOS focuses on two main objective functions: f_{obj1} for compressive strength and f_{obj2} for the cost of RAC, with compressive strength computed using the constructed machine learning model from Section 3.2.

$$f_{Oi} = \left(1 - \frac{i}{n_{org}}\right) f_{obj1} + \left(\frac{i}{n_{org}}\right) f_{obj2}, \in i = 1, 2, 3, \ldots, n_{org} \quad \text{Eq. 4}$$

$$f_O = \begin{cases} f_{O1} = 0.8 f_{obj1} + 0.2 f_{obj2} \\ f_{O2} = 0.6 f_{obj1} + 0.4 f_{obj2} \\ f_{O3} = 0.4 f_{obj1} + 0.6 f_{obj2} \\ f_{O4} = 0.2 f_{obj1} + 0.8 f_{obj2} \\ f_{O5} = 0.0 f_{obj1} + 1.0 f_{obj2} \end{cases} \quad \text{Eq. 5}$$

3. **Determine best organism** ($\boldsymbol{O_{best}}$): The $\boldsymbol{O_{best}}$ is determined and updated at the start of each iteration by randomly selecting an organism from $\boldsymbol{O_{PF}}$**.** While $\boldsymbol{O}$ contains organisms that are repeatedly optimized toward their respective objective functions (f_{Oi}), $\boldsymbol{O_{PF}}$ holds Pareto Front organisms, which are sorted at the end of each organism update. Both $\boldsymbol{O}$ and $\boldsymbol{O_{PF}}$ are subject to updates through organisms interaction phases described in Step 4. The key distinction is that $\boldsymbol{O}$ is updated by replacing organisms with those that have better f_{Oi}, whereas $\boldsymbol{O_{PF}}$ replaces organisms with non-dominated solutions at the end of each phase.
4. **Organisms Interaction Phases**: These phases are executed using the same formulas as the original SOS, as illustrated in Fig. 3. However, for efficiency reasons, the proposed PF-SOS algorithm only utilizes the commensalism and parasitism phases. This study assumes that these two phases are sufficient to cover the exploration requirements of the optimization process. The commensalism

phase is designed to guide organisms toward $\boldsymbol{O_{best}}$, while the parasitism phase introduces randomness, potentially allowing organisms to explore previously unvisited regions of the search space. During these phases, new candidate solutions are generated for both $\boldsymbol{O}$ and $\boldsymbol{O_{PF}}$ by replacing existing organisms and adding into $\boldsymbol{O_{PF}}$, ensuring a balance between convergence and diversity in the optimization proces s.

5. Steps 3 and 4 are repeated until the predetermined iteration number is reached. At the conclusion of the final iteration, $\boldsymbol{O_{PF}}$ will contain the Pareto Front solutions.

In brief, this modified algorithm creates two distinct organism groups: $\boldsymbol{O}$ and $\boldsymbol{O_{PF}}$. The organisms in $\boldsymbol{O}$ are tasked with pursuing the objective f_{Oi}, where any new organism that performs better will replace an existing one. These new organisms are generated through the mutualism, commensalism, and parasitism phases. Concurrently, these candidates are also used to update the organisms in $\boldsymbol{O_{PF}}$, with the goal of finding superior non-dominated solutions. The group $\boldsymbol{O}$ aids in steering the organisms toward the Pareto Front, as depicted by the arrows in Fig. 6. As new candidates are generated during the mutualism, commensalism, and parasitism phases, the organisms in $\boldsymbol{O_{PF}}$ are continuously updated. This method accelerates convergence, and its effectiveness for RAC mixture optimization will be demonstrated by comparing it to the widely recognized NSGA-II in Section 4.2.

Algorithm 3: Multi Objective Optimization PF-SOS

Initialize:
Number of Organism (n_O) and Iteration (n_{ite}).
Machine learning model for compressive strength prediction
Upper and lower bound of each RAC component (u_b and l_b)

Randomly Generate $O_i = [\,x_1 \quad x_2 \quad x_3 \quad \dots \quad x_n\,]$ with $i = 1$ to n_O

For $i = 1$ to n_O
$[f_{obj1}, f_{obj2}]$ = Evaluate objective (O_i)
fitness(O_i) = $weight_{1i} \times f_{obj1}$, $weight_{2i} \times f_{obj2}$ as illustrated in Eqs. (4) and (5).
Check domination of organism
Store non-dominated organism into O_{PF}

For $h = 1$ to n_{ite}
For $i = 1$ to n_O
BestOrganism (O_{best}) = Randomly Select Organism in O_{PF}

//Commensalism Phase
Randomly select other Organisms O_j where $j \neq i$
$O_{i\ new} = O_i + \text{rand}(-1,1) \times (O_{best} - O_j)$
$[f_{obj1}, f_{obj2}]$ = Evaluate objective ($O_{i\ new}$)
fitness(O_i) = $weight_{1i} \times f_{obj1}$, $weight_{2i} \times f_{obj2}$ as illustrated in Eqs. (4) and (5).
If fitness ($O_{i\ new}$) < fitness(O_i)
Replace original Organism-*i*
If $O_{i\ new}$ is non-dominated
Stored $O_{i\ new}$ into O_{PF}
Delete dominated organism inside O_{PF}

//Parasitism Phase
Create New Parasite Organism Randomly ($O_{i\ new}$)
$[f_{obj1}, f_{obj2}]$ = Evaluate objective ($O_{i\ new}$)
fitness(O_i) = $weight_{1i} \times f_{obj1}$, $weight_{2i} \times f_{obj2}$ as illustrated in Eqs. (4) and (5).
If fitness ($O_{i\ new}$) < fitness(O_i)
Replace original Organism-*i*
If $O_{i\ new}$ is non-dominated
Stored $O_{i\ new}$ into O_{PF}
Delete dominated organism inside O_{PF}
End of Algorithm 3

4. Result

This section presents the results of the proposed framework outlined in Section 3. Following the dataset preparation detailed in Section 3.1, the ML-SOS algorithm is executed to identify the optimal machine learning model along with its hyperparameters. The constructed machine learning model then serves as the surrogate objective function for the PF-SOS algorithm, which is tasked with finding the optimal mixture composition of RAC. The results from the ML-SOS algorithm will be discussed in Section 4.1, while the outcomes from the PF-SOS algorithm are presented in Section 4.2. ML-SOS Result.

4.1. ML-SOS result

In this study, the ML-SOS algorithm is executed with 50 iterations and a population of 20 organisms. The objective function employs a 10-fold cross-validation method, with the performance function formulated as described in Eq. (3). The algorithm is tasked

with selecting from three machine learning options: Artificial Neural Network (ANN), Support Vector Machine (SVM), and Random Forest (RF). Each machine learning model is assigned upper and lower bounds for its respective hyperparameters, as detailed in Table 4.

The background theory for Artificial Neural Networks (ANN), Support Vector Machines (SVM), and Random Forest (RF) is not elaborated on in this study but can be referenced in sources [33–35]. The implementation of ANN, SVM, and RF is carried out using the MATLAB machine learning toolbox. The accuracy of the final model is evaluated using four different error indicators: Mean Absolute Percentage Error (MAPE), Coefficient of Correlation (R^2), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE). For comparison, the proposed ML-SOS algorithm is evaluated against standard hyperparameter tuning algorithms with a single machine learning option, denoted as ANN-SOS, SVM-SOS, and RF-SOS. Additionally, ML-SOS is compared to two well-known optimization algorithms, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), denoted as ML-GA and ML-PSO, respectively. This comparison aims to validate the selection of SOS as the optimization algorithm. For GA and PSO, the number of iterations is also set to 50, but with a swarm size of 80 to ensure that the total number of function evaluations per iteration remains equivalent to that of the SOS algorithm. In GA, the crossover and mutation rates are set to 0.9 and 0.1, respectively. A one-point crossover method is adopted, with parental selection using the roulette wheel method. For PSO, the initial velocity is set to zero, and the acceleration parameter is set to two.

Table 5 presents the error metrics for the proposed method alongside those of other compared methods on both the training and testing datasets. For the training data, the errors shown in Table 5 represent the averages obtained from 10-fold cross-validation. Notably, the Random Forest with SOS (RF-SOS) exhibits the lowest error, with only a slight difference compared to the proposed method. The ML-SOS algorithm yields nearly identical error metrics to RF-SOS, as it selects RF as the optimal machine learning model from the three options provided. Similarly, both ML-GA and ML-PSO also identify RF as the best-performing machine learning model. However, during the training process, the SOS algorithm is able to find better hyperparameters that contribute to lower training error. Despite this, ML-GA demonstrates slightly better performance in the validation process, yielding lower error metrics.

Fig. 8 illustrates the convergence of the SOS optimization process for the fitness function, as defined in Eq. (3). From the first iteration, ML-SOS identifies RF as the best organism due to its ability to produce lower errors compared to the other machine learning models. Compared to GA and PSO, the SOS algorithm achieves lower error performance for the training data, as illustrated in Fig. 8(a). This validates the superior optimization performance of SOS over GA and PSO for the current investigated case. A comparison of the predicted versus actual testing data is shown in Fig. 9. The proposed ML-SOS in this study efficiently selects the optimal machine learning algorithm along with its hyperparameters, providing convenience for users by eliminating the need for manual comparisons across various machine learning models. Subsequently, the ML-SOS model will be utilized as the objective function for computing the compressive strength of Recycled Aggregate Concrete (RAC) within the PF-SOS algorithm.

Further uncertainty analysis was conducted on the ML-SOS prediction results. Fig. 10(a) presents the error distribution of ML-SOS predictions across 78 testing data points. The results indicate that the model's errors have a standard deviation of 5.725 MPa, with a 95 % confidence interval (CI) spanning 11.45 MPa. To further analyze the error distribution, the testing data was categorized into three groups based on compressive strength (f_c').

- Group 1: RAC samples with compressive strength below 40 MPa
- Group 2: RAC samples with compressive strength between 40 and 50 MPa
- Group 3: RAC samples with compressive strength above 50 MPa

The number of data points in each group is illustrated in Fig. 10(b), (c), and (d), respectively. The standard deviations for these groups are 4.65 MPa, 4.15 MPa, and 5.75 MPa, with their corresponding 95 % CIs displayed in Fig. 10. Among the three groups, Group 3 (above 50 MPa) exhibits the highest error, whereas Group 2 (40–50 MPa) shows the lowest error. This may be attributed to a higher density of training data within the 40–50 MPa range.

Additionally, this study investigates whether larger prediction errors are associated with specific regions of the input feature space. To assess this, histograms of each input feature in the full testing dataset were compared against histograms of testing data points where prediction errors exceeded 10 %. Among the 78 testing samples, 24 instances exhibited an error greater than 10 %. Fig. 11 illustrates the distribution of input features across all testing data and the subset with high prediction errors (exceed 10 %). From this analysis, no distinct pattern was observed in the high-error subset, suggesting that the errors are not concentrated in a specific input range. Instead, the high errors may stem from missing critical input features or the presence of outliers in the dataset.

Table 4
Upper and lower bound of each machine learning hyperparameters.

Machine Learning Type	Hyperparameters	Lower bound	Upper bound
Artificial Neural Network (ANN)	Number of layer	1	5
	Number of nodes per layer	4	32
Support Vector Machine (SVM)	Box Constraint	1×10^{-4}	1×10^{4}
	Radial-based function parameter	1×10^{-4}	1×10^{4}
Random Forest (RF)	Number of trees	1×10^{-3}	2000
	Learning rate	1×10^{-4}	1

Table 5
Error of each method on training and testing data.

Method	Training Data				Testing Data			
	MAPE (%)	R^2	*MAE*	*RMSE*	*MAPE* (%)	R^2	*MAE*	*RMSE*
Proposed ML-SOS	**12.09**	**0.89**	**4.75**	**6.39**	**8.91**	**0.92**	**4.06**	**5.7**
ML-GA	**12.15**	**0.88**	**4.81**	**6.66**	**8.8**	**0.92**	**4.01**	**5.55**
ML-PSO	**12.12**	**0.88**	**4.77**	**6.54**	**9.23**	**0.92**	**4.14**	**5.59**
ANN-SOS	21.81	0.69	8.35	10.60	20.43	0.67	8.98	11.47
SVM-SOS	20.80	0.70	7.81	10.23	19.25	0.75	7.64	9.97
RF-SOS	**11.92**	**0.89**	**4.75**	**6.39**	**8.46**	**0.93**	**3.78**	**5.27**

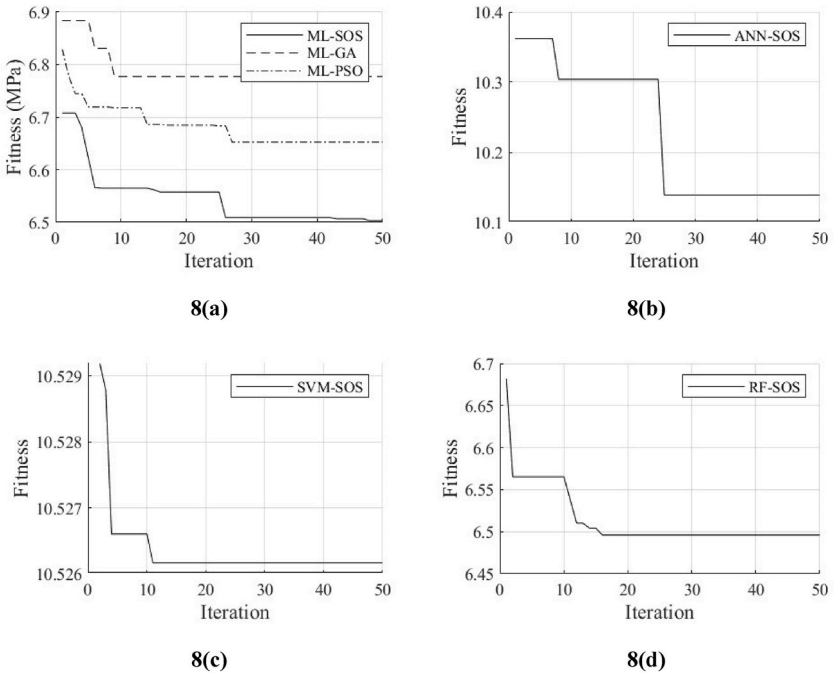


Fig. 8. Convergence of proposed method compared to hyperparameter tuning method.

4.2. PF-SOS result

This section presents the results of the proposed multi-objective PF-SOS algorithm. In this study, two objectives are established: concrete compressive strength (f_c') and cost. For MOO algorithm comparison, this study does not specify the exact costs of the mixture components. Instead, a ratio between each material is provided: concrete is assumed to cost five times that of aggregates, water is considered cost-free, and recycled aggregate is priced at 5 % lower than normal aggregate. Therefore, the cost in this study is treated as unitless.

For the PF-SOS algorithm, the number of iterations and organisms is set to 20 and 100, respectively. Detailed descriptions of the PF-SOS process can be found in Section 3.3, along with the exact framework implemented for RAC optimization. The upper and lower bounds for each mixture component will align with the bounds derived from the data used for training in ML-SOS, as listed in Table 1.

Additionally, performance of the PF-SOS algorithm will be compared them with the notable multi-objective algorithms, NSGA-II and MOPSO. The NSGA-II MATLAB algorithm is directly referenced from Ref. [36] where it utilizes non-dominated sorting and crowding distance techniques to determine the best samples. For a fair comparison, this study will analyze the convergence of the Pareto Front constructed using both PF-SOS and NSGA-II with a similar number of function evaluations. The MOPSO algorithm also utilize the open source algorithm provided by Heris et al. [37].

Fig. 12 illustrates the convergence of the Pareto Front for the proposed PF-SOS compared to NSGA-II and MOPSO across various function evaluation numbers. The results indicate that PF-SOS can achieve a faster convergence process. As shown in Fig. 12(b), after 2000 function evaluations, PF-SOS already provides a more favorable Pareto Front compared to NSGA-II or MOPSO. Unlike conventional multi-objective algorithms, PF-SOS assigns distinct objectives to each organism, as detailed in Eq. (5). This approach is believed to provide clear directional guidance for each organism's movement. By the final iteration (with 4000 function evaluations), several points from PF-SOS and MOPSO overlap; however, PF-SOS consistently offers a superior Pareto Front. Tables 6–8 present

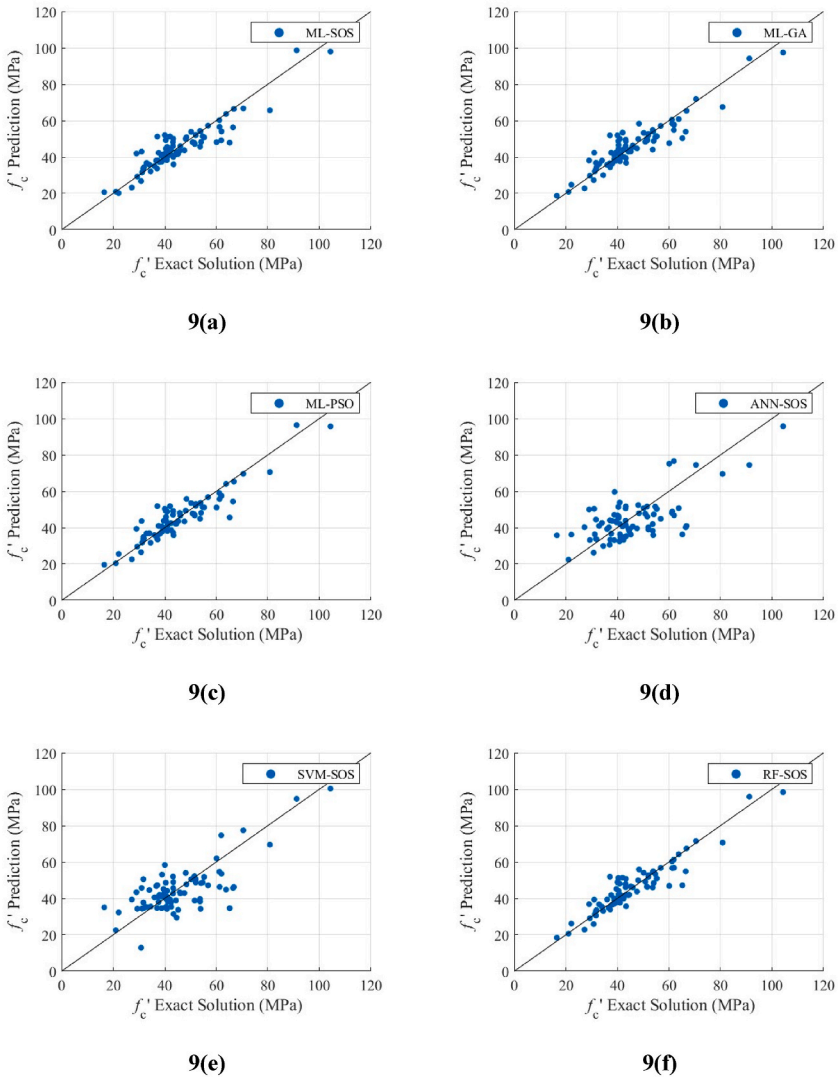


Fig. 9. Prediction on testing data using proposed ML-SOS and several different methods.

samples of optimal mixtures derived from the Pareto Front generated by PF-SOS, NSGA-II and MOPSO, respectively.

Table 6 indicates that the utilization of recycled aggregate can reach up to 90 % for concrete strengths below 70 MPa, with a water-to-cement (w/c) ratio of 0.72. The algorithm leverages aggregate size to enhance compressive strength while maintaining a low w/c ratio and high utilization of recycled aggregate. However, for recycled aggregate concrete (RAC) strengths exceeding 70 MPa, both the w/c ratio and recycled aggregate utilization must be reduced. Result from NSGA-II and MOPSO in Tables 7 and 8 also presents typical results demonstrating that lower w/c ratios and smaller aggregate sizes can effectively boost the compressive strength of RAC. Using smaller aggregates is viewed as a cost-effective alternative to reducing the w/c ratio for increasing compressive strength. It is important to note that the accuracy of the mixture predictions for compressive strengths between 70 and 110 MPa may be lower, likely due to the limited amount of RAC data available within this strength range, as illustrated in Fig. 2(h).

4.3. Cost analysis

To enhance industrial applicability, this study conducts a cost analysis based on realistic cost assumptions obtained from several material suppliers in Indonesia. The cost assumptions adopted are summarized in Table 9. In general, aggregate suppliers in Indonesia do not offer size-specific options; smaller aggregate sizes are typically produced through manual filtering or additional crushing. In this study, it is assumed that the cost of smaller aggregates increases linearly as the particle diameter decreases. Three cost scenarios are considered; Case A assumes that recycled aggregate (RA) is 90 % cheaper than normal aggregate (NA); Case B assumes an 80 % cost reduction; Case C assumes a 70 % cost reduction compared to NA. The PF-SOS Pareto front results for each case are illustrated in Fig. 13. As shown in Fig. 13(a), the cost benefit of using recycled aggregate is significant only for concrete with compressive strength up

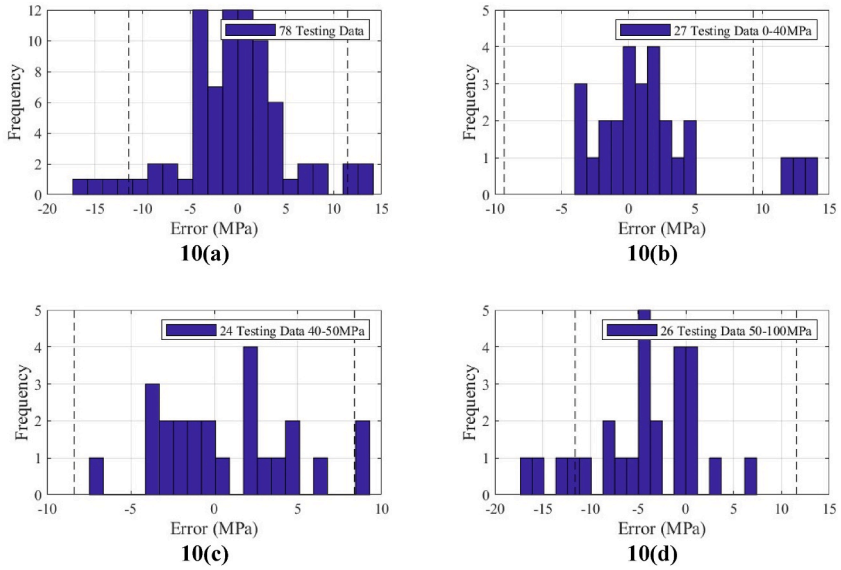


Fig. 10. Error and confidence interval of ML-SOS prediction.

to approximately 65 MPa. For compressive strengths beyond this threshold, the cost advantage of RA diminishes. This trend aligns with expectations; as recycled aggregate is generally less suitable for producing high-strength concrete compared to normal aggregate.

Furthermore, Fig. 13(b) and (c) present 3D scatter plots of the Pareto front, incorporating the *w/c* and *RA*%. These figures highlight the critical role of the *w/c* ratio in influencing both compressive strength and production cost. Achieving higher strength concrete requires a lower *w/c* ratio, which consequently increases production cost. In Fig. 13(c), it is observed that lower-cost designs are associated with higher *RA*%, particularly for lower strength ranges. However, for concrete with compressive strength exceeding 60 MPa, the proportion of RA used decreases to meet the required strength, confirming the earlier observation regarding the limitations of recycled aggregate in high-strength applications.

5. Conclusion

This study focuses on generating optimal mixture compositions for recycled aggregate concrete (RAC), which is recognized as an innovative approach to reducing carbon footprints in the construction industry. To enhance this innovation, a robust and accurate method for determining the mixture is essential. The high complexity of old mortar, water absorption, and the behavior of recycled aggregates contribute to the highly nonlinear nature of RAC compressive strength. To address these challenges, this study proposes two novel algorithms: ML-SOS and PF-SOS. The key conclusions drawn from this study are as follows.

1. **ML-SOS Algorithm**: The proposed ML-SOS algorithm provides a user-friendly solution by automatically selecting the optimal machine learning model and its hyperparameters, eliminating the need for manual comparisons among different models. For the RAC data analyzed, ML-SOS identified Random Forest (RF) as the optimal machine learning method, outperforming Artificial Neural Networks (ANN) and Support Vector Machines (SVM). The RF model achieved the lowest error, as indicated by a Mean Absolute Percentage Error (*MAPE*) of 8.91 % and an R^2 value of 0.92 for testing data predictions (see Table 4).
2. **PF-SOS for Optimal Mixture Generation**: To generate optimal mixtures, this study employs the PF-SOS algorithm to provide Pareto Front solutions. The multi-objective optimization targets two objectives: compressive strength and cost. The performance of PF-SOS is compared against the NSGA-II and MOPSO algorithm. As shown in Fig. 12, PF-SOS demonstrates superior performance in producing well-defined Pareto Front solutions. The assignment of distinct objectives to each organism, as illustrated in Eq. (5), provides clear direction, enabling the algorithm to generate a well-distributed Pareto Front. The results suggest that utilizing smaller aggregate sizes is an effective strategy for enhancing concrete strength while maintaining a high water-to-cement (*w/c*) ratio and maximizing the use of recycled aggregate. For RAC with compressive strengths (f_c') exceeding 65 MPa, it is necessary to reduce both the *w/c* ratio and the proportion of recycled aggregate used.

6. Future direction and limitation

As demonstrated in this study, the integration of artificial intelligence (AI) into concrete mix design particularly for recycled aggregate concrete (RAC) holds significant potential. However, the effectiveness of such approaches is heavily dependent on the availability of high-quality and comprehensive datasets. One of the main challenges hindering industrial implementation is the need for large-scale and diverse data to achieve low prediction error and strong model generalization. Although several platforms have

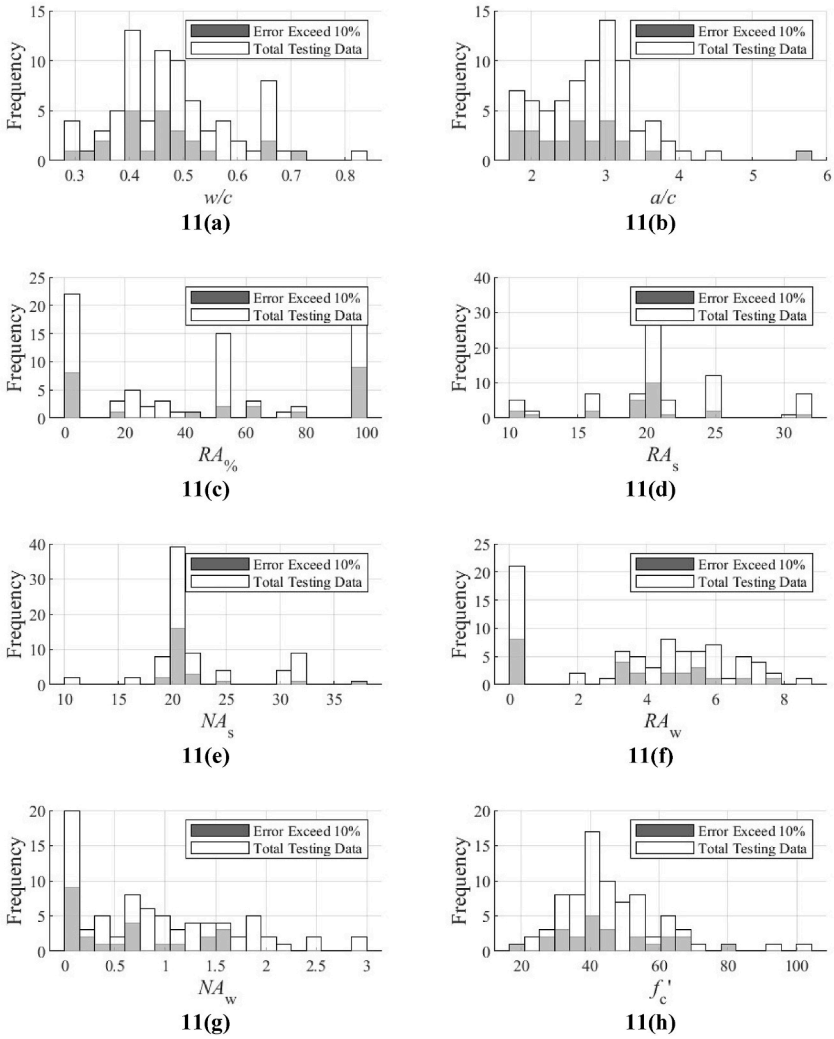
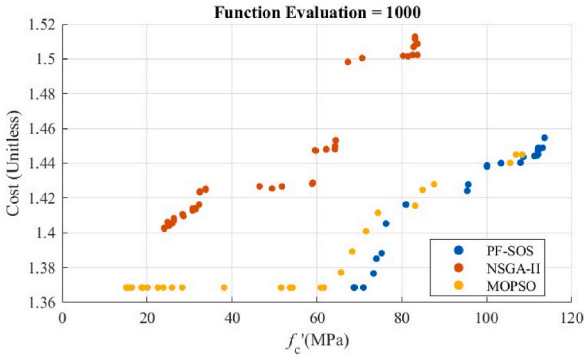


Fig. 11. Histogram of entire testing data and subset with error percentage exceeding 10 %.

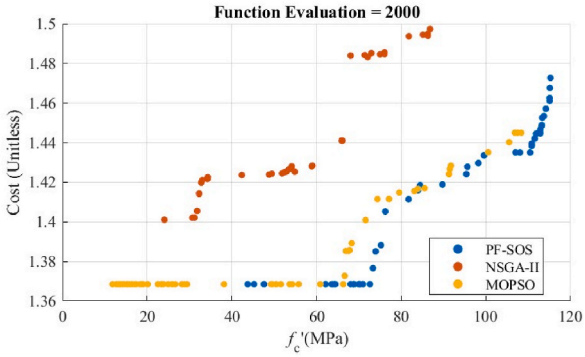
shared concrete mix datasets in an open-source manner, substantial variability still exists in experimental settings across laboratories and countries. Key factors such as aggregate gradation, ambient temperature, curing conditions, and testing protocols can vary significantly. Furthermore, these critical parameters are often not systematically recorded in publicly available datasets, making it difficult to establish a standardized mix design environment for robust AI model development. Another practical barrier is the computational infrastructure required for training machine learning models. In the present study, the ML-SOS algorithm was trained on a system equipped with an Intel 13th Gen i7-13700 processor and 32 GB RAM. With a dataset size of 528 entries, the total computation time was approximately 20 h. Ideally, the training process should be repeated each time a new dataset is introduced. While CPU-based computation remains adequate for datasets of up to around 2000 samples, larger datasets exceeding 10,000 entries are likely to require GPU acceleration and increased memory capacity to maintain reasonable training times.

Several limitations of the developed algorithm are noted as follows.

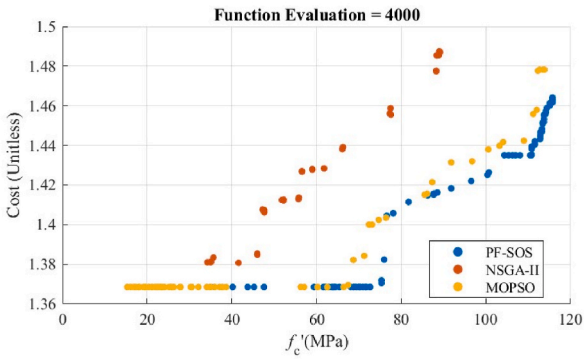
1. The study encountered challenges in acquiring sufficient data, especially concerning the source properties of recycled aggregates such as the characteristics of the parent concrete. Due to the limited availability of such information, these potentially important features could not be incorporated into the model, thereby restricting the range of input variables used for prediction.
2. No data pre-processing was performed to identify and eliminate potential outlier samples, which may have introduced noise or inaccuracies during training and led to prediction errors.
3. The current study does not provide specific guidance on the selection of algorithm parameters for ML-SOS and PF-SOS (e.g., number of iterations, organism size). These parameters may need to be adjusted depending on the characteristics and complexity of different case studies, and further investigation is required to establish more generalized recommendations.



12(a)



12(b)



12(c)

Fig. 12. Convergence of pareto front at different function evaluations.

4. The accuracy of the constructed Pareto Front is highly dependent on the performance of the ML-SOS model, which in turn is influenced by the quality and completeness of the training data.

This study opens several avenues for future research and development.

1. Broader Optimization Objectives: While this study focuses on optimizing cost and compressive strength, future research should consider additional critical objectives in RAC mix design, such as environmental impact, chemical resistance, and long-term durability. Incorporating these broader objectives would enhance the practical relevance and sustainability of the proposed optimization approach.

Table 6
Several optimal mixture from PF-SOS pareto front at final iteration.

No.	f_c' (MPa)	Cost	w/c	a/c	RA%	RA_s	NA_s	RA_w	NA_w
1	24.47	1.37	0.72	6.40	90.00	22.39	10.86	3.26	1.00
2	34.31	1.37	0.72	6.40	90.00	26.09	11.39	3.17	1.00
3	43.77	1.37	0.72	6.40	90.00	11.44	11.39	3.08	1.00
4	56.75	1.37	0.72	6.40	90.00	10.25	10.53	1.91	1.00
5	65.07	1.37	0.72	6.40	90.00	10.25	11.16	3.26	2.53
6	74.80	1.37	0.70	6.40	90.00	10.57	11.81	3.04	1.27
7	89.73	1.42	0.45	6.40	90.00	8.00	10.20	3.27	1.86
8	94.16	1.42	0.40	6.40	90.00	9.77	11.24	3.26	2.02
9	107.41	1.44	0.34	6.40	89.42	10.47	10.24	3.27	1.73
10	114.94	1.46	0.30	6.40	47.58	8.00	10.00	3.05	1.88

Table 7
Several optimal mixture from NSGA-II pareto front at final iteration.

No.	f_c' (MPa)	Cost	w/c	a/c	RA%	RA_s	NA_s	RA_w	NA_w
1	34.22	1.38	0.72	6.40	58.97	19.63	15.48	7.08	1.91
2	34.95	1.38	0.72	6.40	58.69	19.56	15.46	6.99	1.91
3	46.04	1.38	0.70	6.39	58.77	19.48	15.53	7.23	1.91
4	55.77	1.41	0.53	6.40	59.61	19.48	14.93	7.27	1.92
5	66.12	1.44	0.45	6.23	58.71	19.57	14.28	7.03	1.73
6	77.34	1.46	0.32	6.35	56.73	19.91	13.84	7.27	1.82
7	88.29	1.48	0.30	6.22	36.33	14.24	18.03	2.03	1.94

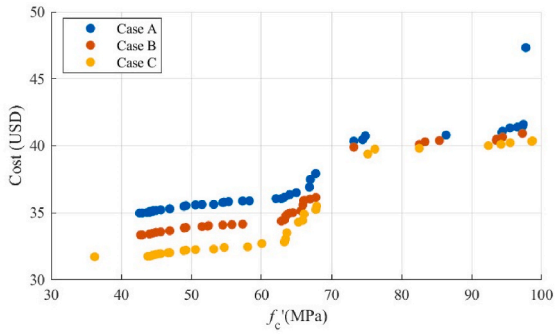
Table 8
Several optimal mixture from MOPSO pareto front at final iteration.

No.	f_c' (MPa)	Cost	w/c	a/c	RA%	RA_s	NA_s	RA_w	NA_w
1	25.19	1.37	0.72	6.40	90.00	32.00	14.96	1.29	2.57
2	35.01	1.37	0.72	6.40	90.00	30.60	17.05	6.70	1.00
3	38.67	1.37	0.72	6.40	90.00	15.38	11.91	10.00	1.06
4	56.26	1.37	0.72	6.40	90.00	8.00	10.00	2.62	1.00
5	66.43	1.37	0.72	6.40	90.00	8.85	12.29	1.01	1.00
6	74.69	1.40	0.70	6.40	11.97	8.00	12.94	2.36	1.34
7	85.55	1.42	0.45	6.40	90.00	8.00	11.55	6.04	1.00
8	96.77	1.43	0.36	6.40	90.00	8.00	11.69	3.61	1.00
9	104.09	1.44	0.31	6.39	90.00	8.00	11.05	1.36	1.00
10	113.84	1.48	0.29	6.40	10.00	8.00	13.06	3.34	1.09

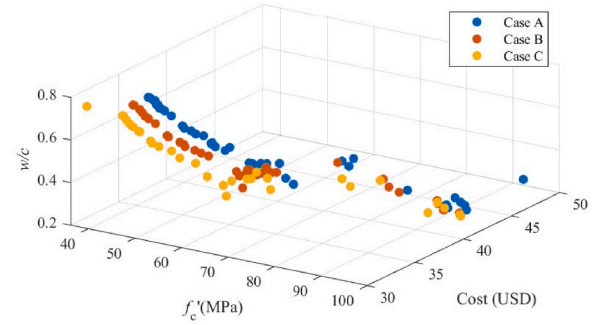
Table 9
RAC component price.

Component		Cement	Normal Aggregate 8 mm	Normal Aggregate 32 mm	Recycle Aggregate Price Compared to Normal Aggregate
Price per m^3 in USD	Case A	150	100	30	0.9
	Case B				0.8
	Case C				0.7

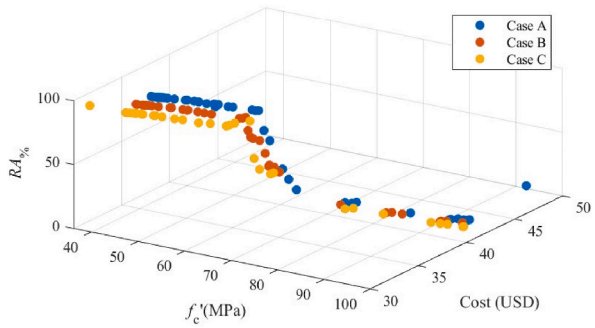
2. Experimental Validation of Results: To bridge the gap between research and industrial implementation, experimental validation of the predicted Pareto front solutions is essential. Conducting laboratory tests based on selected optimal mix designs will strengthen the credibility of the ML-based framework and promote greater adoption of machine learning in real-world concrete mix design practices.
3. Enhancement of PF-SOS Algorithm: The current PF-SOS algorithm has limitations in generating a well-distributed Pareto front across all objectives. Additionally, the weighting strategy for each organism's objective is based on a simple linear distribution. Future work could focus on incorporating adaptive weighting mechanisms and more sophisticated selection rules for the best organism, which may lead to improved convergence behavior and better Pareto front diversity.
4. Broader Applicability of ML-SOS and PF-SOS: The proposed ML-SOS and PF-SOS algorithms have strong potential for application in various engineering optimization problems beyond concrete mix design. The ML-SOS framework, in particular, can be extended to include a wider range of machine learning models beyond ANN, SVM, and RF, such as XGBoost, LightGBM, or deep learning architectures, thereby increasing its adaptability and robustness in solving complex prediction-based optimization tasks.



13(a)



13(b)



13(c)

Fig. 13. Pareto front computed using PF-SOS for several cost scenario.

CRediT authorship contribution statement

Hanna Chintya Febriani Gunawan: Visualization, Resources, Investigation. **John Thedy:** Writing – original draft, Formal analysis, Data curation, Conceptualization. **Bagus Hario Setiadji:** Supervision, Software, Resources, Investigation. **Ay Lie Han:** Writing – review & editing, Project administration, Funding acquisition. **Marc Ottele:** Validation, Software, Methodology, Formal analysis.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Han Ay Lie reports financial support was provided by Diponegoro University. Hanna Chintya Febriani Gunawan reports a

relationship with Diponegoro University that includes: employment. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors gratefully acknowledge the financial support of the World Class University (Visiting Professor), Universitas Diponegoro program through the grant number 174/UN7.A/HK/III/2025.

Data availability

Data will be made available on request.

References

[1] M. Zhaurova, R. Soukka, M. Horttanainen, Multi-criteria evaluation of CO2 utilization options for cement plants using the example of Finland, Int. J. Greenh. Gas Control 112 (2021) 103481, https://doi.org/10.1016/j.ijggc.2021.103481.
[2] H. Unis Ahmed, et al., Geopolymer concrete as a cleaner construction material: an overview on materials and structural performances, Clean. Mater. 5 (2022) 100111, https://doi.org/10.1016/j.clema.2022.100111.
[3] V.W.Y. Tam, M. Soomro, A.C.J. Evangelista, A review of recycled aggregate in concrete applications (2000–2017), Constr. Build. Mater. 172 (2018) 272–292, https://doi.org/10.1016/j.conbuildmat.2018.03.240.
[4] D. Caro, et al., Environmental and socio-economic effects of construction and demolition waste recycling in the European Union, Sci. Total Environ. 908 (2024) 168295, https://doi.org/10.1016/j.scitotenv.2023.168295.
[5] Q. Zhao, W. Gao, Y. Su, T. Wang, J. Wang, How can C&D waste recycling do a carbon emission contribution for construction industry in Japan city? Energy Build. 298 (2023) 113538 https://doi.org/10.1016/j.enbuild.2023.113538.
[6] V.W.Y. Tam, X.F. Gao, C.M. Tam, Microstructural analysis of recycled aggregate concrete produced from two-stage mixing approach, Cement Concr. Res. 35 (6) (2005) 1195–1203, https://doi.org/10.1016/j.cemconres.2004.10.025.
[7] V.W.Y. Tam, C.M. Tam, Y. Wang, Optimization on proportion for recycled aggregate in concrete using two-stage mixing approach, Constr. Build. Mater. 21 (10) (2007) 1928–1939, https://doi.org/10.1016/j.conbuildmat.2006.05.040.
[8] C. Thomas, J. Setién, J.A. Polanco, P. Alaejos, M. Sánchez de Juan, Durability of recycled aggregate concrete, Constr. Build. Mater. 40 (2013) 1054–1065, https://doi.org/10.1016/j.conbuildmat.2012.11.106.
[9] H. Guo, et al., Durability of recycled aggregate concrete – a review, Cem. Concr. Compos. 89 (2018) 251–259, https://doi.org/10.1016/j.cemconcomp.2018.03.008.
[10] G. Bai, C. Zhu, C. Liu, B. Liu, An evaluation of the recycled aggregate characteristics and the recycled aggregate concrete mechanical properties, Constr. Build. Mater. 240 (2020) 117978, https://doi.org/10.1016/j.conbuildmat.2019.117978.
[11] B. Wang, L. Yan, Q. Fu, B. Kasal, A comprehensive review on recycled aggregate and recycled aggregate concrete, Resour. Conserv. Recycl. 171 (2021) 105565, https://doi.org/10.1016/j.resconrec.2021.105565.
[12] E.K. Lauritzen, Demolition and Reuse of Concrete and Masonry, CRC Press, 1993, https://doi.org/10.1201/9781482271270.
[13] S. Nagataki, A. Gokce, T. Saeki, M. Hisada, Assessment of recycling process induced damage sensitivity of recycled concrete aggregates, Cement Concr. Res. 34 (6) (2004) 965–971, https://doi.org/10.1016/j.cemconres.2003.11.008.
[14] M. Etxeberria, A.R. Marí, E. Vázquez, Recycled aggregate concrete as structural material, Mater. Struct. 40 (5) (2007) 529–541, https://doi.org/10.1617/s11527-006-9161-5.
[15] F. Deng, Y. He, S. Zhou, Y. Yu, H. Cheng, X. Wu, Compressive strength prediction of recycled concrete based on deep learning, Constr. Build. Mater. 175 (2018) 562–569, https://doi.org/10.1016/j.conbuildmat.2018.04.169.
[16] E.M. Golafshani, A. Behnood, Estimating the optimal mix design of silica fume concrete using biogeography-based programming, Cem. Concr. Compos. 96 (2019) 95–105, https://doi.org/10.1016/j.cemconcomp.2018.11.005.
[17] H. Dabiri, M. Kioumarsi, A. Kheyroddin, A. Kandiri, F. Sartipi, Compressive strength of concrete with recycled aggregate; a machine learning-based evaluation, Clean. Mater. 3 (2022) 100044, https://doi.org/10.1016/j.clema.2022.100044.
[18] S.R. Salimbahrami, R. Shakeri, Experimental investigation and comparative machine-learning prediction of compressive strength of recycled aggregate concrete, Soft Comput. 25 (2) (2021) 919–932, https://doi.org/10.1007/s00500-021-05571-1.
[19] M. Sabău, J. Remolina Duran, Prediction of compressive strength of general-use concrete mixes with recycled concrete aggregate, Int. J. Pavement Res. Technol. 15 (1) (2022) 73–85, https://doi.org/10.1007/s42947-021-00012-6.
[20] M. Hosseinzadeh, M. Dehestani, A. Hosseinzadeh, Prediction of mechanical properties of recycled aggregate fly ash concrete employing machine learning algorithms, J. Build. Eng. 76 (2023) 107006, https://doi.org/10.1016/j.jobe.2023.107006.
[21] Y. Peng, C. Unluer, Modeling the mechanical properties of recycled aggregate concrete using hybrid machine learning algorithms, Resour. Conserv. Recycl. 190 (2023) 106812, https://doi.org/10.1016/j.resconrec.2022.106812.
[22] Y. Sun, J. Zhang, G. Li, Y. Wang, J. Sun, C. Jiang, Optimized neural network using beetle antennae search for predicting the unconfined compressive strength of jet grouting coalcretes, Int. J. Numer. Anal. Methods GeoMech. 43 (4) (2019), https://doi.org/10.1002/nag.2891.
[23] J.-S. Chou, C.-Y. Liu, H. Prayogo, R.R. Khasani, D. Gho, G.G. Lalitan, Predicting nominal shear capacity of reinforced concrete wall in building by metaheuristics-optimized machine learning, J. Build. Eng. 61 (2022) 105046, https://doi.org/10.1016/j.jobe.2022.105046.
[24] C. Jui-Sheng, N. Ngoc-Tri, P. Anh-Duc, Shear strength prediction in reinforced concrete deep beams using nature-inspired metaheuristic support vector regression, J. Comput. Civ. Eng. 30 (1) (Jan. 2016) 4015002, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000466.
[25] M.W. Falah, et al., Compressive strength prediction using coupled deep learning model with Extreme gradient boosting algorithm: environmentally friendly concrete incorporating recycled aggregate, Complexity (New York, N. Y) 2022 (1) (2022) 5433474, https://doi.org/10.1155/2022/5433474.
[26] I.-C. Yeh, Optimization of concrete mix proportioning using a flattened simplex–centroid mixture design and neural networks, Eng. Comput. 25 (2) (2009) 179–190, https://doi.org/10.1007/s00366-008-0113-2.
[27] C. Min-Yuan, P. Doddy, W. Yu-Wei, Novel genetic algorithm-based evolutionary support vector machine for optimizing high-performance concrete mixture, J. Comput. Civ. Eng. 28 (4) (Jul. 2014) 6014003, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000347.
[28] J. Zhang, Y. Huang, Y. Wang, G. Ma, Multi-objective optimization of concrete mixture proportions using machine learning and metaheuristic algorithms, Constr. Build. Mater. 253 (2020), https://doi.org/10.1016/j.conbuildmat.2020.119208.
[29] X. Yuan, et al., Machine Learning Prediction Models to Evaluate the Strength of Recycled Aggregate Concrete, 2022, https://doi.org/10.3390/ma15082823.
[30] M.-Y. Cheng, D. Prayogo, Symbiotic Organisms Search: a new metaheuristic optimization algorithm, Comput. Struct. 139 (2014) 98–112, https://doi.org/10.1016/j.compstruc.2014.03.007.
[31] S. Helfrich, A. Herzel, S. Ruzika, C. Thielen, Using scalarizations for the approximation of multiobjective optimization problems: towards a general theory, Math. Methods Oper. Res. 100 (1) (2024) 27–63, https://doi.org/10.1007/s00186-023-00823-2.

[32] N. Ryu, S. Min, Multiobjective optimization with an adaptive weight determination scheme using the concept of hyperplane, Int. J. Numer. Methods Eng. 118 (6) (2019) 303–319, https://doi.org/10.1002/nme.6013.
[33] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. 5 (4) (1943) 115–133, https://doi.org/10.1007/BF02478259.
[34] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297, https://doi.org/10.1007/BF00994018.
[35] L. Breiman, Random Forests, Mach. Learn. 45 (1) (2001) 5–32, https://doi.org/10.1023/A:1010933404324.
[36] A. Seshadri, NSGA-II: A Multi-Objective Optimization Algorithm, MAT-Lab Cent. Implementierung, 2009.
[37] M. Heris, Multi-Objective Particle Swarm Optimization (MOPSO), MATLAB Cent. File Exch., 2015 [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/52870-multi-objective-particle-swarm-optimization-mopso.