RESEARCH PAPER

Multi-objective optimization of reinforced concrete cantilever retaining wall: a comparative study

Ali R. Kashani[1] · Amir H. Gandomi[2] · Koorosh Azizi[1] · Charles V. Camp[1]

Abstract

This paper investigates the performance of four multi-objective optimization algorithms, namely non-dominated sorting genetic algorithm II (NSGA-II), multi-objective particle swarm optimization (MOPSO), strength Pareto evolutionary algorithm II (SPEA2), and multi-objective multi-verse optimization (MVO), in developing an optimal reinforced concrete cantilever (RCC) retaining wall. The retaining wall design was based on two major requirements: geotechnical stability and structural strength. Optimality criteria were defined as reducing the total cost, weight, CO_2 emission, etc. In this study, two sets of bi-objective strategies were considered: (1) minimum cost and maximum factor of safety, and (2) minimum weight and maximum factor of safety. The proposed method's efficiency was examined using two numerical retaining wall design examples, one with a base shear key and one without a base shear key. A sensitivity analysis was conducted on the variation of significant parameters, including backfill slope, the base soil's friction angle, and surcharge load. Three well-known coverage set measures, diversity, and hypervolume were selected to compare the algorithms' results, which were further assessed using basic statistical measures (i.e., min, max, standard deviation) and the Friedman test with a 95% level of confidence. The results demonstrated that NSGA-II has a higher Friedman rank in terms of coverage set for both cost-based and weight-based designs. SPEA2 and MOPSO outperformed both cost-based and weight-based solutions in terms of diversity in examples without and with the effects of a base shear key, respectively. However, based on the hypervolume measure, NSGA-II and MVO have a higher Friedman rank for examples without and with the effects of a base shear key, respectively, for both the cost-based and weight-based designs.

Keywords Retaining wall · Multi-objective optimization · Pareto front · Nondominated sorting genetic algorithm II (NSGA-II) · Multi-objective particle swarm optimization (MOPSO) · The strength Pareto evolutionary algorithm II (SPEA2 · Multi-objective multi-verse optimization (MVO)

1 Introduction

One of the major challenges in geotechnical engineering is stabilizing uneven natural and artificial soil slopes. The systems developed to retain such masses include gravity and cantilever retaining walls, sheet piles, anchored earth, and mechanically stabilized earth walls. The difference between

these retaining structures is based on how each can withstand unstable loads. Final cost is essential in determining which type of earth retaining structure is best suited. While reinforced concrete cantilever retaining (CCR) walls are among the most effective retaining structures in a wide range of construction projects, they are relatively costly due to the massive bulk of the required materials. Hence, any effort to decrease the final cost of CCR walls is pertinent.

Artificial intelligence-based algorithms have found wide applications in facilitating complicated civil engineering problems (Najafzadeh and Azamathulla 2015; Najafzadeh and Kargar 2019; Gandomi et al. 2021; Kashani et al. 2021a; Maniat et al. 2021). Optimization algorithms have proven effective in preparing cost-effective designs for engineering problems. Optimization algorithms have been applied successfully to a wide variety of civil engineering

1 Department of Civil Engineering, University of Memphis, Memphis, TN 38152, USA

2 Faculty of Engineering & IT, University of Technology Sydney, Ultimo, NSW 2007, Australia

problems (e.g., structural engineering (Gandomi et al. 2013; Akhani et al. 2019; Bekdaş et al. 2019; Kashani et al. 2020; Kashani et al. 2021b, c; Gholizadeh et al. 2020), water engineering (Azizi et al. 2017; Ebrahimi and Khorram 2021; Ali et al. 2022; Azari et al. 2022; Moeini et al. 2022), geotechnical engineering, transportation engineering (Yang et al. 2012; Gandomi et al. 2015a, b, 2017a; Gandomi and Kashani 2017; Kashani et al. 2016, 2019a, b, 2021d, 2022a, b), construction management (Sahib and Hussein 2019; Panwar and Jha 2019), and structural damage detection (Mishra et al. 2019; Fathnejat and Ahmadi-Nedushan 2020; Akhani and Pezeshk 2022)). Due to their stochastic nature and varied performances, metaheuristic optimization algorithms require constant updates, which can be done by 1- developing new algorithms (Gandomi 2014; Mirjalili and Lewis 2016; Saremi et al. 2017; Mirjalili et al. 2017a), or 2- enhancing the algorithms' performances (Jordehi 2015; Gandomi and Kashani 2016, 2018a, b; Gandomi and Deb 2020; Bigham and Gholizadeh 2020). As a result, research on applying metaheuristic algorithms in a wide range of engineering problems is an active study area.

Numerous studies have applied optimization algorithms to minimize the final cost of CCR walls over the past few years. Optimization approaches simplify the design procedure by satisfying three primary criteria: geotechnical stability, structural strength, and economic efficiency. For example, Khajehzadeh et al. (2010) used particle swarm optimization, and then later employed a modified optimization method. Khajehzadeh and Eslami (2012) applied the gravitational search algorithm, Ceranic et al. (2001) utilized simulated annealing, Camp and Akin (2012) applied Big Bang Big Crunch, Aydogdu (2017) tried biogeography-based optimization algorithms, Gandomi et al. (2015c, 2017b, c) and Gandomi and Kashani (2018a, b) considered evolutionary and swarm optimization algorithms.

These studies' main limitation is the relatively high final cost, weight, or CO_2 emission associated with a single objective during the design optimization procedure. Although the single objective approach provides a cost-effective final design for decision-makers, it may not reflect all aspects of the design since the important efficiency-related features often have conflicting and reciprocal relations. Therefore, it is impossible to find a design that satisfies the optimality criteria for all conflicting objectives. Multi-objective optimization has proved to be a sophisticated approach to identifying solutions, called Pareto optimal solutions, to such conflicts (Deb 2001; Marchionatti and Gambino 1997). Therefore, multi-objective optimization for finding the Pareto front (PF) solutions has become widespread throughout science and engineering (Gunantara 2018; Afshari et al. 2019; Gholizadeh and Fattahi 2021; Behmanesh et al. 2020; Rangaiah et al. 2020).

In most construction projects, particularly retaining walls, the main concern is minimizing the final cost and maximizing safety. The stronger and bulkier the wall, the higher the safety. As a result, the final cost would be much higher than the optimal low-cost design. Multi-objective optimization of retaining walls has been addressed in just a few studies. For instance, (Kaveh et al. 2013) employed a non-dominated sorting genetic algorithm (NSGA-II) for retaining wall optimization that considered bar congestion and cost as two conflicting objectives. In another effort, (Das et al. 2016) utilized NSGA-II for retaining wall optimization considering the final cost and factor of safety.

In this study, multi-objective optimization of retaining walls was investigated, emphasizing two different combinations of objectives based on the study by (Sarıbaş and Erbatur 1996): (1) minimum cost and maximum factor of safety (cost-based design), and (2) minimum weight and maximum factor of safety (weight-based design). Four multi-objective algorithms, i.e., non-dominated sorting genetic algorithm II (NSGA-II) (Deb et al. 2000), multi-objective particle swarm optimization (MOPSO) (Coello and Lechuga 2002), strength Pareto evolutionary algorithm II (SPEA2) (Zitzler et al. 2001), and multi-objective multi-verse optimization (MVO) (Mirjalili et al. 2016), were utilized for retaining wall optimization. Three different measures were considered to compare the performances of the proposed algorithms. A computer program based on ACI 318-05 (2005) and analysis presented by (Das 2010) was developed in MATLAB to analyze retaining wall designs. The second design example also explored the effect of a base shear key. A sensitivity analysis was conducted on the cost-based design of Example 2 for the effective parameters, including surcharge load (q), base soil's friction angle (ϕ), and backfill slope (β). To be more specific, the main contributions of this work include (1) analyzing the trend of variation of cost and weight by changing the factor of safety of the retaining walls using multi-objective optimization, (2) a systematic comparative study of the performance of different classes of multi-objective optimization algorithms based different measures and statistical analysis on solving Reinforced Concrete Cantilever Retaining Wall problems, (3) examining the impact of design parameters on the final objective through a sensitivity analysis, and (4) providing a practical design procedure based on geotechnical and structural regulations provided in ACI 218-05. The results indicate that the NSGA-II and SPEA2 algorithms were more efficient than MOPSO and MVO.

2 Methodology

A feasible retaining wall design should satisfy geotechnical stability and structural strength requirements (Das 2010). In the former, three factors of safety—overturning (FS_O), sliding (FS_S), and the foundation's bearing capacity

(FS$_B$)—were considered to guarantee the serviceability of the structure. In the latter, the shear and moment capacity of each section of the wall (stem, heel, toe, and shear key) were checked with ACI 318–05 (2005) regulations. In formulating the optimal design of the retaining wall, a bi-objective function is proposed for minimizing cost/weight and maximizing the factors of safety (FS$_O$, FS$_S$, and FS$_B$) as follows

$$\begin{cases} f_{\text{cost}} = C_s W_{st} + C_c V_c \\ FS = (FS_O + FS_S + 2 \times FS_B)^{-1} \end{cases}, \tag{1}$$

$$\begin{cases} f_{\text{weight}} = W_{st} + 100 V_c \gamma_c \\ FS = (FS_O + FS_S + 2 \times FS_B)^{-1} \end{cases}, \tag{2}$$

where C_s and C_c are the unit cost of steel and concrete, respectively, W_{st} is the weight of reinforcing steel, and V_c and γ_c are the volume and unit weight of concrete scaled by a factor of 100 as proposed by (Sarıbaş and Erbatur 1996), respectively.

Figure 1 defines the twelve design variables for the retaining wall in this study: width of the base slab (X1), the width of the toe slab (X2), stem thickness at the bottom of the wall (X3), stem thickness at the top of the wall (X4), base slab thickness (X5), distance from the front of the shear key to the front of the toe of the wall (X6), the width of the shear key (X7), the height of the shear key (X8), the vertical steel area in the stem per unit length of the wall (R1), the horizontal steel area of the toe slab (R2), the horizontal steel area of the heel slab (R3), and the vertical steel area of the shear key per unit length of the wall (R4). Discrete variables were considered for steel reinforcement areas, according to Table 1. In this study, the design procedure and constraints are defined based on (Camp and Akin 2012).

3 Description of optimization algorithms

3.1 Non-dominated sorting genetic algorithm II

Unlike a single-objective optimization problem, which provides only a single optimal solution, a multi-objective optimization problem provides solutions representing trade-offs between conflicting objectives known as a Pareto optimal set. Several techniques have been proposed to obtain the Pareto optimal solutions in literature (Coello et al. 2007). Due to their effectiveness and easy implementation,

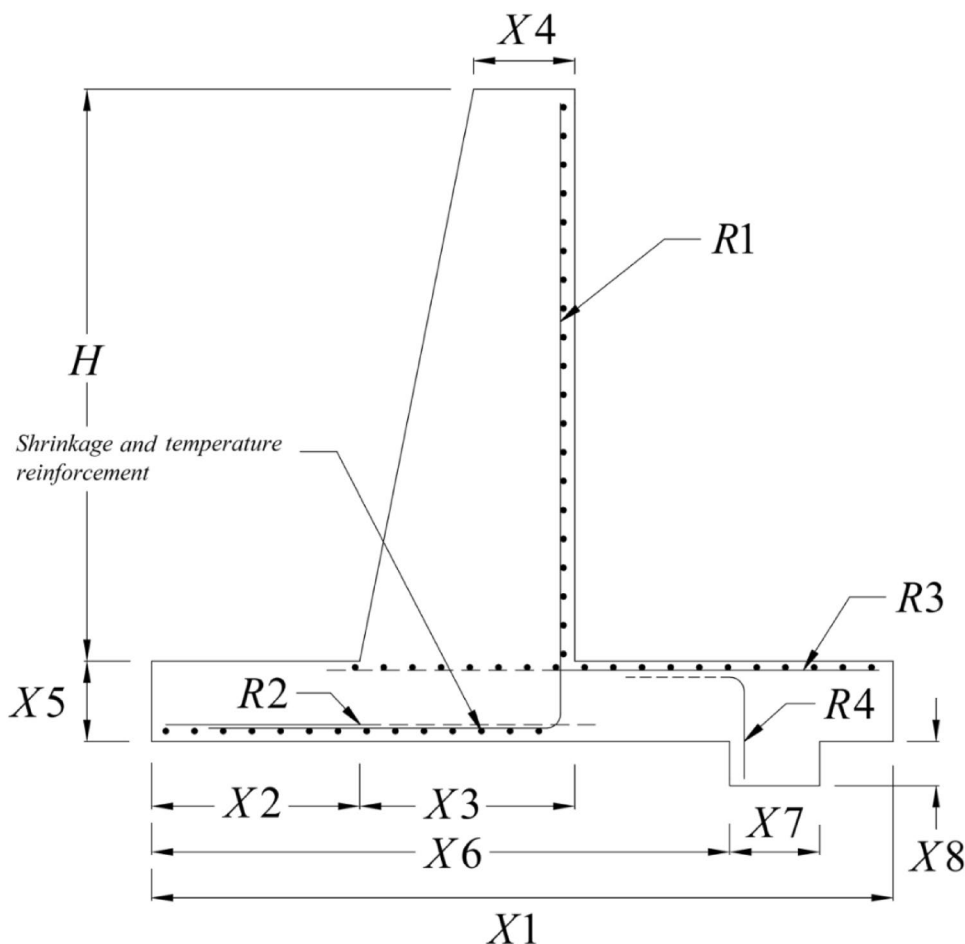**Fig. 1** Design variables for a general retaining wall

Table 1 Steel reinforcement properties for design variables *R1* to *R4*

Index number (η)	Reinforcement		Total A_s (cm^2)
	Quantity	Bar size (mm)	
1	3	10	2.356
2	4	10	3.141
3	3	12	3.392
4	5	10	3.926
5	4	12	4.523
⋮	⋮	⋮	⋮
221	16	30	113.097
222	17	30	120.165
223	18	30	127.234

multi-objective evolutionary algorithms (MOEAs) have gained much attention from researchers. One of the significant metaheuristic algorithms in this category is NSGA-II, an improved version of the non-dominated sorting genetic algorithm (Deb et al., 2000).

NSGA-II applies an elitism-based non-dominated approach for ranking and sorting solutions using a crowding distance method in its selection operator to maintain the diversity in the obtained Pareto optimal set (Deb et al. 2002). First, in non-dominated sorting, each solution's objective functions are evaluated, and the whole population is ranked and sorted into different non-dominated levels based on the dominance count. Second, an infinite crowding distance value is assigned to the solutions after defining the boundary values based on the smallest and largest function values. Finally, the crowding distance between any two neighboring solutions is computed based on the normalized difference in the objective function values (Deb et al. 2002). Compared to NSGA, NSGA-II offers an improved mating mechanism dependent on the crowding distance and performance constraints using an adapted explanation of dominance without the use of penalty functions.

3.2 Multi-objective evolutionary algorithm based on decomposition

The strength pareto evolutionary algorithm 2 (SPEA2), developed by (Zitzler et al. 2001), is a non-dominated genetic algorithm with features like a truncation operator, nearest neighbor, and genetic operators (crossover and mutation) that maintain diversity and best convergence of solutions. Like most evolutionary algorithms, SPEA2 uses a set of solutions encoded as the chain of a chromosomic number in a generational approach. In each generation, all population members that are evaluated receive a fitness value. Better performance results in a high fitness value, and as such, a high probability is chosen to develop new generations using the chromosomes pairs crossing concept (Zitzler et al. 2001).

SPEA2 uses an external archive, including the previously identified non-dominated solutions, and is updated every generation. In the next step, a strength value is computed for each solution. Based on these strength values, the fitness of each individual is calculated. The fitness assignment strategy of SPEA2 considers the number of individuals that an individual dominates and the number of its dominators. The algorithm also uses the nearest neighbor density approach and archive truncation method to maintain diversity and preserve the boundary solutions, respectively (Zitzler et al. 2001).

3.3 Multi-objective particle swarm optimization

Particle swarm optimization (PSO) (Eberhart and Kennedy 1995; Kennedy and Eberhart 1995) was initially proposed as a simulation model based on studying the choreography of a bird flock for optimizing continuous nonlinear functions. There are two definitions in the application of PSO: the individual best and the global best. In swarm optimization, the particles search for the best solution based on their experience and the other particles within the same swarm. Then, each particle compares its fitness value at the current position to the best fitness value it had attained before. The best-known position, with the best fitness, for each particle, is $p$best. The best position for the entire swarm is $g$best. The ith particle velocity (v_i) is updated for the $k+1$ iteration according to the following equation

$$v_i^{k+1} = wv_i^k + c_1 \text{rand}_1 \times \left(p\text{best}_i - s_i^k \right) + c_2 \text{rand}_2 \times \left(g\text{best} - s_i^k \right), \tag{3}$$

where w is a weight function, c_1 and c_2 are acceleration factors, rand$_1$ and rand$_2$ are random numbers $\in [0, 1]$, and s_i^k is the position of each particle. The weight function is

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{\text{iter}_{\max}} \times \text{iter}, \tag{4}$$

where $w_{\max}$ is the initial weight, $w_{\min}$ is the final weight, iter$_{\max}$ refers to the maximum iteration number, and *iter* is the current iteration number.

The new position of each particle is updated using the velocities as

$$s_i^{k+1} = s_i^k + v_i^{k+1}. \tag{5}$$

A new multi-objective PSO (MOPSO) algorithm was introduced by Coello and Lechuga (2002). Optimization can be performed for more than one conflicting objective simultaneously. In the MOPSO algorithm, the swarm is initialized following the identification of a set of leaders with the swarm's non-dominated particles. These are stored in a secondary repository of particles to guide

particle flight (Sha and Lin 2010). A quality measure is computed for all leaders to select one leader for each particle of the swarm. A leader for each particle is selected, and the flight is performed for each generation. After performing the flight, the MOPSO algorithm applies a mutation operator to evaluate the particle and its corresponding personal best experience (*p*best) is updated. A new particle replaces its *p*best particle when this particle is dominated or if both are non-dominated with respect to each other. The set of leaders is updated once all the particles are updated. The quality measure for the leaders is then recalculated, and this process is repeated for a defined number of iterations.

3.4 Multi-objective multi-verse optimization

The multi-verse optimizer (MVO) is an optimization algorithm developed by Mirjalili et al. (2017b) based on multi-verse theory in astrophysics. It emulates the interplay among universes in the Big Bang theory and how they interact with each other through different types of holes, such as black, white, and wormholes.

The MVO computational process includes several iterations to send and receive objects (variable) to and from universes (solutions) based on their inflation rates (fitness values) through wormholes. This function helps the exploration and exploitation processes from becoming trapped in local optima. The mathematical representation of the MVO algorithm is described as

$$x_i^j = \begin{cases} x_k^j, & \text{if } rm_1 < \text{NI}(x_i) \\ x_i^j, & \text{otherwise} \end{cases}, \tag{6}$$

where x_i^j is the jth parameter of the ith universe; x_k^j is the jth parameter of the kth universe chosen by roulette wheel selection; rm_1 indicates a random value $\in [0, 1]$; x_i represents the ith universe; and $\text{NI}(x_i)$ shows the normalized fitness value of the ith solution. The objects of the universes are updated using wormholes to improve the inflation rate, as follows

$$x_i^j = \begin{cases} \begin{cases} x_j^b + \text{TDR} \times \left((ub_j - lb_j) \times rm_4 + lb_j \right) rm_3 \\ x_j^b - \text{TDR} \times \left((ub_j - lb_j) \times rm_4 + lb_j \right) rm_3 \geq 0.5 \end{cases} & rm_2 < \text{WEP} \\ x_i^j & rm_2 \geq \text{WEP} \end{cases}, \tag{7}$$

where x_i^j presents the jth parameter value of jth solution; x_j^b indicates the jth parameter of the best solution, lb_j is the lower bound and ub_j the upper bound of the jth variable, rm_2, rm_3, and rm_4 are random numbers [0, 1], WEP and TDR represent adaptive variables. WEP is the wormhole existence probability and is employed to enhance exploitation. TDR is the traveling distance rate to allow objects to fly to the best

universe through a wormhole. The adaptive formulas for WEP and TDR coefficients are given as follows

$$\text{WEP} = \text{WEP}_{\min} + \text{iter} \times \left(\frac{\text{WEP}_{\max} - \text{WEP}_{\min}}{\text{iter}_{\max}} \right) \tag{8}$$

$$\text{TDR} = 1 - \frac{\text{iter}^{1/p}}{\text{iter}_{\max}^{1/p}} \tag{9}$$

where iter is the given iteration, $\text{WEP}_{\min}$ and $\text{WEP}_{\max}$ are default values set to 0.2 and 1, respectively, and p is a default value set to 6 that shows the accuracy of the exploitation (Mirjalili et al. 2016).

4 Performance indices

In this paper, the performances of the algorithms are compared using several standard measures. Previous studies have proposed various performance measures to evaluate various aspects (e.g., convergence and diversity) of a non-dominated solution set (Deb 2001). As explained in (Zitzler 1999; Zitzler et al. 2000; Azevedo and Araújo 2011), the performance indices quantify the convergence and diversity of final Pareto solutions in multi-objective problems. This study used the following three performance indices to compare different Pareto sets regarding the global Pareto set obtained from all sets.

4.1 Coverage set (CS)

The CS index, proposed by Zitzler et al. (2000), compares the non-dominated degrees of optimal solutions from different iterations. The values of CS (X', X'') are in the range of 0–1. When CS is equal to 1, all X'' are dominated by or equal to X'. To understand the exact non-dominated relationship between two different iterations, two cases of CS need to be analyzed: (X', X'') and (X'', X'). The CS index is expressed as

$$\text{CS}(X', X'') = \frac{\left| \left\{ a'' \in X''; \exists a' \in X' : a' \geq a'' \right\} \right|}{|X''|} \tag{10}$$

where X' and X'' are the optimal solutions from different iterations, and a' and a'' are optimal solutions.

4.2 Diversity (DI)

The DI index, proposed by Zitzler (1999), evaluates the diversity of optimal solutions from the multi-objective optimization

algorithm. It can be expressed using the minimum and maximum values of the objective function as

$$\mathrm{DI} = \sqrt{\frac{1}{M} \sum_{m=1}^{M} \left(\frac{\max f_{\mathrm{m}} - \min f_{m}}{F_{\mathrm{m}}^{\max} - F_{\mathrm{m}}^{\min}} \right)^2} \tag{11}$$

where $F_{\mathrm{m}}^{\max}$ and $F_{\mathrm{m}}^{\min}$ are the maximum and minimum values of Pareto fronts, respectively, f_{m} is the mth value of the objective function, and M is the number of objective functions.

4.3 Hypervolume (HV)

The HV, defined by Zitzler and Thiele (1999), is a popular performance index that measures the proximity and diversity of a Pareto approximate front. Specifically, it measures the volume of the partition of the objective space bound between the Pareto approximate front and a reference point. The HV is a complete unary performance metric in terms of weak dominance relation; HV's solution is not weakly dominated by its opponent Zitzler and Thiele (1999). In this study, as Deb (2001) suggested, HV is calculated in the normalized objective space, where all results are normalized to be inside the unit hypercube [0, 1]m. Therefore, all normalized HV values (or simply *NHV*) are less than or equal to 1. Higher values of *NHV* are desirable, like the other two performance metrics. *HV* is calculated as

$$\mathrm{HV} = \sum_{i=1}^{|Q|} v_i \tag{12}$$

where v_i is a hypercube constructed using reference point W (found by constructing a vector of worst objective function values) at the ith solution, Q is the objective space (total search space), and W is the reference point that can simply be found by constructing a vector of worst objective function values. Therefore, HV is calculated using a union of all hypercubes.

5　Numerical simulation

This section examines the proposed algorithms' efficiency based on two case studies presented by Sarıbaş and Erbatur (1996). Example 1 considers the optimal design of a retaining wall without a base shear key (Sarıbaş and Erbatur 1996), and Example 2 includes the effects of a base shear key. Table 2 lists the parameter values for numerical examples, and Table 3 lists the boundary limitations of the design variables. ACI 318–05 (2005) requirements and discrete variables for steel reinforcement are considered in the design. Both examples use two different sets of objective functions: (1) minimum cost and maximum FOS (cost-based design), and (2) minimum weight and maximum FOS (weight-based design). Under this contextual analysis, each optimization algorithm is run independently 50 times, with a population size of 50 for 1000 generations. The best non-dominated Pareto solution (true Pareto front) is obtained by huddling all the utilized algorithms' Pareto solutions.

In the same way, the best Pareto front for each algorithm is the set of non-dominated solutions from the 50 runs, which were then compared with the true Pareto front using performance indices CS, DI, and *NHV*. The CS index, the

Table 2 Parameter values for Examples 1 and 2

Input parameters	Symbol	Value		Unit
		Example 1	Example 2	
Height of stem	H	3.0	4.5	m
Yield strength of reinforcing steel	f_y	400	400	MPa
Compressive strength of concrete	f_c	21	21	MPa
Concrete cover	C_c	7	7	cm
Shrinkage and temperature reinforcement ratio	ρ_{st}	0.002	0.002	-
Surcharge load	q	20	30	kPa
Backfill slope	β	10	0	°
Internal friction angle of retained soil	ϕ	36	36	°
Internal friction angle of base soil	ϕ'	0	34	°
Unit weight of retained soil	γ_s	17.5	17.5	kN/m³
Unit weight of base soil	γ_s'	18.5	18.5	kN/m³
Unit weight of concrete	γ_c	23.5	23.5	kN/m³
Cohesion of base soil	C	125	0	kPa
Depth of soil in front of the wall	D	0.5	0.75	m
Cost of steel	C_s	0.4	0.4	$/kg
Cost of concrete	C_c	40	40	$/m³

Table 3 Design variable boundary constraints for Examples 1 and 2 (from Camp and Akin (2012))

Design variables	Unit	Example 1		Example 2	
		Lower bound	Upper bound	Lower bound	Upper bound
X_1	m	1.3090	2.3333	1.96	5.5
X_2	m	0.4363	0.7777	0.65	1.16
X_3	m	0.2000	0.3333	0.25	0.5
X_4	m	0.2000	0.3333	0.25	0.5
X_5	m	0.2722	0.3333	0.4	0.5
X_6	m	–	–	1.96	5.5
X_7	m	–	–	0.2	0.5
X_8	m	–	–	0.2	0.5
R_1	–	1	223	1	223
R_2	–	1	223	1	223
R_3	–	1	223	1	223
R_4	–	–	–	1	223

Fig. 2 Pareto front comparison for the cost-based design of Example 1

most important index of the multi-objective optimization solutions, was used to calculate the ratio of the number of non-dominated solutions obtained by a method to the number of Pareto solutions in the total set of solutions. The DI shows how well a method finds a widespread Pareto front. The *NHV* is calculated by the hypercube from the given objective functions to show the convergence aspect of the Pareto front. Also, basic statistical measures were computed (i.e., min, max, mean, standard deviation (STD)) along with a Friedman test with a 95% confidence level considering all 50 runs.

5.1 Example 1: 3-m tall retaining wall design without a base shear key

Figures 2 and 3 show the best aggregate Pareto fronts, resulting from 50 independent runs for each algorithm, for the cost-based and weight-based designs, respectively. Moreover, the true Pareto front is also depicted to better compare the algorithms' performances. For the cost-based design, NSGA-II and SPEA2 have a more significant overlap with the true Pareto front, while MOPSO and MVO Pareto fronts were far from the true Pareto front. There is a negligible difference between the presented solutions and true Pareto for all algorithms for the weight-based design.

The algorithms' performances were further evaluated using three measures, CS, DI, and *NHV* (see Fig. 4). The CS results in Fig. 4 confirm the observations in Figs. 2 and 3 that NSGA-II and SPEA2 participate more than the other two algorithms in forming the true Pareto front in both cost-based and weight-based designs. A comparison of NSGA-II and SPEA2 revealed that NSGA-II contributes more to the true Pareto than SPEA2. DI results suggest that SPEA2 is
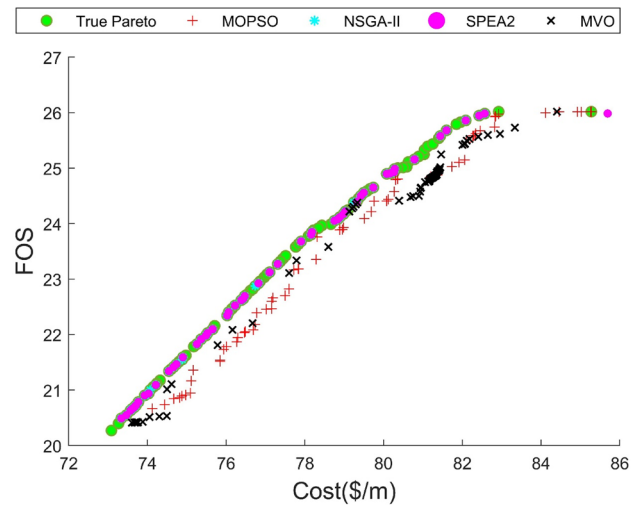
the better solver for the cost-based design, while NSGA-II is better for the weight-based design. *NHV* metrics indicate minor differences between these algorithms.

Tables 4, 5 and 6 list a statistical comparison of the proposed performance measures (i.e., CS, DI, and NHV). In this way, the proposed indices are calculated for every run out of 50 runs. The results are presented as min, mean, max, and standard deviation (STD) and ranked using the Friedman statistical test at a 95% significance level. The CS results, listed in Table 4, indicate that NSGA-II exhibited better performance than the other algorithms due to its higher mean value. Also, the Friedman test results confirm the advantage of NSGA-II over the other approaches. Based on the DI
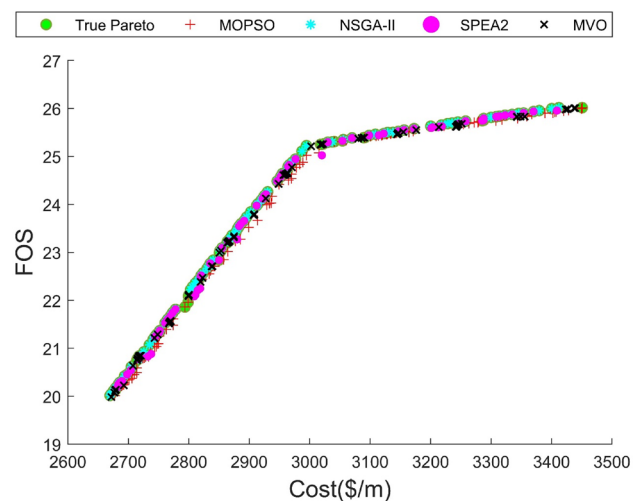

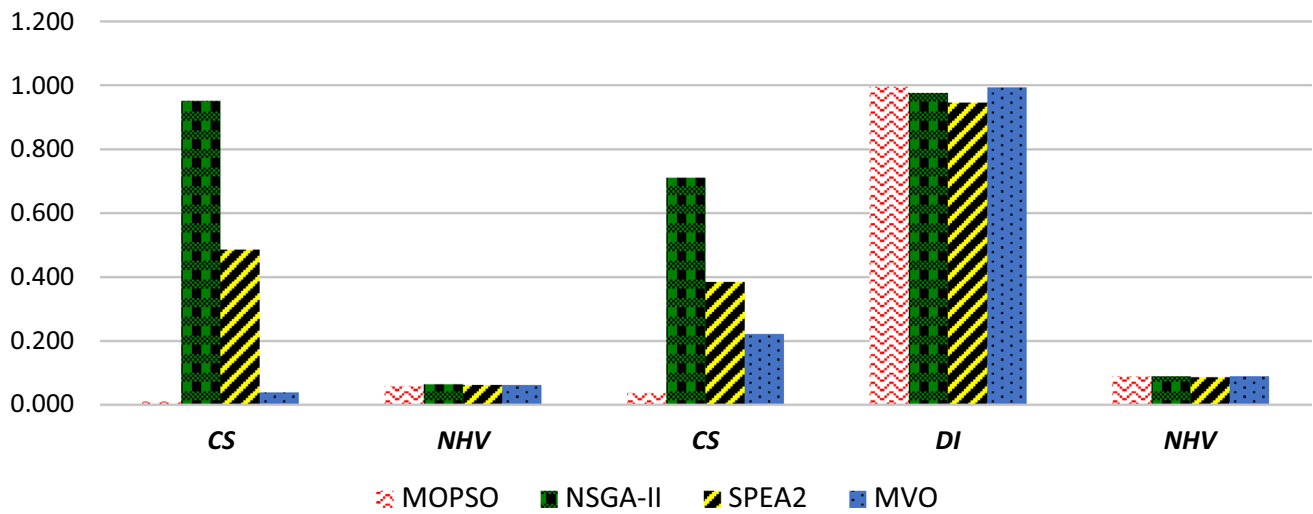**Fig. 3** Pareto front comparison for the weight-based design of Example 1

Fig. 4 Comparison of performance measures for Example 1

Table 4 CS values of the cost-based designs for Example 1

CS	MOPSO	NSGAII	SPEA2	MVO
Min	0	0.072848	0.039735	0
Max	0.05298	0.675497	0.344371	0.02649
Mean	0.001987	0.136954	0.090464	0.001589
STD	0.008084	0.081728	0.041981	0.005054
Friedman rank	3.1	1	2	3.1

Table 6 NHV values of the cost-based designs for Example 1

NHV	MOPSO	NSGA-II	SPEA2	MVO
Min	42.83728	68.98337	67.47319	63.5848
Max	69.46702	70.21706	69.52192	76.40606
Mean	62.79358	69.67386	68.88013	69.73618
STD	5.386051	0.254108	0.552678	2.88383
Friedman rank	3.892857	1.392857	2.607143	2.107143

Table 5 DI values the cost-based designs for Example 1

DI	MOPSO	NSGA-II	SPEA2	MVO
Min	0.745712	0.900899	0.930785	0.874225
Max	1.15339	0.909986	2.95404	1.219239
Mean	0.950427	0.906319	1.391317	0.962048
STD	0.083037	0.002049	0.531021	0.070583
Friedman rank	2.75	3.477273	1.25	2.522727

Table 7 CS values of the weight-based designs for Example 1

CS	MOPSO	NSGA-II	SPEA2	MVO
Min	0	0.007407	0.007407	0
Max	0.051852	0.733333	0.392593	0.259259
Mean	0.008741	0.074074	0.049333	0.030222
STD	0.010123	0.097305	0.05285	0.036102
Friedman rank	3.7	1.14	2	2.6

metric analysis listed in Table 5, the obtained mean values indicate that the SPEA2 algorithm dominated the other cost-based design methods. The NHV test results of the cost-based design, shown in Table 6, again confirm that NSGA-II outperformed the other algorithms.

Tables 7, 8 and 9 compare the proposed measures (i.e., CS, DI, and NHV) for weight-based designs. Results also exhibit the superior performance of NSGA-II over the other algorithms, like in the cost-based design. The CS and NHV results proposed NSGA-II as the best solver due to its lower mean values. The Friedman ranking scores were also

consistent with this observation. However, for the DI metric, SPEA2 demonstrated better performance than NSGA-II.

5.2 Example 2: 4.5-m tall retaining wall design with and without a base shear key

In Example 2, two cases are considered: one with a base shear key (Case I) and one without a base shear key (Case II). In this example, cohesionless soil is considered for the base. Tables 2 and 3 list values for parameters in this example and the domains for the design variables. Figures 5 and 6 show the best Pareto fronts of the utilized algorithms for

Table 8 DI values of the weight -based designs for Example 1

DI	MOPSO	NSGA-II	SPEA2	MVO
Min	0.943343	0.970709	0.947282	0.890416
Max	1.069722	1.010534	1.04477	1.026334
Mean	0.990945	0.984446	0.998352	0.974814
STD	0.023612	0.012098	0.01817	0.022988
Friedman rank	2.25	2.863636	1.704545	3.181818

Table 9 NHV values of the weight -based designs for Example 1

NHV	MOPSO	NSGA-II	SPEA2	MVO
Min	0.082976	0.088529	0.087423	0.078086
Max	0.094875	0.093313	0.092672	0.093059
Mean	0.088059	0.090226	0.089384	0.08917
STD	0.002401	0.001435	0.001446	0.002351
Friedman rank	3.142857	1.571429	2.928571	2.285714

Fig. 6 Pareto front comparison for the weight-based design of Example 2-Case I
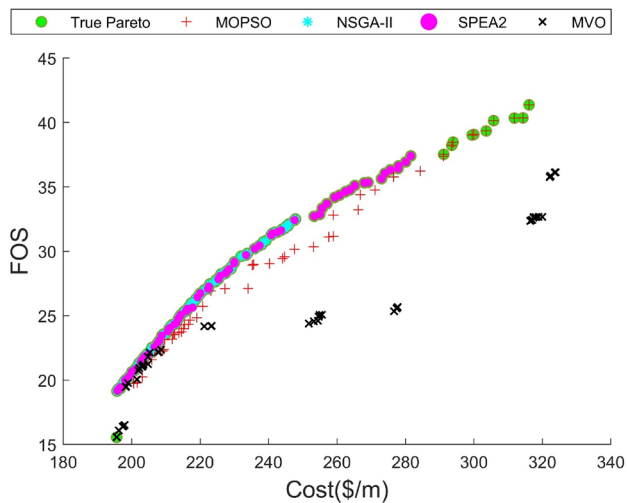
Fig. 5 Pareto front comparison for the cost-based design of Example 2-Case I

cost-based and weight-based designs of Case I. Most of the solutions in the Pareto fronts of NSGA-II and SPEA2 are coincident with the true Pareto. Although all MOPSO's results were not located on the true optimal solution set, they were very close (Fig. 7).

In contrast, most of MVO's Pareto solutions were far from the true Pareto front. Figure 5 shows that SPEA2 and NSGA-II had more contributions to the true Pareto fronts from the lower cost and FOS to intermediate cost and FOS for the cost-based design. However, MOPSO could find solutions in the true optimal Pareto front for the higher FOS and cost values. On the other hand, NSGA-II and SPEA2 were more involved in forming the true Pareto front for higher weight
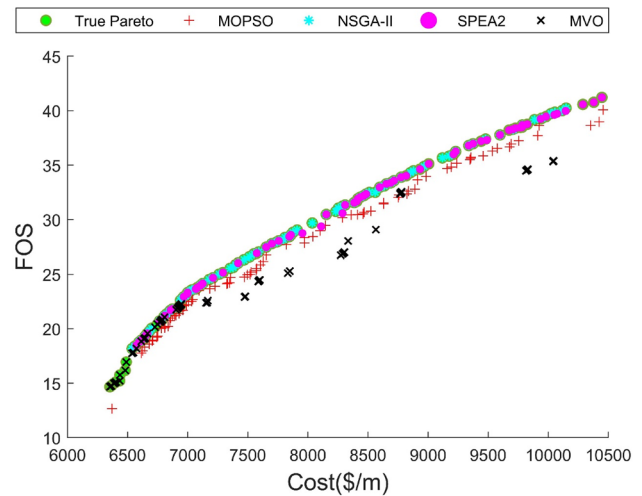
and FOS for the weight-based design. The MVO algorithm was effective for the lower cost and FOS solutions.

The comparison of the CS performance measure reflects the better performance of NSGA-II over the other algorithms for both cost-based and weight-based designs. The second-best algorithm was SPEA2 based on the CS metric. As shown in Fig. 7, although MOPSO and MVO performed better than NSGA-II and SPEA2 in terms of diversity, their solutions did not cover the true Pareto front provided in Fig. 6. Moreover, SPEA2 resulted in more diverse solutions than NSGA-II for both cost-based and weight-based designs. Figure 7 also shows MOPSO and MVO as the best and second-best algorithms in terms of diversity and hyper volume. However, recorded better coverage for both cost and weight designs.

Tables 10, 11, 12, 13, 14 and 15 provide a statistical comparison of CS, DI, and *NHV* for cost-based and weight-based designs for Case I. The CS results for both cost-based and weight-based designs indicate NSGA-II and SPEA2 as the best and second-best methods, confirmed by Friedman test results. Based on this measure, these two algorithms play a more critical role in forming the true Pareto front. Although MVO and MOPSO were the better solvers considering DI and NHV measures, their performances are not satisfactory since they are far from the true Pareto front. Further comparison of NSGA-II and SPEA2 as the best optimizers based on CS measures proves that SPEA2 performs better with higher mean values of DI and *NHV*. The Friedman test ranking results indicate MOPSO and MVO are the best solvers based on DI and NHV measures.

A sensitivity analysis was conducted on the variation of the cost-based designs for the following parameters: (1) backfill soil slope, (2) base soil's friction angle, and (3) surcharge load. Figures 8a–c present the sensitivity analysis
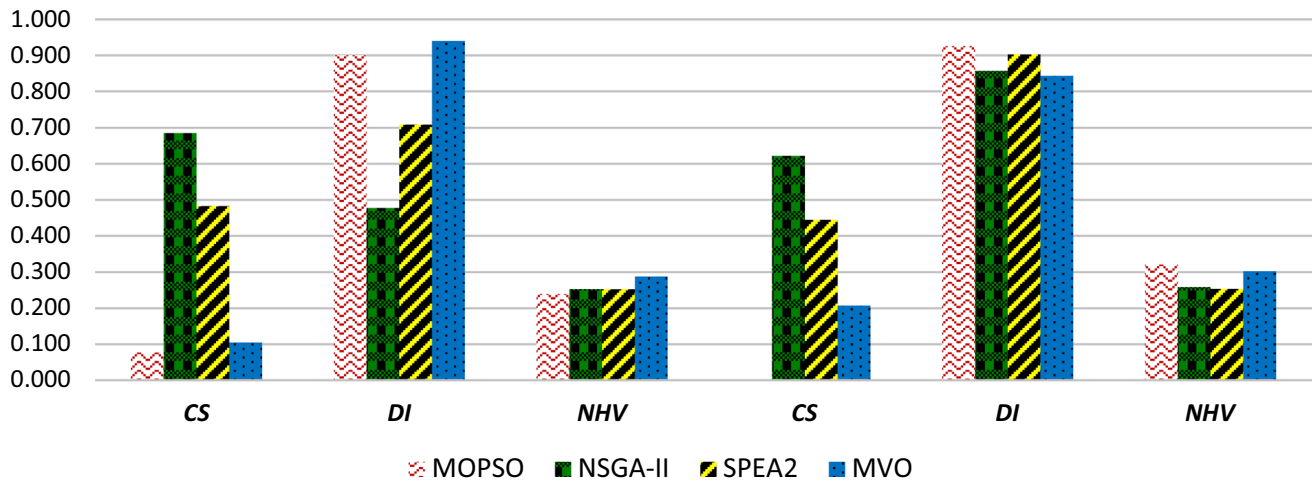
Example 2 (Case 1)

Fig. 7 Comparison of *DI* measures for numerical simulations

Table 10 CS values of the cost-based designs for Example 2 (Case I)

CS	MOPSO	NSGA-II	SPEA2	MVO
Min	0	0.055944	0.020979	0
Max	0.076923	0.811189	0.594406	0.13986
Mean	0.002238	0.238042	0.181958	0.017063
STD	0.010873	0.138482	0.110812	0.031435
Friedman rank	3.42	1.04	2.04	3

Table 11 DI values of the cost-based designs for Example 2 (Case I)

DI	MOPSO	NSGA-II	SPEA2	MVO
Min	0.027597	0.461265	0.708702	0.544536
Max	1.35927	0.783306	1.444243	1.387263
Mean	0.995983	0.621469	0.898854	1.028522
STD	0.302235	0.083627	0.126839	0.186008
Friedman rank	1.75	3.875	2.55	1.825

Table 12 NHV values of the cost-based designs for Example 2 (Case I)

NHV	MOPSO	NSGA-II	SPEA2	MVO
Min	0.105347	0.246328	0.245034	0.243297
Max	0.313575	0.272526	0.270078	0.321297
Mean	0.269526	0.253897	0.251761	0.280271
STD	0.036413	0.005721	0.00524	0.024699
Friedman rank	2.071429	2.571429	3.571429	1.714286

Table 13 CS values of the weight-based designs for Example 2 (Case I)

CS	MOPSO	NSGA-II	SPEA2	MVO
Min	0	0	0	0
Max	0.007407	0.762963	0.474074	0.207407
Mean	0.000148	0.02237	0.014815	0.004593
STD	0.001037	0.106208	0.066204	0.029027
Friedman rank	1.96	1.02	1.28	1.94

Table 14 DI values of the weight-based designs for Example 2 (Case I)

DI	MOPSO	NSGA-II	SPEA2	MVO
Min	0.384083	0.500607	0.802895	0.715858
Max	1.083038	0.856874	1.001166	1.081281
Mean	0.957371	0.691432	0.894992	0.942152
STD	0.157757	0.084455	0.050732	0.093038
Friedman rank	1.612903	3.83871	2.451613	2.096774

Table 15 NHV values of the weight-based designs for Example 2 (Case I)

NHV	MOPSO	NSGA-II	SPEA2	MVO
Min	0.234896	0.255354	0.25346	0.25932
Max	0.334012	0.315642	0.321269	0.335072
Mean	0.296541	0.269353	0.270501	0.311588
STD	0.027239	0.012983	0.014586	0.022558
Friedman rank	2.392857	2.75	3.285714	1.392857

results of the cost-based design of Case I. Figure 8a demonstrates that increasing β values from 5° to 25° resulted in more expensive design values and a right shift in the Pareto front towards higher costs. On the other hand, the decreasing FOS resulted in a downward shift of the Pareto front. It can be seen that the inclination of the Pareto front decreases with increasing β values, meaning that for lower backfill slopes, we can get a higher factor of safeties by increasing the final cost rather than higher backfill slopes. Surcharge loads had the same effect as the backfill slope on the final design. Changing the surcharge load within the considered domain on the final cost was less intensive than the backfill slope. However, this effect caused a higher reduction in FOS for the surcharge load than the backfill slope. It can be seen the final cost varied in the lower domain, while FOS was higher than the ones for backfill soil slopes.

In contrast, Fig. 8b indicates the positive effect of the base soil's friction angle on the final design, where increasing the ϕ values resulted in lower costs and higher FOSs. The Pareto fronts shrank for more intensive cases, which suggests that attaining the higher FOSs was much more expensive for highly intensive cases.

Figure 9 compares the extreme design points for this case study. The observation confirms that increasing β resulted in increased cost. In Fig. 9a, the minimum cost value of 172.78 ($/m) was obtained by NSGA-II, and MOPSO found a maximum of 308.92 ($/m). The variation in cost values was about 78.79%. Moreover, these changes diminished FOSs by about 48.88%. MOPSO achieved the maximum FOS of 48.51 and a minimum of 24.8 (see Fig. 9b). Increasing ϕ from 28° to 38° caused a decrease in cost and increased FOS. Figure 9c shows that the maximum cost value of 241.61 ($/m) is related to the MOPSO algorithm at ϕ equal to 28°, while NSGA-II obtained the minimum cost of 156.70 ($/m) at ϕ equal to 38°. Cost and FOS variations were 35.14% and 213.47%, respectively. Increasing the surcharge load resulted in a 55.19% increase in cost and a 64.05% decrease in FOS. the NSGA-II algorithm found the minimum FOS of 29.60 and the cost of 128.90 ($/m). MOPSO obtained the maximum FOS of 82.33 and the cost of 200.03 ($/m). In summary, FOS experienced the maximum variations by changing ϕ, and the most significant changes in cost were attributed to changing β.

Figures 10 and 11 present the Pareto fronts for each algorithm for Case II's cost-based and weight-based designs. Most of the solutions on the true Pareto front were obtained by NSGA-II and SPEA2, similar to the previous examples. For Case II, MOPSO did not participate in forming the true optimal solution, while MVO did contribute to developing part of the true Pareto front. In particular, MVO was able to find more solutions on the true Pareto front in the area with lower FOS and lower cost for the cost-based designs. MVO successfully found more solutions coincident with the true Pareto front for weight-based designs. A comparison of Figs. 5 and 6 with Figs. 10 and 11 shows that adding a base shear key does not affect the final cost, weight, or FOS values on the Pareto fronts.

Comparing performance measures in Fig. 12 resulted in the same conclusion as the previous examples. Based on CS measures, NSGA-II outperformed the other algorithms, followed by SPEA2; however, SPEA2 outperformance NSGA-II in both cost-based and weight-based designs based on DI measures. There is no significant difference between these algorithms based on NHV values.

Tables 16, 17, 18, 19, 20 and 21 list a statistical comparison of CS, DI, and NHV for the Case II cost-based and weight-based designs. The CS results indicate that NSGA-II and SPEA2 were the best algorithms because of their higher mean values for both designs, with NSGA-II exhibiting better performance. Considering the DI index, SPEA2 performed better than NSGA-II, while NHV revealed no considerable difference between these algorithms.

Figures 8d–f display sensitivity analyses of the Pareto fronts for Case II cost-based designs and show similar results to those of Case I. In general, increasing β and q increased the cost designs, while FOS values decreased. An inverse trend was recorded by increasing ϕ values. It can be seen that the range of cost was maximized by changing β, while FOS varied most with changing q. The inclination of the Pareto front for lower β and q values was significant, meaning that higher FOSs can be obtained by slightly increasing the final cost. However, these inclinations were reduced at higher β and q, where a moderate increase in FOS resulted in a higher cost. The Pareto fronts' inclinations for ϕ are less than those for β and q, meaning choosing a higher FOS value results in higher costs compared to β and q.

Figure 13 compares extreme design points and shows the same trends as those for Case I. It can be observed that increasing β and q resulted in higher costs and lower FOSs, which was the opposite for ϕ variations. Increasing β from 5° to 25° resulted in a 65.27% increase in the cost values and a 53.08% decrease in FOS values. MVO and MOPSO obtained the minimum cost of 171.09 ($/m) and FOS of 23.23. MOPSO found the maximum cost of 282.78 ($/m) and FOS of 49.51. Cost values decreased by 27.19%, and FOS values increased by 125.88% by varying ϕ from 28° to 38°. In this case scenario, the lowest cost of 158.51 ($/m) and FOS of 25.51 were obtained by MVO and MOPSO, respectively. MOPSO and SPEA2 obtained the highest cost of 217.72 ($/m) and FOS of 57.63. Varying q from 0 to 50 (kPa) resulted in a 56.96% increase in cost values and a 64.97% decrease in FOS values. The lowest cost of 130.04 ($/m) and FOS of 29.23 was obtained by MVO and MOPSO, respectively. Finally, the highest cost of 204.11 ($/m) and FOS of 83.46 was determined by MOPSO and NSGA-II, respectively.
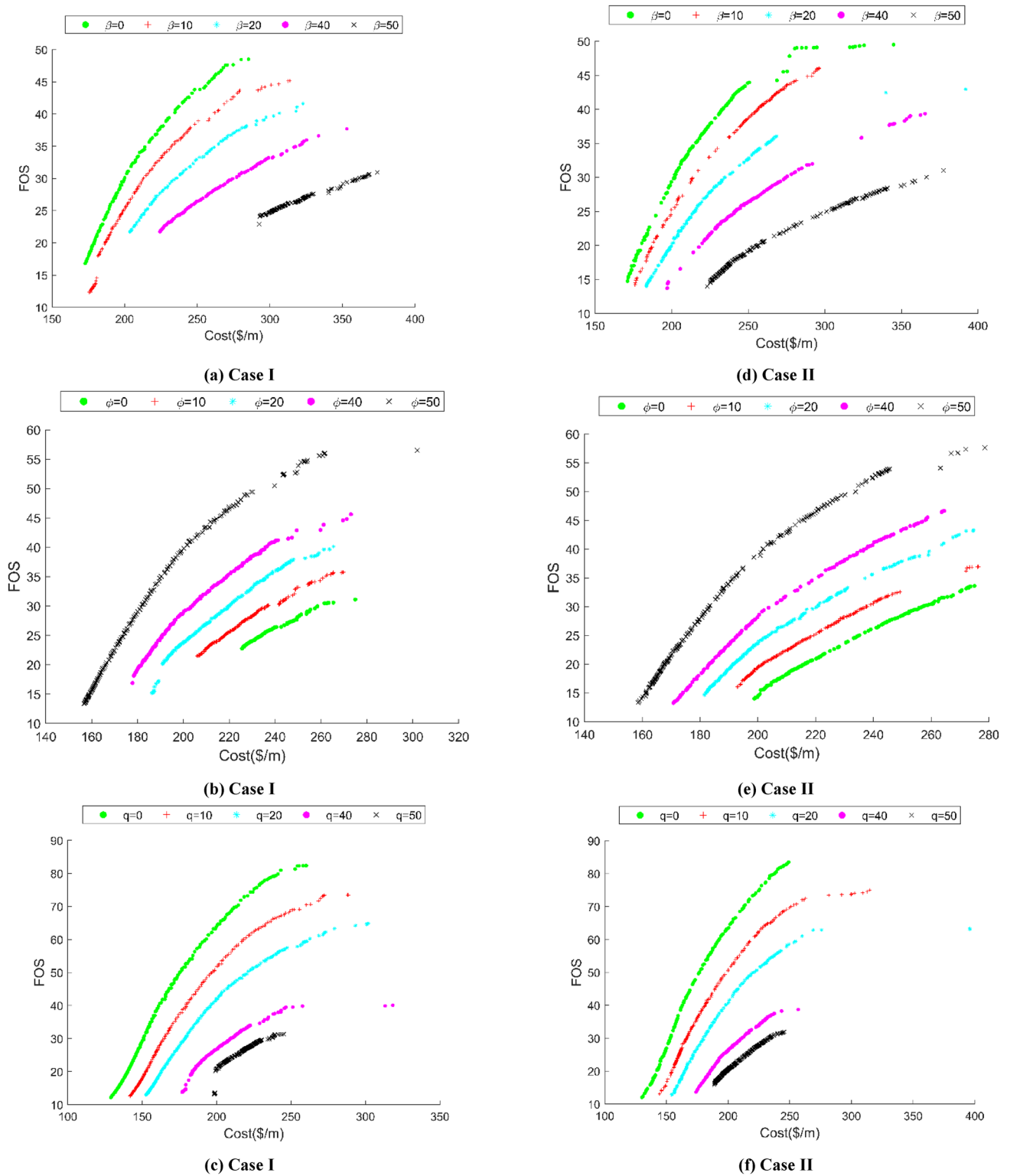
(a) Case I

(d) Case II

(b) Case I

(e) Case II

(c) Case I

(f) Case II

Fig. 8 Cost-based design sensitivity to **a** backfill slope, **b** base soil friction angle, and **c** surcharge load
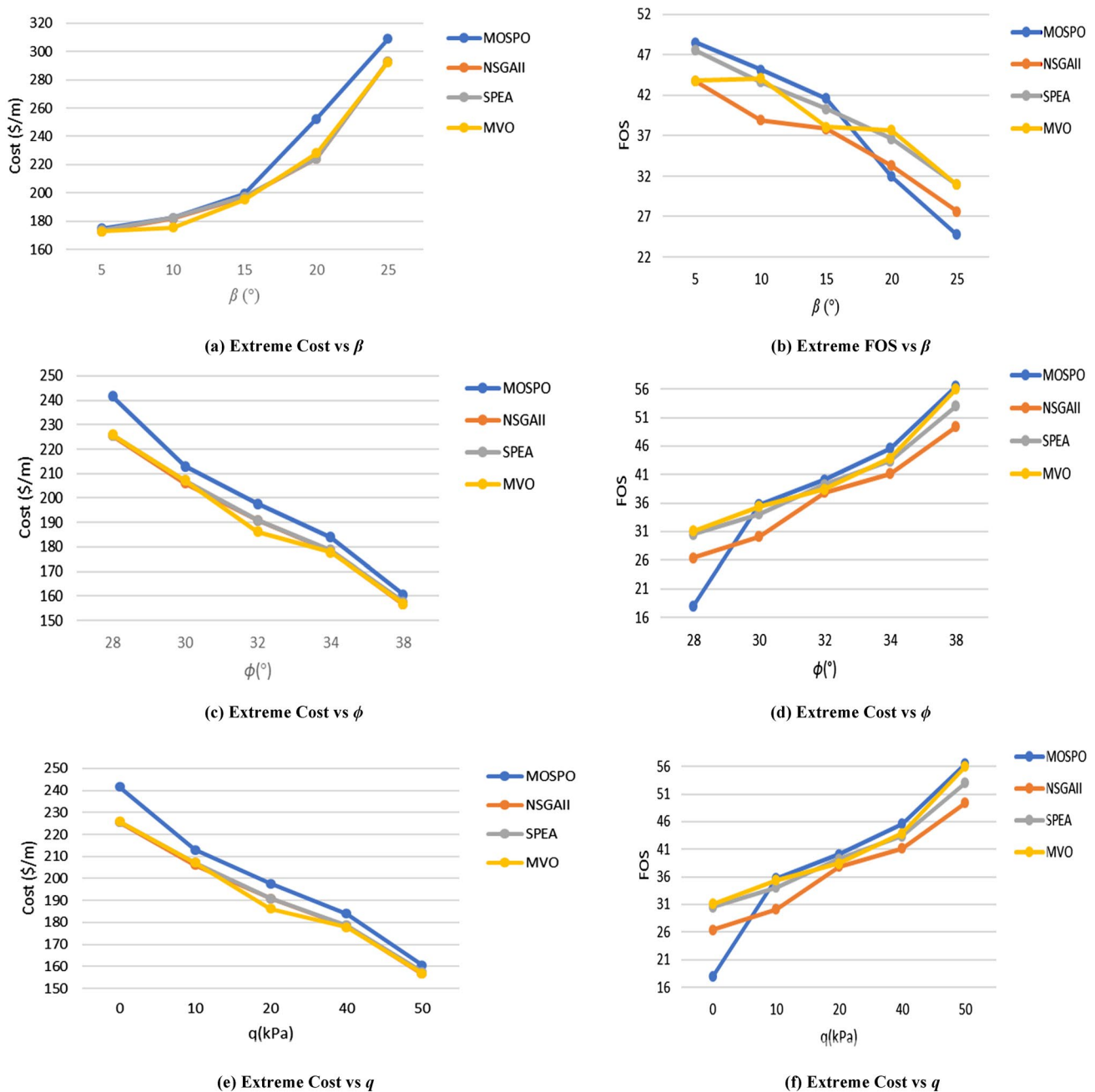
Fig. 9 Comparison of the extreme design points for the cost-based design of Case I

6 Summary and conclusions

In this study, four metaheuristic algorithms, including non-dominated sorting genetic algorithm II (NSGA-II), multi-objective particle swarm optimization (MOPSO), strength Pareto evolutionary algorithm II (SPEA2), and multi-objective multi-verse optimization (MVO), were utilized for multi-objective optimization of retaining walls.
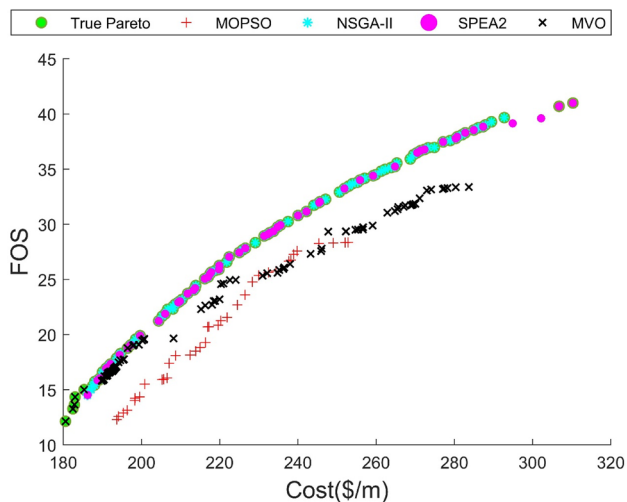
Fig. 10 Pareto front comparison for the cost-based design of Example 2-Case II
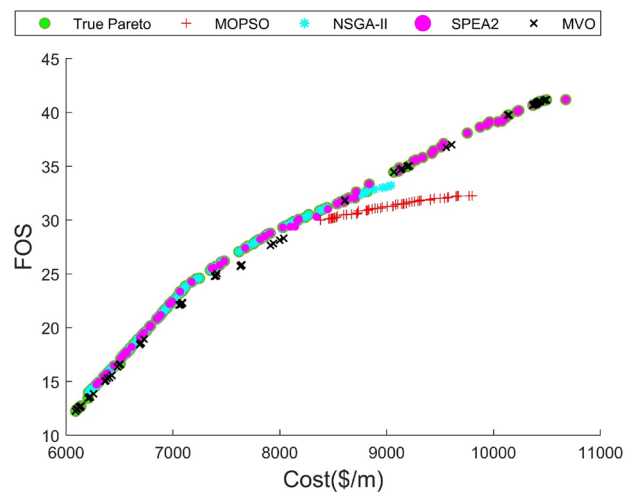
Fig. 11 Pareto front comparison for the weight-based design of Example 2-Case II

Two case studies were considered for two different designs with different sets of objectives: (1) minimum cost and maximum factor of safety (cost-based design), and (2) minimum weight and maximum factor of safety (weight-based design). Moreover, the effect of a base shear key on the final design was studied in the second example. This study aimed to (1) apply different multi-objective optimization methods to the design of a retaining wall, (2) provide an efficient comparison between the algorithms' performances, (3) explore the effect of a base shear key on the final designs, and (4) conduct a sensitivity analysis on critical design parameters, i.e., surcharge load (q), base soil's friction angel (ϕ), and backfill slope (β). The performances of the utilized algorithms were measured with three well-known indices: coverage set (CS), diversity (DI), and hypervolume (NHV).

Comparing the CS values indicated that the NSGA-II and SPEA2 had the most significant contribution in forming the true optimal Pareto front. Based on DI values, SPEA2 provided better performance. Also, observations indicate that CS and DI have inverse relations, whereby the algorithms with higher CS resulted in lower DI. Based on NHV measures, there were negligible differences between the algorithms. While *DI* measure values for MOPSO and MVO were higher than those for NSGA-II and SPEA2, these methods did not significantly contribute to the true Pareto fronts.

Comparing the Pareto fronts for Case I and Case II of Example 2 shows little effect of the base shear key on cost and FOS. Sensitivity analysis demonstrated that decreasing ϕ and increasing β and q resulted in increased cost, causing shrinkage in the Pareto front. The contraction of Pareto fronts in the retaining wall without a base shear key was much more significant than those with a base shear key. Moreover, for more intensive cases, the inclination of Pareto fronts was smaller, which further indicates that high FOSs required a significant increase in the final costs.
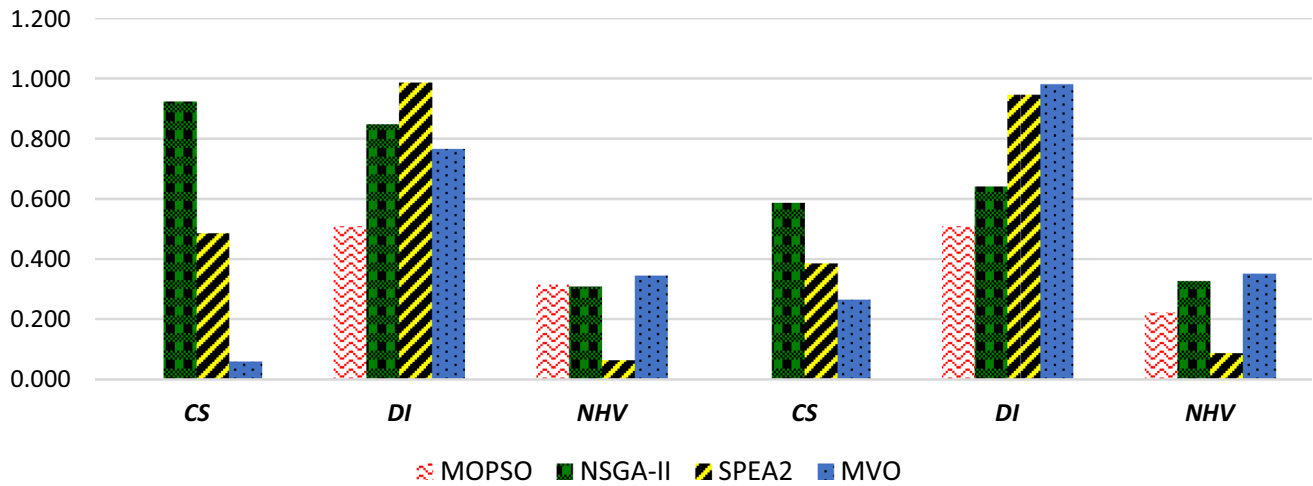
Example 2 (Case 2)

Fig. 12 Comparison of *NHV* measures for numerical simulations

Table 16 CS values for cost-based designs for Example 2 (Case II)

CS	MOPSO	NSGA-II	SPEA2	MVO
Min	0	0.038835	0.019417	0
Max	0	0.941748	0.514563	0.058252
Mean	0	0.177282	0.127184	0.01165
STD	0	0.119458	0.069504	0.016704
Friedman rank	3.42	1	2	3

Table 19 CS values for weight-based designs for Example 2 (Case II)

CS	MOPSO	NSGA-II	SPEA2	MVO
Min	0	0	0	0
Max	0	0.593548	0.445161	0.264516
Mean	0	0.017548	0.012516	0.005548
STD	0	0.082551	0.062075	0.037017
Friedman rank	2	1	1.28	1.9

Table 17 DI values for cost-based designs for Example 2 (Case II)

DI	MOPSO	NSGA-II	SPEA2	MVO
Min	0	0.586228	0.706534	0.477998
Max	1.313303	0.950737	1.358221	1.537728
Mean	0.8175	0.784649	0.950155	0.958777
STD	0.394283	0.086219	0.128496	0.266955
Friedman rank	2.384615	3.230769	2.153846	2.230769

Table 20 DI values for weight-based designs for Example 2 (Case II)

DI	MOPSO	NSGA-II	SPEA2	MVO
Min	0	0.641127	0.801433	0.782222
Max	1.070252	0.945235	1.004358	1.110663
Mean	0.703617	0.811018	0.926126	0.961678
STD	0.282253	0.082238	0.048824	0.070183
Friedman rank	3.105263	3.210526	2.105263	1.578947

Table 18 NHV values for cost-based designs for Example 2 (Case II)

NHV	MOPSO	NSGA-II	SPEA2	MVO
Min	0.029955	0.251887	0.267848	0.329959
Max	0.3365	0.345768	0.34437	0.355047
Mean	0.280312	0.318917	0.317677	0.345141
STD	0.067605	0.018479	0.016153	0.006012
Friedman rank	3.785714	2.142857	2.821429	1.071429

Table 21 NHV values for weight-based designs for Example 2 (Case II)

NHV	MOPSO	NSGA-II	SPEA2	MVO
Min	0.088823	0.272771	0.275661	0.337748
Max	0.359912	0.358022	0.356916	0.363611
Mean	0.316293	0.328966	0.335877	0.35694
STD	0.059365	0.019167	0.012972	0.004416
Friedman rank	2.75	3	2.892857	1.178571

(a) Extreme Cost vs β

(b) Extreme FOS vs β

(c) Extreme Cost vs ϕ

(d) Extreme Cost vs ϕ

(e) Extreme Cost vs q

(f) Extreme Cost vs q
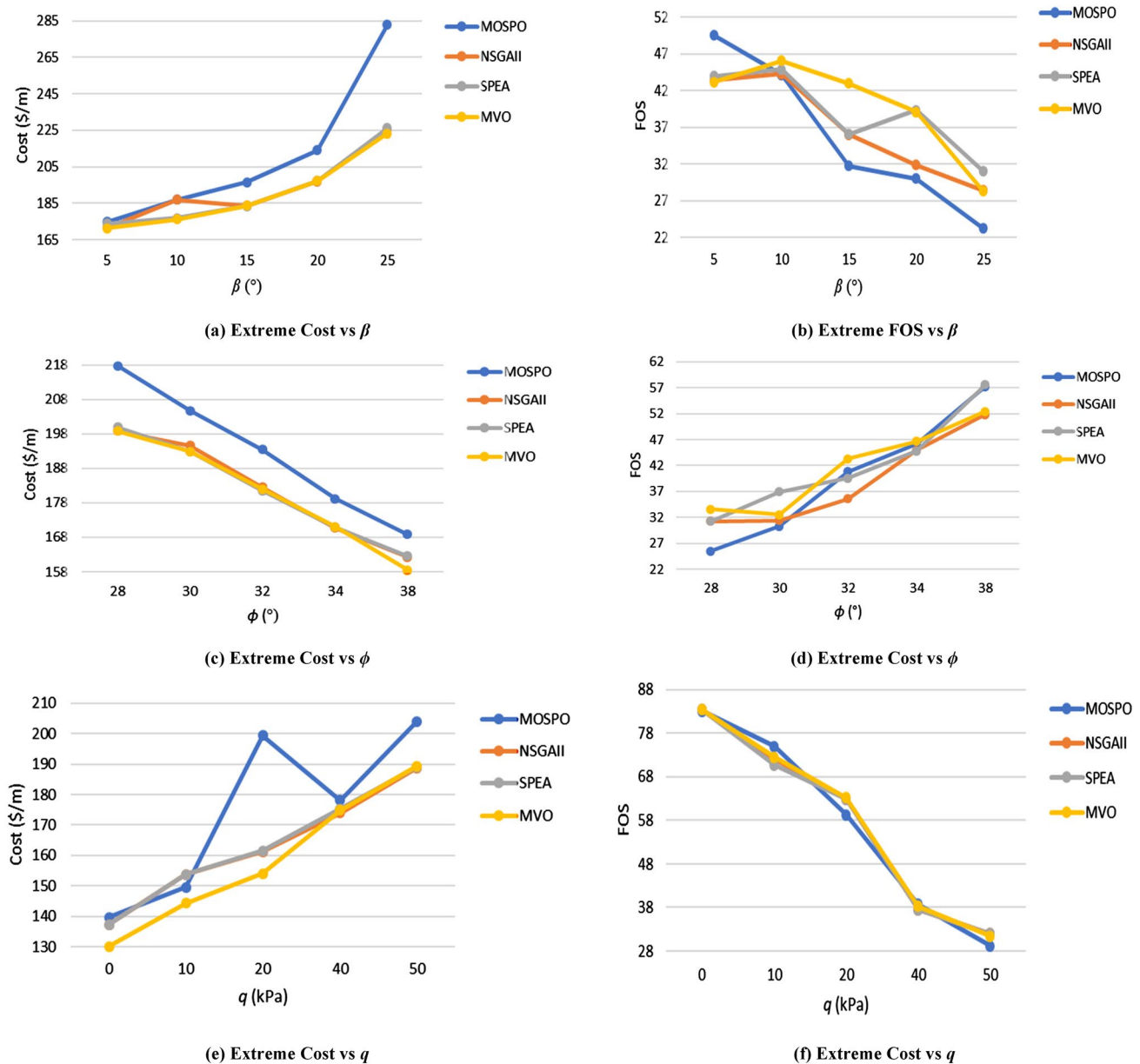
Fig. 13 Comparison of the extreme design points for the cost-based design of Case II

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Replication of results The results presented in this article can be replicated by implementing the data structures and algorithms presented in this article. We will provide the data required for regenerating the figures in GitHub or ResearchGate at the time of publishing the paper and will share the link here.

Research involving human and animal rights This article does not contain any studies with human participants or animals performed by any of the authors.

References

American Concrete Institute (2005) Building code requirements for structural concrete and commentary (ACI 318–05), Detroit

Akhani M, Pezeshk S (2022) Using metaheuristic algorithms to optimize a mixed model-based ground-motion prediction model and associated variance components. J Seismol 26:483

Akhani M, Kashani AR, Mousavi M, Gandomi AH (2019) A hybrid computational intelligence approach to predict spectral acceleration. Measurement 138:578–589

Afshari H, Hare W, Tesfamariam S (2019) Constrained multi-objective optimization algorithms: review and comparison with application in reinforced concrete structures. Appl Soft Comput 83:105631

Ali ASA, Ebrahimi S, Ashiq MM, Alasta MS, Azari B (2022) CNN-Bi LSTM neural network for simulating groundwater level. Environ Eng 8:1–7

Aydogdu I (2017) Cost optimization of reinforced concrete cantilever retaining walls under seismic loading using a biogeography-based optimization algorithm with Levy flights. Eng Optim 49:381–400. https://doi.org/10.1080/0305215X.2016.1191837

Azari B, Hassan K, Pierce J, Ebrahimi S (2022) Evaluation of machine learning methods application in temperature prediction. Environ Eng 8:1–12

Azizi K, Attari J, Moridi A (2017) Estimation of discharge coefficient and optimization of Piano Key Weirs. In: Labyrinth and Piano Key Weirs III: Proceedings of the 3rd International Workshop on Labyrinth and Piano Key Weirs (PKW 2017) (p. 213).

Azevedo CRB, Araújo AFR (2011) Correlation between diversity and hypervolume in evolutionary multiobjective optimization. In: 2011 IEEE Congress of Evolutionary Computation (CEC), pp. 2743–2750

Behmanesh R, Rahimi I, Gandomi AH (2020) Evolutionary many-objective algorithms for combinatorial optimization problems: a comparative study. Arch Comput Methods Eng. https://doi.org/10.1007/s11831-020-09415-3

Bekdaş G, Nigdeli SM, Kayabekir AE, Yang X-S (2019) Optimization in civil engineering and metaheuristic algorithms: a review of state-of-the-art developments. In: Platt GM, Yang X-S, Silva Neto AJ (eds) Computational intelligence, optimization and inverse problems with applications in engineering. Springer, Cham, pp 111–137

Bigham A, Gholizadeh S (2020) Topology optimization of nonlinear single-layer domes by an improved electro-search algorithm and its performance analysis using statistical tests. Struct Multidisc Optim 62(4):1821–1848

Camp CV, Akin A (2012) Design of retaining walls using big bang-big crunch optimization. J Struct Eng 138:438–448. https://doi.org/10.1061/(ASCE)ST.1943-541X.0000461

Ceranic B, Fryer C, Baines RW (2001) An application of simulated annealing to the optimum design of reinforced concrete retaining structures. Comput Struct 79:1569–1581. https://doi.org/10.1016/S0045-7949(01)00037-2

Coello CAC, Lamont GB, Van Veldhuizen DA (2007) Evolutionary algorithms for solving multi-objective problems. Springer, Berlin

Coello CC, Lechuga MS (2002) MOPSO: a proposal for multiple objective particle swarm optimization. In: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600). IEEE, pp. 1051–1056

Das BM (2010) Principles of foundation engineering. Cengage Learning, Boston

Das MR, Purohit S, Das SK (2016) Multi-objective optimization of reinforced cement concrete retaining wall. Indian Geotech J 46:354–368. https://doi.org/10.1007/s40098-015-0178-y

Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, New York

Deb K, Agrawal S, Pratap A, Meyarivan T (2000) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Schoenauer M, Deb K, Rudolph G, Yao X, Lutton E, Merelo JJ, Schwefel HP (eds) Parallel problem solving from nature PPSN VI. Springer, Berlin, pp 849–858

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6:182–197

Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science. IEEE, pp. 39–43

Ebrahimi S, Khorram M (2021) Variability effect of hydrological regime on river quality pattern and its uncertainties: case study of Zarjoob River in Iran. J Hydroinf 23(5):1146–1164

Fathnejat H, Ahmadi-Nedushan B (2020) An efficient two-stage approach for structural damage detection using meta-heuristic algorithms and group method of data handling surrogate model. Front Struct Civ Eng 14:907–929. https://doi.org/10.1007/s11709-020-0628-1

Gandomi AH (2014) Interior search algorithm (ISA): a novel approach for global optimization. ISA Trans 53:1168–1183. https://doi.org/10.1016/j.isatra.2014.03.018

Gandomi AH, Kashani AR (2016) Evolutionary bound constraint handling for particle swarm optimization. In: 2016 4th International Symposium on Computational and Business Intelligence (ISCBI). IEEE, pp. 148–152

Gandomi AH, Kashani AR (2017) Construction cost minimization of shallow foundation using recent swarm intelligence techniques. IEEE Trans Industr Inf 14(3):1099–1106

Gandomi AH, Kashani AR (2018a) Probabilistic evolutionary bound constraint handling for particle swarm optimization. Oper Res 18:801–823

Gandomi AH, Kashani AR (2018b) Automating pseudo-static analysis of concrete cantilever retaining wall using evolutionary algorithms. Measurement 115:104–124

Gandomi AH, Deb K (2020) Implicit constraints handling for efficient search of feasible solutions. Comput Methods Appl Mech Eng 363:112917. https://doi.org/10.1016/j.cma.2020.112917

Gandomi AH, Yang X-S, Talatahari S, Alavi AH (2013) Metaheuristic applications in structures and infrastructures. Newnes, Oxford

Gandomi AH, Kashani AR, Mousavi M (2015a) Boundary constraint handling affection on slope stability analysis. In: LagarosPapadrakakis NDM (ed) Engineering and applied sciences optimization. Springer, Berlin, pp 341–358

Gandomi AH, Kashani AR, Mousavi M, Jalalvandi M (2015b) Slope stability analyzing using recent swarm intelligence techniques. Int J Numer Anal Methods Geomech 39:295–309

Gandomi AH, Kashani AR, Roke DA, Mousavi M (2015c) Optimization of retaining wall design using recent swarm intelligence techniques. Eng Struct 103:72–84

Gandomi AH, Kashani AR, Mousavi M, Jalalvandi M (2017a) Slope stability analysis using evolutionary optimization techniques. Int J Numer Anal Methods Geomech 41:251–264

Gandomi AH, Kashani AR, Roke DA, Mousavi M (2017b) Optimization of retaining wall design using evolutionary algorithms. Struct Multidisc Optim 55:809–825

Gandomi AH, Kashani AR, Zeighami F (2017c) Retaining wall optimization using interior search algorithm with different bound constraint handling. Int J Numer Anal Methods Geomech 41(11):1304–1331

Gandomi M, Kashani AR, Farhadi A, Akhani M, Gandomi AH (2021) Spectral acceleration prediction using genetic programming based approaches. Appl Soft Comput 106:107326

Gholizadeh S, Fattahi F (2021) Multi-objective design optimization of steel moment frames considering seismic collapse safety. Eng Comput 37(2):1315–1328

Gholizadeh S, Danesh M, Gheyratmand C (2020) A new Newton metaheuristic algorithm for discrete performance-based design optimization of steel moment frames. Comput Struct 234:106250

Gunantara N (2018) A review of multi-objective optimization: Methods and its applications. Cogent Eng 5:1502242. https://doi.org/10.1080/23311916.2018.1502242

Jordehi AR (2015) A review on constraint handling strategies in particle swarm optimisation. Neural Comput Appl 26:1265–1275. https://doi.org/10.1007/s00521-014-1808-5

Kashani AR, Gandomi AH, Mousavi M (2016) Imperialistic competitive algorithm: a metaheuristic algorithm for locating the critical slip surface in 2-dimensional soil slopes. Geosci Front 7:83–89

Kashani AR, Gandomi M, Camp CV, Gandomi AH (2019a) Optimum design of shallow foundation using evolutionary algorithms. Soft Comput 24:6809

Kashani AR, Saneirad A, Gandomi AH (2019b) Optimum design of reinforced earth walls using evolutionary optimization algorithms. Neural Comput Appl 24:6809

Kashani AR, Gandomi M, Camp CV, Rostamian M, Gandomi AH (2020) Metaheuristics in civil engineering: a review. Metaheuristic Comput Appl 1(1):19–42 https://doi.org/10.12989/mca.2020.1.1.019

Kashani AR, Akhani M, Camp CV, Gandomi AH (2021a) A neural network to predict spectral acceleration. In: Samui P, Dixon B (eds) Basics of computational geophysics. Elsevier, Amsterdam, pp 335–349

Kashani AR, Camp CV, Rostamian M, Azizi K, Gandomi AH (2021b) Population-based optimization in structural engineering: a review. Artif Intell Rev 55:345

Kashani AR, Chiong R, Dhakal S, Gandomi AH (2021c) Investigating bound handling schemes and parameter settings for the interior search algorithm to solve truss problems. Eng Rep 3(10):e12405

Kashani AR, Chiong R, Mirjalili S, Gandomi AH (2021d) Particle swarm optimization variants for solving geotechnical problems: review and comparative analysis. Arch Comput Methods Eng 28(3):1871–1927

Kashani AR, Camp CV, Akhani M, Ebrahimi S (2022a) Optimum design of combined footings using swarm intelligence-based algorithms. Adv Eng Softw 169:103140

Kashani AR, Camp CV, Azizi K, Rostamian M (2022b) Multi-objective optimization of mechanically stabilized earth retaining wall using evolutionary algorithms. Int J Numer Anal Methods Geomech 46:1433

Kaveh A, Kalateh-Ahani M, Fahimi-Farzam M (2013) Constructability optimal design of reinforced concrete retaining walls using a multi-objective genetic algorithm. Struct Eng Mech 47:227–245. https://doi.org/10.12989/sem.2013.47.2.227

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95—International Conference on Neural Networks. IEEE, Perth, WA, Australia, pp. 1942–1948

Khajehzadeh M, Eslami M (2012) Gravitational search algorithm for optimization of retaining structures. Indian J Sci Technol 5(1):1821–1827

Khajehzadeh M, Taha MR, El-Shafie A, Eslami M (2010) Economic design of retaining wall using particle swarm optimization with passive congregation. Aust J Basic Appl Sci 4(11):5500–5507

Maniat M, Camp CV, Kashani AR (2021) Deep learning-based visual crack detection using Google Street View images. Neural Comput Appl 33(21):14565–14582

Marchionatti R, Gambino E (1997) Pareto and political economy as a science: methodological revolution and analytical advances in economic theory in the 1890s. J Polit Econ 105:1322–1348. https://doi.org/10.1086/516395

Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008

Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-Verse Optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27:495–513. https://doi.org/10.1007/s00521-015-1870-7

Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017a) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191. https://doi.org/10.1016/j.advengsoft.2017.07.002

Mirjalili S, Jangir P, Mirjalili SZ, Saremi S, Trivedi IN (2017b) Optimization of problems with multiple objectives using the multi-verse optimization algorithm. Knowl-Based Syst 134:50–71. https://doi.org/10.1016/j.knosys.2017.07.018

Mishra M, Barman SK, Maity D, Maiti DK (2019) Ant lion optimisation algorithm for structural damage detection using vibration data. J Civ Struct Health Monit 9:117–136. https://doi.org/10.1007/s13349-018-0318-z

Moeini M, Shojaeizadeh A, Geza M (2022) Supervised stacking ensemble machine learning approach for enhancing prediction of total suspended solids concentration in Urban watersheds. J Environ Eng 148(6):04022026

Najafzadeh M, Azamathulla HM (2015) Neuro-fuzzy GMDH to predict the scour pile groups due to waves. J Comput Civ Eng 29(5):04014068

Najafzadeh M, Kargar AR (2019) Gene-expression programming, evolutionary polynomial regression, and model tree to evaluate local scour depth at culvert outlets. J Pipeline Syst Eng Pract 10(3):04019013

Panwar A, Jha KN (2019) A many-objective optimization model for construction scheduling. Constr Manage Econ 37:727–739. https://doi.org/10.1080/01446193.2019.1590615

Rangaiah GP, Feng Z, Hoadley AF (2020) Multi-objective optimization applications in chemical process engineering: tutorial and review. Processes 8:508. https://doi.org/10.3390/pr8050508

Sahib NM, Hussein A (2019) Particle swarm optimization in managing construction problems. Procedia Comput Sci 154:260–266. https://doi.org/10.1016/j.procs.2019.06.039

Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. Adv Eng Softw 105:30–47. https://doi.org/10.1016/j.advengsoft.2017.01.004

Sarıbaş A, Erbatur F (1996) Optimization and sensitivity of retaining structures. J Geotech Eng 122:649–656. https://doi.org/10.1061/(ASCE)0733-9410(1996)122:8(649)

Sha DY, Lin H-H (2010) A multi-objective PSO for job-shop scheduling problems. Expert Syst Appl 37:1065–1070. https://doi.org/10.1016/j.eswa.2009.06.041

Yang X-S, Gandomi AH, Talatahari S, Alavi AH (2012) Metaheuristics in water, geotechnical and transport engineering. Newnes, Oxford

Zitzler E (1999) Evolutionary algorithms for multiobjective optimization: methods and applications, vol 63. Shaker, Ithaca

Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans Evol Comput 3:257–271. https://doi.org/10.1109/4235.797969

Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. Evol Comput 8:173–195. https://doi.org/10.1162/106365600568202

Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength pareto evolutionary algorithm. Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK), Zürich