Applied Soft Computing Journal 83 (2019) 105631

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

Constrained multi-objective optimization algorithms: Review and comparison with application in reinforced concrete structures

Hamid Afshari *, Warren Hare, Solomon Tesfamariam

University of British Columbia, Kelowna, Canada

HIGHLIGHTS

- Multi-objective optimization (MOO) approaches in general and in Reinforced Concrete (RC) structures are reviewed.
- A novel multi-objective model is developed for the optimal design of RC beams.
- A set of algorithms are examined and compared through numerical testing.
- This paper enlightens the merits of advanced MOO methods in structural engineering problems.

ARTICLE INFO

Article history:
Received 16 April 2019
Received in revised form 9 July 2019
Accepted 12 July 2019
Available online 23 July 2019

Keywords:
Multi-objective optimization
Reinforced concrete structures
Reinforced concrete beam
Constrained optimization
Deflection

ABSTRACT

Engineering design problems are often multi-objective in nature, which means trade-offs are required between conflicting objectives. In this study, we examine the multi-objective algorithms for the optimal design of reinforced concrete structures. We begin with a review of multi-objective optimization approaches in general and then present a more focused review on multi-objective optimization of reinforced concrete structures. We note that the existing literature uses metaheuristic algorithms as the most common approaches to solve the multi-objective optimization problems. Other efficient approaches, such as derivative-free optimization and gradient-based methods, are often ignored in structural engineering discipline. This paper presents a multi-objective model for the optimal design of reinforced concrete beams where the optimal solution is interested in trade-off between cost and deflection. We then examine the efficiency of six established multi-objective optimization algorithms, including one method based on purely random point selection, on the design problem. Ranking and consistency of the result reveals a derivative-free optimization algorithm as the most efficient one.

1. Introduction

Design of structures, considering optimal performance of safety and serviceability objectives, is conflicting with cost minimization objective [1,2]. Such optimization can be cast as a multi-objective optimization (MOO) problem (also called multi-criteria optimization or vector optimization). In MOO, several objectives, usually conflicting, have to be optimized simultaneously over a feasible set determined by constraint functions [3]. There is, no unique solution that is simultaneously optimal for all objectives and one can only consider a trade-off among the objectives [4,5]. Application of multi-objective modeling has been used to optimize different engineering structures, e.g. composite structures [6], tall buildings [7], adjacent buildings [8], and structural dampers [9]. The reported applications utilized different methods, with vast majority using some form of evolutionary algorithm (e.g., [10, 11]). As a result, a key question remains to be answered: what is the efficiency of provided solutions affected by the selected approach? In this paper we address this question by examining multi-objective algorithms with respect to the specific test problem of RC beams.

For a decision-maker, who plans to optimize m objectives with no clear preference of objectives, the MOO problem is formulated by a function mapping decision vector to a vector of objective values. The MOO problem can be formulated as:

$$\begin{aligned} &\min F(x) \\ &\text{subject to } x \in S \end{aligned} \tag{1}$$

where $\mathrm{F}: \mathbb{R}^n \to \mathbb{R}^m$, $\mathrm{F}(x) = (f_1(x), f_2(x), \cdots, f_m(x))$ for $m \geq 2$, $S \subseteq \mathbb{R}^n$, is a feasible decision space defined by linear, nonlinear, or box constraints, and $x \in S$ is the decision vector. We denote the image of S by $Z = F(S)$ and we call it a feasible objective space. The mapping between the decision space and objective space, for $m = 2$ and $n = 3$, is illustrated in Fig. 1.

In a single objective minimization problem, a solution $x_1 \in S$ is better than another solution $x_2 \in S$ iff $f(x_1) < f(x_2)$. Unlike the

* Corresponding author.
E-mail addresses: afsharih@myumanitoba.ca, hamid.afshari@ubc.ca (H. Afshari).

https://doi.org/10.1016/j.asoc.2019.105631
1568-4946/

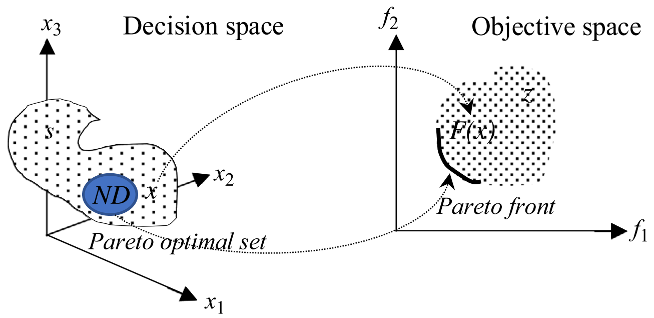

Fig. 1. The objective and decision space and mapping of *Pareto optimal set* (a.k.a. the Nondominated set, *ND*) to *Pareto front*.

single objective optimization, in the multi-objective comparing two solutions x_1 and x_2 is not as straightforward. In this context, the interest is often in the objective space. For this reason, the notion of optimality depends on how decision alternatives are compared and ordered, that is, on the order relation in the objective space. The order relations in objective space can be defined using the concept of *dominance*, which the *Pareto dominance* is most commonly used [12]. For example, x_1 is said to *dominant* x_2 iff $f_j(x_1) \leq f_j(x_2)$ for all $j = 1, 2, \ldots, m$, with at least one inequality being strict. If neither x_1 dominates x_2 nor x_2 dominates x_1, then the points are said to be *indifferent*.

Through generating solution for multi-objective problems, one would agree that a good solution must not be *dominated* by the other feasible alternatives [13]. The set of feasible points that are not dominated by any other feasible points is called the *Pareto optimal set*. The set of function vectors generated by the Pareto optimal set is called the *Pareto front*.

The Pareto optimal set is almost never a single point, rather a set of solutions [3]. Thus, the Pareto optimal set represents the solution to MOO problem [14]. Fig. 1 illustrates the concepts of Pareto optimal set and Pareto front.

Achieving the exact Pareto front of an arbitrary problem is usually quite difficult. Nevertheless, reasonably good approximations of the Pareto front are generally acceptable within limited computational time [15,16].

The structure of S and F influence the types of MOO problems. When S is a polyhedral set and the objective functions are all linear, then the problem in Eq. (1) is called Multi-objective Linear Optimization problem (MOLO) [17]. A variety of solution techniques have been developed due to the special characteristics of MOLO problems [17–19]. If any of the objective or constraint functions is nonlinear, then it is called Nonlinear Multi-objective optimization problem (NMOO). If all the objective functions and the feasible region are convex, the MOO is called convex. The MOO problem is non-differentiable if some of the objective functions or the constraint functions forming the feasible region are non-differentiable [20]. When the objective functions of MOO are subject to noise or computed as the result of a computer simulation, then the problem is categorized as a black-box MOO problem [12,21]. As engineering design problems often consist of a mixture of numerical simulations and analytical calculations, most engineering design problems are of this type. Non-gradient optimization methods are typically used for such problems, e.g. [22].

This research aims to review and compare novel MOO algorithms developed to solve practical problems in industry. It is noticed that there is a lack of comparative studies to evaluate the efficiency of such algorithms, in particular, for the application in the structural engineering. Thus, the addressed problem in this research is the efficiency measurement of a set of the common and recent MOO algorithms (multi-objective gradient-based algorithm, multi-objective derivative-free optimization algorithm, multi-objective genetic algorithm, Non-dominated sorting genetic algorithm-III, multi-objective particle swarm optimization, and a random method). To evaluate the efficiency of selected algorithms, a multi-objective model for the optimal design of RC beams is developed, and 25 variations of test problem are generated. The contributions in this paper include: (i) reviewing state-of-the-art algorithms developed for MOO problems and select a set of algorithms based on popularity, efficiency and recentness to be applied for advanced constrained MOO problems, (ii) incorporating deflection limit states as the second objective function for the RC beam design optimization to bridge the gap of the literature, which mostly focus on cost optimization of ultimate limit states, and (iii) developing a framework to compare algorithms for proposed structural engineering problem, and measure the efficiency of those methods. The comparative study shows that derivative-free optimization algorithm (BiMADS) outperforms other algorithms. In addition, the developed mathematical model for the optimal design of RC beams could reduce the deflection up to 3 time (3000 mm to 1000 mm) by adding 30% to the cost. Such results are valuable for scholars as well as practitioners for the optimal design of structures.

In the following section, the algorithms for solving MOO problems in general are reviewed. This section elaborates different approaches developed for MOO problems using proposed categories. Section 3 reviews state-of-the-art in application of MOO algorithms in RC structures. Section 4 presents the developed MOO problem for the design of RC beams by minimizing cost and deflection. In Section 5, a comparative study of five MOO algorithms and one random algorithm is presented when applied to solve proposed model in Section 4. Results of such comparison are analyzed to determine the efficient algorithms. The paper concludes with research findings and future work.

2. Multi-objective optimization algorithms

In solving MOO problems, selection of a method suitable for the problem at hand is a fundamental first step [23]. There are three general approaches to MOO problems. One is to combine the objective functions into a single composite function using weighted sum method (scalarizing). The optimal solution to the single-objective optimization problem is the Pareto optimal solution to the MOO problem. By varying the weight, the Pareto front can be generated as well. However, it should be emphasized that not all Pareto points can be generated as the solution to a scalarized problem. The second approach is to move all but one objective to the constraint set. These approaches are repeated for different combinations in order to find a variety of Pareto optimal points. The third approach is to attempt to determine the entire Pareto optimal solution set or a representative subset at one time [24]. This approach is exemplified by heuristics and evolutionary strategies, such as the genetic algorithms discussed in this paper.

Heuristic algorithms attempt to find a good solution to an optimization problem using some form of trial-and-error. While there is no guarantee of success, heuristic algorithms often find a better or improved solution than an educated guess [25,26]. Heuristic algorithms belong to a broad family of non-deterministic optimization methods aimed at finding accurate solutions to complex optimization problems when exact methods are not applicable [27]. They are considered as higher-level techniques or strategies that intend to combine lower-level techniques and tactics. The lower-

Algorithm 1 Multi-objective Tabu Search (MOTS)

Initialize
  Select the tabu list $(TL_i) = \emptyset$, and $ND = \emptyset$
  Add a random feasible solution to ND
Begin
  While (the stopping criteria is NOT satisfied) **do**
    For each objective in the MOO problem
      Search the neighborhood of all possible defined moves
      Choose the non-tabu candidate solution with the best objective function value as the best candidate
    End for
    Set the weights of objective functions
    Find the solution to minimize weighted sum of objective functions by neighborhood search
    Compare each feasible candidate solution with current ND
    Update the TL_i
    If TL_i is full
      Remove the oldest tabu-list entry
    End if
  End while
  Return ND and P_t
End

level techniques (e.g., local search) explore and draw on ideas from other disciplines (e.g., physics) to help with solving the modeled system [25,28]. The increasing acceptance of these algorithms is due to their ability to find multiple solutions in a single run, work without derivatives, converge to Pareto optimal solutions, and handle both continuous function and combinatorial optimization problems [29].

A variety of algorithms are applied for the optimization of multi-objective problems. Such algorithms can be briefly categorized into *neighborhood search*, *evolutionary*, *physical*, *swarm*, and *deterministic algorithms*. Each category is discussed in the subsequent sections.

2.1. Neighborhood search algorithms

The algorithms described in this category are mostly global optimization algorithms and metaheuristics that involve an embedded neighborhood (local) exploring procedure [30]. The algorithms in this class, with the exception of Stochastic Hill Climbing and Random Search algorithms, use a multi-start search mechanism to generate "better" and "varied starting points". Such starting points are the inputs of a neighborhood searching techniques for exploring either potential improvement or unvisited areas. To name a few algorithms in this class, we refer to the Variable Neighborhood Search [31], Greedy Randomized Adaptive Search Procedure (GRASP) [32], Scatter Search [33], and Tabu Search [34]. Two algorithms including Multi-objective Tabu Search and Multi-objective GRASP are described in detail as follows.

2.1.1. Multi-objective Tabu Search (MOTS)

The Tabu search (TS) algorithm was designed to help with searching difficult regions of search space, such as escape from local minima, using the move operator and the tabu list [35,36]. The move operator generates new (candidate) solutions through slightly perturbing a current solution. The set of all candidates is called the neighborhood of the current solution. The tabu list stores some aspect(s) of recent moves to forbid search in those directions for a certain number of iterations. Using short-term memory, the last visited points are recorded to avoid being revisited as tabu. TS has two other features including intensification and diversification. Intensification refers to the medium-term memory to store the best solutions. Diversification is linked to the long-term memory to record the areas of the search space that have been searched reasonably thoroughly [37]. Since the application of TS for MOO problems by Hertz et al. [38], several extensions have been applied to the algorithm [37,39–42]. Hansen [39] developed a MOTS using a weighted average of the objectives. A pseudo code of the MOTS is presented in **Algorithm 1** [34].

2.1.2. Multi-objective Greedy Randomized Adaptive Search Procedure (MOG)

The greedy randomize adaptive search procedure (GRASP) algorithm consists of two phases [32]. First, a randomized greedy algorithm is applied to construct a solution. Second, the local search procedure is applied to improve the constructed solution. At each step, candidate elements are evaluated and ranked using a greedy evaluation function to create a restricted candidate list. One element is then randomly added to the solution from the restricted candidate list. There are several multi-objective algorithms developed using GRASP. The difference between the algorithms is in their efficiency and the method they handled multiple objectives. For example, **Algorithm 2** is proposed by considering a random weighting method for the objective functions [43].

In **Algorithm 2**, the local search is conducted by a randomly selection of a solution from *ND*. By searching the neighborhood of the selected solution, the *ND* is updated and dominated solutions are removed from the list. The algorithm returns *ND*as the solution. For further applications and local search methods, we refer to the Marti et al. [44] and Arroyo and Souza Pereira [45].

Algorithm 2 Multi-objective GRASP (MOG)

Initialize
    $P_t = \emptyset$
Begin
    For $t = 1{:}G$ (number of times a PF is built and optimized)
        For $j = 1{:}N$ (number of solutions constructed as initial PF)
            Generate a random weight vectors of objectives (λ)
            Build solution using λ (weights of objective functions) and P_t
        End for
        Return new PF
        For $i = 1{:}I$ (iterations of local search to improve new PF)
            Local search using new PF
        End for
        Return best neighborhood of new PF
        $P_t = P_t \cup$ new PF
        Remove dominated solutions
    End for
    Return ND and P_t
End

2.2. Evolutionary algorithms

Evolutionary algorithms (EAs) are a class of non-gradient population-based algorithms that widely used in the literature. These algorithms are often nature-inspired using four major steps including reproduction, mutation, recombination and selection. The use of a population-based framework allows EAs to generate several elements of Pareto optimal set in a single run. This is a major motivation to use these algorithms for solving multi-objective optimization problems [46]. In addition, the multi-objective evolutionary algorithms (MOEAs) are less susceptible to the shape and continuity of the Pareto front and require less specific domain information to operate [15]. There are several MOEAs in the literature such as MOO using Genetic Algorithm (MOGA) [24], Non-dominated Sorting Genetic Algorithm-III (NSGA-III) [47], Strength Pareto Evolutionary Algorithm 2 (SPEA2) [48], Pareto Archived Evolution Strategy (PAES) [49], Pareto Envelope-based Selection Algorithm (PESA) [50], Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) [51], multi-objective vortex search [52], and multi-objective artificial algae algorithm [53]. Two popular algorithms in this class are introduced in this section.

2.2.1. Multi-objective Genetic Algorithm (MOGA)

The genetic Algorithm (GA) is a common heuristics algorithm to solve engineering problems [20]. As the first developed algorithm that uses Pareto-based ranking and niching techniques together, MOGA (**Algorithm 3**) encourages the search toward the Pareto front while maintaining diversity in the population and fitness sharing [54].

A major difference of MOGA with single-objective GA is the fitness evaluation process. In single-objective GA, the fitness evaluation step is accounted for measuring objective function value, however, in MOGA the evaluation is more complicated. First, a rank is assigned to each solution $x \in P_t$. Then, a fitness value is assigned to each solution based on the ranking of the solution. The rest of the steps include calculating niche count and shared fitness value for each solution, and normalizing fitness values based on shared fitness values [24].

Algorithm 3 Multi-objective Genetic Algorithm (MOGA)

Initialize
    Select a random initial population $P_t \subset S$
Begin
    While (stopping criteria are NOT satisfied) **do**
        Evaluate fitness of the population
        Select parents using a stochastic selection method
        Apply crossover and mutation,
        Update P_t, ND and Set $t = t + 1$
    End while
    Return ND and P_t
End

MOGA has been widely adopted in the research to optimize the design of structures. For example, Liu et al. [55] developed a multi-objective model to optimize initial and life cycle costs (LCCs) of seismic steel-resisting frame structures. Using a multi-objective GA, a set of alternative designs is produced as the result of tradeoff between objectives. Finally, designers decide a solution based on the degree of design complexity (in terms of the number of different standard steel section types) as well as initial and LCCs. Barone and Frangopol [56] proposed a mathematical model to measure the life-cycle maintenance performance of deteriorating buildings. The model compared a set of performance indicators when applied to assess the cost-efficiency of the associated optimal solution. The objectives include the cost of repair and the system reliability. The model was not compared with other solution approaches. Mitseas et al. [57] developed a multi-objective design optimization framework considering the initial cost and LCC of the structural systems subject to evolutionary stochastic excitations. A building structure consisting of the versatile Bouc–Wen (hysteretic) model was used as a numerical example to solve the model. Greco and Marano [58] considered two sub-structures, the wall and the frame, modeled as single

Algorithm 4 Non-dominated Sorting Genetic Algorithm-III (NSGA-III)

Initialize

Select a random initial population $P_t \subset S$,
Set N_p as the size of population, and $t = 0$ (t_{max} is the maximum iteration)

Begin

While ($t < t_{max}$) **do**
Generate offspring Q_t
Mutate on Q_t
Set $R_t = P_t \cup Q_t$
Apply non-dominated sorting on R_t to find different levels of non-dominance
Generate the next generation P_{t+1}
Normalize objective functions
Generate reference points
Associate each member of next generation to a reference point
Choose $N_p - |P_{t+1}|$ members from non-dominance levels by niche-preserving operator
$t = t + 1$
End while
Return ND and P_t

End

degree of freedom systems under ground motion excitation. Using a passive strategy, the aim was to protect both sub-structures by developing a multi-objective design optimization approach. In this regard, displacement of the frame and the shear in the wall are considered as objectives of the model. None of the above papers compared the algorithms with other solution approaches.

Barraza et al. [59] developed MOO model of two- and three-dimensional moment resisting steel structures subjected to earthquake loads. The model minimizes the structural weight and the maximum inter-story drift. Two evolutionary algorithms including GA and particle swarm optimization (PSO) were used and the efficiency of the algorithms was compared. It is concluded that the structural buildings obtained by PSO are in general better solution compared to the GA results. Bhattacharjya and Chakraborty [60] proposed an improved robust MOO of structure with random parameters. The aim was to improve the robustness of the performance by defining a new index. The model is applied to optimize a vibrating platform for maximum frequency and minimum cost. The result is compared with deterministic multi-objective model to measure the quality of the Pareto front approximation produced by these methods.

2.2.2. Non-dominated Sorting Genetic Algorithm-III (NSGA-III)

Srinivas and Deb [61] developed the NSGA using the Goldberg's notion of a non-dominated sorting procedure [62]. NSGA uses the sorting procedure as a ranking selection method, and a fitness sharing niching method to maintain stable sub-populations across the Pareto front [30]. Deb at al. [63] extended the NSGA (called NSGA-II) to improve its criticism in terms of computational complexity, the lack of elitism, and the need to specify a sharing parameter. This algorithm is widely applied in many optimization problems in engineering. Recently, a reference-point based multi-objective NSGA-II algorithm (called NSGA-III, **Algorithm 4**) was developed to solve the problems with multiple objectives in an efficient way [47,64]. The application of NSGA-III algorithm is growing in different contexts in the literature for its properties as discussed above [65].

As discussed before, the NSGA-II algorithm and its improved versions has been widely applied in different structural engineering problems, especially in RC structures [22,66,67]. In Section 3 applications of NSGA-II algorithm in RC optimization problems are presented. However, based on best of our search, NSGA-III has not been used in structural engineering optimization problems.

2.3. Physical algorithms

Physical algorithms are inspired by a physical process ranging from metallurgy, music, and complex dynamic systems such as avalanches [30]. They are generally stochastic optimization algorithms with the mixtures of local (neighborhood-based) and global search techniques. Some examples of such algorithms include Harmony Search [68], Memetic Algorithm [69], and Simulated Annealing [70] for MOO problems. Two commonly used algorithms in the literature are presented in this section.

2.3.1. Multi-objective Simulated Annealing (MOSA)

As a probabilistic heuristic algorithm, Simulated Annealing (SA) mimics the annealing process as discussed in material science [71]. This algorithm has been adapted in a multi-objective framework due to its simplicity and capability of generating Pareto front in single run at minimum time [72]. Since the first multi-objective SA proposed by Serafini [70], several improved algorithms have been developed in the literature such as the method of Suppapitnarm and Parks (SMOSA) [73], the method of Ulungu and Teghem (UMOSA) [74], Pareto simulated annealing (PSA) [75], multi-objective simulated annealing using constraints violation in acceptance criterion (WMOSA) [76], and pareto-domination-based acceptance criterion (PDMOSA) [77]. Recently, Ekbal and Saha [78] and Singh et al. [79], and Antunes et al. [80] proposed improvements to the classic MOSA algorithm. **Algorithm 5** is proposed by Antunes et al. [80].

Algorithm 5 Multi-objective Simulated Annealing (MOSA)

Initialize
Create the set of initial random solutions $P_t \subset S$,
Set t_{max}, and boundary of temperature $[T_{min}, T_{max}]$,
Begin
For t =1: t_{max}
$T = T_{max}$
While $(T > T_{min})$ **do**
For i =1: size (ND)
Pick a solution s_i from ND
Search a solution s_i' in the neighbourhood of s_i
If s_i' dominates s_i,
Add s_i' to ND
Else if (rand $\in [0,1] < APF(s_i, s_i', T)$)
$s_i' \rightarrow s_i$
Else
Ignore s_i'
End for
Reduce T
End while
Update the set of ND
End for
Return ND and P_t
End

In this algorithm, δ_j represents the difference of performance between the competing solutions (s and s') in the objective function $j = 1, \ldots, p$ ($\delta_j = f_j(s') - f_j(s)$). Using a weighted sum (w_j), the aggregation of differences is calculated ($\Delta = \sum_{j=1}^{p} w_j \delta_j$). Such Δ is used to measure the acceptance probability function (APF) according to distinct rules (e.g., scalar linear where P=min$\left(1, e^{-\Delta/T_t}\right)$, T_t =temperature at iteration t).

2.3.2. Multi-objective Harmony Search (MOHS)

The Harmony Search algorithm was first introduced by Geem et al. [81]. The algorithm imitates the evolution of a harmony relationship between sound waves in deferent frequencies when played at the same time. The harmony search algorithm uses a stochastic random search, instead of using a gradient search. This algorithm has a simple concept, few parameters such as harmony memory (HM), harmony memory consideration rate (HMCR), and pitch adjustment rate (PAR) to search solutions, and could be easily implemented [81,82]. For multi-objective version of the harmony search algorithm (MOHS) there are several proposed algorithms. In this section, a version with minimum changes compared to the single objective HS algorithm is presented [83]. In **Algorithm 6**, BW refers to the bandwidth for searching and improvising the solution space.

Algorithm 6 Multi-objective Harmony Search (MOHS)

Initialize
Set parameters (HMS, HMCR, PAR, BW)
Randomly select a sample from HM
Begin
While (stopping criteria are NOT satisfied) **do**
Improvise a new solution $x^{im} \in S$
Calculate the Pareto ranking of x^{im} considering HM
If x^{im} has a better ranking than the worst solution in HM
Update HM with x^{im}
End if
End while
Return ND and P_t
End

The difference between algorithms could be addressed in their approach for searching the solution space, ranking the solutions, and dealing with the multi-objective problem. For example, weighting mechanism [84] and non-dominated ranking (as used in NSGA-II) [85] is used to deal with multiple objectives. For further studies, we refer to recent extensions of MOHS including Ayala et al. [86], Dai et al. [87], and Kougias and Theodossiou [88].

2.4. Swarm algorithms

Swarm algorithms mimic the processes of decentralized, self-organized systems, which can be either natural or artificial in nature [89]. Swarm intelligence is applied into many areas concerned with the development of multi-agent systems such as optimization [90]. The concept of swarm intelligence is based on many simple identities that work with each other to exhibit a desired behavior. For example, though the individual behavior of members of society (e.g., ants, fishes, bees) is not sophisticated, their mutual cooperation and coordinated behavior helps them to achieve complex tasks. Due to the efficacy of the swarm algorithms, there are many applications of these algorithms in the structural engineering [91]. Recently, new algorithms such as Krill Herd algorithm has been developed, which is capable of solving a wide range of optimization problems [92]. In addition, new algorithms such as Whale optimization algorithm [93], Polar bear algorithm [94], and Dragonfly algorithm [95] have been developed recently to optimize multi-objective problems. These algorithms could provide efficient results in comparative studies and are aimed for future work by the authors.

2.4.1. Multi-objective Particle Swarm Optimization (MOPSO)

Particle swarm optimization (PSO) resembles GAs in some features such as using stochastic process, working with a set of potential solutions, and fitness functions. Candidate solutions (particles) move iteratively through the search space and improve the fitness value based on a given quality measure. Because each particle is influenced by its neighbor (leader), the position and velocity of the particle is updated in each iteration to reach a better solution [89].

Single-objective PSO has been widely used in the literature as an efficient metaheuristic algorithm. By solving single-objective PSO, the leader that each particle uses to update its position is completely determined once a neighborhood topology is established. Using MOPSO, however, each particle should select just one out of a set of different leaders to update its position [96]. A repository, usually stored in a different location (called external archive), keeps the non-dominated solutions found so far. The solutions contained in the external archive are used as the set of leaders when the positions of the particles are updated, and the content of external archive is usually reported as the final output of the **Algorithm 7**.

Parpinelli et al. [97] compared the swarm intelligence algorithms for structural engineering optimization. The algorithms include the bacterial foraging optimization (BFO), particle swarm optimization (PSO), and artificial bee colony (ABC), which are all applied for optimization of three optimization problems in structural engineering. The result shows that PSO presented the best balance between the quality of solutions and the number of function evaluations. Such research needs to be extended for a comprehensive study. We select MOPSO for the comparative study in our research.

2.4.2. Multi-objective Ant Colony Optimization (MOACO)

The algorithm is inspired by the mechanism of an ant colony searching for food. As a stochastic optimization method, the algorithm uses mathematical principles from graph theory. Thus, a mathematical graph consisting of dots (nodes or vertices) and lines (edges) that connect nodes represents the graph. The optimization problem in ACO is to find the shortest path (a set of edges) [98], and summarized in **Algorithm 8**.

Algorithm 7 Multi-objective Particle Swarm Optimization (MOPSO)

Initialize
  Initialize swarm
  Initialize leaders in an external archive
  Calculate the quality measure of leaders
  $t = 0$
Begin
  While (stopping criteria are NOT satisfied) **do**
    For each particle
      Select leaders
      Update position (flight)
      Mutation
      Evaluation
      Update the best solution
    End for
    Update leaders in the external archive
    Calculate the quality measure of leaders
    $t = t + 1$
  End while
  Return ND and P_t from external archive
End

Algorithm 8 Multi-objective Ant Colony Optimization (MOACO)

Initialize
  Initialize Pheromone trails $\tau(\tau^k)$
  Initialize heuristic matrix $\eta(\eta^k)$
  Initialize Pareto set $P_t = \emptyset$
  Determine the weights for each objective (e.g., randomly)
Begin
  While (the stopping criteria are NOT met) **do**
    For each path
      Evaluate the probability to take
      Update the best solution and remove dominate ones
    End for
    For each objective
      Evaluate the weighted sum of objectives
      Determine the best solution
      Update the Pheromone trails
    End for
  End while
  Return ND and P_t
End

In this algorithm, $\tau_{ij}^{k}(t)$ refers to the kth pheromone matrix entry associated with the ith design variable and the jth alternative (or from node i to j) at iteration t. The ants move from their current randomly selected location to the other places using a probability function, which is a function of pheromone trail intensities and the objective functions. A popular edge increases the pheromone trail intensities, and such intensities increase the chance of an edge for passing by the ants. When all solutions (paths) are generated, the Pareto set is updated. As the next step, the pheromone matrices are updated using the solutions stored in Pareto set. Finally, the algorithm returns the Pareto set, when the stopping criteria are met.

There are several versions of multi-objective ant colony optimization algorithm. Such revisions of MOACO are based on considering pheromone and heuristics matrix (single or multiple matrices for τ, η, and colony), solution construction (targeted, dynamic and fixed), evaluation (Pareto and non-Pareto), update and decay (individual and global update of pheromone matrices), and Pareto archival methods (offline storage, online storage, elite, and no storage). For further studies, please refer to Angelo et al. [99], Benhala et al. [100], Saken et al. [101], Angus and Woodward [102].

2.5. Deterministic algorithms

An alternative to heuristic methods that use random processes to generate a solution is deterministic methods. In this class of algorithms, the same solution is obtained by multiple run of the algorithm. Derivative-free optimization (DFO) methods also strive to minimize without the use of higher-order information (gradients). DFO methods use mathematical convergence analysis to ensure algorithms converge to an optimal point (under appropriate conditions [89]). The DFO is a challenging area for optimizing functions, because from one hand, it has a growing demand for many practical optimization problems in industries, and on the other hand, valuable gradient information is missing compared to gradient-based optimization methods. Such problems are common in many areas of real-life problems including engineering design, management, finance, and others.

When higher-order information is available, gradient-based methods can be employed. Like DFO, gradient-based methods use mathematical convergence analysis to ensure algorithms converge to an optimal point under appropriate conditions [103]. For single objective optimization, it is widely accepted that gradient-based methods will outperform DFO methods in most situations [103]. However, this attribute has not been explored for MOO.

2.5.1. Bi-objective Mesh Adaptive Direct Search (BiMADS)

The core of BiMADS is the Mesh Adaptive Direct Search (MADS) method. The method iterates on a set of meshes with varying sizes. A mesh is defied to discretize the space of variables. By the term Adaptive, it means that the algorithm performs an adaptive search on meshes through controlling the refinement of the meshes [104]. Each iterate of MADS generates a trial point on the mesh that improves the current best solution. If an iterate fails, the next iteration is initiated on a better mesh. Each iteration includes a search step and a poll step. The search step is very flexible and returns a point that improves current best solution. The poll step rigidly generates trial mesh points in the vicinity of the best current solution. The MADS in a high-level is presented in **Algorithm 9**.

Algorithm 9 Mesh Adaptive Direct Search (MADS)

```
Initialize
    Select the initial point (x_0)
Begin
  While (stopping criteria are NOT satisfied) do
        Search on the mesh to find a better solution than x_t
        Poll on the mesh if Search failed
          If a better solution is found (either Search or Poll)
            Update the best solution
            The mesh size is coarsened (optional)
          Else
             Refine the mesh
          End if
    End while
End
```

BiMADS algorithm is used in Nonlinear Optimization by Mesh Adaptive Direct Search (NOMAD) application package [105,106]. BiMADS approximates the Pareto front of a bi-objective problem by solving a series of single-objective optimization problem. Each single objective problem, relies on a reference point (r) in the objective space to obtain a uniform coverage of the Pareto front.

The BiMADS algorithm solves two single objective problems independently using MADS algorithm form a starting point $x_0 \in S$. The initial set of non-dominated solutions (ND) and the Pareto front (P_t) are built using the combination of the points found during the two MADS iterations. Both ND and P_t are sorted in ascending order of f_1 value to compare the gaps between non-dominated solutions using Euclidian distance. The aim is to ensure a uniform coverage of the Pareto front. A BiMADS iteration includes three major steps: (i) define a reference point (r), (ii) solving the scalarized MOP using the MADS algorithms, and (iii) updating the set of non-dominated solutions and removing dominated solutions. After finding a reference point (r), ϕ_r: $\mathbb{R}^m \to \mathbb{R}$ is parameterized with respect to some reference point $r \epsilon \mathbb{R}^m$. A stopping condition is reached when maximum number of objective function evaluation has been conducted. The pseudo code of the BiMADS is presented in **Algorithm 10** [107].

Based on the reviewed literature, there is no research conducted on the optimal design of structures that applied NOMAD as a multi-objective optimization solver. However, NOMAD has been applied as a solver for the single-objective optimal design of building energy systems. In particular, Eisenhower et al. [108] compared different algorithms for sampling the parameters space of the building around its baseline. The aim was to optimize building energy costs using a cost function that minimizes energy consumption while maintaining or improving comfort. The presented meta-model was solved using a gradient-based model (called IPOPT) and NOMAD. In almost all cases, both algorithms performed similarly.

2.5.2. Multi-objective gradient-based

The available literature on gradient-based MOO is limited, and currently appears primarily theoretical (no widely distributed gradient-based MOO solver exists). In 2000, Fliege and Svaiter published a gradient-based MOO method based on steepest descent [109]. In 2009, Fliege, Drummond, and Svaiter advanced the gradient-based MOO by adapting Newton's method [110]. More recently, Bento et al. [111] presented a (sub)gradient-based algorithm for MOO that works for smooth or nonsmooth optimization functions. We examine Bento, et al.'s method in this paper. The Weighting Subgradient Algorithm (WSA) finds a single point on the Pareto optimal set. The pseudo-code, **Algorithm 11**, simplifies the method by fixing some parameters to their default values (see [111] for full method and further details).

The term t_{max} refers to the max number of iterations to run the algorithm. In order to approach the entire Pareto optimal set, random restarts are used. In the following algorithm, is *budget* defined as the total number of allowed function calls.

Several studies exist in the literature that utilized gradient-based methods to solve a single objective optimization problem; however, there is no research addressing the application of multi-objective gradient-based method as the principal approach to solve structural engineering optimization problem. Thus, this approach remains as the gap of existing methods to optimize problems.

3. Multi-objective optimization applied to RC structures

Since the mid 19th, there have been several studies focused on the optimal design of RC beams in different contexts (e.g., Norman [112]; Sandhu, [113]). This section reviews most recent papers and highlights some research gaps as presented in Table 1. There are several studies addressing the optimization of reinforced structures including either single or multi-objective models. The existing research for optimization of reinforced structures is summarized as follows:

- Minimization of costs is considered as the major objective in the design of reinforced structures. In most cases, technical issues are treated as constrains. However, such technical features (i.e., deflection) deserve to be optimized as an objective function rather than being monitored as a constraint. Thus, a technical constraint such as deflection could contribute in optimization instead of being monitored for violation.

Algorithm 10 Bi-objective Mesh Adaptive Direct Search (BiMADS)

Initialize
  Select the starting point x_0
  Set $ND = \emptyset, P_t = \emptyset$
Begin
  Apply MADS using (f_1, S, x_0) as inputs, and set the output as ND_{f_1}
  Apply MADS using (f_1, S, ND_{f_1}) as inputs, and set the output as ND_{f_2}
  Update the Pareto set (ND, ND_{f_1}, ND_{f_2})
  Update the Pareto front $(P_t, ND_{f_1}, ND_{f_2})$
  While (stopping criteria are NOT satisfied) **do**
    Find the reference point (r) for P_t
    Apply MADS using (ϕ_r, S, ND) as inputs, and set the output as ND_{ϕ_r}
    Update the Pareto set (ND, ND_{ϕ_r})
    Update the Pareto front $(P_t, ND_{\phi_r}, S, ND)$
  End while
  Return ND and P_t
End

Algorithm 11 Weighting Subgradient Algorithm (WSA)

Begin
  Initialize
    Select the starting point x_0
    Select stopping parameter t_{max}
    Set $ND = \emptyset, P_t = \emptyset, t = 0$
  While $t < t_{max}$
    Gradient Computation:
      Compute a (sub)gradients for each objective function f_i at x^t
    Subproblem:
      Create proximal-style subproblem
        $v^t \in \operatorname{argmin}\left\{m(v) + \frac{1}{2}|v - x^t|^2 : \ v \in S\right\}$
      where m is a weighted sum model made from the objective functions' values and subgradient information
    Update:
      Set $x^{t+1} = x^t + 0.95\, v^t$
      Set $t = t + 1$
  End while
End

- Application of evolutionary algorithms is common in the literature. However, with the exception of this paper, there is a lack of direct comparison between heuristic and deterministic methods.
- The literature also lacks in applying deterministic multi-objective approached in optimization of RC structures. Such approaches have great potential to model the complex objectives and constraints in structural engineering.

4. Comparative study

In order to compare the algorithms reviewed, we present a comparative study using MOO for RC beam design. The proposed mathematical model for the numerical study is solved using 5 algorithms (MOGA, NSGA-III, MOPSO, BiMADS, and MOGB) that were reviewed in Section 2 and a random method (**Algorithm 13** below). The main reason behind selecting such algorithms are either their popularity in the literature or lack of study in the literature. The addition of the random method is to provide a baseline for what can be accomplished.

The aim is to measure the accuracy and consistency of the algorithms for a range of test problems. The study uses optimal design of simply supported reinforced concrete beams. A set of 25 test problems are then generated by varying length of a beam (L) and loads on the beam (W), as presented in Fig. 2.

The properties of RC beam in terms of its high compressive strength, durability, and resistance to fire and water damages have made it to be widely used structural component. The design of a RC beam is carried out iteratively by selecting trial sections,

Algorithm 12 Multi Objective Gradient Based method (MOGB)

begin
  Initialize
    Select *budget*
  Distribute budget
    Set t_{max}=round ($\sqrt{budget}$)
    Set *unusedbudget=0*
  While *unusedbudget* < *budget*
    Randomly generate a starting point x^0
    Run WSA initialized with x^0 and $maxit$
    Storing *all* iteration points x^t generated by WSA
    Increase *unusedbudget* by function calls used
  End while
  Evaluate date
    Examine all iteration points for non-dominance and output ND and P_t
End

Table 1
Summary of the reviewed literature on optimization of RC structures.

References		Objective Function(s)	Optimization Approach(es)[a]	Structures[b]
Coello et al. [114]	S	Cost	GA	RC Beams
Leps and Sejnoha [115]	S	Cost	Augmented SA	RC Beams
Guerra and Kiousis [116]	S	Cost	SQPA	RC Structures
Babu and Venkataramana [117]	S	Cost	SUMT	RC Beams
Ozrurk et al. [118]	S	Cost	Artificial Bee Colony	RC Beams
Rahmanian et al. [119]	S	Cost	Exhaustive Enumeration	RC Beams
Shariat et al. [120]	S	Cost	Lagrangian Multiplier Method	RC Beams
Barakat et al. [121]	M	Cost; Reliability; Sustainability; Flexural Strength Reliability	ε-Constraint	Prestressed Concrete Beam
Zou et al. [122]	M	Cost; Seismic Performance	ε-Constraint	RC Frames
Lagaros and Papadrakakis [123]	M	Cost; Interstorey Drift	NSES-II	RC Structures
Paya et al. [124]	M	Cost; Constructability; Sustainability; Safety	MOSA	RC Frames
Fragiadakis and Lagaros [125]	M	Initial Cost; Life Cycle Cost	MOPSO	RC Structures
Mitropoulou et al. [126]	M	Initial Cost; Life Cycle Cost	NSES-II	RC Structures
Martinez-Martin et al. [127]	M	Cost; CO_2 Emission	MOSA	RC Bridge Piers
Camp and Assadollahi [128]	M	Cost; CO_2 Emission	Big Bang-Big Crunch	RC Footings
Kaveh et al. [129]	M	Cost; Reinforcing Bar Congestion	NSGA-II	RC Retaining Wall
Khajehzadeh et al. [130]	M	Cost; CO_2 Emission	AGSA-PS	RC Retaining Wall
Choi et al. [22]	M	Cost; Seismic Performance	NSGA-II	RC Frames
Das et al. [66]	M	Cost; Safety Factor	NSGA-II	RC Retaining Wall
Choi [67]	M	Number of Bracings; Dissipated Energy	NSGA-II	RC Frames
This paper	M	Cost; Deflection	MOGA; NSGA-III; MOPSO; MOGB; BiMADS	RC Beams

[a]Sequential Quadratic Programming Algorithm (SQPA); Sequential Unconstrained Minimization Technique (SUMT); Nondominated Sorting Evolution Strategies II (NSES-II); Adaptive Gravitational Search Algorithm with Pattern Search (AGSA-PS).
[b]Reinforced Concrete (RC).

Algorithm 13 Random search (Random)

Begin
  Initialize
    Select *budget*
  Prepare points
    Randomly generate budget feasible points using a uniform distribution
  Evaluate date:
    Examine all points for non-dominance, and output ND and P_t
End

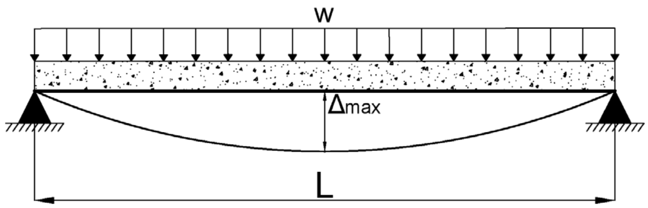

Fig. 2. Schematic view of loads and deflection on RC beam.

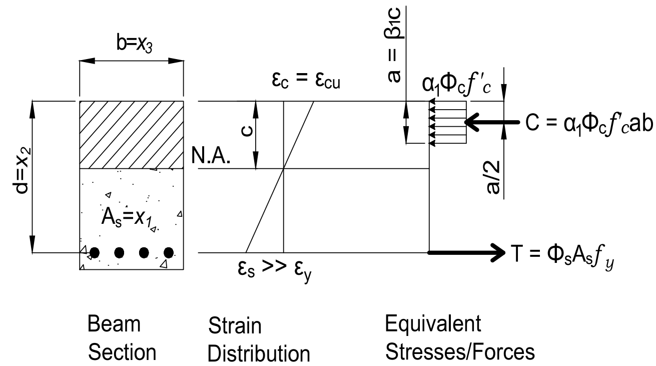

Fig. 3. Actual and equivalent rectangular stress distribution.

which is not often necessarily optimal. The design process should satisfy the safety requirements (at ultimate load) and deflections at serviceability load. This design process is not necessarily optimal, and in this paper, the ultimate design and deflections are considered as a MOO problem.

4.1. Proposed mathematical model

This paper minimizes the construction cost for the steel and concrete materials. Thus, the cost function is measured as the factor of steel and concrete volume consumed in the beam design. Assuming the length of beam as L, Eq. (2) presents the structure of cost objective to be minimized.

$$C = L\,(UC_s)\,x_1 + L\,(UC_c)\,x_2x_3. \tag{2}$$

where C is total RC beam construction cost, UC_s and UC_c are the unit cost of steel and concrete, respectively. Variables $(x_1, \ldots, x_4)$ are illustrated in Fig. 3, and defined as:

- x_1 Cross-sectional area if steel reinforcement (A_s),
- x_2 Effective depth of beam (d),
- x_3 Beam width (b),
- x_4 Compressive strength of concrete (f'_c) that impacts the UC_c.

The beam should be designed to resist ultimate moment, M_u, by satisfying different constraints. Four constraints are defined based on Canadian Standard Association (CSA) [131]. First, minimum amount of steel reinforcement is required (Eq. (3)) to avoid possible sudden failure after the initial cracking of the concrete section.

$$\frac{x_1}{x_3h} \geq \frac{0.2\sqrt{f'_c}}{f_y}. \tag{3}$$

where f'_c is the compressive strength of steel, f_y is the yield strength of steel reinforcement. In order to measure h, it is divided into three sections including effective depth of beam (x_2), radius of steel bar, and concrete cover ($h = x_2 + dbar/2 + cover$).

The second constraint considers the maximum amount of steel to be used in the RC beam, to ensure ductile failure. The constraint is formulated based on the material properties of concrete and steel reinforcements. The stress–strain relationship of concrete is nonlinear curve, while the stress–strain diagram for steel can be presented by elastoplastic diagram. Since application of nonlinear curve for concrete is not possible for design purposes, CSA allows replacing nonlinear stress distribution with an equivalent rectangular stress block. Fig. 3 presents the strain distribution and equivalent stress block of the beam.

As presented in Fig. 3, by assuming compression reinforcement of concrete equal to the tension force in the steel reinforcement ($C_c = T_r$), the depth of the compression stress block (a) is calculate in Eq. (4):

$$a = \frac{\phi_s x_1 f_y}{\alpha_1 \phi_c f'_c x_3}. \tag{4}$$

The ductile failure is ensured with the check shown in Eq. (5). By simplifying the Equations, the maximum ratio of steel reinforcement over the concrete is measured in Eq. (6).

$$\frac{c}{x_2} \leq \frac{700}{700 + f_y}, \tag{5}$$

$$\frac{x_1}{x_2x_3} \leq \frac{700}{700 + f_y} * \frac{\phi_c f'_c \alpha_1 \beta_1}{\phi_s f_y}. \tag{6}$$

where $c = a/\beta_1$. For the third constraint, we consider the factored bending moment resistant to be at least equal to factored design bending moment ($M_r \geq M_f$), as presented as:

$$\phi_s x_1 f_y \left(d - \frac{\phi_s x_1 f_y}{2\alpha_1 \phi_c f'_c x_3} \right) \geq \frac{w_f}{8} L^2. \tag{7}$$

The last constraint is to define a limit for the width over height of RC beam. Based on industrial experience, the width of the beam should be at least longer than one third of RC beam height, as presented in Eq. (8):

$$x_3 \geq \frac{h}{3}. \tag{8}$$

Besides the cost objective function, we considered the deflection as the second objective to be evaluated. For the simply supported beam, the maximum deflection is at the midpoint (Fig. 2) and computed using Eq. (9). The deflection objective function should be re-written in terms of problem variables and parameters, further substitution and simplification is required to prepare the deflection as an objective function. In addition to the variables presented in Fig. 3, deflection needs f'_c to be treated a variable (x_4). Thus, Eq. (9) is substituted with decision variables using a recursive procedure:

$$\Delta = \frac{5wL^4}{384E_cI_e}, \tag{9}$$

where E_c is the modulus of elasticity of concrete, I_e is the effective moment of inertia, and w is the applied load with consideration of both dead and live load.

It is obvious that the deflection equation (Eq. (9)) is nonlinear. In addition to the presented constraints, the fifth constraint is added to limit the f'_c as:

$$20 \text{ MPa} \leq x_4 \leq 40 \text{ MPa}. \tag{10}$$

The multi-objective model is presented as follows:

$$\min F(x) = (f_1(x), f_2(x)) \tag{11}$$

where $f_1(x)$ and $f_2(x)$ represents the deflection limit state and material cost of the RC beam, respectively. The first objective

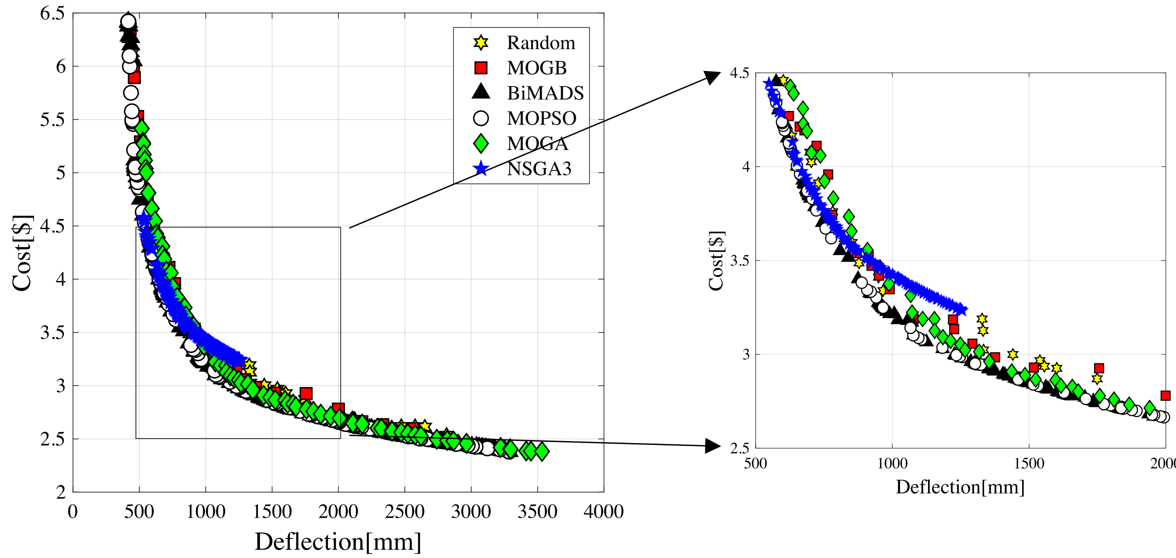

Fig. 4. Pareto fronts for Length = 12000 mm and Load =100 kN/m.

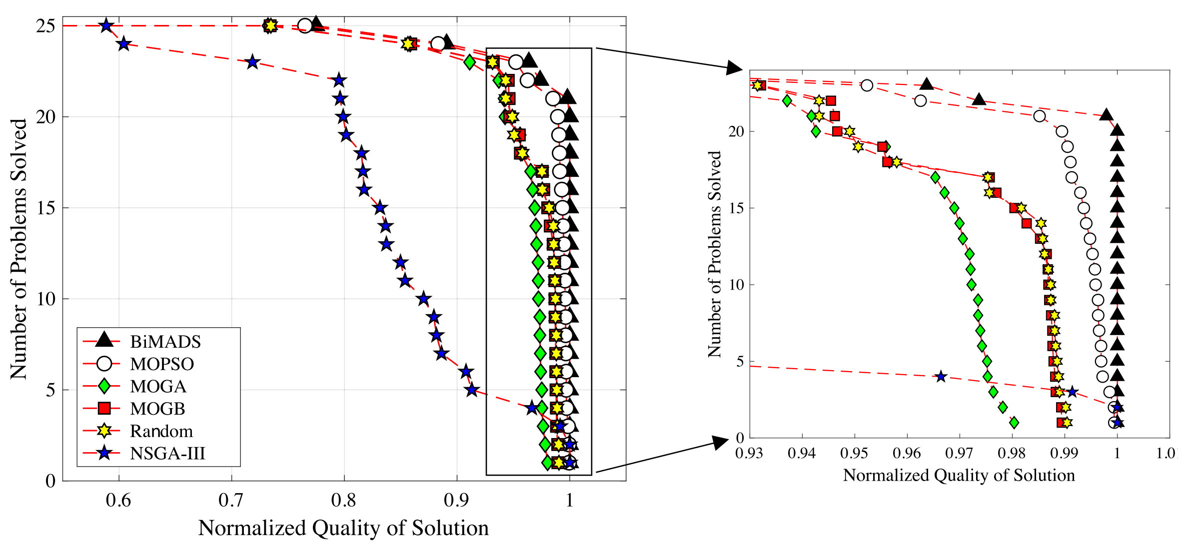

Fig. 5. Accuracy graph for the six algorithms (MOGA, NSGA-III, MOPSO, BiMADS, MOGB, and Random method).

function is computed by expanding and simplifying Eq. (9) as:

$$f_1(x) = \frac{5L^4(DL+LL)}{G\sqrt{x_4}}. \tag{12}$$

where G is:

$$G = \frac{(2844444x_1G_2 - 6320987G_3)}{x_3^2x_4^{3/2}} + \frac{x_3^3x_4^{3/2}G_1^6\left(\frac{x_3G_1^3}{12000} + \frac{(8000G_3-1200x_1G_2)}{2187x_3^2x_4^{3/2}}\right)}{9L^6(DL+LL)^3},$$

$$G_1 = 10x_2 + 5dbar + 602,$$

$$G_2 = (400x_1 + G_5 - 28.3x_2x_3x_4^{1/2}G_4)^2,$$

$$G_3 = (20x_1 - 1.41x_2x_3x_4^{1/2}G_4)^3,$$

$$G_4 = \left(\frac{x_1(200x_1+G_5)}{x_2^2x_3^2x_4}\right)^{1/2},$$

$$G_5 = 9x_2x_3x_4^{1/2}.$$

Second objective function is computed via:

$$f_2(x) = 1.5*10^{-3}x_1 + 1.2*10^{-5}x_2x_3 + 8.4*10^{-4}x_3. \tag{13}$$

Subject to:

$$400x_1 - 14x_3x_4^{1/2} - 0.2x_2x_3x_4^{1/2} \geq 0, \tag{14}$$

$$\frac{x_1}{x_2x_3} - 0.0009x_4 \leq 0, \tag{15}$$

$$\frac{x_1x_2x_3x_4(187-0.34x_4) - 57800x_1^2}{0.55x_3x_4 - 0.001x_3x_4^2} \geq 157500, \tag{16}$$

$$3x_3 - x_2 \geq 70, \tag{17}$$

$$20 \leq x_4 \leq 40. \tag{18}$$

The proposed model is applied to design a simply supported rectangular cross-section RC beam. The beam is reinforced with 4-20M bars at the bottom. Other parameters include the effective depth of 430 mm, and yield strength of steel reinforcement equal to 400 MPa. A set of 25 test problems has been defined to evaluate the examined algorithms. In particular, the RC beam design problem is solved for the combination of 5 different lengths of

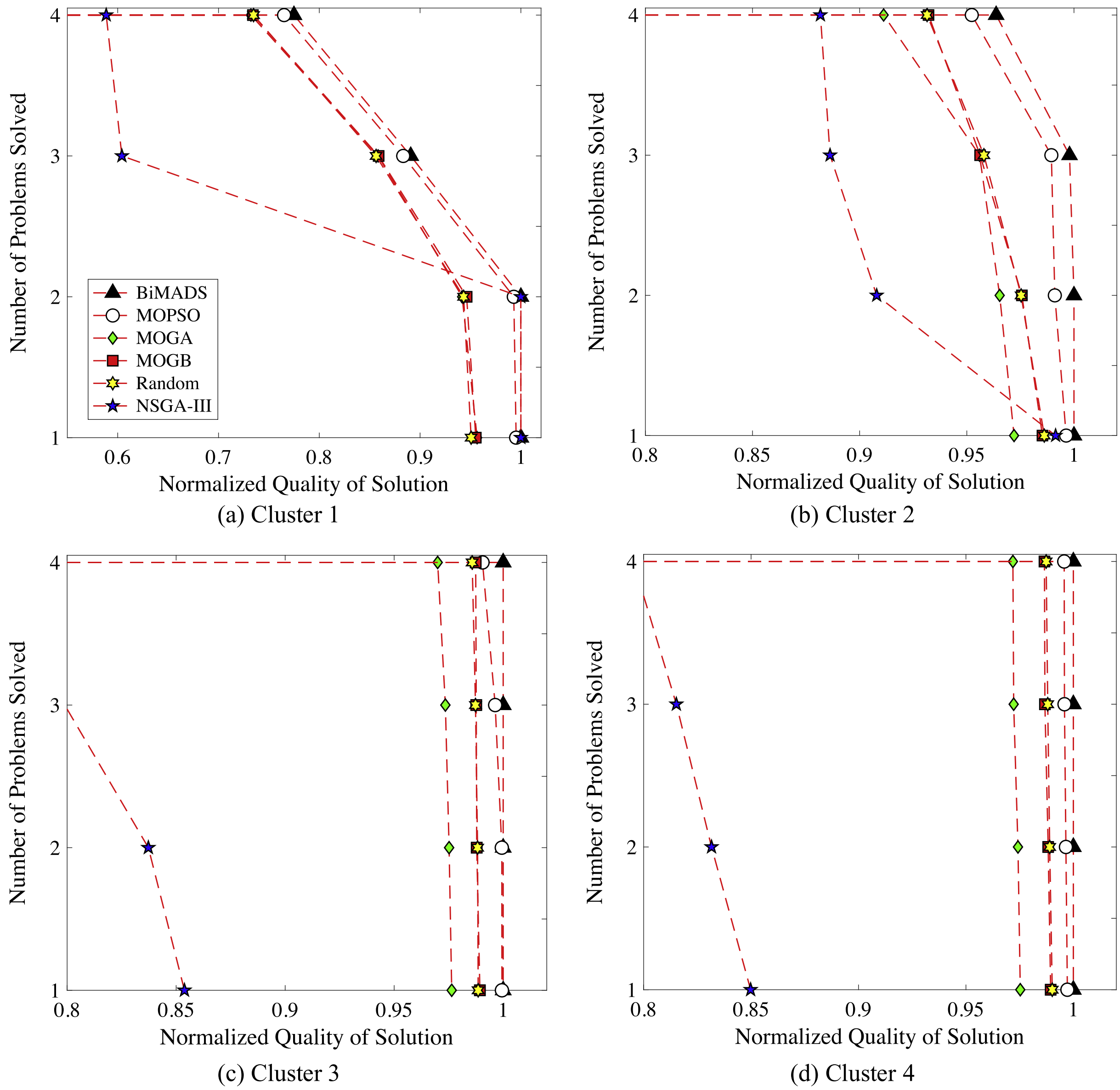

Fig. 6. Accuracy graph for the examined algorithms (MOGA, NSGA-III, MOPSO, BiMADS, MOGB, and Random method) under 4 defined clusters.

beam (3, 4.5, 6, 10, 12 m) and 5 uniform loads (20, 26, 40, 50, 100 kN/m).

4.2. Dealing with constraints

Lack of access to the algorithms that directly handle constraints in the solution approach creates a major challenge so that some of the existing multi-objective algorithms are unable to handle constrained problems. A common method for constrained optimization is replacing the original problem by a set of sub problems in which the constraints are represented by terms added to the objective function. There are three common approaches in the literature to deal with constraints using penalty method including the extreme barrier, quadratic penalty, and log-barrier. In direct-search methods, there are other efficient techniques to handle constraints [132–135]. This paper uses *extreme barrier* method. In this method, a constant infinite penalty value is added to the objective function if a constraint is violated [104]. Thus, the unconstrained problem is reformulated as:

$$E(x) = \min\{f_1(x) + \iota(x), f_2(x) + \iota(x), \ldots, f_n(x) + \iota(x)\} \tag{19}$$

where the *penalty parameter*, $\iota(x)$, is defined as:

$$\iota(x) = \begin{cases} 0 & \text{if } x \in S, \\ Inf & \text{if } x \notin S. \end{cases} \tag{20}$$

In practice, *Inf* can be replaced with any sufficiently large number (e.g., 10^{12}).

5. Results and discussion

The six algorithms (MOGA, NSGA-III, MOPSO, BiMADS, and MOGB, and a Random method) are tested to measure their efficiency and accuracy of the algorithms. Each algorithm is applied to the formulated problem with a penalty term to handle the constraints (Eqs. (14) to (18)). There are several extensions of the algorithms as discussed in Section 2; however, we used the original version of the algorithms for a fare comparison. The aim is to measure if there is any meaningful difference between the Pareto fronts obtained by each algorithm, as well as measuring the efficiency of the penalty methods when applied for the algorithms. Fig. 4 illustrates an example of the Pareto fronts generated by the examined algorithms.

5.1. Accuracy analysis for the examined algorithms

In order to remove the effect of randomness on the optimization results in non-deterministic algorithms, each of the 25 test problems are solved 10 times. The algorithms have been coded in MATLAB so that each algorithm has been treated equally to 10000 functions call (for BiMADS, NOMAD version 3.8.1 is used). The accuracy of algorithms is then measured based on the average values of 10 runs. We adopted a coverage-based method to compare the Pareto fronts of the algorithms [136]. In this method, a common area is defined and the Pareto fronts are compared based the coverage of the common area. Therefore, the algorithms are ranked based on the maximum coverage of the common area. Since the Pareto fronts are numerically compared, we can visualize the efficiency of the algorithms using an accuracy graph. The best algorithm in terms of accuracy is selected when it could reach to higher normalized quality of solution among the solved problems.

Fig. 5 depicts the accuracy graph of the examined algorithms using "*extreme barrier*" method. Fig. 5 shows that BiMADS provides a better result compare to other algorithms. The next better algorithms are MOPSO, MOGA, MOGB, and NSGA-III, respectively. For this particular problem and range of parameters, none of the MOGA, MOGB, and NSGA-III are better than the random algorithm, and NSGA-III does particularly poorly. The performance of NSGA-III was not expected before the analysis. We note that we used the default parameters for each algorithm. Although this presents a fair comparison, it is possible to individually tune the algorithms for a higher performance.

Table 2 presents the summary of comparative study of the examined algorithms. The comparative analysis in Table 2 is based on the normalized performance of algorithms. BiMADS could outperformance other algorithms in 22 out of 25 variations of the RC beam design problem. NSGA-III is the second in terms of number of outperformances, however, due to larger standard deviation, the average normalized performance of NSGA-III is less than all examined algorithms.

5.2. Consistency analysis across the test problems

Another important analysis is to ensure the consistency of the ranking of the algorithms across the test problems. In this regard, 25 variations of the test problems are divided into 4 clusters. Each cluster represents specific characteristics of the test problem. For example, cluster 1 denotes the beams with short length and light loads. The results of consistency test are illustrated in Fig. 6.

To assess the consistency of results, the ranking of the examined algorithms in each cluster is investigated. As depicted in Fig. 6, all six algorithms (BiMADS, MOPSO, MOGA, MOGA, Random, and NSGA-III) remain the same in all clusters. Because the ranking in each cluster remains the same as the ranking in the entire test problems, the results are considered as consistent and further conclusions are made based on the entire test problems.

5.3. Analysis of pareto front

Fig. 7 illustrates the Pareto front of the presented model using BiMADS algorithm considering under specific length and load. The trade-off between cost and deflection helps to find either a design at desired cost or the cost of a certain deflection threshold. More importantly, the design of the RC beams could be adjusted using the sensitivity analysis. In other words, the Pareto front indicates how much reduction in the deflection could be reached by increasing a fair amount of design budget. For example, Fig. 7 shows that the deflection of the RC beam could be reduced up to 3 time (3000 mm to 1000 mm) by adding 30% to the cost (2.45$ to 3.2 $).

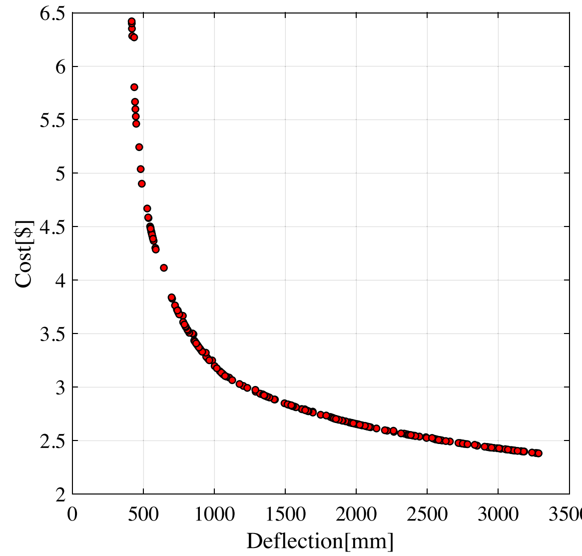

Fig. 7. Pareto fronts for (Length, Load) = [12000 mm, 100 kN/m] Using BiMADS.

After reviewing the accuracy and efficiency of the examined algorithms and penalty methods, another technical analysis is conducted to evaluate the behavior of Pareto front with different values of compressive strength of concrete. Thus, variable x_4 is substituted with deterministic values in the multi-objective model. Fig. 8 illustrates the Pareto front for different values of x_4 \in {25, 30, 35, 40} MPa. Fig. 8 shows that the new multi-objective model has similar behavior when solved with different values of comprehensive strength of concrete. However, by increasing the value of x_4, a more cost-efficient design is obtained from the model. Such results are technically consistent, which makes the model valid to be utilized in construction industries. We also conducted the sensitivity analysis to evaluate the effects of different values of the yield strength of steel reinforcement (f_y) on the Pareto front and solution results. The study off_y \in {350, 400, 450, 500} MPa shows that the Pareto front remains the same for all values but $f_y = 350$ MPa.

Besides the discussion of the model and the algorithms to solve the proposed model, there are several practical implications for the developed method in this paper. First, the trade-off between cost and technical specifications of a structure is a fundamental step before implementing the design. This paper highlighted utility of different optimization algorithms to be used as a tool to minimize the cost of RC beams while satisfying the stringent design constraints. Second, accurate measurement of such trade-off is also of values of the research for practical implementation. Construction industry witnesses huge investments and expenditures each year, however, over-design and under-design are still a challenge for the industry. The proposed method and the comparison of the algorithms reduces the need for spending extra costs (over-design) as well as any collapse (under-design). An efficient and accurate algorithm is the key for saving for designing structures, especially for large-scale construction.

6. Conclusions

Multi-objective optimization models have huge potential to support complex decisions in construction industry, however, methods and approaches are required to facilitate the modeling

Table 2
Summary of comparative analysis of examined algorithms.

	BiMADS	MOPSO	MOGA	NSGA-III	MOGB	Random
Computational time (seconds)	26.16	10.10	5.59	44.86	113.11	4.28
Number of outperformances	22	0	0	3	0	0
Average normalized performance	0.984	0.978	0.951	0.842	0.961	0.960
Standard deviation of normalized performance	0.0492	0.0505	0.0527	0.1018	0.0555	0.0558

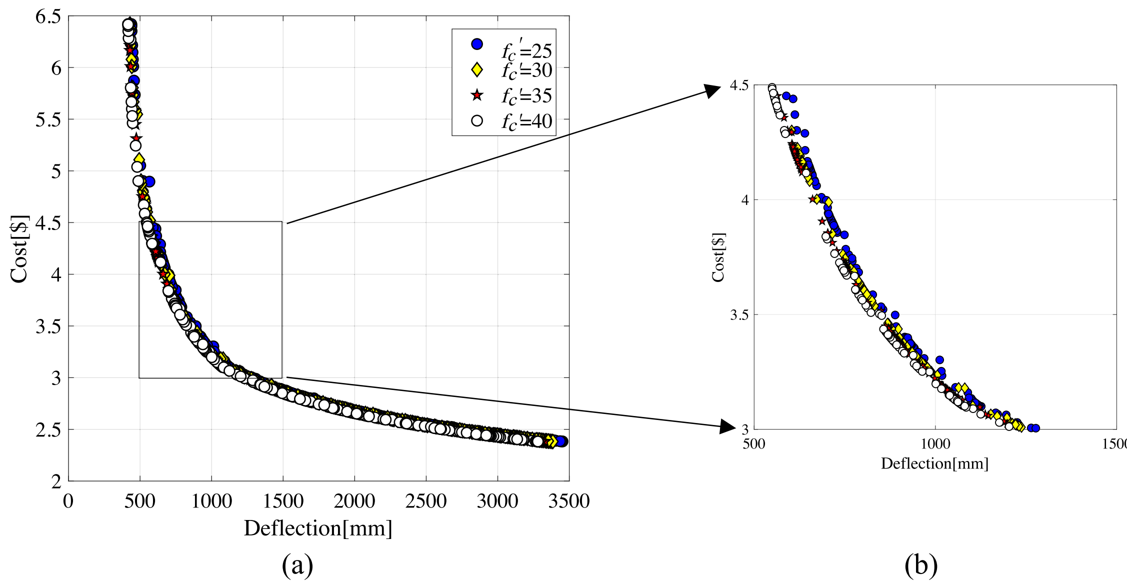

Fig. 8. Effect of the decision variable f_c' (x_4) on the Pareto Front for Length = 12000 mm, Load = 100 kN/m using BiMADS.

and implementation of such models. In this paper, the multi-objective optimization (MOO) concepts and algorithms are reviewed to highlight the gap in the literature for comparative study of efficient algorithms. A multi-objective model is then developed to minimize the cost and deflection of reinforce concrete (RC) beams. To address the efficiency of the algorithms for solving the multi-objective models, five MOO algorithms and one random algorithm are chosen. Using a comparative study, reviewed algorithms are examined to conclude the most efficient ones across a range of test problems. It is worth mentioning that we used the original MOO algorithms for comparing the algorithms. There are several combined algorithms developed in the literature (e.g., combining PSO and GA [26,137,138]) that are not selected for this paper. The paper bridges the gap of the literature in the following directions:

- The developed multi-objective model provides trade-off between cost and deflection when designers decide for the technical specifications of the RC beam. Setting deflection as the objective function is important because the deflection has been mostly addressed as constraint which has limited effect on producing optimal solutions compared to objective function. In addition, the Pareto front of the model enables the decision-makers with adjusting the design parameters to avoid under- and over-design problems.
- Evolutionary algorithms are extensively utilized for the solution of multi-objective model. However, this paper examines derivative-free optimization (DFO) and gradient-based methods as well to conclude the efficient algorithms using the test problem. Thus, a set of measures such as each of developing, efficiency and applicability for structural problems are considered to compare the algorithms. In addition to the efficiency, deterministic methods are superior to evolutionary algorithms due to lack of randomness in the provided solution.

For future research, the authors plan to extend the proposed model to more complex structures in construction industry. Such extension improves the limitation of the paper in the comparative study with more case studies. Uncertainty is available in all systems. Proposing either a second objective as reliability or a robust optimization model as well as comparative study of results is considered for future work. From the optimization perspective, it is sought to develop own approach in dealing with multi-objective optimization problems and compare the efficiency of algorithms over different problems compared to the DFO, gradient-based and the evolutionary algorithms. In addition, this paper could be extended to evaluate the efficacy of other methods to deal with constraints (e.g., log-barrier, quadratic penalty) as future direction of study.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.asoc.2019.105631.

Acknowledgments

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) [Discovery Grant, reference numbers #2018-03865, RGPIN-2014-05013].

References

[1] V. Mokarram, M.R. Banan, An improved multi-objective optimization approach for performance-based design of structures using nonlinear time–history analyses, Appl. Soft Comput. 73 (2018) 647–665.
[2] S.Y. Ok, J. Song, K.S. Park, Optimal design of hysteretic dampers connecting adjacent structures using multi-objective genetic algorithm and stochastic linearization method, Eng. Struct. 30 (5) (2008) 1240–1249.

[3] C.A.C. Coello, A comprehensive survey of evolutionary-based multiobjective optimization techniques, Knowl. Inf. Syst. 1 (3) (1999) 269–308.
[4] L.G. Drummond, B.F. Svaiter, A steepest descent method for vector optimization, J. Comput. Appl. Math. 175 (2) (2005) 395–414.
[5] G. Levitin (Ed.), Computational Intelligence in Reliability Engineering: Evolutionary Techniques in Reliability Analysis and Optimization, vol. 39, Springer Science & Business Media, 2006.
[6] S.N. Omkar, J. Senthilnath, R. Khandelwal, G.N. Naik, S. Gopalakrishnan, Artificial Bee Colony (ABC) for multi-objective design optimization of composite structures, Appl. Soft Comput. 11 (1) (2011) 489–499.
[7] H.S. Kim, J.W. Kang, Semi-active fuzzy control of a wind-excited tall building using multi-objective genetic algorithm, Eng. Struct. 41 (2012) 242–257.
[8] M.E. Uz, M.N. Hadi, Optimal design of semi active control for adjacent buildings connected by MR damper based on integrated fuzzy logic and multi-objective genetic algorithm, Eng. Struct. 69 (2014) 135–148.
[9] F. Amini, P. Ghaderi, Hybridization of harmony search and ant colony optimization for optimal locating of structural dampers, Appl. Soft Comput. 13 (5) (2013) 2272–2280.
[10] A.İ. Ölçer, C. Tuzcu, O. Turan, An integrated multi-objective optimisation and fuzzy multi-attributive group decision-making technique for subdivision arrangement of Ro–Ro vessels, Appl. Soft Comput. 6 (3) (2006) 221–243.
[11] J.G. Iniestra, J.G. Gutiérrez, Multicriteria decisions on interdependent infrastructure transportation projects using an evolutionary-based framework, Appl. Soft Comput. 9 (2) (2009) 512–526.
[12] C. Audet, G. Savard, W. Zghal, Multiobjective optimization through a series of single-objective formulations, SIAM J. Optim. 19 (1) (2008) 188–210.
[13] P.L. Yu, Cone convexity, cone extreme points, and nondominated solutions in decision problems with multiobjectives, J. Optim. Theory Appl. 14 (3) (1974) 319–377.
[14] H. Tamaki, H. Kita, S. Kobayashi, Multi-objective optimization by genetic algorithms: A review, in: Proceedings of IEEE International Conference on Evolutionary Computation, IEEE, 1996, pp. 517–522.
[15] C.A.C. Coello, G.B. Lamont, D.A. Van Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems, vol. 5, Springer, New York, 2007.
[16] M. Wagner, K. Bringmann, T. Friedrich, F. Neumann, Efficient optimization of many objectives by approximation-guided evolution, European J. Oper. Res. 243 (2) (2015) 465–479.
[17] H.P. Benson, An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem, J. Global Optim. 13 (1) (1998) 1–24.
[18] P. Armand, C. Malivert, Determination of the efficient set in multiobjective linear programming, J. Optim. Theory Appl. 70 (3) (1991) 467–489.
[19] R. Benayoun, J. De Montgolfier, J. Tergny, O. Laritchev, Linear programming with multiple objective functions: Step method (STEM), Math. Progr. 1 (1) (1971) 366–375.
[20] K. Miettinen, Nonlinear multiobjective optimization, in: International Series in Operations Research and Management Science, vol. 12, 1999.
[21] A.L. Custódio, J.A. Madeira, A.I.F. Vaz, L.N. Vicente, Direct multisearch for multiobjective optimization, SIAM J. Optim. 21 (3) (2011) 1109–1140.
[22] S.W. Choi, Y. Kim, H.S. Park, Multi-objective seismic retrofit method for using FRP jackets in shear-critical reinforced concrete frames, Composites B 56 (2014) 207–216.
[23] S. Greco, J. Figueira, M. Ehrgott, Multiple criteria decision analysis, in: Springer's International Series, 2005.
[24] A. Konak, D.W. Coit, A.E. Smith, Multi-objective optimization using genetic algorithms: A tutorial, Reliab. Eng. Syst. Saf. 91 (9) (2006) 992–1007.
[25] X.S. Yang, Harmony search as a metaheuristic algorithm, in: Music-Inspired Harmony Search Algorithm, Springer, Berlin, Heidelberg, 2009, pp. 1–14.
[26] T. Weise, R. Chiong, K. Tang, Evolutionary optimization: Pitfalls and booby traps, J. Comput. Sci. Tech. 27 (5) (2012) 907–936.
[27] G.R. Zavala, A.J. Nebro, F. Luna, C.A.C. Coello, A survey of multi-objective metaheuristics applied to structural optimization, Struct. Multidiscip. Optim. 49 (4) (2014) 537–558.
[28] D.F. Jones, S.K. Mirrazavi, M. Tamiz, Multi-objective meta-heuristics: An overview of the current state-of-the-art, European J. Oper. Res. 137 (1) (2002) 1–9.
[29] B. Suman, Study of simulated annealing based algorithms for multiobjective optimization of a constrained problem, Comput. Chem. Eng. 28 (9) (2004) 1849–1871.
[30] J. Brownlee, Clever algorithms: nature-inspired programming recipes. Jason Brownlee, 2011.
[31] J.E.C. Arroyo, R. dos Santos Ottoni, A. de Paiva Oliveira, Multi-objective variable neighborhood search algorithms for a single machine scheduling problem with distinct due windows, Electron. Notes Theor. Comput. Sci. 281 (2011) 5–19.
[32] A.P. Reynolds, B. De la Iglesia, A multi-objective GRASP for partial classification, Soft Comput. 13 (3) (2009) 227–243.
[33] A.J. Nebro, F. Luna, E. Alba, B. Dorronsoro, J.J. Durillo, A. Beham, AbYSS: Adapting scatter search to multiobjective optimization, IEEE Trans. Evol. Comput. 12 (4) (2008) 439–457.
[34] S. Kulturel-Konak, A.E. Smith, B.A. Norman, Multi-objective tabu search using a multinomial probability mass function, European J. Oper. Res. 169 (3) (2006) 918–931.
[35] F. Glover, Tabu search—Part I, ORSA J. Comput. 1 (3) (1989) 190–206.
[36] F. Glover, Tabu search—Part II, ORSA J. Comput. 2 (1) (1990) 4–32.
[37] M. Hansen, Tabu search for multiobjective combinatorial optimization: TAMOCO, Control Cybernet. 29 (2000) 799–818.
[38] A. Hertz, B. Jaumard, C.C. Ribeiro, W.P. Formosinho Filho, A multi-criteria tabu search approach to cell formation problems in group technology with multiple objectives, RAIRO-Oper. Res. 28 (3) (1994) 303–328.
[39] M.P. Hansen, Tabu search for multiobjective optimization: MOTS, in: Proceedings of the 13th International Conference on Multiple Criteria Decision Making, 1997, pp. 574–586.
[40] S. Carcangiu, A. Fanni, A. Montisci, Multiobjective tabu search algorithms for optimal design of electromagnetic devices, IEEE Trans. Magn. 44 (6) (2008) 970–973.
[41] T. Ghisu, G.T. Parks, D.M. Jaeggi, J.P. Jarrett, P.J. Clarkson, The benefits of adaptive parametrization in multi-objective tabu search optimization, Eng. Optim. 42 (10) (2010) 959–981.
[42] J. Pacheco, R. Martí, Tabu search for a multi-objective routing problem, J. Oper. Res. Soc. 57 (1) (2006) 29–37.
[43] D.S. Vianna, J.E.C. Arroyo, A GRASP algorithm for the multi-objective knapsack problem, in: XXIV International Conference of the Chilean Computer Science Society, IEEE, 2004, pp. 69–75.
[44] R. Martí, V. Campos, M.G. Resende, A. Duarte, Multiobjective GRASP with path relinking, European J. Oper. Res. 240 (1) (2015) 54–71.
[45] J.E.C. Arroyo, A.A. de Souza Pereira, A GRASP heuristic for the multi-objective permutation flowshop scheduling problem, Int. J. Adv. Manuf. Technol. 55 (5–8) (2011) 741–753.
[46] C. Blum, et al., Evolutionary optimization, in: R. Chiong, T. Weise, Z. Michalewicz (Eds.), Variants of Evolutionary Algorithms for Real-World Applications, Springer, Berlin, Heidelberg, 2012.
[47] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints, IEEE Trans. Evol. Comput. 18 (4) (2014) 577–601.
[48] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm, TIK-report, 103, 2001.
[49] J.D. Knowles, D.W. Corne, Approximating the nondominated front using the pareto archived evolution strategy, Evol. Comput. 8 (2) (2000) 149–172.
[50] D.W. Corne, J.D. Knowles, M.J. Oates, The pareto envelope-based selection algorithm for multiobjective optimization, in: International Conference on Parallel Problem Solving from Nature, Springer, Berlin, Heidelberg, 2000, pp. 839–848.
[51] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on evolutionary computation 11 (6) (2007) 712–731.
[52] A. Özkış, A. Babalık, A novel metaheuristic for multi-objective optimization problems: The multi-objective vortex search algorithm, Inform. Sci. 402 (2017) 124–148.
[53] A. Babalik, A. Ozkis, S.A. Uymaz, M.S. Kiran, A multi-objective artificial algae algorithm, Appl. Soft Comput. 68 (2018) 377–395.
[54] C.M. Fonseca, P.J. Fleming, Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation, IEEE Trans. Syst. Man Cybern. A 28 (1) (1998) 26–37.
[55] M. Liu, Y.K. Wen, S.A. Burns, Life cycle cost oriented seismic design optimization of steel moment frame structures with risk-taking preference, Eng. Struct. 26 (10) (2004) 1407–1421.
[56] G. Barone, D.M. Frangopol, Life-cycle maintenance of deteriorating structures by multi-objective optimization involving reliability, risk, availability, hazard and cost, Struct. Saf. 48 (2014) 40–50.
[57] I.P. Mitseas, I.A. Kougioumtzoglou, M. Beer, An approximate stochastic dynamics approach for nonlinear structural system performance-based multi-objective optimum design, Struct. Saf. 60 (2016) 67–76.
[58] R. Greco, G.C. Marano, Multi-objective optimization of a dissipative connection for seismic protection of wall-frame structures, Soil Dyn. Earthq. Eng. 87 (2016) 151–163.
[59] M. Barraza, E. Bojórquez, E. Fernández-González, A. Reyes-Salazar, Multi-objective optimization of structural steel buildings under earthquake loads using NSGA-II and PSO, KSCE J. Civ. Eng. 21 (2) (2017) 488–500.
[60] S. Bhattacharjya, S. Chakraborty, An improved robust multi-objective optimization of structure with random parameters, Adv. Struct. Eng. (2018) 1369433217752626.
[61] N. Srinivas, K. Deb, Muiltiobjective optimization using nondominated sorting in genetic algorithms, Evol. Comput. 2 (3) (1994) 221–248.

[62] D.E. Golberg, Genetic Algorithms in Search, Optimization, and Machine Learning, vol. 1989, no. 102, Addion Wesley, 1989, p. 36.

[63] K. Deb, A. Pratap, S. Agarwal, T.A.M.T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.

[64] H. Jain, K. Deb, An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach, IEEE Trans. Evol. Comput. 18 (4) (2014) 602–622.

[65] M. Tavana, Z. Li, M. Mobin, M. Komaki, E. Teymourian, Multi-objective control chart design optimization using NSGA-III and MOPSO enhanced with DEA and TOPSIS, Expert Syst. Appl. 50 (2016) 17–39.

[66] M.R. Das, S. Purohit, S.K. Das, Multi-objective optimization of reinforced cement concrete retaining wall, Indian Geotech. J. 46 (4) (2016) 354–368.

[67] S.W. Choi, Investigation on the seismic retrofit positions of FRP jackets for RC frames using multi-objective optimization, Composites B 123 (2017) 34–44.

[68] J. Ricart, G. Hüttemann, J. Lima, B. Barán, Multiobjective harmony search algorithm proposals, Electron. Notes Theor. Comput. Sci. 281 (2011) 51–67.

[69] J. Knowles, D. Corne, Memetic algorithms for multiobjective optimization: issues, methods and prospects, in: Recent Advances in Memetic Algorithms, Springer, Berlin, Heidelberg, 2005, pp. 313–352.

[70] P. Serafini, Simulated annealing for multi objective optimization problems, in: Multiple Criteria Decision Making, Springer, New York, NY, 1994, pp. 283–292.

[71] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.

[72] B. Suman, P. Kumar, A survey of simulated annealing as a tool for single and multiobjective optimization, J. Oper. Res. Soc. 57 (10) (2006) 1143–1160.

[73] A. Suppapitnarm, G.T. Parks, Simulated annealing: an alternative approach to true multiobjective optimization, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 1999, Morgan Kaufmann Publishers, 1999, pp. 406–407.

[74] E.L. Ulungu, J.F.P.H. Teghem, P.H. Fortemps, D. Tuyttens, MOSA method: a tool for solving multiobjective combinatorial optimization problems, J. Multi-Criteria Decis. Anal. 8 (4) (1999) 221–236.

[75] P. Czyzżak, A. Jaszkiewicz, Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization, J. Multi-Criteria Decis. Anal. 7 (1) (1998) 34–47.

[76] B. Suman, Simulated annealing-based multiobjective algorithms and their application for system reliability, Eng. Optim. 35 (4) (2003) 391–416.

[77] B. Suman, Study of self-stopping PDMOSA and performance measure in multiobjective optimization, Comput. Chem. Eng. 29 (5) (2005) 1131–1147.

[78] A. Ekbal, S. Saha, Combining feature selection and classifier ensemble using a multiobjective simulated annealing approach: application to named entity recognition, Soft Comput. 17 (1) (2013) 1–16.

[79] H.K. Singh, A. Isaacs, T. Ray, W. Smith, A simulated annealing algorithm for constrained multi-objective optimization, in: Evolutionary Computation, 2008 CEC 2008. IEEE World Congress on Computational Intelligence. IEEE Congress on, IEEE, 2008, pp. 1655–1662.

[80] C.H. Antunes, P. Lima, E. Oliveira, D.F. Pires, A multi-objective simulated annealing approach to reactive power compensation, Eng. Optim. 43 (10) (2011) 1063–1077.

[81] S. Sivasubramani, K.S. Swarup, Multi-objective harmony search algorithm for optimal power flow problem, Int. J. Electr. Power Energy Syst. 33 (3) (2011) 745–752.

[82] L.M. Abualigah, A.T. Khader, E.S. Hanandeh, A.H. Gandomi, A novel hybridization strategy for krill herd algorithm applied to clustering techniques, Appl. Soft Comput. 60 (2017) 423–435.

[83] J. Ricart, G. Hüttemann, J. Lima, B. Barán, Multiobjective harmony search algorithm proposals, Electron. Notes Theor. Comput. Sci. 281 (2011) 51–67.

[84] X.Z. Gao, X. Wang, S.J. Ovaska, Harmony search methods for multi-modal and constrained optimization, in: Music-Inspired Harmony Search Algorithm, Springer, Berlin, Heidelberg, 2009, pp. 39–51.

[85] S. Sivasubramani, K.S. Swarup, Environmental/economic dispatch using multi-objective harmony search algorithm, Electr. Power Syst. Res. 81 (9) (2011) 1778–1785.

[86] H.V.H. Ayala, L. dos Santos Coelho, V.C. Mariani, M.V.F. da Luz, J.V. Leite, Harmony search approach based on ricker map for multi-objective transformer design optimization, IEEE Trans. Magn. 51 (3) (2015) 1–4.

[87] X. Dai, X. Yuan, Z. Zhang, A self-adaptive multi-objective harmony search algorithm based on harmony memory variance, Appl. Soft Comput. 35 (2015) 541–557.

[88] I.P. Kougias, N.P. Theodossiou, Multiobjective pump scheduling optimization using harmony search algorithm (HSA) and polyphonic HSA, Water Resour. Manag. 27 (5) (2013) 1249–1261.

[89] W. Hare, J. Nutini, S. Tesfamariam, A survey of non-gradient optimization methods in structural engineering, Adv. Eng. Softw. 59 (2013) 19–28.

[90] A. Nayyar, D.N. Le, N.G. Nguyen (Eds.), Advances in Swarm Intelligence for Optimizing Problems in Computer Science, CRC Press, 2018.

[91] P. Vasant (Ed.), Handbook of Research on Modern Optimization Algorithms and Applications in Engineering and Economics, IGI Global, 2016.

[92] L.M.Q. Abualigah, Feature Selection and Enhanced Krill Herd Algorithm for Text Document Clustering, Springer, 2019.

[93] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.

[94] D. Połap, Polar bear optimization algorithm: Meta-heuristic with fast population movement and dynamic birth and death mechanism, Symmetry 9 (10) (2017) 203.

[95] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Comput. Appl. 27 (4) (2016) 1053–1073.

[96] M. Reyes-Sierra, C.C. Coello, Multi-objective particle swarm optimizers: A survey of the state-of-the-art, Int. J. Comput. Intell. Res. 2 (3) (2006) 287–308.

[97] R.S. Parpinelli, F.R. Teodoro, H.S. Lopes, A comparison of swarm intelligence algorithms for structural engineering optimization, Internat. J. Numer. Methods Engrg. 91 (6) (2012) 666–684.

[98] A.Y. Zomaya (Ed.), Handbook of Nature-Inspired and Innovative Computing: Integrating Classical Models with Emerging Technologies, Springer Science & Business Media, 2006.

[99] J. Angelo, H.S. Bernardino, H.J.C. Barbosa, Multi-objective ant colony approaches for structural optimization problems, in: Proceedings of the Eleventh International Conference on Computational Structures Technology, Civil-Comp Press, Stirlingshire, (UK), 2012.

[100] B. Benhala, A. Ahaitouf, A. Mechaqrane, B. Benlahbib, Multi-objective optimization of second generation current conveyors by the ACO technique, in: Multimedia Computing and Systems (ICMCS), 2012 International Conference on, IEEE, 2012, pp. 1147–1151.

[101] A. Sallem, B. Benhala, M. Kotti, M. Fakhfakh, A. Ahaitouf, M. Loulou, Application of swarm intelligence techniques to the design of analog circuits: evaluation and comparison, Analog Integr. Circuits Signal Process. 75 (3) (2013) 499–516.

[102] D. Angus, C. Woodward, Multiple objective ant colony optimisation, Swarm Intell. 3 (1) (2009) 69–85.

[103] J. Nocedal, S.J. Wright, Numerical Optimization, second ed., Springer, 2006.

[104] C. Audet, J.E. Dennis Jr., Mesh adaptive direct search algorithms for constrained optimization, SIAM J. Optim. 17 (1) (2006) 188–217.

[105] C. Audet, S.Le. Digabel, C. Tribes, NOMAD User Guide, Technical Report G-2009-37, Les cahiers du GERAD, 2009.

[106] S.Le. Digabel, Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm, ACM Trans. Math. Software 37 (4) (2011) 44:1–44:15.

[107] G. Stracquadanio, V. Romano, G. Nicosia, Semiconductor device design using the BiMADS algorithm, J. Comput. Phys. 242 (2013) 304–320.

[108] B. Eisenhower, Z. O'Neill, S. Narayanan, V.A. Fonoberov, I. Mezić, A methodology for meta-model based optimization in building energy models, Energy Build. 47 (2012) 292–301.

[109] J. Fliege, B.F. Svaiter, Steepest descent methods for multicriteria optimization, Math. Methods Oper. Res. 51 (2000) 479–494.

[110] J. Fliege, L.G. Drummond, B.F. Svaiter, Newton's method for multiobjective optimization, SIAM J. Optim. 20 (2) (2009) 602–626.

[111] G.C. Bento, J.C. Neto, P.S.M. Santos, S.S. Souza, A weighting subgradient algorithm for multiobjective optimization, Optim. Lett. 12 (2) (2018) 399–410.

[112] D.G. Norman, Economic aspects in the design of some reinforced concrete structural members, J. Proc. 61 (4) (1964) 419–440.

[113] B.S. Sandhu, Economical design of reinforced concrete slabs and walls, Civ. Eng. 41 (7) (1971) 68.

[114] C.C. Coello, F.S. Hernández, F.A. Farrera, Optimal design of reinforced concrete beams using genetic algorithms, Expert Syst. Appl. 12 (1) (1997) 101–108.

[115] M. Lepš, M. Šejnoha, New approach to optimization of reinforced concrete beams, Comput. Struct. 81 (18–19) (2003) 1957–1966.

[116] A. Guerra, P.D. Kiousis, Design Optimization of Reinforced Concrete Structures (Doctoral dissertation), Colorado School of Mines, 2004.

[117] N. Babu, K. Venkataramana, Shape optimization of steel reinforced concrete beams, Comput. Concrete 4 (4) (2007) 317–330.

[118] H.T. Ozturk, A. Durmus, A. Durmus, Optimum design of a reinforced concrete beam using artificial bee colony algorithm, Comput. Concrete 10 (3) (2012) 295–306.

[119] I. Rahmanian, Y. Lucet, S. Tesfamariam, Optimal design of reinforced concrete beams: A review, Comput. Concrete 13 (4) (2014) 457–482.

[120] M. Shariat, M. Shariati, A. Madadi, K. Wakil, Computational Lagrangian multiplier method by using for optimization and sensitivity analysis of rectangular reinforced concrete beams, Steel Compos. Struct. 29 (2018) 243–256.
[121] S. Barakat, K. Bani-Hani, M.Q. Taha, Multi-objective reliability-based optimization of prestressed concrete beams, Struct. Saf. 26 (3) (2004) 311–342.
[122] X.K. Zou, C.M. Chan, G. Li, Q. Wang, Multiobjective optimization for performance-based design of reinforced concrete frames, J. Struct. Eng. 133 (10) (2007) 1462–1474.
[123] N.D. Lagaros, M. Papadrakakis, Seismic design of RC structures: a critical assessment in the framework of multi-objective optimization, Earthq. Eng. Struct. Dyn. 36 (12) (2007) 1623–1639.
[124] I. Paya, V. Yepes, F. González-Vidosa, A. Hospitaler, Multiobjective optimization of concrete frames by simulated annealing, Comput.-Aided Civ. Infrastruct. Eng. 23 (8) (2008) 596–610.
[125] M. Fragiadakis, N.D. Lagaros, An overview to structural seismic design optimisation frameworks, Comput. Struct. 89 (11–12) (2011) 1155–1165.
[126] C.C. Mitropoulou, N.D. Lagaros, M. Papadrakakis, Life-cycle cost assessment of optimally designed reinforced concrete buildings under seismic actions, Reliab. Eng. Syst. Saf. 96 (10) (2011) 1311–1331.
[127] F.J. Martinez-Martin, F. Gonzalez-Vidosa, A. Hospitaler, V. Yepes, Multi-objective optimization design of bridge piers with hybrid heuristic algorithms, J. Zhejiang Univ. 13 (6) (2012) 420–432.
[128] C.V. Camp, A. Assadollahi, CO 2 and cost optimization of reinforced concrete footings using a hybrid big bang-big crunch algorithm, Struct. Multidiscip. Optim. 48 (2) (2013) 411–426.
[129] A. Kaveh, M. Kalateh-Ahani, M. Fahimi-Farzam, Constructability optimal design of reinforced concrete retaining walls using a multi-objective genetic algorithm, Struct. Eng. Mech. 47 (2) (2013) 227–245.
[130] M. Khajehzadeh, M.R. Taha, M. Eslami, Multi-objective optimisation of retaining walls using hybrid adaptive gravitational search algorithm, Civ. Eng. Environ. Syst. 31 (3) (2014) 229–242.
[131] Canadian Standards Association, Design of Concrete Structures, Canadian Standards Association, Mississauga, Ont., 2004.
[132] C. Audet, J.E. Dennis Jr., A pattern search filter method for nonlinear programming without derivatives, SIAM J. Optim. 14 (4) (2004) 980–1010.
[133] C. Audet, J.E. Dennis Jr., A progressive barrier for derivative-free nonlinear programming, SIAM J. Optim. 20 (1) (2009) 445–472.
[134] C. Audet, J.E. Dennis Jr., Mesh adaptive direct search algorithms for constrained optimization, SIAM J. Optim. 17 (1) (2006) 188–217.
[135] J.E. Dennis, C.J. Price, I.D. Coope, Direct search methods for nonlinearly constrained optimization using filters and frames, Opt. Eng. 5 (2) (2004) 123–144.
[136] E. Zitzler, L. Thiele, An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach, TIK-report, 43, 1998.
[137] L.M. Abualigah, A.T. Khader, Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering, J. Supercomput. 73 (11) (2017) 4773–4795.
[138] M. Esmaelian, H. Shahmoradi, M. Vali, A novel classification method: A hybrid approach based on extension of the UTADIS with polynomial and PSO-GA algorithm, Appl. Soft Comput. 49 (2016) 56–70.