Wiley
International Journal of Energy Research
Volume 2026, Article ID 9120610, 25 pages
https://doi.org/10.1155/er/9120610

WILEY

Research Article

Many Objective Salinity Profile Optimization for Salinity Gradient Solar Ponds

Hamed Rafiei,[1] **Mohammad-R. Akbarzadeh-T.**,[1] **Naser Pariz**,[1] **and Aliakbar Akbarzadeh**[2]

[1]*Department of Electrical Engineering, Center of Excellence on Soft Computing and Intelligent Information Processing (SCIIP), Ferdowsi University of Mashhad, Mashhad 9177948944, Iran*
[2]*Department of Mechanical and Manufacturing Engineering, Royal Melbourne Institute of Technology (RMIT) University, Victoria 3083, Australia*

Correspondence should be addressed to Mohammad-R. Akbarzadeh-T.; akbazar@um.ac.ir

Received 16 October 2023; Revised 21 October 2023; Accepted 9 December 2025

Academic Editor: Rajashik Paul

Salt gradient solar ponds (SGSPs) are a highly economical and sustainable renewable energy technology. However, their optimal design is not trivial due to often conflicting objectives presented in their intended utility, environmental operational settings, and stability requirements. Here, we present the problem of finding the optimal salt density profiles (SDPs) as a vital aspect of SGSP design as a many-objective optimization problem. We then propose a systematic integer-coded, order-preserving mutation and crossover evolutionary operator-based nondominated sorting genetic search algorithm (ONSGA) that automatically generates optimal SDPs. Specifically, the proposed method aims to maximize solar insolation transmission and thermal efficiency while minimizing changes in SDP and total salt consumption, all while meeting the overall criterion for pond stability. The proposed approach is general; in addition to the mentioned criteria, any other relevant criteria, such as climatic conditions, could also be considered. The proposed approach is compared with the traditional multiobjective covariance adaptive evolution strategy (MOCMA) and the competitive and cooperative swarm optimization constrained multiobjective optimization algorithm (CMOCSO) in terms of hypervolume, spread, and Pareto optimal solutions, using the initial SDP measured at the RMIT SGSP pond and studied under five typical initial temperature scenarios. The proposed algorithm outperforms the others in optimizing the SDPs, achieving 45 wins for hypervolume and 43 wins for spread across 65 experiments. Additionally, the optimal profiles yield improvements of up to 94% in some aspects over the currently implemented SDPs. The variety of optimal solutions also confirms that optimal profiles are not unique, and different in-situ criteria surrounding a given solar pond lead to other desirable profiles.

Keywords: density profile; evolutionary algorithms; large-scale optimization; many objective optimization; solar ponds

1. Introduction

Salt gradient solar ponds (SGSPs) are a highly economical and resilient technology for absorbing and storing solar energy regardless of climatic conditions [1]. Various instances of SGSP implementations have confirmed the efficacy of these structures in a variety of meteorological settings, such as the case studies in Barcelona (Spain) [2], Makkah (Saudi Arabia) [3], Melbourne (Australia) [4], El Paso (USA) [5], and Sabzevar (Iran) [6]. Expectedly, these implementations also come with different intended utilities, such as emphasizing their long-term storage capacity versus their energy conversion efficiency and their robustness against environmental disturbances. Besides, these objectives may change over time to serve different purposes during the winter and summer seasons. However, the current research has mainly focused on the pond's *physical* design and has addressed these objectives only in isolation. As a result, a soft approach with adjustable control variables that would concurrently consider the cumulative effect of multiple

goals could be a vital step towards enduring and optimal SGSP designs.

The SGSPs consist of three layers, namely the lower convective zone (LCZ), the nonconvective zone (NCZ), and the upper convective zone (UCZ). Convection maintains uniform temperature and salinity levels in LCZ and UCZ, whereas the salinity gradient in the NCZ suppresses convection. Salinity increases gradually from UCZ (almost freshwater) down to the bottom LCZ (saturated water). Thereby, UCZ, NCZ, and LCZ act as a disturbance rejector, an insulator, and an energy accumulator, respectively [7]. Recently, [7] showed that these salt density profiles (SDPs) could be automatically controlled using artificial intelligence and expert systems. Hence, a critical step in SGSP design is determining the depth and salt density of the layers to meet the actual pond's usage policy and performance. For instance, a pond on a site with a high radiation rate and strong winds needs to have higher robustness than higher energy storage. Similarly, a pond operating in cold seasons in an area with limited salt availability should achieve stability and energy generation efficiency while keeping low salt consumption. The utilization policy for greenhouse preheating could also significantly differ in heating residences. In short, as will also be shown in this research, there is no unique and optimal global profile that concurrently meets all of the SGSPs' utility criteria in all places and at all times. Hence, the SDPs should be designed for each pond based on its unique and frequently conflicting criteria. However, due to a lack of an automated procedure, the SDPs are often determined based on the expert's knowledge for general purposes and situations.

The primary contribution of this work is hence addressing the above problem and introducing SGSP design as an in-situ discrete (MOSO) problem. Specifically, we consider objectives such as stability, control effort, solar radiation transmission, and salt consumption criteria to achieve the highest sustainability and performance of the SGSPs. Notably, these profiles often conflict, raising the need for many objective searches. Furthermore, they are not unique, and different criteria, such as the environmental conditions and circumstances surrounding a pond installation, could lead to varied profiles. To this end, we then propose integer-coded order-preserving mutation and crossover operators for a nondominated sorting genetic algorithm (ONSGA) to reach optimal profiles. These operators play a crucial role in avoiding infeasible solutions for the SGSP profiles, thereby leading to a more effective search process. Figure 1 illustrates the focus of the proposed method in the context of SGSPs, where the proposed ONSGA provides a desired reference profile based on the pond condition and desired objectives. The control part then manipulates the pond by pumps in either automated [7] or manual configurations [8] to reach this profile. As needed, this profile design process could be repeated based on new criteria at scheduled times. So, the MOSO problem and its associated solution strategy (ONSGA) are important steps toward fully automated optimal control of SGSPs.

In summary, the main contributions of this work are twofold, as follows:

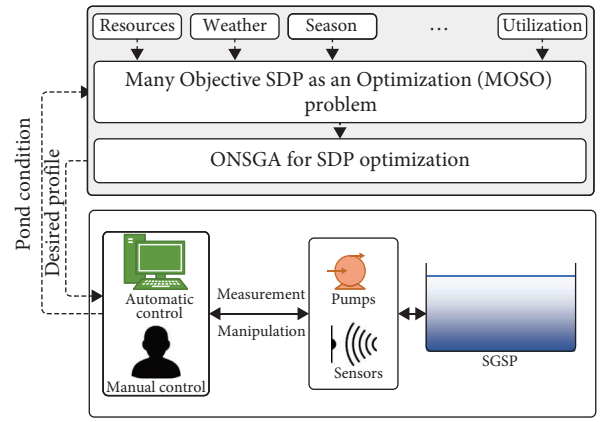


FIGURE 1: The proposed many-objective salt-density optimization (MOSO) and its corresponding ONSGA many-objective optimization. The dashed arrows (optimization) occur on a different timescale from the solid arrows (control cycle).

- The design process is formulated as a many-objective salt-density optimization (MOSO) problem by concurrently considering various criteria such as stability, robustness, and energy storage. This problem definition contrasts with previous works, which mostly consider pond stability for manually designing SDPs.
- The ONSGA algorithm is proposed for automatically solving the MOSO without requiring gradient computation using novel order-preserving mutation and crossover operators.

Furthermore, we consider five scenarios for pond design to examine the feasibility and validity of the proposed method and its resulting profiles.

The remainder of this article is organized as follows. A literature review on possible SGSP objectives is offered in Section 2. Section 3 offers preliminaries on the pond discretization concept. Section 4 defines the salt gradient profile design as a many-objective optimization problem. Section 5 explains the proposed evolutionary search strategy. Simulation results are then provided in Section 6. Finally, Sections 7 and 8 give the discussion and conclude the article, respectively.

2. Related Works

Several studies on SGSPs have addressed their various design objectives, such as stability, controllability, performance, novel combinations, operating cost, and energy efficiency. In particular, extensive research has explored the stability of SGSPs from a mathematical perspective. In [9], the solar pond stability is investigated under variable stratifications. It is shown that there are two types of solutions based on high wavenumbers and small solute Rayleigh numbers. The relevance of wavenumbers is also confirmed in [10]. Furthermore, an analytical solution for nonlinear equations and the stability of SGSPs is presented in [11]. This method converts the nonlinear equations to linearized ones under a specific gradient profile, similar to time-independent Schrödinger equations, which account for the system's nonlinearity.

The stability of SGSPs is also explored in [12] by plotting the ratios of temperature differential in height against change in salt density concentration. This work showed that stability is obtained when the salinity of LCZ and UCZ are at a minimum (0%) and maximum (at the saturation point), respectively. In [5], the operation and stability of SGSPs are investigated numerically and subsequently validated by measurements from the solar pond at El Paso. In these works, pond stability is numerically investigated under different SDPs.

Some studies have investigated SGSP stability experimentally. For example, in [13], the stability and double diffusion of temperature and salt gradient are explored in a laboratory-scaled SGSP. It is shown that instability occurs more in the NCZ compared to the other layers. In [14], a laboratory-scaled solar pond is also constructed to investigate the stability of the solar pond according to stability criteria. In short, regarding the internal stability of NCZ, the stability of the boundaries (LCZ-NCZ and NCZ-UCZ) is a critical concern that affects the overall stability of the SGSP. In [15], it is noted that convection and upward salt diffusion make UCZ and LCZ thicker; therefore, we must enforce a boundary.

For pond stability, the stability of boundaries is crucial. In [16], the dynamics of LCZ are explored under an equilibrium condition of the lower interface. This analysis is subject to turbulence and diffusion, based on a new model that encompasses micro-convection and thermal bursts. In [17], a 3D model of SGSPs is analyzed, while its thermal conductivity, salinity density, and thermal and salt diffusion coefficients change as a function of temperature and salinity. This model is subsequently verified with the experimental results from a pond in Tunisia. In [18], the analysis of the stability of an industrial pond reveals that instability of the upper and lower boundaries, caused by environmental conditions and heat extraction or salt addition, can effectively lead to instability of the entire pond. In [19], the stability of boundaries is studied. It is shown that the erosion of boundaries is directly related to diffusivities and the density stability ratio. Also, solar absorption and thermal diffusion effects are studied to achieve a more realistic boundary condition. In [20], a 3D numerical simulation is carried out to explore stability in the NCZ. Also, temperature distribution, salinity distribution, and velocity fields are studied in SGSP's height and width. Furthermore, it is shown that linear stability criteria could be assumed for a large-scale SGSP. In [21], the effect of porous media on the stability of SGSPs is explored numerically. This porous media region induces more stability and, in some cases, higher temperatures at the LCZ. In [3], the stability of the Makkah solar pond is studied numerically in the presence of perturbations in temperature and salinity with a change in UCZ thickness and heat extraction.

Several works focus on SGSP's thermal performance and its effect on stability. In [22], the diffusive coefficients and radiation absorption are assumed to be depth-dependent variables, and the stability criteria are then explored accordingly. It is concluded that only nonlinear profiles for temperature and salinity satisfy the stability criteria. In [23], variable turbulent Prandtl–Schmidt numbers are used to model the SGSP by a 2D numerical method. In [24], the transient behavior of an SGSP is investigated using a 2D numerical method. It is demonstrated that heat extraction, water transparency, and solar radiation each have a critical impact on SGSP stability, particularly during warm seasons. Additionally, it is noted that the top of the LCZ and the bottom of the NCZ are more susceptible to instability. In [20], the stability of NCZ in SGSPs is investigated using a 3D numerical method, and stability parameters are obtained by solving a series of nonlinear equations. In [25], the stability of the equilibrium solution is explored under thermal diffusion. In [26], layer thickness is optimized for simultaneous heat storage and Artemia cultivation based on the Crank–Nicolson model.

Salt consumption, along with stability, could be a significant economic factor for SGSPs. In [27], the minimum required concentration for maintaining stability is calculated. In [27], the least salt concentration is investigated for each layer according to stability criteria. In [28], a typical SGSP is compared with some SGSPs of different UCZ concentrations ranging from 0% to 10%. The numerical simulation and experimental results indicate that any increase in salt at the surface reduces evaporation and the temperature in the LCZ.

SGSPs are very useful when a stable, low-temperature heat stream is required. As mentioned above, the LCZ is for heat storage, and the heat is often extracted from this layer; however, auxiliary heat extraction with NCZ and other combined technologies improves the overall SGSP performance. In [29], the difference between the temperatures of LCZ and UCZ drives a thermoelectric generator. In [30], the extracted heat from the LCZ of a pond combined with a chimney is used for power generation. The temperature in the NCZ-LCZ boundary is usually higher than in adjacent layers. In [2, 15], the heat is extracted from both LCZ and NCZ instead of LCZ solely; as a result, the thermal efficiency of the SGSP is increased by up to 55%. Additionally, the NCZ extraction poses a threat to the stability of SGSP due to the imposed density disturbance at the boundary between NCZ and LCZ [31]. In [32], a solar collector is added to an industrial SGSP in Spain to boost the temperature of the LCZ layer and enhance the efficacy of the SGSP.

Ambient temperature also affects the performance of SGSPs. While minimum salt consumption and maximum energy extraction are directly related to the pond objectives, ambient conditions pose a concern regarding the pond's stability and robust performance. Some reports indicate that SGSPs can be used in low-temperature climates, as evidenced by the El Paso SGSP, which experienced a period of frozen surface and observed no significant decrease in efficiency [33, 34]. Therefore, a question arose on whether freezing the surface has a positive or negative effect. The answer was simple: the impact of wind rejection is positive, while the increased surface reflection and decreased radiation penetration have adverse effects. In [35], the effect of different ice thicknesses on the performance of SGSPs is explored. The thermal performance of an SGSP in winter is analyzed in [36]. It is demonstrated that ambient temperature can significantly impact the efficiency of the SGSP, and the salinity gradient plays a crucial role in

determining storage capacity, the temperature of the LCZ, and the overall efficiency of the SGSP.

As the above literature illustrates, several studies have addressed the pond performance and the SDP design under different circumstances through conducting experiments and following heuristic procedures. However, many of these objectives, such as pond stability, operating cost, and energy efficiency, conflict. In other words, there is no superior solution that could meet all of these goals. Furthermore, the objectives form a multimodal and multidimensional function that is not necessarily differentiable. Therefore, the traditional derivative-based procedures for determining optimal profiles are inappropriate.

In the current literature on derivative-free optimization, several approaches could address conflicting objectives. Most of these approaches center on the basic notion of nondominated sorting of solutions and population-based search techniques such as genetic algorithms [37]. NSGA-III [38], in particular, is among the potential strategies for solving such nonlinear many (more than 3) objective optimization problems [39]. In genetic algorithms, the crossover and mutation operators cause the evolved crowd to approach an optimal answer over several generations. Various works have modified the operators of NSGA-III for specific applications [40]. More details on crossover and mutation operators can be found in [41, 42]. However, NSGA-III is not directly applicable to the MOSO because a monotonically increasing salt profile is essential in the solar pond, and current operators do not support it. The search space for layers' salinity is also vast, covering the range of 0kg/m^3 for pure water at the top to $360\,\text{kg/m}^3$ for saline water at the bottom. Furthermore, optimizing pond and layer characteristics could considerably increase the number of decision variables.

The facts above suggest that SGSP has the potential to be considered as a large-scale optimization problem. However, despite significant progress, many-objective optimization algorithms often require numerous function evaluations to obtain optimal solutions [43]. For better performance in big data optimization problems, adaptive mutation and crossover operators are introduced in [44, 45] for the NSGA-III, respectively. Problem-dependent crossover operators incorporate domain knowledge or heuristics into the recombination process, which helps generate offspring that better respect problem constraints and structure. This often leads to higher-quality solutions compared to traditional, problem-independent crossovers [46]. Therefore, we must design suitable evolutionary operators to satisfy the necessary SDP conditions.

For this purpose, integrating problem-specific knowledge into the above optimization process could be a successful strategy for reducing function evaluations and improving offspring selection within the search space. In the context of SDP, as demonstrated in this research, attending to SGSP's unique layered structure offers promising opportunities for algorithm optimization through targeted operator modifications.

3. Preliminaries

3.1. Pond Discretization. Here, the concept of pond discretization [7] is briefly discussed. Based on this concept, several boundaries in the SGSP are defined, and the volume between each pair of consecutive boundaries is referred to as a layer. The boundaries are determined based on the available diffuser locations in the pond. The higher the number of layers, the finer the SDP resolution. The pond is assumed to be large, and each layer is considered to have a uniform temperature and salinity. Finally, since mass and heat transfer are slow in the SGSP, we assume that the layer specifications (temperature and salinity) are constant during optimal profile search and layer manipulations.

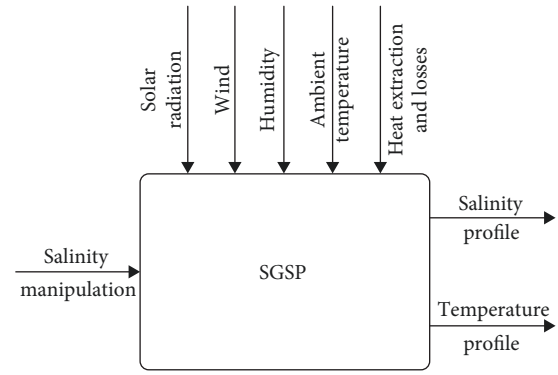


FIGURE 2: Input, outputs, and disturbances of a typical SGSP [7].

3.2. Pond's Stimuli. From a systematic perspective, many stimuli affect the performance of SGSPs [7]. These can be categorized into two main groups: disturbances and inputs. Inputs are those that are manipulated through manual or automatic control, whereas disturbances are not controlled. Wind, solar radiation, humidity, heat losses, weather conditions, water turbidity, and heat extraction are instances of disturbances. In contrast, the salinity injection/suction in/from each layer are inputs (Figure 2) [7, 47]. In [48], a neural network predicts the temperature of extracted water from LCZ and the efficiency of a laboratory-scaled SGSP. The first and most crucial step in implementing and maintaining the SGSP is determining the appropriate density profile considering these stimuli [15].

Based on the above preliminaries, we formulate the optimal salinity search of each layer as an optimization problem, presented in the following section.

4. Proposing Salt Density Profile Design as a Many-Objective Optimization Problem (MOSO)

In the following, several design criteria for the optimal SDP are first presented. The design strategy toward such a suitable SDP is then defined as an optimization problem. Several criteria affecting the performance of the SGSP, including stability criteria, control effort, salt consumption, and thermal performance, are determined here.

The first criterion is pond stability. The stability of SGSP's interfaces R_{ρ_i} between layers i and $i+1$ can be analyzed based on the dimensionless stability parameter from [49]:

$$R_{\rho_i} = \frac{\beta(S_{i+1} - S_i)}{\alpha(T_{i+1} - T_i)}, \tag{1}$$

where α is the coefficient of thermal expansion. β is the coefficient of saline expansion. S_i and S_{i+1} are salinities of layers i and $i+1$. T_i, T_{i+1} are temperatures of layers i and $i+1$.

The parameter R_{ρ_i} quantifies the relative importance of salinity-induced density stratification versus thermally induced density changes. Higher values of R_{ρ_i} indicate stronger stability, as the stabilizing salinity gradient dominates over the destabilizing thermal gradient. We aim to stabilize the interface R_{ρ_i} by driving its value toward a constant desired number γ. Therefore, the cost function for stability can be defined as follows:

$$f_1 : \sum_{i=1}^{N-1} \max\left(0, \gamma - R_{\rho_i}\right). \tag{2}$$

A second aspect is minimizing the control effort by reducing the number of salt diffusion actions. Based on the pond discretization concept [7], SGSPs are discretized into several layers on top of each other. The more similar the initial profile is to the optimal profile, the less effort is needed to reach it. Therefore, the least number of consecutive actions required for stability is crucial to the optimal SDP. In other words, the candidate salinity for each layer must be close to the current salinity. For this purpose, we define the cost function as follows:

$$f_2 : N - (\text{the number of matched layers}). \tag{3}$$

Third is minimizing total salt consumption and maximizing radiation transmission. The salt-related costs could be reduced by:

- Selecting a solar pond site in an area with abundant salt and solar insolation.
- Using the most appropriate type of salt.
- Decreasing salt diffusivity in the solar pond. The diffusion coefficient of salt directly increases with layer temperature [50].
- Having the least layer salinity changes. Salt diffusion between the two layers is directly proportional to the difference between their salt concentrations [7].

If we can minimize density under constant temperature, radiation transmission rises, and salt consumption decreases. The following cost function hence satisfies these aspects:

$$f_3 : \sum_{i=1}^{N} \rho_i. \tag{4}$$

The fourth criterion is maximizing the heat capacity of layers. The heat capacity of layer i (C_{p_i}) could be calculated using the following equation [51]:

$$C_{p_i} = 4180 + 4.396C_i + 0.0048C_i^2, \tag{5}$$

where C_i is the salt concentration of layer i. The heat storage in each layer depends on the heat capacity of the solution. Therefore, as the salinity of layers increases, the heat capacity and energy storage increase. The fourth criterion is minimizing the inverse heat capacity, that are defined as follows:

$$f_4 : \sum_i \frac{1}{C_i}. \tag{6}$$

The index i determines the heat extraction site of the pond and the layers that we want to have high heat capacity.

Lastly, we seek the minimum salinity inter-layer variation. As the salinity of adjacent layers is closer to each other, the layer's salinity change is smaller, needing less frequent modification [7]. Therefore, the cost function can be defined as follows:

$$f_5 : \sum_{i=1}^{N-1} C_i - C_{i+1}. \tag{7}$$

According to the above criteria, we define the optimization problem as follows:

Definition 1. An SGSP with discretized layers as an optimization problem is formulated as follows:

$$\min_{\mathbf{C}} \{f_1, f_2, f_3, f_4, f_5\} \text{s.t. } C_i < C_{i+1}, \tag{8}$$

where $\mathbf{C} = [C_1, \ldots, C_N]^T$ is the concentration of layers. These concentrations should be increasing because of convection prevention.

In this optimization problem, the objective functions are minimized according to layer concentrations.

There are occasions where some of these criteria are in agreement. For instance, maximizing the heat capacity of the lower layers also leads to better stability by strengthening the LCZ–NCZ boundary. Also, less salt consumption in SDP is desired for minimum density (f_3) and salinity inter-layer density variation (f_5) criteria.

Considering Equation (8), a question arises on whether these five criteria conflict. Although there are occasions where some of these criteria are in agreement, such as maximizing the heat capacity of the lower layers that also leads to improving stability by strengthening the LCZ–NCZ boundary or a less salt consumption SDP is desired for minimum density (f_3) and salinity inter-layer density variation (f_5) criteria, we focus on the conflicting situations. Figure 3 shows the relations between criteria. The stability criterion (f_1) distances the layers' salinity to suppress the convection, while the minimum density criterion (f_3) favors less salinity, and the minimum salinity intra-variation criterion (f_5) makes them closer. The control effort (f_2) resists any manipulation that the other criteria may seek. Finally, the

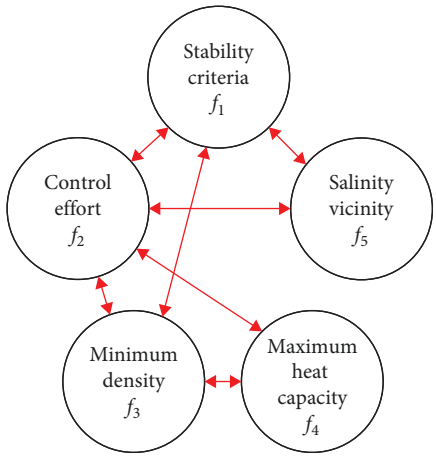


FIGURE 3: Relation of the criteria. Red arrows show the conflict relation.

minimum density criterion (f_3) decreases the lower layers' concentration and heat capacity (f_4). The above conflicting objectives are addressed by the many-objective ONSGA optimization algorithm proposed in the following section.

There are also physical meanings underlying these criteria. Again, f_1 directly affects the pond stability. f_2 concerns the pond repair duration (t_{total}) and the pumps' power consumption (P). So one can conclude that $f_2 \propto t_{\text{total}} \propto P$. Furthermore, $f_3 \propto S_c$, where S_c is the overall pond salt consumption. Finally, $f_4 \propto \eta$ and $f_5 \propto D$, where η and D are pond thermal efficiency and diffusivity, respectively.

5. The Proposed ONSGA

The salinity of layers in SGSPs changes in quantized levels; therefore, we should replace operators in the traditional NSGA-III with integer ones. Also, an optimal SDP should be incremental from top to bottom; otherwise, the proposed profile is inapplicable, and pond stability is threatened. For this case, we modify the operators to be order-preserving, as the main difference from the NSGA-III. The overall procedure of ONSGA is explained in Figure 4 and the Algorithm 1. Discrete *Crossover* and *Mutation* operators are provided in Algorithms 2 and 3. The crossover operator combines parts of parent solutions to create new offspring. It starts by randomly selecting two parent solutions. Then it picks a random cut point where it will split and combine the parents' gene sequences. After mixing the parents' genes at this cut point, it sorts the resulting offspring's genes to maintain proper order. The sorting step guarantees the new solutions will be valid. This method allows good traits from parents to be passed to children while keeping solutions correctly ordered. The mutation operator also helps create new solutions while keeping them valid. It works by randomly picking two solutions from the population and selecting random positions in their gene sequences. The operator then swaps the values at these positions between the two solutions. After swapping, it sorts the genes in both solutions to maintain the correct order. This approach ensures the solutions stay valid while introducing helpful changes. The sorting step is what makes it "order-preserving"—it always keeps the solutions properly ordered, to be implementable in SGSPs. The proposed algorithm uses a similar procedure as in NSGA-III [38] regarding population diversity by predefining the reference points set. The reference points are placed on a normalized hyperplane that is equally inclined to all objective axes. Considering M objective functions and p divisions per objective, H reference points are calculated as follows:

$$H = \binom{M+p-1}{p}, \tag{9}$$

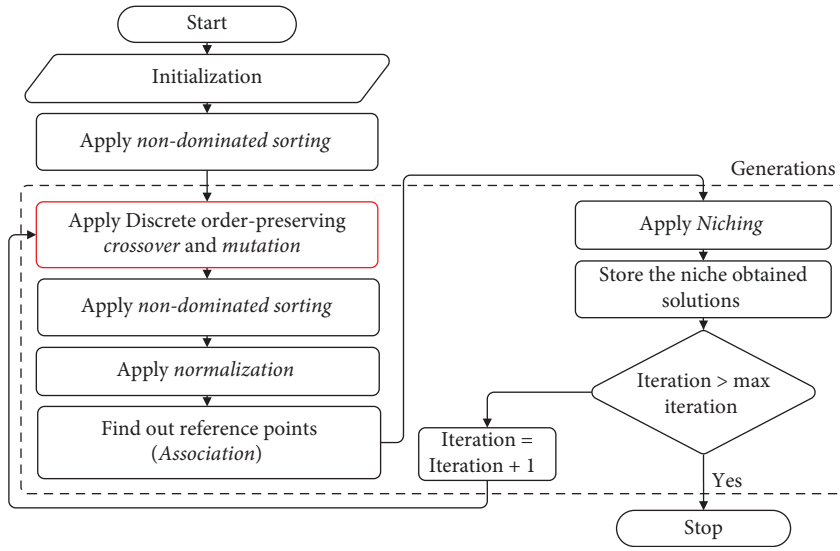
where $\binom{\ }{\ }$ is the choose operator. The other parts of ONSGA are similar to NSGA-III and are described briefly here for completeness. The *Nondominated-sorting* compares solutions to find which solutions are nondominant, known as the Pareto Front (Algorithm 4). *Normalization* lets ONSGA deal with differently scaled objective functions of the Pareto-optimal front (Algorithm 5). *Association* associates each individual to the related reference point (Algorithm 6). *Niching* provides the population for the next generation (Algorithm 7).

6. Results

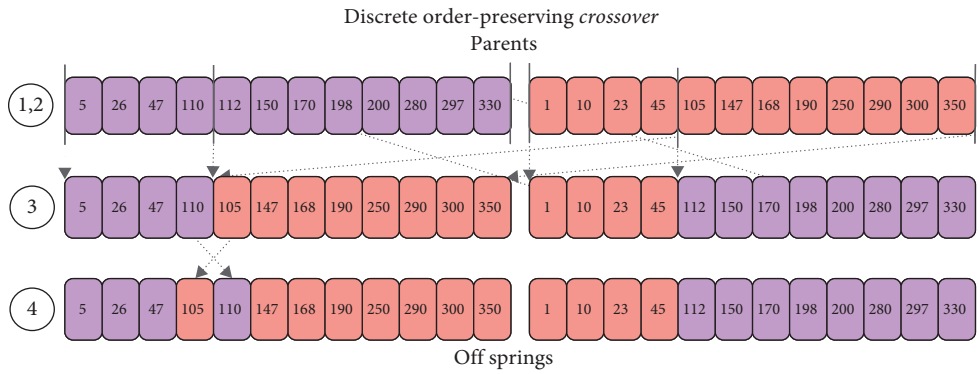
Five scenarios are used to examine the proposed concepts. One layer in UCZ, 10 layers in NCZ, and one layer in LCZ are supposed for a typical SGSP. The optimal SDP is calculated under five initial temperature profiles shown in Figure 5. In [49], γ is supposed to be greater than 1.14, therefore, γ in this article is chosen at 3. It is supposed that in all scenarios, the temperature of layers does not change after setting up the optimal SDP. Otherwise, the thermal balance should be calculated. The current SDP is as follows: 20, 53, 86, 119, 152, 185, 218, 251, 284, 317, 340, and 350 kg/m^3. Please note that here the optimal salinity profile for SGSP's settings is proposed. Therefore, there is no need to model thermal and salinity over time.

6.1. Simulation. The proposed method produces an optimal SDP based on the current salinity profiles of the RMIT's SGSP and the following expected temperature scenarios:

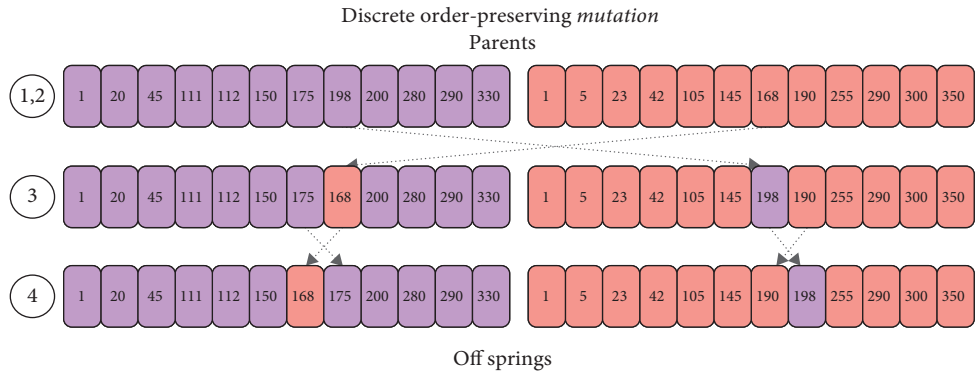
- Scenario 1, semi-constant temperature profile: A semi-constant temperature profile $\in$ [298 K, 310 K] is considered for the current temperature profile. This scenario could be used in the setup of SGSPs so that all layers have almost the same temperature.
- Scenario 2, linear temperature profile: A linear temperature profile distributed in $\in$ [298 K, 388 K] is considered. This scenario could be used when the SGSPs warm up before utilization.
- Scenario 3, exponential temperature profile: An exponential temperature profile in SGSP distributed in $\in$ [298 K, 387 K] is considered. This scenario is very close to the operational temperature profile in such ponds.
- Scenario 4, reference temperature profile: A temperature profile from [27] is applied. This profile is related



(a)



(b)



(c)

FIGURE 4: (a) Flowchart of ONSGA. (b) Proposed *Crossover* operator including (1) Select two individuals randomly, (2) Get a cut-point randomly, Crossover two individuals, and (4) Sort individuals genes. (c) Proposed *Mutation* operator including (1) Select two individuals randomly, (2) Get two indices randomly, Exchange two genes, and (4) Sort individuals genes.

to 90°C at LCZ with zero bottom reflectivity and $\in$ [298 K, 368 K]. As stated before, heat can be extracted from the NCZ layer. In this scenario, we assume that heat is continuously extracted from LCZ and NCZ, consequently maintaining a linear temperature profile throughout the pond.

- Scenario 5, the temperature profile of small SGSP's RMIT (Figure 6): This Scenario is for testing the proposed method on a real available temperature profile when the gradient of the pond is destroyed. The optimal SDP for the pond is proposed based on stability, control effort, minimum diffusivity between layers,

Data: Reference points Z^s, Supplied aspiration points Z^a, Parent population P_t.
Result: P_{t+1}
$S_t = \emptyset, i = 1;$
Q_t = *Discrete order-preserving Crossover*(P_t) + *Discrete order-preserving Mutation*(P_t);
$R_t = P_t \cup Q_t;$
$(F_1, F_2, \cdots) = Non - dominated - sort(R_t);$
while $|S_t| \geq N$ **do**
  $S_t = S_t \cup F_i$ *and* $i = i + 1;$
end
The last front to be included: $F_l = F_i;$
if $|S_t| = N$ **then**
  $P_{t+1} = S_t$, *break;*
else
  $P_{t+1} = \cup_{j=1}^{l-1} F_j;$
  Points to be chosen from $F_l : K = N - |P_{t+1}|;$
  Normalize objectives and create a reference set
    $Z_r : Normalize(f^n, S_t, Z^r, Z^s, Z^a);$
  Associate each individual s of S_t *with a reference point:*
    $[\pi(s), d(s)] = Associate(S_t, Z^r)$; $\pi(s)$: *closest reference point, d: distance between s and* $\pi(s)$;
  Compute niche count of reference point
    $j \in Z^r : \rho_j = \sum_{\mathbf{s} \in S_t / F_l} ((\pi(\mathbf{s}) = j)\ ?\ 1\ :\ 0);$
  Choose K individuals from F_l *to construct*
    $P_{t+1} : Niching(K, \rho_j, \pi, d, Z^r, F_l, P_{t+1})$
end

Algorithm 1: Generation t of ONSGA.

Data: P_t
Result: $CrossedP_t$
for *Number of Parents* **do**
  Select two individuals randomly;
  Get a cut-point randomly;
  Crossover two individuals;
  Sort individuals' genes;
end

Algorithm 2: Discrete Order-Preserving Crossover.

Data: P_t
Result: $MutatedP_t$
for *Number of Mutants* **do**
  Select two individuals randomly;
  Get exchange indices randomly;
  Exchange two genes;
  Sort individuals' genes;
end

Algorithm 3: Discrete Order-Preserving Mutation.

Data: R_t
Result: F_i
for each $p \in R_t$ **do**
  $S_p = \emptyset$; $n_p = 0$;
  for each $q \in R_t$ **do**
    if $p < q$ **then**
      $S_p = S_p \cup \{q\}$;
    end
    else if $q < p$ **then**
      $n_p = n_p + 1$;
    end
  end
  if $n_p = 0$ **then**
    $p_{\text{rank}} = 1$; $F_1 = F_1 \cup \{p\}$;
  end
end
$i = 1$;
while $F_i \neq \emptyset$ **do**
  $Q = \emptyset$;
  for each $p \in F_i$ **do**
    for each $q \in S_p$ **do**
      $n_q = n_q - 1$;
      if $n_p = 0$ **then**
        $q_{\text{rank}} = i + 1$; $Q = Q \cup \{Q\}$;
      end
    end
  end
  $i = i + 1$; $F_i = Q$;
end

Algorithm 4: *Nondominated–Sort*.

Data: S_t, Z^s (structured points) or Z^a (supplied points)
Result: f^n, Z^r (reference points on normalized hyper-plane)
for $j = 1$ to M **do**
  Compute ideal point: $z_j^{\min} = \min_{\mathbf{s} \in S_t} f_j(\mathbf{s})$;
  Translate objectives: $f_j(\mathbf{s}) = f_j(\mathbf{s}) - z_j^{\min} \forall \mathbf{s} \in S_t$;
  Compute extreme points $(z^{j,\max}, j = 1, \cdots, M)$ of S_t;
end
Compute intercepts a_j for $j = 1, \cdots, M$;
Normalize objectives (f^n);
if Z^a is given, **then** Map each (aspiration) point on the normalized hyper-plane and save the points in the set Z^r;
else
  $Z^r = Z^s$;
end

Algorithm 5: *Normalization* (f_n, S_t, Z^r, Z^s/Z^a).

minimum salt consumption, and maximum thermal capacitance in the lower layers to store more thermal energy. The current salinity and temperature profiles are considered the initialization of our method. The optimal SDP for a 2-m-height pond is proposed when the salinity of its upper layers is zero.

Data: Z^r, S_t
Result: $\pi(\mathbf{s} \in S_t), d(\mathbf{s} \in S_t)$
for each reference point $\mathbf{z} \in Z^r$ **do**
  Compute reference line $\mathbf{w} = \mathbf{z}$;
end
for each $\mathbf{s} \in S_t$ **do**
  for each $\mathbf{w} \in Z^r$ **do**
    Compute $d^{\perp}(\mathbf{s}, \mathbf{w}) = ||(\mathbf{s} - \mathbf{w}^T \mathbf{s}\mathbf{w}/||\mathbf{w}||2)||$;
  end
  Assign $\pi(\mathbf{s}) = \mathbf{w} : argmin_{\mathbf{w} \in Z^r} d^{\perp}(\mathbf{s}, \mathbf{w})$;
  Assign $d(\mathbf{s}) = d^{\perp}(\mathbf{s}, \pi(\mathbf{s}))$;
end

Algorithm 6: *Association* (S_t, Z^r).

Data: $K, \rho_j, \pi(\mathbf{s} \in S_t), d(\mathbf{s} \in S_t), Z^r, F_l$
Result: P_{t+1}
$k = 1$;
while $k \leq K$ **do**
  $J_{\min} = \{j: \text{argmin}_{j \in Z^r} \rho_j\}$; $\bar{j} = \text{random}(J_{\min})$;
  $I_{\bar{j}} = \{\mathbf{s} : \pi(\mathbf{s}) = \bar{j}, \mathbf{s} \in F_l\}$;
  if $I_{\bar{j}} \neq \phi$ **then**
    if $\rho_{\bar{j}} = 0$ **then**
      $P_{t+1} = P_{t+1} \cup (\mathbf{s})$:$\text{argmin}_{\mathbf{s} \in I_{\bar{j}}} d(\mathbf{s})$);
    else
      $P_{t+1} = P_{t+1} \cup \text{random } (I_{\bar{j}})$
    end
    $\rho_{\bar{j}} = \rho_{\bar{j}} + 1$; $F_l = F_l / \mathbf{s}$; $k = k + 1$;
  else
    $Z^r = Z^r / \{\bar{j}\}$;
  end
end

Algorithm 7: *Niching* (K, ρ_j, π, d, Z^r, F_l, P_{t+1}).

The step-by-step instruction to apply the proposed method is provided in Figure 7.

In the remainder of this section, we first compare the proposed ONSGA with multi-objective covariance adaptive evolution strategy (MOCMA) and competitive and cooperative swarm optimization constrained multi-objective optimization algorithm (CMOCSO) in obtaining an optimal salinity profile considering various object combinations. Afterward, we compare the optimal profiles offered by the proposed ONSGA with an empirically established salinity profile. Finally, we show that the proposed optimal profiles meet the layer stability criteria of SGSP.

6.2. Comparing Optimization Algorithms. For solving the MOSO problem, we implement the proposed algorithm using the PlatEMO platform v4.5 [52] developed in MATLAB (2022b) software on a system with a Core i8 CPU and 16 GB RAM. Several algorithms, such as NSGA,[1] MOEAD,[2] LMOCSO,[3] MOEADAWA,[4] MOEADDE,[5] MOPSO,[6] NMPSO,[7] and S3CMAES,[8] have been examined. However, they all fail in most scenarios; therefore, we refrained from reporting their results in comparison. Specifically, we define more than 30 algorithms for optimizing 70 problems, including all objective combinations for all scenarios. After each problem, exclude the algorithms with "NaN" for hypervolume and spread for the rest of the problems. Finally, the proposed algorithm is compared with MOCMA and CMOCSO through the platform because they converge in all scenarios and present a better solution than other algorithms. The population size, number of runs, and maximum function evaluation are 100, 10, and 10,000, respectively.[9] The algorithms are compared over hyper volume, spread, and Pareto solutions.

Tables 1 and 2 show the performance of algorithms for different scenarios per objective functions. The proposed ONSGA is superior in most cases (45 for hypervolume and 43 for spread out of 65 experiments across all scenarios). The

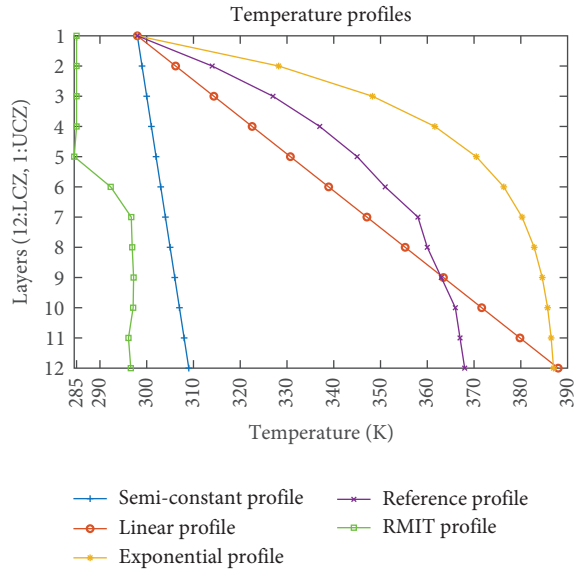


FIGURE 5: Five scenarios of temperature for each layer in the SGSP.

FIGURE 6: SGSP's RMIT [7].

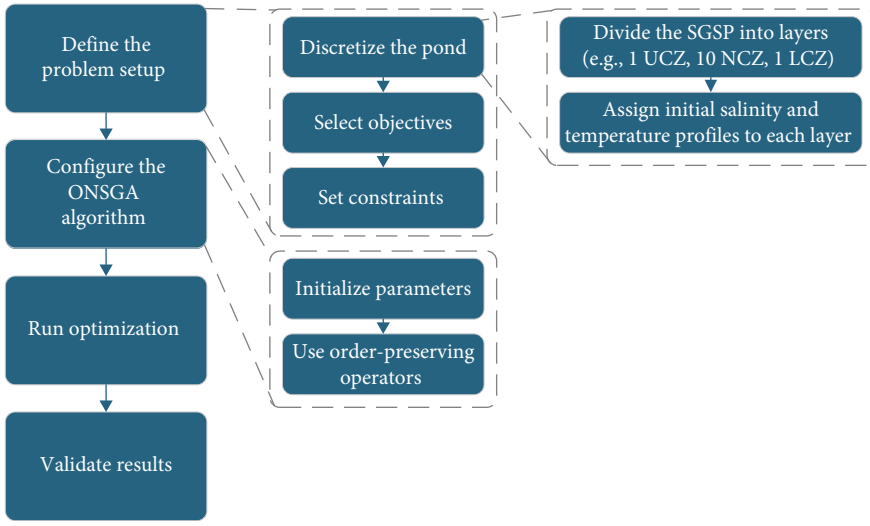


FIGURE 7: The step-by-step instruction to apply the proposed method.

Table 1: Performance of optimization algorithms for semi-constant, linear, and exponential scenarios.

Scenario	Objectives	MOCMA			CMOCSO			ONSGA		
		Object values	HV mean (std)	Spread mean (std)	Object values	HV mean (std)	Spread mean (std)	Object values	HV mean (std)	Spread mean (std)
Semi-constant	f_1	(0)	—	—	(0)	—	—	(0)	—	—
	f_1, f_2	(0,12)	2.4231e–1 (5.36e–2)	1.0008e + 0 (9.17e–4)	(0,0)	9.0146e–1 (2.37e–1)	1.0001e0 (3.42e–4)	(0,0)	**1.0000e0 (0.00e0)**	**1.0000e + 0 (0.00e + 0)**
	f_1, f_3	(0,305)	9.2671e–1 (3.30e–2)	1.0002e + 0 (3.58e–4)	(0,28)	**9.8644e–1 (2.73e–2)**	1.0039e0 (4.28e–3)	(0,59)	9.8610e–1 (4.49e–3)	**1.0004e + 0 (1.08e–3)**
	f_1, f_4	(0,1.30e4)	2.1513e–1 (6.63e–4)	1.0392e + 0 (1.27e–2)	(3,1.2138e4)	**2.2047e–1 (7.22e–4)**	1.0109e0 (1.81e–2)	(0,1.20e4)	2.1606e–1 (8.48e–4)	**1.0007e + 0 (4.72e–3)**
	f_1, f_5	(0,0.0016)	**9.7228e–1 (3.86e–3)**	**9.9987e–1 (2.28e–4)**	(0,0.0014)	9.1971e–1 (1.09e–3)	1.0000e0 (1.66e–5)	(0,0.0014)	9.5960e–1 (2.34e–16)	1.0000e + 0 (0.00e + 0)
	f_1, f_2, f_3	(0,12,284)	1.0282e–1 (3.75e–2)	1.0070e + 0 (2.81e–3)	(0,12,41)	2.6668e–1 (1.15e–1)	1.0037e0 (3.11e–3)	(0,8,132)	**6.5132e–1 (1.28e–1)**	**1.0030e + 0 (1.22e–3)**
	f_1, f_2, f_4	(0,12,1.3081e4)	2.2642e–2 (5.89e–3)	1.0087e + 0 (7.13e–3)	(0,7,1.3286e4)	4.5447e–2 (1.67e–2)	1.0094e0 (9.19e–3)	(0,8,1.2744e4)	**9.4871e–2 (2.00e–2)**	**1.0071e + 0 (1.98e–2)**
	f_1, f_2, f_5	(0,11,0.0014)	1.5885e–1 (7.23e–2)	1.0002e + 0 (3.58e–4)	(0,5,0.0014)	7.2373e–1 (2.41e–1)	1.0000e0 (9.10e–5)	(0,2.0,0.0014)	**8.9176e–1 (3.33e–2)**	**1.0000e + 0 (9.54e–5)**
	f_1, f_3, f_4	(3,177,1.3689e4)	**2.1549e–1 (6.58e–4)**	1.0324e + 0 (3.37e–2)	(0,55,1.2098e4)	2.1254e–1 (6.18e–4)	**8.8673e–1 (1.28e–1)**	(0,44,1.2178e4)	2.0556e–1 (9.04e–3)	9.8041e–1 (2.06e–2)
	f_1, f_3, f_5	(0,325,0.0015)	8.5517e–1 (3.00e–2)	1.0056e + 0 (4.93e–3)	(0,90,0.0014)	6.7196e–1 (3.81e–1)	1.0024e0 (1.50e–3)	(0,178,0.0014)	**9.0933e–1 (8.60e–3)**	**9.9309e–1 (1.02e–2)**
	f_1, f_4, f_5	(0,336,0.0015)	8.6002e–1 (2.34e–2)	1.0028e + 0 (2.26e–3)	(0,102,0.0014)	7.3194e–1 (3.86e–1)	1.0012e0 (1.66e–5)	(0,134,0.0015)	**8.9239e–1 (4.82e–3)**	**9.9774e–1 (2.59e–3)**
	f_1, f_2, f_3, f_4	(0,12,302,1.3204e4)	2.0896e–2 (4.96e–3)	1.0786e + 0 (1.56e–1)	(0,11,347,1.2601e4)	5.1711e–2 (1.63e–2)	**9.7449e–1 (6.97e–2)**	(0,8,297,1.3313e4)	**9.7849e–2 (1.20e–2)**	9.7851e–1 (1.53e–2)
	f_1, f_2, f_3, f_5	(0,12,285,0.0014)	1.0227e–1 (3.94e–2)	1.0087e + 0 (7.13e–3)	(0,11,280,0.0014)	1.7605e–1 (1.33e–1)	1.0016e0 (4.36e–3)	(0,7,177,0.0015)	**6.4462e–1 (9.97e–2)**	**9.9208e–1 (1.99e–3)**
	f_1, f_2, f_3, f_4, f_5	(0,11,315,1.3385e4,0)	1.4156e–2 (2.36e–3)	NaN (NaN)	(27,12,28,1.1955e4,0.0022)	8.7778e–4 (2.59e–3)	1.0022e0 (3.89e–3)	(0,9,331,1.3097e4,0.0014)	**8.0863e–2 (5.12e–3)**	**9.2813e–1 (1.70e–2)**
Linear	f_1	(0.2715)	—	—	(0.22)	—	—	(0)	—	—
	f_1, f_2	(0.3664,10)	1.6285e–1 (6.26e–2)	**1.0000e + 0 (0.00e + 0)**	(0,1)	1.6285e–1 (6.26e–2)	1.0008e0 (2.62e–3)	(0,1)	**9.7704e–1 (5.10e–2)**	1.0003e + 0 (9.92e–4)
	f_1, f_3	(0,360)	8.6219e–1 (5.04e–2)	9.9492e–1 (3.66e–2)	(6.59,178)	8.6219e–1 (5.04e–2)	9.9492e–1 (3.66e–2)	(0,258)	**9.6143e–1 (5.75e–3)**	**9.9102e–1 (7.39e–3)**
	f_1, f_4	(0,1.3150e4)	**2.0632e–1 (4.39e–3)**	1.1886e + 0 (1.21e–1)	(24.45,1.1792e4)	1.6093e–1 (5.31e–3)	1.1886e0 (1.21e–1)	(0,1.2546e4)	1.9480e–1 (4.94e–3)	**1.0080e + 0 (4.14e–2)**
	f_1, f_5	(1.0622,0.0016)	8.8983e–1 (9.61e–3)	**9.9986e–1 (2.42e–4)**	(0.23,0.0014)	6.0249e–1 (2.69e–3)	1.0004e0 (8.95e–4)	(0,0.0014)	**9.4911e–1 (4.14e–2)**	1.0000e + 0 (0.00e + 0)
	f_1, f_2, f_3	(1.357,12,320)	1.2066e–1 (6.24e–2)	1.0036e + 0 (5.94e–3)	(29.02,12,20)	2.2083e–1 (1.09e–1)	9.8496e–1 (1.49e–2)	(0,6,282)	**6.9338e–1 (1.04e–1)**	**9.8140e–1 (2.78e–3)**
	f_1, f_2, f_4	(4,12,1.2876e4)	1.8585e–2 (8.41e–4)	1.1688e + 0 (4.07e–1)	(17.84,11,1.2106e4)	2.5802e–2 (1.46e–2)	9.8098e–1 (8.76e–2)	(0,10,1.2927e4)	**9.5491e–2 (1.64e–2)**	**8.0850e–1 (3.89e–2)**
	f_1, f_2, f_5	(1.0523,12,0.0014)	1.1904e–1 (5.47e–2)	1.0062e + 0 (4.23e–3)	(1.19,0,0.0014)	4.8641e–1 (1.17e–1)	**1.0022e0 (4.02e–3)**	(0,2,0.0014)	**9.0821e–1 (3.89e–2)**	1.0043e + 0 (4.11e–3)

Table 1: Continued.

Scenario	Objectives	MOCMA			CMOCSO			ONSGA		
		Object values	HV mean (std)	Spread mean (std)	Object values	HV mean (std)	Spread mean (std)	Object values	HV mean (std)	Spread mean (std)
	f_1, f_3, f_4	(3.048,298,1.2855e4)	**2.0374e–1 (4.54e–3)**	1.0703e + 0 (1.62e–1)	(17.77,360,1.2342e4)	1.7302e–1 (3.99e–3)	**7.1646e–1 (1.45e–1)**	(0,293,1.2871e4)	1.7909e–1 (7.89e–3)	9.2247e–1 (3.76e–2)
	f_1, f_3, f_5	(0.8039,339,0.0015)	8.3201e–1 (2.86e–2)	1.0167e + 0 (1.38e–2)	(32.0,5,0.2)	3.8772e–1 (3.36e–1)	9.9287e–1 (2.85e–2)	(0,285,0.0015)	**8.9783e–1 (3.42e–2)**	**9.8585e–1 (3.91e–3)**
	f_1, f_4, f_5	(3.578,1.2995,e4,0.0015)	1.5427e–1 (6.11e–2)	2.2535e + 0 (1.64e + 0)	(30,1.1756e4,0.0031)	0.0e0(0.0e0)	1.0011e0 (1.09e–3)	(0,1.2764e4,0.0016)	**1.7456e–1 (6.59e–3)**	**9.0242e–1 (9.65e–2)**
	f_1, f_2, f_3, f_4	(2.354,12,328,1.3428e4)	1.8394e–2 (5.13e–3)	1.0095e + 0 (8.36e–3)	(15.24,11,101,1.4112e4)	2.9599e–2 (5.85e–3)	**6.8468e–1 (1.54e–1)**	(0,9,318,1.3037e4)	**8.9967e–2 (1.00e–2)**	9.3945e–1 (1.62e–2)
	f_1, f_2, f_3, f_5	(3.890,12,335,0.0015)	8.6341e–2 (5.11e–2)	1.0091e + 0 (6.78e–3)	(5.02,9,231,0.0014)	2.2627e–1 (1.61e–1)	**9.6061e–1 (3.21e–2)**	(0,8,268,0.0015)	**7.1928e–1 (7.11e–2)**	9.7212e–1 (5.61e–3)
	f_1, f_2, f_3, f_4, f_5	(8.871,12,342,1.29541,0)	1.5537e–2 (5.52e–3)	1.0604e + 0 (4.19e–2)	(31.61,12,7,1.1741e4,0.0014)	0.0e0(0.0e0)	1.0020e0 (1.27e–3)	(0,12,261,1.2697e4,0.0021)	**7.4436e–2 (8.33e–3)**	**9.2809e–1 (7.55e–3)**
Exponential	f_1	(0.4929)	—	—	(5.91)	—	—	(0)	—	—
	f_1, f_2	(3.0470,12)	9.0440e–2 (3.10e–4)	NaN (NaN)	(0.09,2)	8.5578e–1 (1.91e–1)	**1.0006e0 (7.02e–4)**	(0,4)	**8.8626e–1 (6.44e–2)**	1.0006e + 0 (8.60e–4)
	f_1, f_3	(5.695,395)	8.9264e–1 (8.00e–3)	1.0102e + 0 (0.00e + 0)	(6.7,185)	**9.6727e–1 (3.07e–2)**	1.0142e0 (1.09e–2)	(0,208)	9.6356e–1 (1.01e–2)	**1.0063e + 0 (2.13e–3)**
	f_1, f_4	(3.502,1.308e4)	**2.1646e–1 (7.97e–4)**	1.0529e + 0 (4.22e–2)	(20,1.1647e4)	1.6926e–1 (1.68e–3)	1.0409e0 (4.53e–2)	(0,1.3060e4)	1.8804e–1 (5.05e–3)	**1.0066e + 0 (3.12e–2)**
	f_1, f_5	(2.8478,0.0014)	9.3487e–1 (5.00e–3)	1.0001e + 0 (9.81e–5)	(0.24,0.0014)	7.5212e–1 (3.62e–3)	1.008e0 (1.58e–3)	(0,0.0014)	**9.3864e–1 (2.87e–2)**	**1.0000e + 0 (0.00e + 0)**
	f_1, f_2, f_3	(3.485,12,264)	9.0440e–2 (3.10e–4)	NaN (NaN)	(7.8,12,121)	3.3925e–1 (1.53e–1)	1.0012e0 (8.66e–3)	(0,9,250)	**7.0896e–1 (1.00e–1)**	**9.8950e–1 (5.25e–3)**
	f_1, f_2, f_4	(4.062,12,1.2737e4)	1.9808e–2 (8.50e–5)	1.0645e + 0 (1.10e–1)	(12.7,12,1.2330e4)	3.7264e–2 (1.18e–2)	1.0343e0 (3.98e–2)	(0,10,1.2941e4)	**1.0021e–1 (1.07e–2)**	**9.5773e–1 (3.23e–2)**
	f_1, f_2, f_5	(8.5848,12,0.0014)	NaN (NaN)	NaN (NaN)	(1.23,4,0.0014)	2.5767e–1 (6.22e–2)	1.0029e0 (1.99e–3)	(0,4,0.0014)	**8.9297e–1 (6.77e–2)**	**1.0022e + 0 (2.75e–3)**
	f_1, f_3, f_4	(5.819,360,1.2773e4)	**2.1633e–1 (7.07e–4)**	1.0229e + 0 (2.02e–2)	(24.17,5,1.1633e4)	1.9514e–1 (1.73e–3)	**8.8608e–1 (1.72e–1)**	(0,294,1.3392e4)	1.7587e–1 (8.10e–3)	9.5622e–1 (1.70e–2)
	f_1, f_3, f_5	(0.7975,360,0.0015)	2.1633e–1 (7.07e–4)	1.0005e + 0 (1.19e–3)	(4.9,313,0.0014)	4.4849e–1 (1.62e–1)	**9.7864e–1 (3.89e–2)**	(0,299,0.0014)	**8.9479e–1 (2.92e–2)**	9.8970e–1 (3.02e–3)
	f_1, f_4, f_5	(8.88,1.2652e4,0.0016)	1.3406e–1 (8.37e–2)	1.0365e + 0 (0.00e + 0)	(30,1.1627e4,0.5)	0.0e0(0.0e0)	1.0007e0 (5.51e–4)	(0.0012,1.2941e4,0.0016)	**1.7736e–1 (8.96e–3)**	**9.3211e–1 (4.37e–2)**
	f_1, f_2, f_3, f_4	(8.39,12,429,1.2830e4)	9.0440e–2 (3.10e–4)	1.0156e + 0 (1.81e–2)	(3.7,12,254,1.3341e4)	4.8328e–2 (1.23e–2)	**9.0201e–1 (9.28e–2)**	(0,12,316,1.3193e4)	**8.3092e–2 (9.66e–3)**	9.6258e–1 (1.25e–2)
	f_1, f_2, f_3, f_5	(1.405,12,287,0.0017)	1.2251e–1 (5.34e–2)	1.0028e + 0 (3.57e–3)	(10.7,10,257,0.0013)	1.7483e–1 (1.34e–1)	9.8071e–1 (2.41e–2)	(0,10,285,0.0017)	**7.4205e–1 (5.48e–2)**	**9.6144e–1 (7.56e–3)**
	f_1, f_2, f_3, f_4, f_5	(12.38,11,350,1.3499e4,0)	NaN (NaN)	NaN (NaN)	(30,12,11,1.1633e4,0.09)	0.0e0(0.0e0)	1.0021e0 (1.46e–3)	(0,10,311,1.3298e4,0.0015)	**8.0626e–2 (9.11e–3)**	**9.4117e–1 (1.22e–2)**

Note: The best results are shown in bold.

Table 2: Performance of optimization algorithms for reference and RMIT scenarios.

Scenario	Objectives	MOCMA			CMOCSO			ONSGA		
		Object values	HV mean (std)	Spread mean (std)	Object values	HV mean (std)	Spread mean (std)	Object values	HV mean (std)	Spread mean (std)
Reference	f_1	(0)	—	—	(0)	—	—	(0)	—	—
	f_1, f_2	(5.911,10)	2.4038e–1 (0.00e + 0)	NaN (NaN)	(0,0)	8.6359e–1 (1.95e–1)	1.0000e0 (0.0e0)	(0,0)	**1.0000e + 0 (0.00e + 0)**	**1.0000e + 0 (0.00e + 0)**
	f_1, f_3	(1.359,291)	9.0590e–1 (4.38e–2)	**9.9286e–1 (8.55e–3)**	(30.87,0)	9.4895e–1 (2.13e–2)	1.0125e0 (1.04e–2)	(0,168)	**9.5192e–1 (5.64e–3)**	1.0021e + 0 (2.41e–3)
	f_1, f_4	(2.383,1.2903e4)	**2.1807e–1 (1.26e–3)**	1.1246e + 0 (4.93e–2)	(25.91,1.1728e4)	1.6278e–1 (1.66e–3)	1.0970e0 (1.12e–1)	(0,1.2735e4)	1.4141e–1 (4.15e–3)	9.8356e–1 (2.67e–2)
	f_1, f_5	(0.6684,0.0014)	**8.7126e–1 (7.87e–3)**	1.0000e + 0 (0.00e + 0)	(1.49,0.0013)	8.1085e–1 (2.07e–3)	1.0001e0 (2.19e–4)	(0,0.0014)	5.2399e–1 (0.00e + 0)	**1.0000e + 0 (0.00e + 0)**
	f_1, f_2, f_3	(2.640,12,246)	8.2806e–2 (0.00e + 0)	1.0117e + 0 (0.00e + 0)	(29.62,12,8)	2.5268e–1 (1.82e–1)	1.0014e0 (8.92e–3)	(0,5,325)	**6.7315e–1 (8.42e–2)**	**9.8825e–1 (5.48e–3)**
	f_1, f_2, f_4	(1.421,11,1.3046e4)	1.9907e–2 (3.24e–3)	1.0394e0 (2.70e–2)	(33,12,1.1720e4)	3.5470e–2 (7.71e–3)	1.0094e0 (1.49e–2)	(0,7,1.2970e4)	**6.0659e–2 (5.07e–3)**	**9.2790e–1 (1.40e–2)**
	f_1, f_2, f_5	(2.4584,12,0.0014)	2.2254e–1 (1.97e–1)	1.0049e + 0 (0.00e + 0)	(0,2,0.0013)	**8.5363e–1 (1.24e–1)**	1.0014e0 (1.61e–3)	(0,2,0.0014)	5.5972e–1 (2.26e–2)	**1.0001e + 0 (2.06e–4)**
	f_1, f_3, f_4	(0,302,1.3274e4)	**2.1535e–1 (1.64e–3)**	1.0349e + 0 (4.72e–2)	(17.83,41,1.2121e4)	1.5853e–1 (1.51e–2)	**8.4941e–1 (1.88e–1)**	(0,205,1.2666e4)	1.6521e–1 (1.05e–2)	9.5419e–1 (1.67e–2)
	f_1, f_3, f_5	(0.6640,360,0.0014)	**8.0892e–1 (3.43e–2)**	1.0241e + 0 (1.75e–2)	(21.14,47,0.0013)	4.8172e–1 (3.02e–2)	9.9603e–1 (1.82e–2)	(0,268,0.0014)	8.0753e–1 (9.11e–3)	**9.8024e–1 (6.34e–3)**
	f_1, f_4, f_5	(7.942,1.2581e4,0.0021)	**1.6269e–1 (6.66e–2)**	1.0803e + 0 (1.03e–1)	(31.37,1.1721e4,1)	0.0e0 (0.0e0)	1.0003e0 (5.35e–4)	(0,1.2877e4,0.0015)	5.6339e–2 (5.67e–4)	**9.1906e–1 (4.32e–2)**
	f_1, f_2, f_3, f_4	(5.922,12,326,1.2828e4)	1.8262e–2 (1.31e–3)	1.0699e + 0 (1.04e–1)	(33,12,0,1.1720e4)	3.2180e–2 (4.44e–3)	**9.1108e–1 (1.72e–1)**	(0,11,340,1.2822e4)	**6.4156e–2 (8.13e–3)**	9.2189e–1 (1.62e–2)
	f_1, f_2, f_3, f_5	(1.42,12,337,0.0014)	1.4825e–2 (0.00e + 0)	1.0017e + 0 (6.24e–4)	(16.64,12,120,0.0013)	1.3500e–1 (7.43e–2)	**9.5191e–1 (3.47e–2)**	(0,7,325,0.0014)	**4.7510e–1 (3.44e–2)**	9.8093e–1 (4.14e–3)
	f_1, f_2, f_3, f_4, f_5	(11.18,12,219,1.3222e4,0.0017)	1.4825e–2 (0.00e + 0)	NaN (NaN)	(30,12,3,1.1722e4,0.3)	0.0e0(0.0e0)	1.0038e0 (2.87e–3)	(0,6,330,1.3235e4,0.0014)	**4.6217e–2 (5.50e–3)**	**8.6444e–1 (2.68e–2)**

Table 2: Continued.

Scenario	Objectives	MOCMA			CMOCSO			ONSGA		
		Object values	HV mean (std)	Spread mean (std)	Object values	HV mean (std)	Spread mean (std)	Object values	HV mean (std)	Spread mean (std)
RMIT	f_1	(3)	—	—	(11.80)	—	—	(0)	—	—
	f_1, f_2	(1.395,11)	NaN (NaN)	NaN (NaN)	(13.44,3)	6.5488e–1 (1.82e–1)	**1.0000e0 (4.28e–8)**	(2.8098,2)	**9.9174e–1 (2.61e–2)**	1.0000e + 0 (2.04e–7)
	f_1, f_3	(0,270)	9.4040e–1 (5.37e–2)	1.0000e + 0 (2.84e–7)	(27.54,2)	**9.6954e–1 (4.03e–2)**	1.0000e0 (5.11e–7)	(2.8098,221)	9.5935e–1 (8.59e–3)	**1.0000e + 0 (2.08e–7)**
	f_1, f_4	(15,1.2786e4)	**2.2071e–1 (4.94e–3)**	1.0000e + 0 (3.93e–6)	(23.99,1.1978e4)	2.0728e–1 (9.90e–8)	1.0000e0 (9.23e–6)	(2.8098,1.2717e4)	1.6898e–1 (2.85e–3)	**1.0000e + 0 (1.02e–6)**
	f_1, f_5	(0,0.0015)	**9.8990e–1 (0.00e + 0)**	NaN (NaN)	(2.80,0.0013)	9.7601e–1 (1.04e–7)	1.0000e0 (0.0e0)	(2.8098,0.0014)	3.5227e–1 (0.00e + 0)	**1.0000e + 0 (0.00e + 0)**
	f_1, f_2, f_3	(0,12,370)	8.3465e–2 (0.00e + 0)	NaN (NaN)	(22.79,12,5)	2.6619e–1 (1.35e–1)	**1.0000e0 (4.62e–7)**	(2.8098,10,252)	**7.2741e–1 (7.33e–2)**	1.0000e + 0 (5.85e–8)
	f_1, f_2, f_4	(18,12,1.2281e4)	1.8856e–2 (1.60e–3)	1.0000e + 0 (0.00e + 0)	(231.6,11,1.2330e4)	2.6837e–2 (8.67e–3)	1.0000e0 (9.29e–6)	(2.8098,10,1.290e4)	**6.2050e–2 (7.35e–3)**	**1.0000e + 0 (1.19e–6)**
	f_1, f_2, f_5	(2.8098,11,0.0013)	1.8856e–2 (1.60e–3)	1.0000e + 0 (0.00e + 0)	(5.29,9,0.0013)	2.9463e–1 (8.93e–2)	1.0000e0 (8.61e–8)	(2.8098,2,0.0014)	**4.1372e–1 (4.57e–5)**	**1.0000e + 0 (8.32e–9)**
	f_1, f_3, f_4	(0,290,1.2899e4)	**2.2165e–1 (4.30e–3)**	1.0000e + 0 (6.16e–6)	(11.98,16,1.2015e4)	1.6799e–1 (2.88e–2)	**9.9998e–1 (2.94e–5)**	(2.8098,330,1.2793e4)	1.7119e–1 (6.13e–3)	1.0000e + 0 (1.29e–6)
	f_1, f_3, f_5	(0,336,0.0014)	8.4840e–1 (4.57e–2)	1.0000e + 0 (0.00e + 0)	(98.0,40,0.025)	5.2397e–1 (1.93e–1)	1.0000e0 (1.61e–7)	(2.8098,203,0.0014)	**9.1178e–1 (1.40e–2)**	**1.0000e + 0 (3.93e–7)**
	f_1, f_4, f_5	(0.5854,1.3256e4,0.0016)	2.2071e–1 (4.94e–3)	NaN (NaN)	(65.50,1.1981e4,0.05)	**1.886e–1 (8.40e–2)**	1.0000e0 (4.31e–7)	(2.8098,1.2795e4,0.0014)	5.5812e–2 (2.39e–4)	**9.9999e–1 (6.19e–6)**
	f_1, f_2, f_3, f_4	(18,12,635,1.2654e4)	1.9965e–2 (1.47e–3)	1.0001e + 0 (0.00e + 0)	(31.24,12,3,1.3487e4)	2.9761e–2 (3.59e–3)	**9.9999e–1 (1.35e–5)**	(2.8098,11,340,1.2950e4)	**7.2566e–2 (8.09e–3)**	9.9999e–1 (1.59e–6)
	f_1, f_2, f_3, f_5	(0.3445,12,3060.0018)	1.3816e–1 (0.00e + 0)	NaN (NaN)	(2.80,12,80,0.0013)	1.7650e–1 (1.69e–1)	1.0000e0 (1.43e–6)	(2.8098,8,196,0.0014)	**3.8252e–1 (4.28e–2)**	**1.0000e + 0 (6.65e–7)**
	f_1, f_2, f_3, f_4, f_5	(9.973,11,333,1.3807e4,0.0016)	1.9965e–2 (1.47e–3)	NaN (NaN)	(67.12,12,21,1.1982e4,0.04)	7.6055e–4 (2.41e–3)	1.0000e0 (3.56e–6)	(2.8115,12,350,1.3018e4,0.0014)	**2.5222e–2 (1.96e–3)**	**9.9999e–1 (3.37e–6)**

Note: The best results are shown in bold.

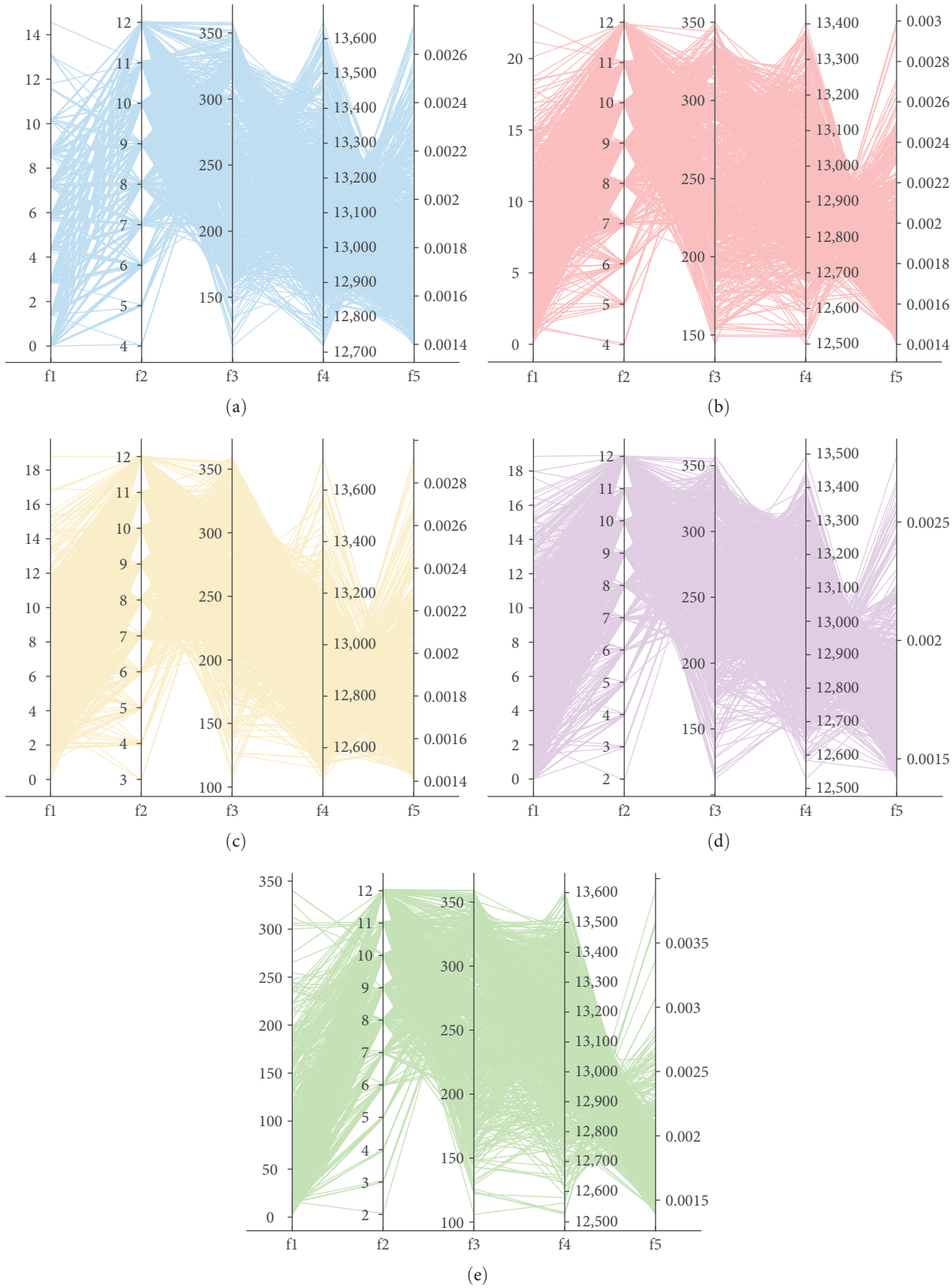


FIGURE 8: The parallel coordinate plot of the Pareto solutions in (a) constant, (b) linear, (c) exponential, (d) reference, and (e) RMIT temperature profile scenario with the f_1, f_2, f_3, f_4, f_5 objective order.

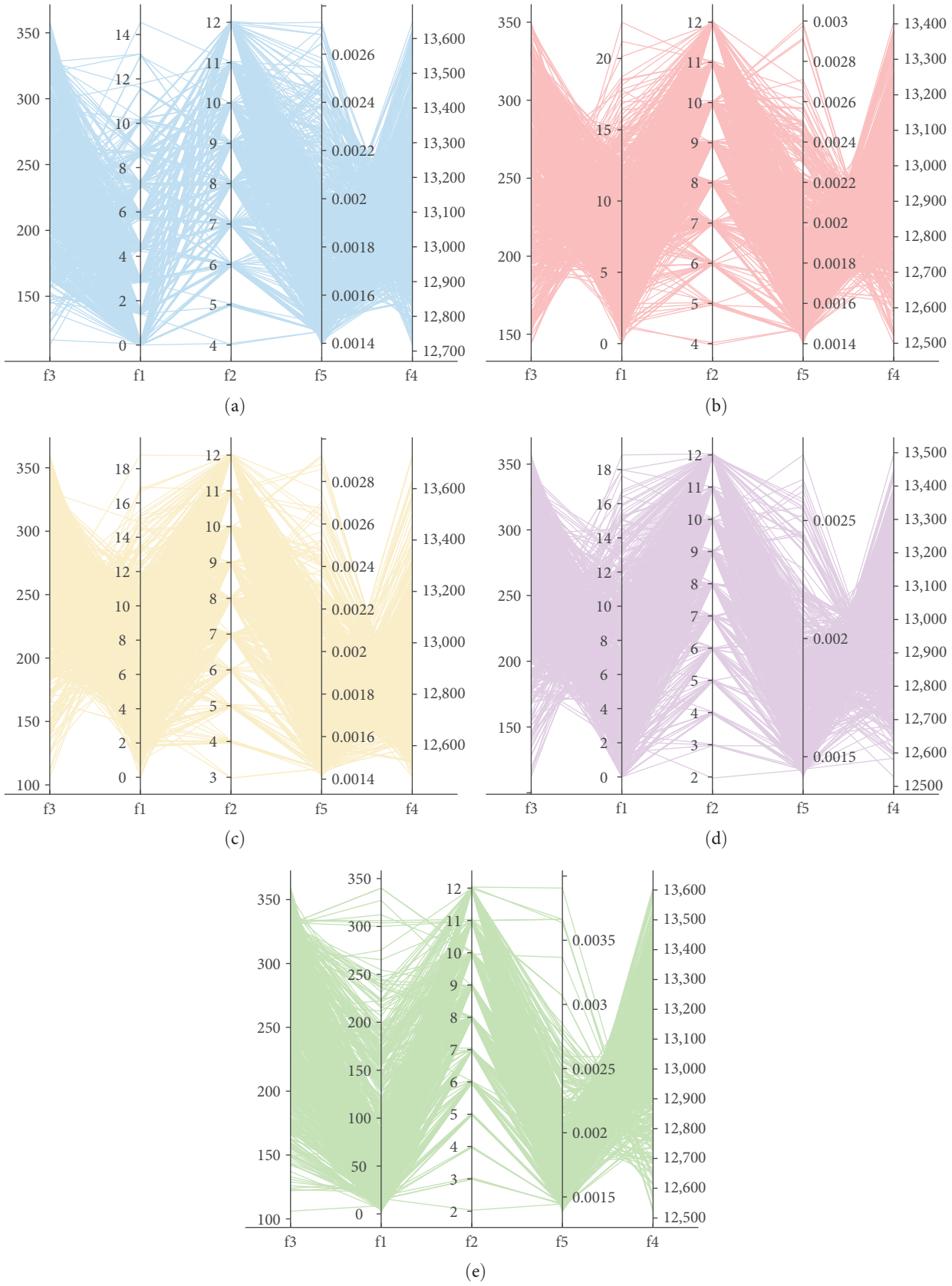


FIGURE 9: The parallel coordinate plot of the Pareto solutions in (a) constant, (b) linear, (c) exponential, (d) reference, and (e) RMIT temperature profile scenario with the f_3, f_1, f_2, f_5, f_4 objective order.

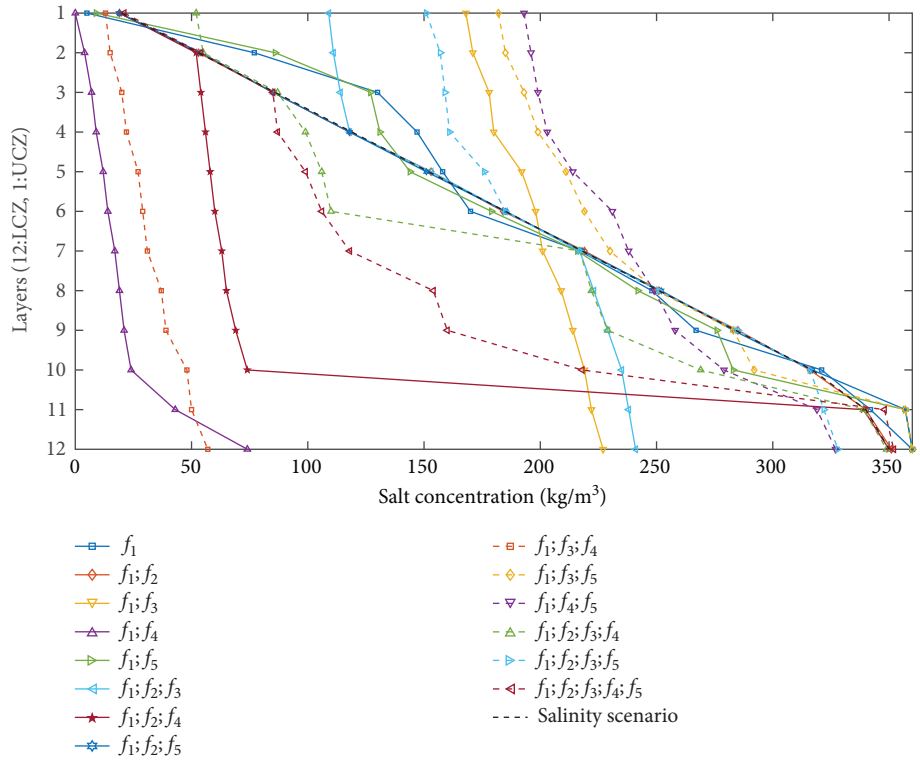


FIGURE 10: The proposed optimal SDPs for Scenario 1: semi-constant temperature profile.

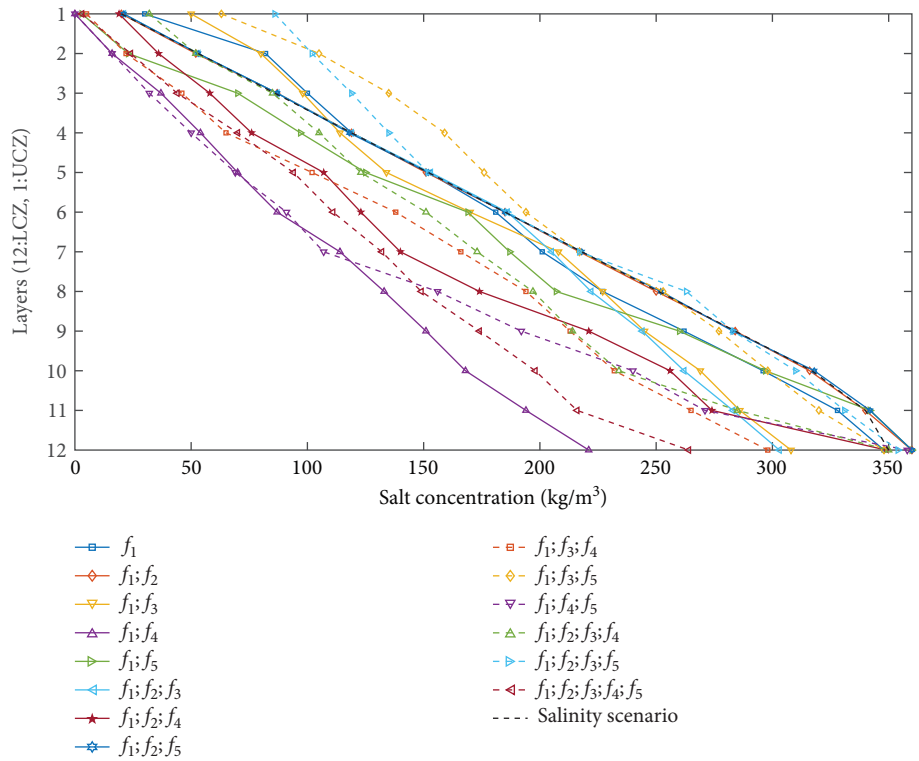


FIGURE 11: The proposed optimal SDPs for Scenario 2: linear temperature profile.

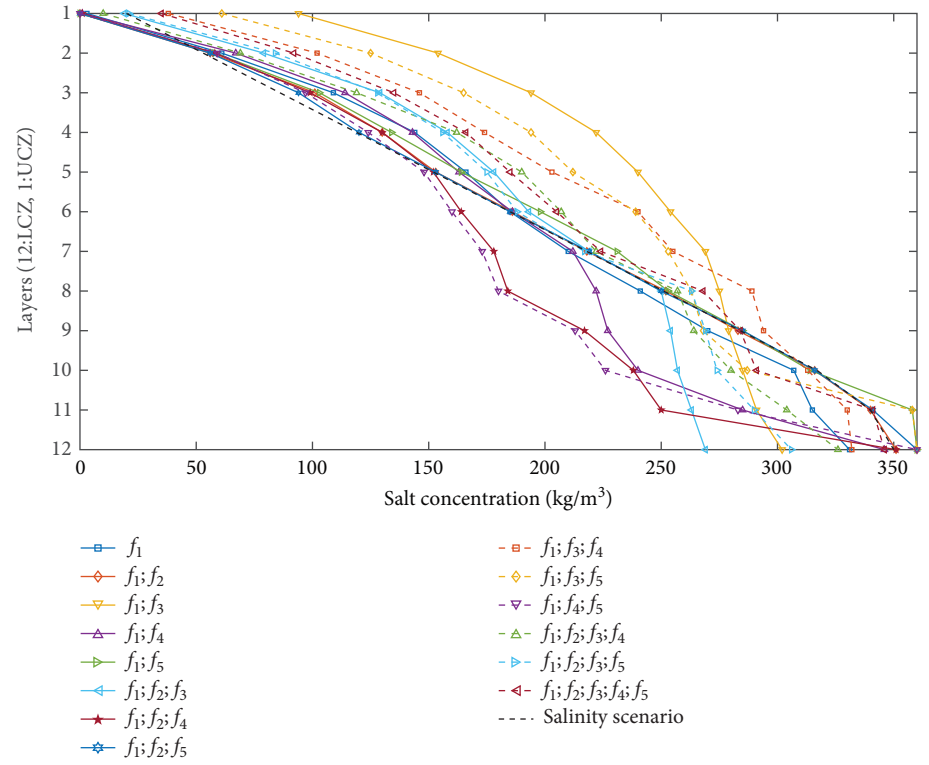


FIGURE 12: The proposed optimal SDPs for Scenario 3: exponential temperature profile.

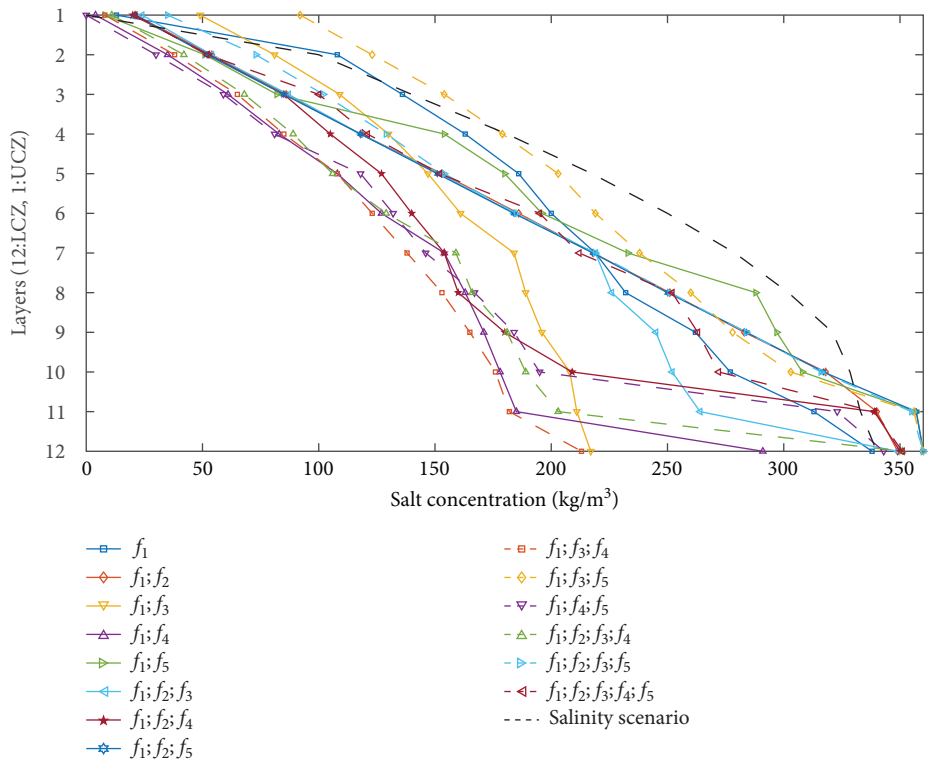


FIGURE 13: The proposed optimal SDPs for Scenario 4: reference temperature profile.

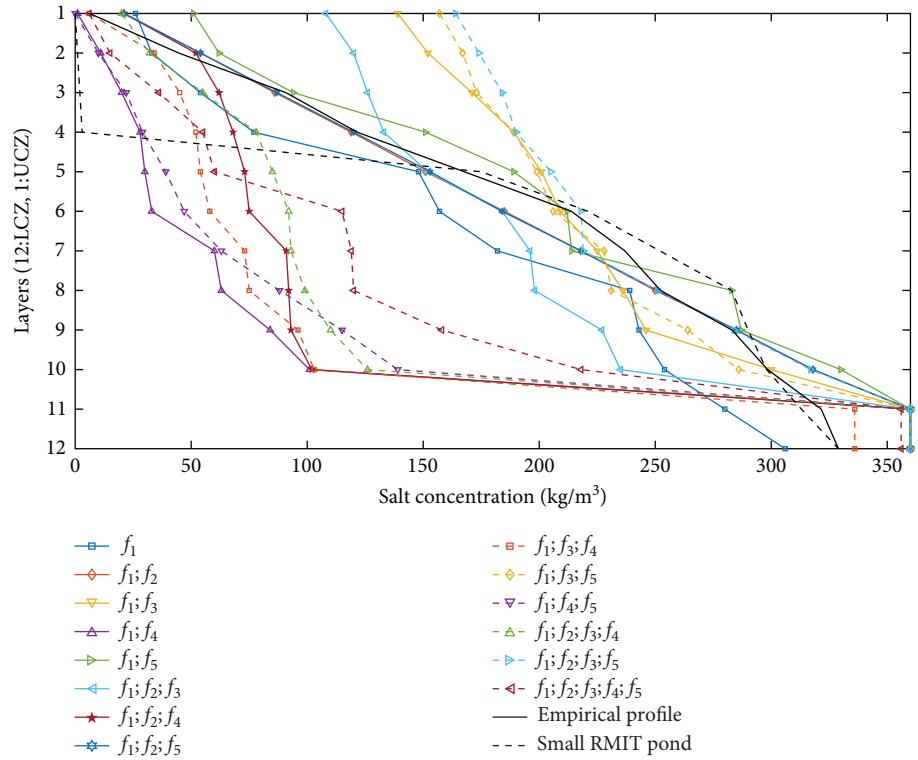


FIGURE 14: The proposed optimal profile for Scenario 5: small RMIT pond with 2 m height.

TABLE 3: Runtime comparison (mean/std) of optimization algorithms.

Sceneraio	MOCMA	CMOCSO	ONSGA
Semi-constant	6.4097e+0 (1.98e–1)	4.9857e+0 (2.20e–1)	**2.7759e+0 (4.19e–2)**
Linear	6.5321e+0 (1.50e–1)	5.0085e+0 (2.16e–1)	**3.3444e+0 (3.12e–2)**
Exponential	6.5153e+0 (1.92e–1)	4.9677e+0 (2.12e–1)	**3.3656e+0 (2.55e–2)**
Reference	7.2061e+0 (1.92e–1)	5.9056e+0 (2.06e–1)	**3.4088e+0 (1.19e–1)**
RMIT	8.1311e+0 (2.83e–1)	9.6711e+0 (1.81e+0)	**3.3702e+0 (4.10e–2)**

Note: ONSGA consistently outperforms the other algorithms, achieving the lowest runtime (bolded) across all scenarios. The best results are shown in bold.

reason for obtaining better results by the proposed ONSGA is the order-preserving operators, since they automatically repair the infeasible solutions. Figures 8 and 9 show the parallel coordinate plots of the Pareto solutions under different temperature profiles and objective orders of f_1, f_2, f_3, f_4, f_5 and f_3, f_1, f_2, f_5, f_4, respectively. Each parallel coordinate representation shows 4 relationships at a time out of the total $\binom{4}{2} = 6$ possible pairwise relationships. The preferred optimal Pareto solutions with importance order of f_1, f_2, f_3, f_4, and f_5 are shown in Figures 10–14 for constant, linear, exponential, reference, and RMIT scenarios, respectively.

Table 3 presents the runtime comparison of the proposed ONSGA with state-of-the-art methods under the most computationally demanding case, that is, optimization across all five objectives (f_1 to f_5). ONSGA consistently outperforms the other algorithms, achieving the lowest runtime (bolded) across all scenarios. The standard deviation (std) for ONSGA is also the smallest, indicating more stable performance compared to MOCMA and CMOCSO.

6.3. Comparison of Optimal Profiles. Table 4 shows the comparison of the empirical SDP, the implemented profile in RMIT [4], and proposed optimal profiles on Scenario 5. The proposed method has the best solutions for determining the salinity of SGSP's layers. In the f_1 column, the optimized solutions have the closer stability criteria to 3. In the f_2 column, SDP considering f_1 and f_2 objectives has maximum similarity to the current salinity. The implemented profile in [4] has lower salt consumption. Also, the heat capacity of the pond, in most cases, is the highest value. The value of f_5 in the case of optimizing objectives f_1, f_2, f_3, and f_5 is the minimum value. The proposed method outperforms the SDP presented in [4] in all criteria except for f_3. Additionally, the optimal profiles yield improvements of up to ~94% in some aspects over the implemented SDPs. Empirical profiles pay more attention to the stability, salt consumption, and heat capacity of the pond, showing less negative values than the control effort and the salinity vicinity.

6.4. Validation of Optimal Profiles. After obtaining optimized profiles, they should be validated to be implemented in

Table 4: Criteria analysis of SDPs against 14 combinations of objective functions for the proposed ONSGA.

Methods	Values of criteria (∼ decline compared to the best)				
	f_1	f_2	f_3	f_4	f_5
Empirical SDP[a]	**0**	**10 (−80%)**	**13,506 (−7%)**	**<u>0.0015</u> (−7%)**	**323.07 (−39%)**
Implemented SDP in [4][a]	**1.18e6**	**12 (−83%)**	**12444**	**0.0018 (−22%)**	**3583 (−94%)**
f_1	0.2	12 (−83%)	13,268 (−6%)	0.0017 (−17%)	280 (−30%)
$f_1; f_2$	**0**	**2**	13,533 (−8%)	**0.0014**	339 (−42%)
$f_1; f_3$	**0**	12 (−83%)	13,781 (−9%)	**0.0014**	221 (−11%)
$f_1; f_4$	**0**	12 (−83%)	<u>12,717</u> (−2%)	**0.0014**	359 (−45%)
$f_1; f_5$	**0**	12 (−83%)	13,654 (−8%)	**0.0014**	309 (−36%)
$f_1; f_2; f_3$	**0**	10 (−80%)	13,528 (−8%)	**0.0014**	252 (−22%)
$f_1; f_2; f_4$	**0**	10 (−80%)	12,910 (−3%)	**0.0014**	339 (−42%)
$f_1; f_2; f_5$	**0**	**2**	13,536 (−8%)	**0.0014**	339 (−42%)
$f_1; f_3; f_4$	**0**	12 (−83%)	12,793 (−2%)	<u>0.0015</u> (−7%)	330 (−40%)
$f_1; f_3; f_5$	**0**	12 (−83%)	13,801 (−9%)	**0.0014**	<u>203</u> (−3%)
$f_1; f_4; f_5$	**0**	12 (−83%)	12,795 (−2%)	**0.0014**	360 (−45%)
$f_1; f_2; f_3; f_4$	**0**	11 (−81%)	12,950 (−3%)	**0.0014**	340 (−42%)
$f_1; f_2; f_3; f_5$	**0**	<u>8</u> (−75%)	13,870 (−10%)	**0.0014**	**196**
$f_1; f_2; f_3; f_4; f_5$	**0**	12 (−83%)	13,018 (−4%)	**0.0014**	350 (−44%)

Note: The out-of-scope criteria for each scenario are calculated after optimization. The bold and underlined entries indicate the best and second-best values, respectively, for each objective function, excluding out-of-scopes.

[a]Note that the values are calculated based on the reported salinity profile.

SGSPs. Figure 15 shows the stability criteria of 11 boundaries resulting from 12 layers for the last two scenarios. As it shows, all of the obtained SDPs guarantee the stability of the SGSP. Because the temperature profile is constant at the bottom of Scenario 5, the stability criteria for columns 6–11 are not calculated, and we are content with the increasing salinity in these layers.

7. Discussion

Looking at Figure 8, we can see that the adjacent objectives conflict in all scenarios, especially f_4 and f_5, because the number of intersection lines between them is greater. Additionally, many solution sets do not span the full range of values for each adjacent objective, suggesting limited coverage of extremum points. Figure 9 shows that adjacent objectives (with the order of f_3, f_1, f_2, f_5, f_4) are less conflicting. Looking at both figures, f_1 and f_3 show a clear inverse relationship—when f_1 values are low, f_3 values tend to be high, and vice versa. The RMIT profiles (Figures 8 and 9) show distinct patterns compared to others, implying that temperature dynamics significantly impact objective relationships. Specifically, f_4–f_5 and f_1–f_3 show line intersections than others, indicating weaker trade-offs. Each colored scenario (constant, linear, exponential, reference, RMIT) produces different trade-off patterns, showing how problem conditions affect the available solutions. These interpretations are consistent with prior knowledge, as shown in Figure 3, confirming that different problem formulations create different trade-off landscapes for decision-makers to navigate.

Figures 10–13 show that the overall shapes of profiles are similar to their related temperature profiles. f_2 moves the optimal SDP closer to the current profile. f_3 draws the profile to the left of the salinity axis to decrease salt consumption and increase the transparency of water. Criterion f_4 maximizes the heat capacity of layers. f_5 causes the salinity of the layers to come as close together as possible. The SDP, considering the control effort and stability, is the closest profile to the reference profile. In Scenario 5, R_p can not be calculated because of the constant temperature of layers 6 – 12. For example, in Figure 13, the red-starred profile is the optimal SDP considering stability, minimum control effort, and maximum heat capacity.

In Figure 14, the proposed optimal SDP for various desired criteria is shown. The results show that if we consider the stability and energy storage, the proposed optimal profile is close to the empirical profile used for RMIT's pond. The orange and light blue (dashed) profiles are useful when the evaporation pond, and consequently, saline water is available. If the pond site is far from the salt source, the purple triangle and the red-starred profiles are better.

In the case of maximum energy storage, the lower layer in the optimal profile has high salinity because the heat capacity of saline water is more than that of freshwater. Also, the profile shown by the yellow curve is better when we want to store more energy, and salt resources are limited.

As mentioned in the literature, for improving efficiency, we sometimes extract heat from the NCZ [2, 15]. Also, the boundary of LCZ and NCZ often has the maximum temperature. Therefore, we should keep the storage capacity of part of the NCZ high, which could be ensured by increasing its salinity. In Figure 14, the salinities of the lower layer and bottom of NCZ in the proposed optimal SDP remain high to store more energy.

Considering the MOSO problem optimization, although we examine many algorithms, the MOCMA and CMOCSO

Sec.		Boundaries index										
		1	2	3	4	5	6	7	8	9	10	11
Semi-constant	f_1	116.61	82.04	25.54	16.39	17.76	68.83	44.23	26.70	74.83	28.36	24.07
	$f_1; f_2$	54.58	50.29	50.85	52.87	45.88	50.87	45.60	47.59	42.60	32.48	14.80
	$f_1; f_3$	4.49	10.33	3.01	17.55	8.77	4.43	11.60	7.26	7.24	4.38	7.22
	$f_1; f_4$	6.49	4.86	3.23	4.85	3.23	4.83	3.22	3.22	4.82	30.41	49.02
	$f_1; f_5$	124.39	63.13	6.07	19.53	52.06	53.92	37.14	47.83	9.78	101.59	4.14
	$f_1; f_2; f_3$	3.10	4.61	6.11	52.87	47.36	46.49	8.66	8.63	8.60	4.35	4.35
	$f_1; f_2; f_4$	52.97	3.17	3.17	3.17	3.16	4.72	3.16	6.27	7.81	410.75	14.80
	$f_1; f_3; f_4$	3.23	8.05	3.22	8.02	3.21	3.20	9.57	3.19	14.28	3.18	11.04
	$f_1; f_3; f_5$	4.46	11.70	8.77	17.37	11.54	15.76	28.37	46.22	12.49	88.82	4.14
	$f_1; f_2; f_3; f_4$	4.74	50.23	18.50	10.74	6.14	162.26	7.23	10.05	56.65	96.85	13.47
	$f_1; f_2; f_3; f_5$	8.96	3.04	3.03	22.14	13.21	46.49	48.50	47.59	42.60	8.24	8.22
	$f_1; f_2; f_3; f_4; f_5$	52.91	48.69	3.13	18.50	10.74	18.30	54.37	8.94	85.40	184.97	5.47
Linear	f_1	10.15	3.44	3.60	6.15	5.32	3.65	4.67	6.16	5.89	5.46	3.43
	$f_1; f_2$	6.28	6.75	6.06	5.96	6.23	5.76	5.85	5.93	5.50	4.10	3.41
	$f_1; f_3$	5.80	3.44	3.04	3.76	6.65	6.87	3.43	3.23	4.23	3.01	3.82
	$f_1; f_4$	3.17	4.13	3.32	3.10	3.27	5.11	3.57	3.36	3.15	4.74	4.85
	$f_1; f_5$	3.96	9.21	5.17	5.28	8.14	3.31	3.64	9.37	6.40	7.61	3.08
	$f_1; f_2; f_3$	6.28	6.56	6.06	6.33	6.04	3.46	3.09	3.93	3.21	3.69	3.49
	$f_1; f_2; f_4$	3.34	4.29	3.48	5.91	3.03	3.19	6.26	8.46	6.18	3.19	12.71
	$f_1; f_3; f_4$	3.36	4.71	3.69	7.08	6.76	5.17	5.10	3.45	3.42	5.80	5.70
	$f_1; f_3; f_5$	8.05	5.64	4.46	3.15	3.30	4.15	6.37	4.22	3.67	3.80	4.74
	$f_1; f_2; f_3; f_4$	3.91	6.37	3.81	3.40	5.21	4.05	4.37	3.10	3.60	8.88	10.97
	$f_1; f_2; f_3; f_5$	3.06	3.22	3.02	3.18	6.22	5.58	8.11	3.52	4.66	3.62	3.91
	$f_1; f_2; f_3; f_4; f_5$	4.16	3.92	5.04	4.60	3.23	3.94	3.18	4.60	4.37	3.27	8.43
Exponential	f_1	3.11	3.77	4.04	3.77	4.81	9.37	17.09	23.70	44.36	14.33	42.19
	$f_1; f_2$	3.01	3.54	3.37	3.96	8.34	11.95	18.65	25.17	39.30	42.45	28.67
	$f_1; f_3$	3.09	3.03	3.13	3.00	3.46	5.48	3.32	3.34	7.29	10.91	29.44
	$f_1; f_4$	3.54	3.68	3.35	3.43	5.81	9.73	5.59	4.23	16.05	82.91	163.09
	$f_1; f_5$	3.11	3.54	3.59	5.13	8.54	12.24	12.03	25.96	34.51	77.76	5.32
	$f_1; f_2; f_3$	3.20	3.89	3.33	3.41	3.80	9.70	17.00	3.37	3.76	11.07	16.37
	$f_1; f_2; f_4$	3.06	3.23	3.60	3.79	3.09	5.35	3.45	27.84	26.00	22.23	275.37
	$f_1; f_2; f_5$	3.01	3.00	3.03	5.68	8.09	12.70	17.00	28.43	36.88	44.21	49.37
	$f_1; f_3; f_4$	3.38	3.40	3.19	4.87	9.10	5.51	18.27	4.11	22.56	30.17	5.39
	$f_1; f_3; f_5$	3.34	3.06	3.28	3.04	6.64	5.16	5.48	4.16	22.88	127.28	5.32
	$f_1; f_2; f_3; f_4$	3.15	3.91	4.92	4.73	4.27	5.60	19.15	5.78	19.33	43.30	58.30
	$f_1; f_2; f_3; f_5$	3.35	3.42	3.22	3.25	3.31	10.83	25.20	4.96	6.13	29.01	42.77
	$f_1; f_2; f_3; f_4; f_5$	3.01	3.33	3.54	3.23	5.01	7.07	24.01	12.95	8.45	89.48	13.13
Reference	f_1	9.57	3.33	4.10	4.31	3.48	4.00	9.34	14.20	7.06	49.48	33.74
	$f_1; f_2$	3.31	3.89	4.97	6.66	8.20	6.71	23.52	14.98	16.09	29.63	12.09
	$f_1; f_3$	3.18	3.37	3.25	3.25	3.54	4.91	3.72	3.47	5.86	4.40	8.66
	$f_1; f_4$	3.14	3.19	3.47	4.86	4.86	5.85	6.72	4.00	3.50	10.26	153.32
	$f_1; f_5$	4.34	3.78	11.15	4.88	3.97	7.71	38.79	4.24	5.13	66.15	4.12
	$f_1; f_2; f_3$	3.01	4.01	4.81	6.47	7.96	7.34	5.08	9.07	3.39	16.82	117.48
	$f_1; f_2; f_4$	3.32	3.89	3.13	4.24	3.32	3.05	4.51	9.88	14.12	185.38	14.74
	$f_1; f_2; f_5$	3.11	4.02	5.13	6.28	8.21	7.13	22.81	15.92	15.17	53.77	4.12
	$f_1; f_3; f_4$	3.03	3.31	3.15	4.47	3.85	3.29	11.24	5.98	5.46	8.82	44.99
	$f_1; f_3; f_5$	3.02	3.66	3.77	4.46	3.93	3.97	15.55	8.45	11.57	71.74	5.43
	$f_1; f_2; f_3; f_4$	3.13	3.18	3.30	3.31	5.88	6.49	5.23	7.41	3.97	20.30	211.77
	$f_1; f_2; f_3; f_5$	3.79	3.50	4.17	4.75	7.70	7.13	22.80	15.44	14.72	52.46	6.75
	$f_1; f_2; f_3; f_4; f_5$	3.11	5.83	3.26	5.90	10.66	3.60	28.58	5.24	4.29	92.22	16.07
RMT	f_1	2.8	11.15	12.06	109.90	13.42	—	—	—	—	—	—
	$f_1; f_2$	13.23	16.77	16.98	48.32	50.38	—	—	—	—	—	—
	$f_1; f_3$	4.92	9.45	8.87	17.47	11.60	—	—	—	—	—	—
	$f_1; f_4$	4.06	4.84	4.29	3.20	4.80	—	—	—	—	—	—
	$f_1; f_5$	4.35	16.70	29.15	56.30	33.38	—	—	—	—	—	—
	$f_1; f_2; f_3$	4.62	3.08	3.58	28.49	47.39	—	—	—	—	—	—
	$f_1; f_2; f_4$	12.43	5.27	3.16	7.82	3.14	—	—	—	—	—	—
	$f_1; f_2; f_5$	13.23	17.30	16.97	49.80	45.89	—	—	—	—	—	—
	$f_1; f_3; f_4$	11.33	5.85	3.71	3.17	6.31	—	—	—	—	—	—
	$f_1; f_3; f_5$	3.77	3.02	7.89	14.57	10.18	—	—	—	—	—	—
	$f_1; f_2; f_3; f_4$	4.82	12.22	12.06	10.87	10.82	—	—	—	—	—	—
	$f_1; f_2; f_3; f_4; f_5$	3.64	11.27	10.07	7.87	86.04	—	—	—	—	—	—

FIGURE 15: Stability criteria (R_p) of scenarios. Missing values could not be calculated because of equal layers' temperatures, leading to a zero division.

are comparable with the proposed method because they are powerful algorithms for balancing the minimization of objectives while satisfying constraints. A deeper analysis of their algorithmic operators reveals that order-preserving mechanisms play a fundamental role in ensuring convergence. These mechanisms inherently correct infeasible solutions by maintaining solution feasibility throughout the optimization process. This observation aligns with the well-established understanding that evolutionary operators are among the most critical components of optimization algorithms, as they directly influence solution quality, convergence behavior, and constraint handling.

Here, we focus on the SGSP optimization problem definition and show that the proposed optimized salinity profiles are consistent with an empirical profile in the literature. As mentioned before, the decision variables of this problem could include the pond specification, layer thickness, and energy harvesting schedule, among others.

Here, we implicitly define SGSP's behavior during operation in the objectives by including expert knowledge to have a general view and bypass specific numerical modeling. Specifically, it is assumed that the layers' salinity could reach the new salinities without temperature changes. In other words, the temperature before and after the salinity change is the same, while some temperature change is inevitable due to salinity injection/suction. Also, vertical double diffusion is considered to be a basis for objective functions. However, the objectives may need a little modification to include more real dynamics of SGSPs, such as time-lasting profiles, obtained by explicitly including numerical modeling of SGSPs in the objectives.

The main limitation of this article is that we use algorithms with standard settings and a maximum of 10 runs because of resource limitations, while further investigation could be beneficial.

8. Conclusion

SDPs have a considerable effect on the stability and performance of SGSPs, as confirmed by several credible works in this domain. However, the current research has primarily addressed SGSP's stability only. Here, we take the next step toward SGSP design by including other critical in situ performance measures. Specifically, we concurrently consider several desirable objectives, in addition to stability, such as minimizing the control effort, salt consumption, and diffusivity while maximizing robustness, radiation transmission, and thermal efficiency as a MOSO problem. In this process, we focus on designing an optimal salinity gradient profile due to its considerable effect on the pond's efficiency and resiliency. What is essential to consider in this process is that these objectives are often in conflict. In other words, achieving the best thermal efficiency does not necessarily lead to a more stable pond and vice versa.

Subsequently, we propose a multi-objective search process for these objectives. What complicates this design process is that the search space becomes exponentially more complex and possibly non-convex as the number of objectives increases. For this purpose, we propose a many-objective integer-based ONSGA that benefits from novel discrete mutation and crossover operators. Specifically, we have studied the SDP design for a 12-layer SGSP under five temperature scenarios. The result shows that these criteria could substantially affect the sustainability of the SDP.

The resulting strategy, we hope, will enable SGSP sustainable design based on in-situ usage criteria and environmental circumstances of a pond. It is a soft step that complements the hardware developments to improve pond-specific SDPs. The proposed method is a general framework in which any heuristic criteria could be added to the optimization problem as an objective. Of course, as the number of objectives increases, solving the problem also becomes computationally more complex. We hope to address this complexity in the future by studying function approximation and surrogate techniques to reduce computations. Furthermore, we plan to study SGSP's behavior and energy loss under environmental disturbances. Finally, the comparative analysis of optimization algorithms' adaptability and scalability within control loop implementations under disturbance and dynamic changes is recommended.

Nomenclature

α: Coefficient of thermal expansion ($°C^{-1}$)
β: Coefficient of saline expansion (m^3/kg)
T: Temperature (K)
S: Salinity (g/kg)
R_ρ: Stability coefficient of the system
N: Number of the SGSP layers
ρ: Density (kg/m^3)
C_i: Salt concentration of the layer i (kg/m^3)
γ: Desired R_ρ of each interfaces
C_p: heat capacity of saline water (J/kg°C).

Data Availability Statement

All data to reproduce the results of this research are provided in the manuscript. All codes are available at https://github.com/Hamed-Rafiei/SGSP-ONSGA.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Funding

No funding was received for this manuscript.

Acknowledgments

The authors would like to thank Mohammed Bawahab and Hosam Faqeha from RMIT University for their valuable insight and details of the SGSP at the RMIT Laboratory.

Endnotes

[1]Non-dominated sorting genetic algorithm.
[2]Multiobjective evolutionary algorithm based on decomposition.

[3]Scalable small subpopulations based covariance matrix adaption evolution strategy.

[4]MOEA/D with adaptive weight adjustment.

[5]MOEA/D based on differential evolution.

[6]Multi-objective particle swarm optimization.

[7]Novel multi-objective particle swarm optimization.

[8]Scalable small subpopulations based covariance matrix adaption evolution strategy.

[9]For more details about variable definitions and other default values, readers are strongly recommended to check the PlatEMO documentation.

References

[1] O. K. Ahmed, S. Algburi, R. W. Daoud, H. Khaoula, K. I. Hamada, and H. N. Shubat, "Hybrid Salinity Gradient Solar Ponds: A Short Review," *Renewable and Sustainable Energy Reviews* 226 (2026): 116418.

[2] A. Alcaraz, C. Valderrama, J. Cortina, A. Akbarzadeh, and A. Farran, "Enhancing the Efficiency of Solar Pond Heat Extraction by Using Both Lateral and Bottom Heat Exchangers," *Solar Energy* 134 (2016): 82–94.

[3] A. Abdullah and K. Lindsay, "Assessing the Maximum Stability of the Nonconvective Zone in a Salinity-Gradient Solar Pond," *Journal of Solar Energy Engineering* 139, no. 4 (2017): 041010.

[4] H. Faqeha, M. Bawahab, Q. L. Vet, A. Faghih, A. Date, and A. Akbarzadah, "An Experimental Study to Establish a Salt Gradient Solar Pond (SGSP)," *Energy Procedia* 160 (2019): 239–245.

[5] M. C. Hicks and P. Golding, "One-Dimensional Transient Finite Difference Model of an Operational Salinity Gradient Solar Pond," in *The Fourth Annual Thermal and Fluids Analysis Workshop*, (1992).

[6] M. Farrokhi, M. Jaefarzadeh, M. Bawahab, H. Faqeha, and A. Akbarzadeh, "Integration of a Solar Pond in a Salt Work in Sabzevar in Northeast Iran," *Solar Energy* 244 (2022): 115–125.

[7] H. Rafiei, M.-R. Akbarzadeh-T, N. Pariz, and A. Akbarzadeh, "Expert Systems and the Prospects of Artificial Intelligence for the Automatic Supervisory Control of Salinity Gradient Solar Ponds," *Solar Energy* 246 (2022): 281–293.

[8] T. Newell, R. Cowie, J. Upper, M. Smith, and G. Cler, "Construction and Operation Activities at the University of Illinois Salt Gradient Solar Pond," *Solar Energy* 45, no. 4 (1990): 231–239.

[9] F. Zangrando and L. A. Bertram, "The Effect of Variable Stratification on Linear Doubly Diffusive Stability," *Journal of Fluid Mechanics* 151 (1985): 55–79.

[10] I. Walton, "Double-Diffusive Convection with Large Variable Gradients," *Journal of Fluid Mechanics* 125 (1982): 123–135.

[11] M. Sodha and A. Kumar, "Stability Analysis of Double-Diffusive Solar Ponds With Non-Constant Temperature and Salinity Gradients," *Energy Conversion and Management* 25, no. 4 (1985): 463–468.

[12] Y. Katti and P. Bansal, "Stability Analysis of Solar Ponds With Non-Constant Temperature Gradients," *Energy Conversion and Management* 27, no. 1 (1987): 91–97.

[13] A. Kirkpatrick, R. Gordon, and D. Johnson, "Double Diffusive Natural Convection in Solar Ponds With Nonlinear Temperature and Salinity Profiles," *Journal of Solar Energy Engineering* 108, no. 3 (1986): 214–218.

[14] E. Busquets, V. Kumar, J. Motta, R. Chacon, and H. Lu, "Thermal Analysis and Measurement of a Solar Pond Prototype to Study the Non-Convective Zone Salt Gradient Stability," *Solar Energy* 86, no. 5 (2012): 1366–1377.

[15] J. Leblanc, A. Akbarzadeh, J. Andrews, H. Lu, and P. Golding, "Heat Extraction Methods from Salinity-Gradient Solar Ponds and Introduction of a Novel System of Heat Extraction for Improved Efficiency," *Solar Energy* 85, no. 12 (2011): 3103–3142.

[16] K. Sreenivas, J. H. Arakeri, and J. Srinivasan, "Modeling the Dynamics of the Mixed Layer in Solar Ponds," *Solar Energy* 54, no. 3 (1995): 193–202.

[17] R. B. Mansour, C. T. Nguyen, and N. Galanis, "Numerical Study of Transient Heat and Mass Transfer and Stability in a Salt-Gradient Solar Pond," *International Journal of Thermal Sciences* 43, no. 8 (2004): 779–790.

[18] M. Montalà, J. Cortina, A. Akbarzadeh, and C. Valderrama, "Stability Analysis of an Industrial Salinity Gradient Solar Pond," *Solar Energy* 180 (2019): 216–225.

[19] N. Kaushika, P. Avanti, M. Arulanantham, and T. Singh, "Double Diffusive Zone-Boundary Migration in Solar Ponds," *International Journal of Energy Research* 20, no. 12 (1996): 1027–1035.

[20] H. Wang, M. Xie, and W.-C. Sun, "Nonlinear Dynamic Behavior of Non-Convective Zone in Salt Gradient Solar Pond," *Solar Energy* 85, no. 9 (2011): 1745–1757.

[21] A. A. Hill and M. Carr, "The Influence of a Fluid-Porous Interface on Solar Pond Stability," *Advances in Water Resources* 52 (2013): 1–6.

[22] M. Giestas, A. Joyce, and H. Pina, "The Influence of Non-Constant Diffusivities on Solar Ponds Stability," *International Journal of Heat and Mass Transfer* 40, no. 18 (1997): 4379–4391.

[23] K. Hanjalić and R. Musemić, "Modeling the Dynamics of Double-Diffusive Scalar Fields at Various Stability Conditions," *International Journal of Heat and Fluid Flow* 18, no. 4 (1997): 360–367.

[24] R. B. Mansour, C. T. Nguyen, and N. Galanis, "Transient Heat and Mass Transfer and Long-Term Stability of a Salt-Gradient Solar Pond," *Mechanics Research Communications* 33, no. 2 (2006): 233–249.

[25] D. Akrour, M. Tribeche, and D. Kalache, "A Theoretical and Numerical Study of Thermosolutal Convection: Stability of a Salinity Gradient Solar Pond," *Thermal Science* 15, no. 1 (2011): 67–80.

[26] M. Alanazi, A. Alanazi, Y. A. Alamri, E. A. Ali, S. Alqahtani, and A. E. Anqi, "Thermal Optimization of Salinity Gradient Solar Ponds in Natural Salt Lakes for Simultaneous Renewable Heat Storage and Artemia Cultivation," *Solar Energy* 301 (2025): 113955.

[27] P. Bansal and Y. Katti, "Salt Concentration Profiles for Marginal Stability in Solar Ponds," *Energy Conversion and Management* 26, no. 2 (1986): 227–232.

[28] Y. Keren, H. Rubin, J. Atkinson, M. Priven, and G. Bemporad, "Theoretical and Experimental Comparison of Conventional and Advanced Solar Pond Performance," *Solar Energy* 51, no. 4 (1993): 255–270.

[29] B. Singh, L. Tan, A. Date, and A. Akbarzadeh, "Power Generation From Salinity Gradient Solar Pond Using

Thermoelectric Generators for Renewable Energy Application," in *2012 IEEE International Conference on Power and Energy (PECon)*, (IEEE, 2012): 89–92.
[30] A. Akbarzadeh, P. Johnson, and R. Singh, "Examining Potential Benefits of Combining a Chimney With a Salinity Gradient Solar Pond for Production of Power in Salt Affected Areas," *Solar Energy* 83, no. 8 (2009): 1345–1359.
[31] M. O. Dah, M. Ouni, A. Guizani, and A. Belghith, "The Influence of the Heat Extraction Mode on the Performance and Stability of a Mini Solar Pond," *Applied Energy* 87, no. 10 (2010): 3005–3010.
[32] S. Ganguly, A. Date, and A. Akbarzadeh, "On Increasing the Thermal Mass of a Salinity Gradient Solar Pond With External Heat Addition: A Transient Study," *Energy* 168 (2019): 43–56.
[33] H. Bryant and I. Colbeck, "Solar Pond for London," *Solar Energy* 19 (1977).
[34] P. Lund and J. Routti, "Feasibility of Solar Pond Heating for Northern Cold Climates," *Solar Energy* 33, no. 2 (1984): 209–215.
[35] N. Li, C. H. Zhang, and W. C. Sun, "Transmission and Temperature for the Surface Icing Solar Pond," *Applied Mechanics and Materials* 291 (2013): 105–108.
[36] A. Alcaraz, M. Montalà, C. Valderrama, J. Cortina, A. Akbarzadeh, and A. Farran, "Thermal Performance of 500 m^2 Salinity Gradient Solar Pond in Granada, Spain Under Strong Weather Conditions," *Solar Energy* 171 (2018): 223–228.
[37] Z. Zhou, X. Yan, and D. Li, "Hydrophysics-NSGA-II: A Physics-Informed Multi-Objective Evolutionary Algorithm for Groundwater Contaminant Emergency Scheduling," *Swarm and Evolutionary Computation* 99 (2025): 102210.
[38] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints," *IEEE Transactions on Evolutionary Computation* 18, no. 4 (2014): 577–601.
[39] T. P. França, T. F. de Queiroz Lafetá, L. G. A. Martins, and G. M. B. de Oliveira, "A Comparative Analysis of Moeas Considering Two Discrete Optimization Problems," in *2017 Brazilian Conference on Intelligent Systems (BRACIS)*, (IEEE, 2017): 402–407.
[40] M. Davoodi, F. Panahi, A. Mohades, and S. N. Hashemi, "Multi-Objective Path Planning in Discrete Space," *Applied Soft Computing* 13, no. 1 (2013): 709–720.
[41] T. D. Gwiazda, *Crossover for Single-Objective Numerical Optimization Problems*, 1 (Tomasz Gwiazda, 2006).
[42] T. D. Gwiazda, "Genetic Algorithms Reference Volume Ii Mutation Operator for Numerical Optimization Problems," in *Genetic Algorithms Reference*, (Simon and Schuster, 2007).
[43] R. Landa-Becerra, L. V. Santana-Quintero, and C. A. C. Coello, "Knowledge Incorporation in Multi-Objective Evolutionary Algorithms," in *Multi-Objective Evolutionary Algorithms for Knowledge Discovery from Databases*, (Springer, 2008): 23–46.
[44] J.-H. Yi, S. Deb, J. Dong, A. H. Alavi, and G.-G. Wang, "An Improved NSGA-III Algorithm with Adaptive Mutation Operator for Big Data Optimization Problems," *Future Generation Computer Systems* 88 (2018): 571–585.
[45] J.-H. Yi, L.-N. Xing, G.-G. Wang, et al., "Behavior of Crossover Operators in NSGA-III for Large-Scale Optimization Problems," *Information Sciences* 509 (2020): 470–487.
[46] C. Yang, W. Liu, Q. Chen, H. Wang, and B. Niu, "Incorporating Exhibitor Competitiveness and Price Sensitivity: A Staged Crossover Hybrid Genetic Algorithm (schga) for Optimizing Booth Pricing," *Swarm and Evolutionary Computation* 99 (2025): 102180.
[47] A. Saxena, E. Cuce, A. Kabeel, M. Abdelgaied, and V. Goel, "A Thermodynamic Review on Solar Stills," *Solar Energy* 237 (2022): 377–413.
[48] K. Karunamurthy, R. Manimaran, and M. Chandrasekar, "Prediction of Solar Pond Performance Parameters Using Artificial Neural Network," *International Journal of Computer Aided Engineering and Technology* 11, no. 2 (2019): 141–150.
[49] M. Giestas, H. Pina, and A. Joyce, "The Influence of Radiation Absorption on Solar Pond Stability," *International Journal of Heat and Mass Transfer* 39, no. 18 (1996): 3873–3885.
[50] C. Angeli and E. Leonardi, "A One-Dimensional Numerical Study of the Salt Diffusion in a Salinity-Gradient Solar Pond," *International Journal of Heat and Mass Transfer* 47, no. 1 (2004): 1–10.
[51] M. R. Jaefarzadeh, "Thermal Behavior of a Large Salinity-Gradient Solar Pond in the City of Mashhad," *Iranian Journal of Science and Technology Transactions of Mechanical Engineering* 29 (2005): 219–229.
[52] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization," *IEEE Computational Intelligence Magazine* 12, no. 4 (2017): 73–87.