Highlights

A Novel Approach of Many-Objective Particle Swarm Optimization with Cooperative Agents based on an Inverted Generational Distance Indicator

Najwa Kouka,Fatma BenSaid,Raja Fdhila,Rahma Fourati,Amir Hussain,Adel M. Alimi

- The IGD indicator is employed as selection criteria to face the challenge of the large ratio of non-dominated solutions that exist in MaPs.
- A multi-agent system is used to model the intelligent behavior of multi-swarms.
- A new strategy of knowledge sharing based on automated negotiation is proposed.
- Empirical studies demonstrate the effectiveness of the proposed algorithm in solving MaOPs.

A Novel Approach of Many-Objective Particle Swarm Optimization with Cooperative Agents based on an Inverted Generational Distance Indicator

Najwa Kouka[a,*,1], Fatma BenSaid[a], Raja Fdhila[a], Rahma Fourati[a], Amir Hussain[b] and Adel M. Alimi[a,c]

[a]Research Groups in Intelligent Machines, National Engineering School of Sfax (ENIS), University of Sfax, , BP 1173, 3038, Sfax, Tunisia
[b]School of Computing, Edinburgh Napier University, Edinburgh,, United Kingdom.
[c]Department of Electrical and Electronic Engineering Science, Faculty of Engineering and the Built Environment, University of Johannesburg, South Africa

ARTICLE INFO

Keywords:
Many−objective optimization problems
Particle swarm optimization
Multi-agent system
Automated negotiation

ABSTRACT

Most evolutionary algorithms, including particle swarm optimization (PSO), use Pareto dominance as a major selection criterion and face significant challenges when dealing with many-objective problems. To tackle this issue, this paper proposes a novel algorithm, termed: Many-Objective PSO with Cooperative Agents (MaOPSO-CA). This exploits an Inverted Generational Distance (IGD) indicator in two innovative ways: firstly, as a leader selection method to select the preferable solution in terms of convergence and diversity, and, secondly, as an archiving method to decide which non-dominated solutions are kept in a bounded archive. The proposed strategy significantly promotes selection pressure toward the Pareto front. The results indicate that the IGD-based selection circumvents the issue of a large ratio of non-dominated solutions that exist in MaOPs. Moreover, a multi-swarm is investigated and modeled as a Multi-Agent System (MAS), so that knowledge sharing among different sub-swarms is easily improved through automated negotiation. The effectiveness of our proposed algorithm is validated with numerous experimental studies in solving 110 benchmark testing instances with up to twenty objectives. Experimental results demonstrate the effectiveness of the new algorithm compared to recent state-of-the-art methods. Finally, the application of MaOPSO-CA to a challenging, real-worldwater resource-management problem is shown to produce very encouraging results.

1. Introduction

In real life, many optimization problems have multiple conflicting objectives to optimize simultaneously. These optimization problems are known as Multi-Objective Optimization problems (MOPs). Since the objective functions are conflicting in nature, the improvement of one objective may lead to the deterioration of other objectives. Thus, there is no single perfect solution for the objective function, but a set of trade-off solutions that present the Pareto-optimal Set (PS). The mapping of the PS in an objective space is termed a Pareto-optimal Front (PF).

Over the past decades, nature-inspired heuristic algorithms, e.g., Multi-Objective Evolutionary Algorithms (MOEAs), have been recognized as a suitable method to approximate the PS, owing to their ability to provide a set of trade-off solutions in a single run. Unfortunately, these algorithms encounter several challenges when the optimization problems have more than three objectives which are referred to as a Many-objective Optimization Problem (MaOPs).

One main issue behind the degraded performance of most MOEAs is the use of Pareto dominance as the selection criterion. When the number of objectives increases, the proportion of non-dominated solutions increases rapidly, leading to the inability to differentiate which solutions should be selected to survive into the next generation. Thus, the population will lose selection pressure toward the true PF. The second issue, since the population members are typically far away from each other in MaOPs, the generated offspring will be widely distant from each other and even farther from the PF, thereby applying the crossover and mutation operators are ineffective. Another issue remains in the

*Corresponding author
najwa.kouka.tn@ieee.org (N. Kouka); fatma.bensaid.tn@ieee.org (F. BenSaid); raja.fdhila@ieee.org (R. Fdhila); rahma.fourati@ieee.org (R. Fourati); a.hussain@napier.ac.uk (A. Hussain); adel.alimi@ieee.org (A.M. Alimi)
ORCID(s):

difficulty of diversity estimation. In high-dimensional space, the determination of neighbors in a population becomes difficult and could affect the distribution of solutions in the search space.

To effectively handle these issues, MaOPs have caught wide attention in the evolutionary community in the last decade. These algorithms can be divided into four categories, which are described as follows. The first category refers to preference ordering relation-based approaches. Since the loss of selection pressure toward the true PF is mainly caused by the inability of the Pareto dominance to differentiate solutions, significant efforts have been made to provide new ordering relations that allow comparison between non-dominating solutions. Examples of modified Pareto-dominance definitions include ϵ-dominance, grid-dominance, norm of dominance difference, k-optimality, L-optimality, preference order ranking, and fuzzy dominance [37]. The second category is decomposition-based approaches that divide the problem into a set of sub-problems and optimize them simultaneously. One representative algorithm is MOEA/D, in which the problem is decomposed through the weighted vector. It performed well in MaOPs, which initiated the proposition of other variants such as MOEA/DD [45]. The third category covers reference-based approaches, which aim to maintain a good diversity of solutions. These approaches perform predefined multiple-targeted searches rather than decompose the problem into a set of sub-problems, by investigating predefined reference points. The third generation Non-dominated Sorting Genetic Algorithm NSGAIII is one of the most representative algorithms employing reference points to enhance diversity among the solutions by selecting the closest solutions to these reference points. Inspired by NSGAIII, several algorithms were designed as Many-Objective Evolutionary Algorithm with Adaptive Reference Vector (MaOEA-ARV) [47]. The fourth category is performance indicator-based environmental selection [42] which is widely used to solve MaOPs. The performance indicator is not only used to evaluate the obtained PF but also to guide the search process. For example, several Hyper-Volume (HV) based approaches were proposed such as SIBEA [3]. The main issue that arises when using the HV indicator to solve MaOPs is the high computational complexity. To deal with this issue, the Hype method was proposed [4] to quickly approximate the HV value using Monte Carlo simulation. Other indicators were also investigated in MOEAs. For instance, R2 indicator is used in various algorithms such as R2-MOPSO [14]. The Δ_p indicator is used in the evaluation of fitness function [31]. The indicator I^+_{SDE} combines the sum of objectives, and shift-based density estimation [27]. An epsilon indicator $I\epsilon^+$ is used in MOPSO [25] to update the particle memory and archive. In addition, Inverted Generational Distance (IGD) indicator and its variants were investigated as a selection mechanism in MOEAs. The variants include IGD$^+$-EMOA [23], and MaOEA/IGD [37]. The performance indicator-based environmental selection has been considered a promising method in solving problems with many objectives as they can improve the selection pressure towards the PF (against Pareto dominance).

In addition to the aforementioned algorithms, Particle swarm optimization (PSO) is one of the most well-regarded swarm-based algorithms in the literature. Although the original PSO has shown good performance for single optimization, it suffers from premature convergence with MOPs. Consequently, many researchers conducted modifications and improvements [12] following Coello's first proposal of the Multi-Objective Particle Swarm Optimization (MOPSO). Recently, there are a few works focused on improving the performance of PSO to tackle the MaOPs (MaOPSOs). For example, in NMPSO [18], a balanceable fitness estimation (BFE) method was introduced to improve both convergence and diversity. A Bottleneck Objective Learning (BOL) strategy was employed in CPSO [20], to generate a diverse solution to different parts of PF and to improve convergence on all objectives. Furthermore, a solution reproduction procedure with both an Elitist Learning Strategy (ELS) and a Juncture Learning Strategy (JLS) was introduced in the CPSO algorithm to improve the quality of archiving solutions. An improved Competitive Particle Swarm Optimization (CCMaPSO) is proposed in [10] for solving MaOPs. This algorithm is inspired by the cooperative competitive mechanism, which is based on three main components: a multi-step initialization mechanism, a competition-based learning strategy, and environmental selection. In the initialization step, decision variables are divided into convergence-related and diversity-related decision variables, which are optimized individually. Moreover, the main component is improved by a competitive learning strategy, where particles are updated via leader information from winner particles with good convergence and diversity. For the environment selection, one by one selection strategy suggested in 1by1EA [10] is performed by the selection mechanism.

The performance indicators were also incorporated in MaPSOs to balance convergence and diversity, respectively. One of the representative algorithms is R2-MOPSO [14] that makes use of the R2 contribution of the archiving solutions to select the global best leaders. MaOPSO/vPF [44] suggested a virtual IGD indicator to evaluate the comprehensive quality of a solution in the external archive according to a constructed virtual Pareto front. A unary epsilon indicator $I\epsilon^+$ is used in IDMOPSO algorithm [25] for updating the personal best $pBest$ and archive, respectively. The $pBest$ is updated according to the Pareto dominance relationship and $I\epsilon^+$ indicator. A combination of the $I\epsilon^+$ indicator and the direction vectors is applied to update and maintain the archive.

In particular, to improve the performance of MaOPSOs there are at least two fundamental issues to be addressed. The first one is how to select the leader $gBest$ (global best solution) and how to define personal best $pBest$. Since the $gBest$ and $pBest$ are used to guide the search direction of particles in the swarm, they strongly influence the performance of MaPSOs algorithms, especially in solving MaOPs. The second issue is how to balance the convergence and diversity of the swarm. On one side, the contribution to convergence and diversity performances of some solutions may be in conflict with one other, i.e, where many solutions might be located close to the ideal objectives trade-off. Those solutions with good convergence may collectively have poor diversity. On the other side, the MOPSO is very likely to be trapped in the local optimum due to its fast convergence. In other words, the balance between convergence and diversity is critical to the performance of MaPSOs.

To overcome the above-mentioned issues especially the increasing of non-dominated solutions when dealing with many-objective optimization problems, a novel Many-Objective Particle Swarm Optimization with Cooperative Agents (MaOPSO-CA) is proposed. More specifically, the IGD indicator is considered the selection criterion to enhance both convergence and diversity, thereby increasing the selection pressure toward the PF. Instead of performing optimization in a single swarm, multiple sub-swarms are used. The latter aims to promote diversity, and prevent premature convergence. The different sub-swarms can be located in different regions, which ensures the diversity of the search space. Therefore, cooperation between the sub-swarms is required. Basically, it can be handled by migration or sharing knowledge (best information) among sub-swarms. In this context, the Multi-Agent System (MAS) attempts to ensure cooperation [33] by exchanging knowledge among sub-swarms and coordination by defining the required actions to achieve the optimization process.

The main contributions of our proposed method are highlighted as follows:

1. The IGD indicator is employed as selection criteria for both leader selection and updating archive. The IGD-based selection turns out to be advantageous since it handles the challenge of an increasing number of dominating solutions, whilst facilitating convergence and diversity at the same time.
2. In leader selection, the solution with the best IGD value is chosen as a leader, in this way particles will learn from the best solution that balances convergence and diversity, respectively.
3. A new archival method is suggested. Non-dominated solutions are saved by using strong ϵ-dominance as a comparator. When the archive becomes full, the IGD-based selection is used to remove the worst solution.
4. A multi-agent system is used to model the intelligent behavior of multi-swarms to benefit from MAS features such as cooperation, and coordination.
5. A new strategy of knowledge-sharing-based multilateral and automated negotiation is proposed to select the best global solution among different sub-swarms (agents). The proposed strategy begins with a pre-negotiation phase that determines the trusted agents for the negotiation. Next, it is followed by a negotiation phase to decide which solution is the best.
6. Extensive experiments are conducted to verify the performance of MaOPSO-CA by comparing it with state-of-the-art MaOEAs, MOPSOs, and MaOPSO algorithms on 110 test instances and a practical application: Water resource management. Experimental results validate MaOPSO-CA.

The remainder of this paper is organized as follows. Section 2 overviews related works including IGD indicator and MAS. The proposed MaOPSO-CA approach is explained in section 3. Next, section 4 details the experimental study. Section 5 demonstrates the performance of the proposed approach on a real water management problem. Finally, the last section is devoted to conclusions and some potential future work recommendations.

2. Preliminaries and related works

2.1. Particle swarm optimization algorithm

Particle Swarm Optimization is a population-based search algorithm based on the simulation of the social behavior of birds within a flock. In the PSO algorithm, a swarm of particles flies in a D-dimensional search space searching for the optimal solution. Particles update their positions through the search space according to their previous flying experience $Pbest$ and the social experience $Gbest$ to emulate the success of other particles. Each particle i has a current velocity vector $\vec{V_i} = (\vec{V_{i1}}; \vec{V_{i2}}, ..., \vec{V_{id}})$, which represents the moving direction and speed of the particle P_i in dimension d. It has also a current position vector $\vec{X_i} = (\vec{X_{i1}}; \vec{X_{i2}}, ..., \vec{X_{iD}})$ that represents a potential solution of the problem to be solved. The position of P_i is changed by adding a velocity $\vec{V}_i(t)$ according to equation (1) to the current position as given in equation (2).

$$\vec{V}_i(t+1) = w\vec{V}_i(t) + c1r1(Pbest_{id}(t) - \vec{X}_i(t)) + c2r2(Gbest_{id}(t) - \vec{X}_i(t)) \tag{1}$$

$$\vec{X}_{id}(t+1) = \vec{X}_{id}(t) + \vec{V}_{id}(t) \tag{2}$$

where $\vec{V}_{id}(t)$ represents the dth dimension of the velocity of particle i at iteration t. $\vec{X}_{id}(t)$ denotes the dth dimension of the position of particle i at t-th iteration. The $Pbest_{id}$ is the dth dimension of the personal best of particle i. The $Gbest_{id}$ is the dth dimension of the global best position. w is the inertia weight used to keep the balance between the exploration and exploitation abilities. Its value is typically set between 0 and 1. Learning factors $c1$ and $c2$ are the cognitive and social acceleration coefficients, which represent how much each coefficient influences velocity. $r1$ and $r2$ are random values uniformly distributed in [0, 1]. The process of PSO starts with the random initialization of particles. Then, each particle updates its velocity and its position according to its own flying experience $Pbest$ and the whole swarm's best experience $Gbest$. The search process is repeated until a stopping criterion is satisfied.

2.2. IGD indicator-based MOEAs

Recently, a wide number of indicator-based approaches have been proposed [17] due to the complexity and variety of real-world problems. The performance indicator is not only used to assess the quality of the solution set but it is also applied as a major selection criterion.Typically, some performance indicators are able to simultaneously measure diversity and convergence as the HV and IGD indicators. These indicators were well used in MOEAs for solving MaOPs. Though, the high computational complexity is a major issue of the HV indicator. Despite the fact that the Monte Carlo simulation has been explored to reduce the complexity, the calculation is still ineffectual when the objective number is up than 10, while the calculation of IGD is scalable without these inadequacies [37]. Table 1 presents more details about IGD and HV indicators [8]

Table 1
Comparison between IGD and HV indicators

Criterion	IGD	HV
Performance quality	Convergence and diversity	
Knowledge dependent	Reference set	Reference point
Parameter dependent	No	
Scalable	Yes	
Computational complexity	$O(NM)$	$O(NlogN + N^{d/2})$

Currently, the IGD Indicator has been integrated into MOEAs to deal with the loss of selection pressure when tackling MaOPs. The integration of IGD indicator can be divided into two categories: IGD-based mating selection, and IGD-based selection.

2.2.1. IGD-based selection

The IGD-based selection aims to select which solutions should survive at each generation on one hand. On the other hand, it is used to decide which solution of the non-dominated solutions is stored in an archive. Examples include the IGD$^+$-EMOA algorithm: Lopez et al. [24] used an environmental selection based on the modified IGD indicator called IGD$^+$. This algorithm does not perform well on MOPs that have degenerated and disconnected PF. An improvement version of IGD$^+$-EMOA (IGD$^+$-EMOA II) was proposed in [22]. Its main contribution is the generation method of the reference set that will be used in the IGD measure. This method used non-dominated solutions saved in the archive to be part of the reference set. Regarding the experiment, the IGD$^+$-EMOA II can solve MaOPs with complex PF shapes such as MaF problems [42].

A Many-Objective Differential Evolution with Mutation Restriction (MyO-DEMR) was proposed in [6]. It integrated the IGD indicator as a second sorting criterion. Experimental results revealed that the MyO-DEMR can produce well-converged and diversified solutions for problems with up to 20 objectives, having linear and concave PF shapes. An enhanced IGD indicator termed IGD with non-contributing solution detection (IGD-NS), was involved in the environmental selection method of MOEA (MOEA/IGD-NS) [41]. This indicator is employed as the second selection criterion to distinguish the worst contributing solutions that have the least contribution to the IGD-NS indicator.

The IGD indicator is also investigated in a Many-Objective Particle Swarm Optimization (MaOPSO) on the basis of the virtual Pareto Front (vPF) [44]. The vPF was constructed from the external archive to play a role similar to a true PF in calculating the IGD value. This algorithm, called vPF/MaOPSO used the IGD to evaluate the degree of contribution of each particle in order to select the global best of the population and the personal best of the particle. Experimental results illustrated that the vPF/MaOPSOt achieved good results in dealing with MaOPs owing to the incorporation of the IGD indicator. The IGD^{+}-Many-Objective Evolutionary Algorithm (IGD^{+}-MaOEA) [7] use the IGD^{+} to estimate the density of the entire population.

2.2.2. *IGD-based mating selection*

The mating selection consists in selecting parents to create offspring solutions. Examples of this category include AR-MOEA [38], in which the IGD-NS is in the maintaining selection method. In the latter, solutions with larger contributions win the competition and are added to the gene pool to crease in the suite of the offspring. In the MaOEA/IGD algorithm [37], the IGD was used to compare solutions in the binary tournament selection method. The comparison within solutions was made based on their rank and their distance value to the reference set for the IGD indicator. In case both solutions had the same rank, their distance values are compared and the solution with the minimum value was selected. Owing to the good performance of IGD indicator in solving MaOPs, it is integrated as a selection criterion to select leader and also to archive solutions.

2.3. Multi agent system

A multi-agent system is a loosely coupled network of problem-solving agents that work together to find solutions to problems that are beyond the individual capabilities or knowledge of each agent. According to the definition of Wooldridge and Jennings [43], an agent is a computational system interacting with an environment that can be endowed with the following features:

- Autonomy: each agent acts without the direct control of human beings or other devices.
- Social ability: interactions occur among entities through a communication language in order to satisfy the objectives.
- Reactivity: agents answer in a precise way to signals coming from the environment.
- Pro-activeness: agents do not simply act in response to their environment, they take the initiative in order to satisfy their goal.

2.3.1. *Agents-based approaches to optimization*

A great deal of research attention has been devoted to multi-agent systems in the last decade, on account of their attractive features which can be advantageously applied in real-world problems. These approaches differ in terms of the context of using MAS, agents' behaviors, and the problem to optimize. Key representative approaches are summarized and described in this section.

In [48], a multi-agent optimization algorithm (MAOA) was proposed for solving the resource-constrained project scheduling problem. In the MAOA, a set of agents work in a grouped environment, in which each agent represents a feasible solution. Agents were characterized by social behavior, autonomous behavior, self-learning, and environment adjustment in order to explore and exploit the search space. Additionally, for information sharing, agents perform migration among various groups which adjust the environment dynamically. A continuous-time multi-agent system for optimal resource allocation problems was proposed in [15]. All agents were divided into different groups, where each group was defined as a new dual agent. New agents cooperated with primal agents to find the optimal solution. Primal agents aimed to seek their own optimal solutions and dual agents represented primal agents to communicate with others to meet load constraints. Primal and dual agents cooperated with each other to find optimal solutions to the primal problem and dual problem.

Distributed optimization algorithms with a fixed step size over multi-agent networks were proposed in [19] for distributed convex optimization. In this method, each agent was allocated to optimize a term of the objective function subject to local constraints. In the network, all agents are connected with a given topological structure, where each agent exchanges information with its neighbors locally, and all agents fulfill the task cooperatively. A multi-agent-based optimization method applied to the quadratic assignment problem was presented in [32]. Different agents were divided into four types: a decision-making agent, local search agents, crossover agents, and a perturbation agent. The decision-making agent was responsible to activate the suitable agent and maintain a shared archive to save elite solutions. Such decisions were made based on a reinforcement learning strategy. Local search agents were introduced for the intensification of specific search zones while perturbation agents and crossover agents were used to diversify the search. In [12], a hybridization between MAS and MOPSO algorithm was proposed to solve multi-objective problems. A set of sub-swarms was modeled as a multi-agent system where each agent was regarded as a sub-swarm. After the evolutionary process of each sub-swarm, agents save the best solutions discovered during the search in the shared archive. This method has been recently adopted to solve dynamic multi-objective problems [13].

The aforementioned works presented a brief description of recent multi-agent systems, illustrating the growing trend in agents' contribution to build intelligent systems for optimization of various problems such as scheduling, smart grid, resource allocation, and other practical applications summarized in [2]. It can be concluded that MAS can be performed in two different ways [30]. First, the MAS is used as an agent-based evolutionary algorithm, in which the optimization behavior is embedded in agents' behaviors. Second, the MAS is used as an agent-based modeling method that can be applied in a complex task as strategic planning of complex tasks, task coordination, and information sharing.

3. Proposed approach

In this section, the proposed approach Many-Objective PSO with Cooperative Agents (MaOPSO-CA) is illustrated as a flowchart (see Figure 1), and detailed in Algorithm 1.

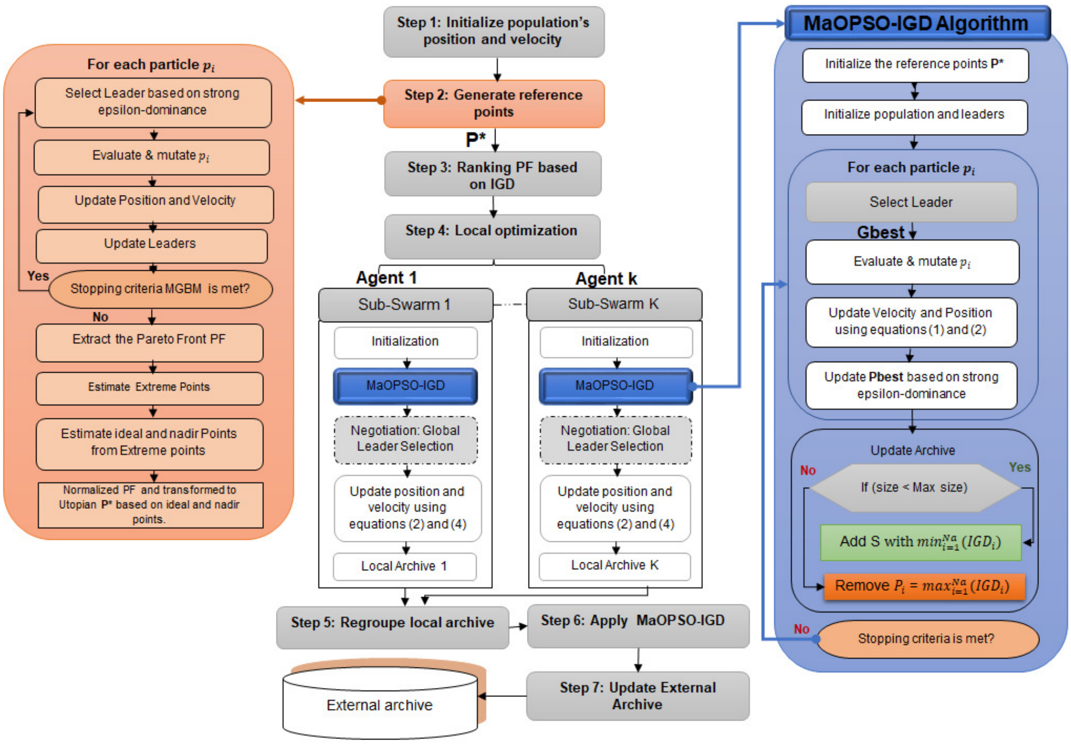


Figure 1: An illustration of MaOPSO-CA is devised. The middle column represents the basic method. Two sides describe the initialization phase and the MaOPSO-IGD algorithm

Algorithm 1: MaOPSO-CA algorithm

```
Input:  Population P with size N
Output: External Archive EA
for i = 1 to N do
    Random initialization of particle P_i : i = 1, ..., N
end for
for i = 1 to N do
    Select leader
    Update velocity and position
    Mutation
    Evaluate f_j, j = 1, ..., M
    Save non-dominated solutions of P in the archive EA
end for
P = Generated Particles
P* = Generate reference points
G_max = number of evaluations defined by MGBM;
K sub-swarm = Ranking (P)
for i = 1 to K do
    Create agent: LSA_i
    Affect sub-swarm i to LSA_i
end for
for i = 1 to K do
    g = 0
    while g ≤ G_max do
        Apply MaOPSO-IGD with Algorithm 2
        g = g + 1
        Save solutions in LA_i
        if g%NG then
            Share best Knowledge (SubSection 3.5)
        end if
    end while
    EA = EA ∪ LA_i
end for
```

3.1. General framework of MaOPSO-CA

In this work, our previous algorithm MOPSO-CA [12] is extended for solving MaOPs. The MOPSO-CA was suggested for solving MOPs, where the leader selection and shared information are handled by the dominance operator, and the updating archive is based on crowding distance. The originality of the proposed method is as follows:

- The first major difference is a new selection criterion based on the IGD indicator, which is employed in the leader selection and archive updating. With this indicator, the proposed approach can handle the challenge of the increasing number of non-dominated solutions, facilitating convergence and diversity, respectively.
- The second major difference is the new strategy of sharing information which is designed based on a new protocol of negotiation.

In the present work, the MAS contains two heterogeneous agents which collaborate, cooperate and negotiate with each other as depicted in Figure 2:

- Global Search Agent (GSA): holds the first level of the architecture, including swarm initialization, sub-swarms generation, and External Archive (EA) updating.
- Local Search Agent (LSA): holds the second level of the architecture in which each LSA performs the optimization of its own sub-swarm through MaOPSO-IGD Algorithm 2.

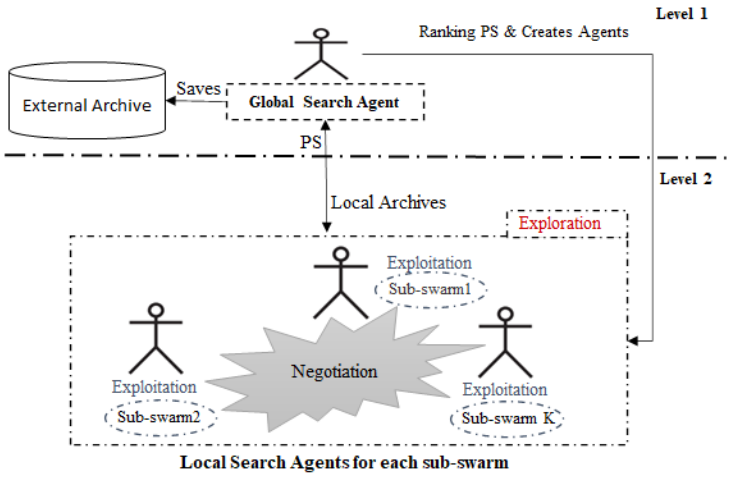


Figure 2: Social behavior of LSAs and GSA

The collaboration is ensured by the coordination of the optimization process among different agents. The cooperation between agents is improved by knowledge sharing where the decision-making of the best solution is carried out through automated negotiation.
Furthermore, the cooperation between these two kinds of agents is improved by communicating local archives of LSAs to GSA.

Algorithm 2: MaOPSO-IGD algorithm

Input: Population P with size N, reference points P^* with Q size;
Output: Archive A

for $i = 1$ to N **do**
  Initialize $\vec{X}_i$ and $\vec{V}_i$ for particle P_i
  Evaluate fitness for P_i
  Initialize $Pbest_i = P_i$
end for
Save non-dominated solutions of P in the archive
$g = 0$
repeat
  for $i = 1$ to N **do**
    Select leader with Algorithm 4
    Update velocity with equation (1)
    Update position with equation (2)
    Mutation
    Evaluation
    if $Pbest_i$ strongly dominate P_i **then**
      $Pbest_i = P_i$
    end if
    Update archive A with Algorithm 5
  end for
  $g = g + 1$
until Stopping criteria is met

3.1.1. Basic steps of MaOPSO-CA

- **Step 1:** First of all, the swarm (population P) and the external archive EA are randomly initialized.

- **Step 2:** A set of optimal solutions is provided which is required for the initialization of multi-swarm as well as the generation of reference points. The generation of these solutions starts with the random initialization of swarm P and EA. Then, for each particle, the leader is selected and the flight is performed. This step is repeated until reaching the MGBM stopping criterion [26]. This criterion relies on the basic dominance comparator which is not suitable for solving MaOPs. Therefore, in our work, we adopted the strong ϵ-dominance. Once the evolutionary process is terminated, generated solutions are saved in the external archive. Then, the set of reference point P^* is generated following Sun et al. [37] method, which is necessary for the IGD indicator as described in Section 3.3.

- **Step 3:** After the environment initialization, the swarm P is subdivided into K sub-swarms. To be specific, an IGD-based non-dominated sorting is used to rank the swarm into a set of sub-swarms. The main process of ranking starts with the calculation of IGD values. Then, the swarm member is sorted in ascending order with respect to their IGD values. Thus, the population is divided into K (size of PF divided by the defined size of the sub-swarm) sub-swarms, simply, according to particle order.

- **Step 4:** Generated sub-swarms are distributed among local search agents. Then, each LSA performs the optimization of its own sub-swarm by the new algorithm denoted as Many-Objective Particle Swarm Optimization based on IGD (MaOPSO-IGD) detailed in Subsection 3.4, Algorithm 2. In order to increase the diversity of solutions, sharing knowledge is performed after a certain number of generations (NG). This step is carried out by the automated negotiation as illustrated in Subsection 3.5.

- **Step 5:** Each LSA shares its local archive with the Global Search Agent GSA in order to update previous leaders in the external archive.

- **Step 6:** Local archives of each LSA are regrouped to start a new search using the MaOPSO-IGD algorithm, whereby the obtained information by each sub-swarm is shared among swarms for the purpose of increasing solutions' diversity.

- **Step 7:** Finally, the external archive EA is reported as the MaOPSO-CA output.

3.2. MaOPSO-IGD algorithm

The pseudocode of MaOPSO-IGD is illustrated in Algorithm 2. The main loop of this algorithm is explained as follows. First, the population member and archive are initialized.
Next, the selection of leader $Gbest$ is performed as detailed in Algorithm 4. Once the leader is selected, each particle updates its velocity and position with the guidance of its personal $Pbest$ and the selected leader $Gbest$ through equations (1) and (2), respectively.

After that, each particle is evaluated and its $Pbest$ is updated when the new one dominates the previous. Once all particles are updated, the archive is also updated with new non-dominated solutions found so far as described in Algorithm 5. The above evolutionary process is repeated until the termination condition is reached. To this extent, the archive is reported as the final PF approximation.

3.3. Generating references points

Since the proposed algorithm is based on the IGD indicator, a set of reference points must be provided first. There are several existing methods to generate reference points. In our work, the method proposed by Sun *et al.* [37], is considered. The main steps of this method are illustrated in Algorithm 3.

With the set of optimal solutions P defined in the first level, the extreme points Z^{ext} are identified, followed by the extraction of the ideal point Z^I and the nadir point Z^N. Next, a set of solutions is generated and placed in $(M-1)$ dimensional hyperplane as proposed in the enhanced NSGAIII algorithm [37]. To this end, these solutions are normalized and transformed to Utopian PF $P*$ based on ideal and nadir points.

3.4. IGD for solution selection

The IGD indicator is based on Euclidean distances mean between elements of the true Pareto front P^* and the Pareto Front PF generated by the evolutionary algorithm. It takes into consideration only elements of PF which are

Algorithm 3: Generation of reference points P^* algorithm

Input: initial population P, k: size of P^*;
Output: P^*
1: Z^{ext}= Estimate the extreme points of $f(x)$ [37]
2: $Z^I = Z_1^I, ...Z_M^I \leftarrow$ extract the ideal points from Z^{ext}
3: $Z^N = Z_1^N, ...Z_M^N \leftarrow$ extract the nadir points from Z^{ext} $P^* \leftarrow$ generate Q reference points from the constrained hyperplane
4: **for** $i < Q$ **do**
5: **for** $j < M$ **do**
6: $(P^*)_j^i \leftarrow (P^*)_j^i * (Z_j^{nad} - Z_j^I) + Z_j^I$
7: **end for**
8: **end for**

close to at least one element of P^*. Some PF solutions may not participate in the IGD value, but some other elements may have diverse participation. Therefore, the individual's IGD is mathematically defined as follows:

$$IGD(P_i) = \sqrt{\sum_{S \in P^*} dis(S, P_i)} \tag{3}$$

Where P_i is the particle and dis is the nearest Euclidean distance from $S \in P^*$ to Euclidean distance. The particle with a smaller IGD value is better than those with higher values. Note that, the IGD indicator measures how good is a solution in terms of convergence and diversity. Therefore, it is suitable to compare the quality of solutions. The particularity of our proposed MaOPSO-CA remains in using IGD indicator as a selection criterion in the leader selection as well as in the archive updating.

3.4.1. Leader selection

The main steps to decide which leader to follow are presented in Algorithm 4. First, the IGD of each particle is measured, so that the particle with the minimum IGD value is selected as the leader. In this way, the preferable solution closer to the P^* with respect to both convergence and diversity, is selected.

Algorithm 4: IGD-based leader selection algorithm

Input: Archive A with size Na, reference points P^*
Output: $Gbest$
1: **for** $i = 1$ to N **do**
2: Calculate IGD of S_i, $S \in A$ with equation 3
3: **end for**
4: $Gbest$ = solution with $min_{i=1}^{size} IGD(S_i, P^*)$

3.4.2. Updating archive

In the updating archive, the IGD indicator is used to keep only the best solutions in the archive as presented in Algorithm 5.

Let us consider that the new solution S is incoming. First, the dominance relation is checked for each solution in the archive. In the present paper, a strong ϵ-dominance is used to compare solutions. The solution $\vec{X}_1$ is a strong-dominated solution $\vec{X}_2$ if solution $\vec{X}_1$ is strongly better than solution $\vec{X}_2$ in all M objectives or in most of objectives.

However, if S is non-dominated to all solutions in the archive, it will be saved in it. In the case of reaching the maximum size of the archive, the IGD value of each solution in the archive is measured to remove the solution with the larger value.

Algorithm 5: IGD-based archiving algorithm

Input: Incoming solution S, reference points P^*
Input: Size of archive N
for $i = 1$ to N **do**
  $nb - dominated_{P_i} = 0$
  $nb - dominated_S = 0$
  for $j = 1$ to M **do**
    if $f_j(S) < f_j(P_i)$ **then**
      $nb - dominated_{P_i} = nb - dominated_{P_i} + 1$
    else
      $nb - dominated_S = nb - dominated_S + 1$
    end if
  end for
  if $nb - dominated_S > nb - dominated_{P_i}$ **then**
    marked P_i as dominated solution
  end if
end for
remove the marked solution
if $|EA| == N$ **then**
  remove the solution P_i with $max_{i=1}^{N} IGD(P_i, P^*)$
end if
add the new solution S to EA

3.5. Knowledge sharing-based on multilateral negotiation

To make a cooperative search, sharing knowledge about the search between particles must be introduced. The main steps of the sharing knowledge strategy are illustrated in Algorithm 6.

Algorithm 6: Knowledge share strategy algorithm

Input: K= Number of Local Search Agent LSA
$AT = \emptyset$
for $i = 1$ to K **do**
  Measure the $ATV(LSA_i)$ with equation 5
end for
avg= $\sum_{i=1}^{i=K} ATV(LSA_i)$
for $i = 1$ to K **do**
  if $ATV(LSA_i) \geq avg$ **then**
    $AT = AT \bigcup LSA_i$
  end if
end for
G_{Best} = outcome of negotiation with Algorithm 7
for $i = 1$ to K **do**
  N= size of $subswarm_k$
  for $j = 1$ to N **do**
    Update velocity with equation (4)
    Update position with equation (2)
  end for
end for

Here, we take advantage of the automated negotiation to decide and select the best global solution G_{Best} among the different sub-swarms. The solution G_{Best} presents the new search direction in the search space that is applied in the velocity equation to promote diversity. Hence, each particle will perform its flight based on its best personal experience $Pbest$, the experience of its neighbors $Gbest$, and the experience of the best global solution G_{Best}. The velocity equation is defined as follows:

$$\vec{V}_i(t+1) = W\vec{V}_i(t) + c1r1(Pbest_{id}(t) - \vec{X}_i(t)) + c2r2(Gbest_{id}(t) - \vec{X}_i(t)) + c3r3(G_{Best,id}(t) - \vec{X}_i(t)) \quad (4)$$

where t is the iteration number; W is the inertia weight; $c1$, $c2$ and $c3$ are learning factors; $r1$, $r2$ and $r3$ are random values uniformly distributed in $[0, 1]$.

The process of sharing knowledge is mainly based on two phases. The first phase is the pre-negotiation which defines trusted agents for the negotiation. The trust of the agent is evaluated by an Agent Trust Value (ATV) formulated by equation 5. A comparison between the ATV of each local search agent and the mean ATV of all agents easily determines trusted agents for proceeding with the negotiation. The second phase is automated negotiation.

$$ATV(LSA_i) = \frac{N_{dominated}}{|PF|} \quad (5)$$

where $N_{dominated}$ is a number of solutions in true PF that are dominated by at least one solution in the local archive (PF) of LSA_i.

3.5.1. Automated negotiation

Indeed, automated negotiation [11] can be viewed as a method of conflict resolution that aims to help negotiators to reach an agreement rapidly. To build a negotiation protocol, four main board topics must be taken into consideration.

- The first is the type of interaction that could be bilateral or multilateral.
- The second is the object presenting the range of issues over which an agreement must be reached. This object could be one or many issues.
- The third one is the negotiation protocol which presents a set of rules to govern the interaction.
- Finally, the negotiation strategy consists of a set of actions planned by a negotiator to undertake during the negotiation process.

In the proposed negotiation protocol, the object of negotiation holds one issue presenting the best global solution. Since the negotiation protocol involves many negotiators, multilateral negotiation is adopted. More specifically, each negotiator has complete information about its opponents, including state (trusted agent or not), and preferences (utility functions). The strategy and the protocol are further detailed in the following parts.

3.5.2. Negotiation strategy

The decision-making is an important step in the coordination and negotiation between agents [46]. The agent's decision during the negotiation process depends on a utility function that evaluates the quality of solutions offered by negotiators to determine the "winner" and the "loser". In our negotiation process, we used three different utility functions. The main utility function is the IGD indicator for offers comparison. In the case of an equality of IGD values, the second utility function is the generational distance GD that is incorporated to measure the offer convergence. In particular, the individual's GD (p_i) is defined by the Euclidean distance from p_i to its nearest member of P^*. Since some solutions may have equal IGD and GD at the same time, the third utility has been incorporated to measure the performance of each agent in the negotiation session. It is the Agent Confidence Weight (ACW) as given in equation 6. ACW is a new parameter used to ensure that the agreement is reached at the end of each negotiation session.

$$ACW(LSA_i) = \frac{W_i}{\sum_{j=0}^{N} W_j} \quad (6)$$

Where W_i denotes a success number that increases when the LSA_i is considered as trusted or as a winner in the negotiation process. In fact, a simple comparison between IGD, GD, and ACW allows our system to determine the negotiation outcome. To this end, the strategy adopted by LSAs is as follows. Firstly, the agent can accept the offer of its opponent if one of the following cases is reached:

- The opponent has a lower IGD value,

- The opponent has equal IGD values, and a lower GD value,
- The opponent has equal values of IGD and GD, and a higher ACW.

Secondly, the agent can make a counter offer if the above-mentioned conditions are not provided. Finally, the agent could walk away if the agreement is reached.

3.5.3. Negotiation protocol

To govern the interaction between agents, a new negotiation protocol denoted by Multi-Trusted Negotiators with Confidence Weight (MTNCW) is proposed as illustrated in Algorithm 7. The MTNCW protocol has a bidding phase followed by a voting phase. In the formal, the initiator with minimum IGD value puts its offer on the agenda. In the latter, all participants evaluate the offer on the negotiation table using the utility function, then return an acceptance or rejection by making a counter proposal. Thus, if the offer is accepted by all the parties, the negotiation ends with this offer. Otherwise, the negotiation starts a new round with only participants that make a counter offer, using the GD and ACW as utility functions.

Algorithm 7: Negotiation protocol algorithm

Input: K= Number of Trusted agents AT
Output: Outcome of negotiation= $Global_{Best}$

Agreement= False
Utility function= IGD indicator
while !Agreement **do**
  initiator= Trusted agent with minimum IGD value
  for $i = 1$ to K **do**
    proposal = $Gbest$
    offer = $Gbest$ of AT_i
    Send proposal to AT_i
    if Utility (proposal) better than Utility (offer) **then**
      Send Accept proposal to initiator
    else
      Send Reject proposal to initiator
    end if
  end for
  if Number of accepted response == $K - 1$ **then**
    Agreement = True
    Trusted agents = agents that reject the proposal
  else
    Utility = GD indicator and ACW
  end if
end while
Update $ACW(LSA_i)$ with equation 6
$Global_{Best}$ = proposal accepted by all trusted agents

3.6. Computational complexity of MaOPSO-CA

This subsection gives an upper bound of the computation complexity within one generation of MaOPSO-CA. There are three main parts in one generation: (1) leader selection; (2) archive updating, and (3) knowledge-sharing strategy. Suppose the number of objectives is M, the size of the particles is N, the size of reference points is Q, and the size of the leader set is N_a. In our case, the size of the leader set is equal to the population size ($N = N_a$).

- In the leader selection: the IGD value for each solution will be calculated and compared to others. In the calculation of IGD, the nearest distance between each leader and reference points needs to be calculated. So the computational complexity for selecting a leader will cost $O(N_a Q)$.

- In the archive updating: when the archive is not full, the computational complexity of archive updating costs $O(N) * (N_a M) = O(N^2 M)$ because the dominance operator is applied for comparing each particle in population with the entire leaders in the archive. In the case when the archive is full, the comparison-based dominance operator is applied first, then the IGD value for each solution is measured in order to remove the worst solution, so the complexity is equal to $O(N^2 M) + O(NQ) = O(N^2 M)$.

- In the knowledge sharing strategy, the global best solution $gBest$ is selected based on negotiation between K-LSA, where K denoted the number of sub-swarm. Each pair of solutions (the offer and proposed) is compared based on the utility function (IGD), so the complexity costs $O(KR)$. When all solutions are equal, the assessment will be based on GD and ACW. Thus, the time complexity of knowledge sharing is $O(k^2 R)$.

To summarize, the overall computational complexity of MaOPSO-CA for one generation is $O(K^2 R)$, which indicates that MaOPSO-CA is computationally similar to most state-of-the-art algorithms.

4. Experimental study on several benchmark problems

To highlight the robustness of the proposed approach in dealing with high computational problems, three experiments were conducted.

- The first is a comparative analysis with eight state-of-the-art MaOEAs (hpaEA [4], NSGA-II/SDR [39], DEAGNG [21], AdaW [49], MOEA/D-UR [9], MultiGPO [50], PeEA [16] and ACDB-EA [49]) on the DTLZ and WFG benchmark suites. The obtained results of these algorithms are extracted from the work in [49] (2022).

- The second experiment is a comparative analysis of the MaF benchmark suite, which was used owing to its complicated features. The proposed MaOPSO-CA is compared with six state-of-the-art for many-objective optimization, i.e., IDMOPSO ([25], 2020), and CCMaPSO ([10], 2022), FDEAII ([29], 2021), DEAGNG ([21], 2020), and VMEF ([28], 2022).

- The third experiment is an ablation study that investigates the contribution of each component in enhancing the MaOPSO-CA performance.

4.1. Benchmark test problems

A set of benchmark functions with diverse properties were investigated for empirical comparison in this work. The summary of parameter settings and PF shapes of test functions are shown in **Section 1 of the supplementary material**.

1. **DTLZ benchmark family** including 7 scalable problems DTLZ1-DTLZ7, and the inverted DTLZ1 (IDTLZ) [49]. These tests can be scaled with any number of objectives and decision variables $D = M - 1 + K$. The decision variable number is set as follows: for DTLZ1 $K = 5$, for DTLZ2-DTLZ6 $K = 10$ and for DTLZ7 $K = 20$.
2. **WFG benchmark family** includes 9 problems (WFG1-WFG9) with decision variables $D = M - 1 + 10$.
3. **MaF benchmark suite** involves 15 problems that cover a good representation of various real-world scenarios. According to [42], the setting of the decision variables is $D = M + K - 1$. Note that $k = 10$ for MaF01-MaF06 and MaF10-MaF11, $K = 20$ for MaF07. For MaF08 and MaF09, the number of decision variables is set as $D = 2$, and $D = 5$ for MaF13. To this end, $D = 20 \times M$ should be considered for MaF14 and MaF15.

4.2. Performance metrics

As a performance metric, we choose two widely used metrics, i.e. IGD and HV indicators. Both of them can not only measure the convergence, but also the diversity of the solution. Let P^* represents a set of solutions sampled uniformly on the true Pareto front, and Q is the set of optimal solutions found by an algorithm. Mathematically, the IGD value of Q is calculated by equation 7.

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, Q)}{|P^*|} \tag{7}$$

where $d(v, Q)$ represents the Euclidean distance between a point v in P^* and its closest neighbor point in Q. $|P^*|$ denotes the number of solutions in P^*. The low value of IGD proves that the Q is distributed uniformly and is close to the true PF.

The hypervolume indicator (equation 8) is described as the volume of the space in the objective space dominated by the Pareto front approximation Q and delimited from above by a reference point. Let $Z^r = (z_1^r, z_2^r, ..., z_m^r)$ be the reference point in the objective space that is dominated by all Pareto-optimal objective spaces.

$$HV(Q) = VOL(\bigcup_{x \in Q} [f_1(x), z_1^r]x...[f_m(x), z_m^r]) \tag{8}$$

Where $VOL(.)$ is the m-dimensional Lebesgue measure. The larger the HV value is, the better the quality of Q approaching the true PF.

The true PF of MaF problems was introduced in CEC 2018 [42] for measuring the performance in terms of IGD.

For the two benchmarks DTLZ and WFG, 10000 reference points are sampled by Das and Dennis's method to calculate IGD. While the reference points used for the calculation of HV values is $(1.2, 1.2, ...)^T$

4.3. General parameter settings

The parameter settings of the compared algorithms over the considered benchmarks are detailed. Specifically, the parameter settings of the first part of the experiment (DTLZ and WFG benchmark suites) are adopted as the original setting in [49]. For the second experiment, the parameter settings are as follows:

- **Number of Objectives**: The MaF problems are composed of 5, 10, and 15 objectives.
- **Number of Evaluations**: All compared algorithms are individually executed 20 independent times. For each independent run, the maximum iteration number is equal to 10^3.
- **Statistical Analysis**: The Mann Whitney U test with a significance level of 0.05 is carried out to conduct the results of performance indicators due to the stochastic nature of the peer algorithms.
- **Population size**: depends on the size of the reference points as illustrated in Table 2.
- **Sub-Swarm size**: the size of the sub-swarm adopted in the proposed approach is 20.
- **Genetic Operators**: The crossover probability is 0.9 and its distribution index is 20. The mutation probability is $\frac{1}{D}$ and its distribution index is 20.
- **MOPSOs Settings**: The settings for velocity updating are related to the MOPSOs algorithms in which $c1$, $c2$ and $c3 = \text{Rand}\ (1.5, 2.0)$, and $w \in [0.1, 05]$. The setting related to IDMOPSO and CCMaPSO is the same as the original settings defined in [25], [10].

Table 2
Number of reference points and corresponding population sizes used in algorithms: NSGAIII, IDMOPSO and MaOPSO-CA

No. of objectives	Divisions	Size of reference points	Population Size of NSGAIII	Population size of IDMOPSO and MaOPSO-CA
5	6,0	210	212	210
10	3,2	275	276	275
15	2,1	135	136	135

4.4. Experimental studies on DTLZ and WFG problems

The generated results of recent competitive algorithms based on IGD and HV indicators, including hpaEA, NSGA-II/SDR, DEAGNG, AdaW, MOEA/D-UR, MultiGPO, PeEA, and ACDB-EA over DTLZ and WFG problems upon 5, 10, 15 and 20 objectives are listed in **Section 2 of the supplementary material**. In the latter, the best and the second-best IGD and HV values for each test function are highlighted with gray shade and light gray shade, respectively. Besides, the comparison result between MaOPSO-CA and each peer competitor is marked by the symbols (+,-). The symbol '+' refers to that the MaOPSO-CA performs better than the peer algorithm, and '-' means the opposite.

The obtained results based on IGD and HV indicators which are detailed in the supplementary material are summarized in Table 3. In addition, the number of best and second-best in all instances in terms of IGD and HV indicators are presented in Figure 3 and Figure 4.

Table 3
Comparison results of MaOPSO-CA and each competitor on test instances in terms of IGD and HV indicators

MaOPSO-CA vs	DTLZ		WFG	
	IGD (+/-)	HV (+/-)	IGD (+/-)	HV (+/-)
hpaEA [4]	23/9	32/0	36/0	36/0
NSGA-II/SDR [39]	19/13	32/0	36/0	36/0
DEAGNG [21]	19/13	32/0	36/0	36/0
AdaW [49]	18/14	32/0	36/0	36/0
MOEA/D-UR [9]	17/15	32/0	36/0	36/0
MultiGPO [50]	14/18	32/0	36/0	36/0
PeEA [16]	15/17	32/0	36/0	36/0
ACDB-EA [49]	14/18	32/0	36/0	36/0

As presented in Table 3, it can be observed that the proposed MaOPSO-CA has significant superiority over competitors on most of the test instances in terms of IGD indicator as well as HV indicator. Specifically, for DTLZ test problems, proportions of test instances where MaOPSO-CA achieves better IGD results than hpaEA, NSGA-II/SDR, DEAGNG, AdaW, MOEA/D-UR, MultiGPO, PeEA, and ACDB-EA are 23/32, 19/32, 19/32, 18/32, 17/32, 14/32, 15/32, and 14/32, respectively. In other words, proportions of hpaEA, NSGA-II/SDR, DEAGNG, AdaW, MOEA/D-UR, MultiGPO, PeEA, and ACDB-EA are 9/32, 13/32, 13/32, 14/32, 15/32, 18/32, 17/32, and 18/32, respectively. For the HV indicator, MaOPSO-CA is better than hpaEA, NSGA-II/SDR, DEAGNG, AdaW, MOEA/D-UR, MultiGPO, PeEA, and ACDB-EA. To this end, based on these two indicators, it can be found that the MaOPSO-CA performs on 136 out of 256 comparisons.

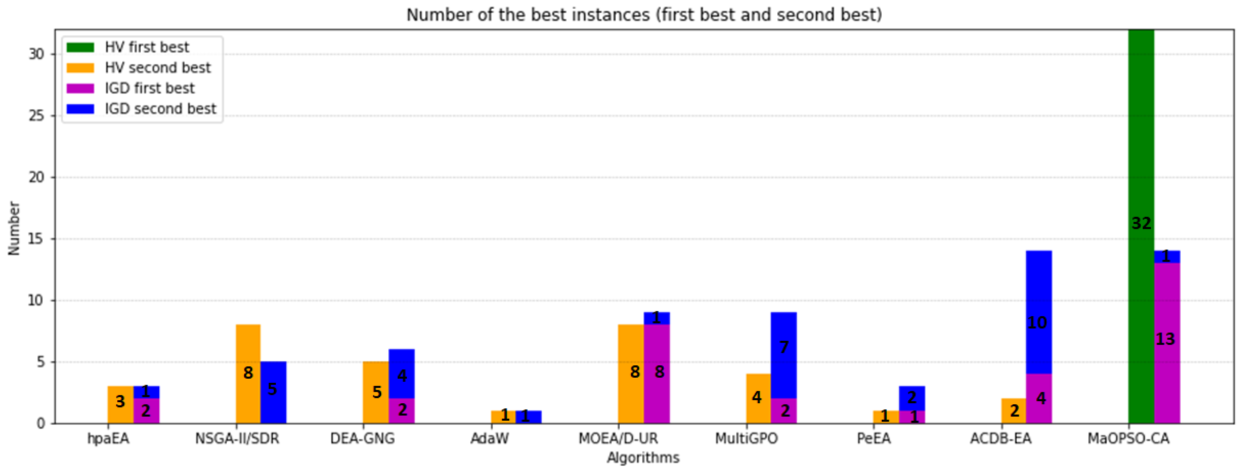


Figure 3: The best and second-best number of DTLZ instances obtained by eight compared algorithms is shown. The bottom of each bar indicates the number of best instances, and the top of each bar indicates the number of second-best instances.

According to Figure 3 and Figure 4, each algorithm is associated with two bars, where the first and the second bars consider HV and IGD, respectively. For each bar in the figure, its lower and its upper columns denote the number of the best instance and the second-best instance, respectively.

Along with the graphical representation, Table 4 further reports the tabular form of the number of the best instance and the second-best instance in terms of IGD and HV indicators for DTLZ and WFG problems.

When compared with recent state-of-the-art algorithms, the proposed approach has shown to be superior to the compared algorithms within most test functions of DTLZ and WFG test problems as shown in Table 5.

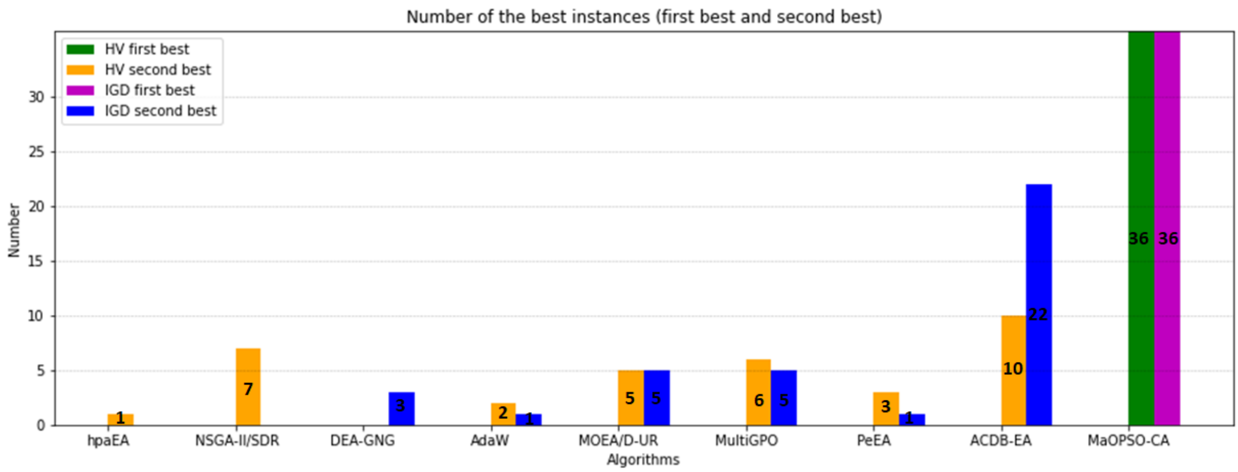


Figure 4: The best and second-best number of WFG instances obtained by eight compared algorithms is shown. The bottom of each bar indicates the number of best instances, and the top of each bar indicates the number of second-best instances.

Table 4
Number and proportion of the best and second-best instances obtained by algorithms on DTLZ and WFG

Metric	Instance	hpaEA [4]	NSGA-II/SDR [39]	DEAGNG [21]	AdaW [49]	MOEA/D-UR [9]	MultiGPO [50]	PeEA [16]	ACDB-EA [49]	MaOPSO-CA
					DTLZ					
IGD	1_{st} and 2_{st}	2/1	0/5	2/4	0/1	8/1	2/7	1/2	4/10	13/1
HV	1_{st} and 2_{st}	0/3	0/8	0/5	0/1	0/8	0/4	0/1	0/2	32/0
					WFG					
IGD	1_{st} and 2_{st}	0/0	0/0	0/3	0/1	0/5	0/5	0/1	0/22	36/0
HV	1_{st} and 2_{st}	0/1	0/7	0/0	0/2	0/5	0/6	0/3	0/10	36/0

Table 5
Comparison results of all considered algorithms on DTLZ and WFG test problems with 5, 10, and 15 objectives

MaOPSO-CA vs	hpaEA [4]	NSGA-II/SDR [39]	DEAGNG [21]	AdaW [49]	MOEA/D-UR [9]	MultiGPO [50]	PeEA [16]	ACDB-EA [49]
IGD	59/9	55/13	55/13	54/14	53/15	50/18	51/17	50/18
HV	68/0	68/0	68/0	68/0	68/0	68/0	68/0	68/0

4.5. Experimental studies on MaF Problems

This subsection is devoted to the experimental results performed on the MaF benchmark [42]. The mean and standard deviation results of IGD and HV indicators on each test instance are detailed in **Section 3 of the supplementary material**. The best results and the second-best results are highlighted with gray shade and light gray shade, respectively. Besides, the Mann-Whitney U test is used to find the significant difference within the proposed approach against MaOEAs with the symbols +, -, and = denoting whether our performance results are statistically better than, equal to (no significant difference), or worse than those of the corresponding peer competitors, respectively.

The comparison results between MaOPSO-CA and each peer competitor are summarized in Table 6 and Table 7. Considering the IGD indicator, MaOPSO-CA is better than IDMOPSO, CCMaPSO, FDEAII, VMEF and DEAGNG, are 42/42, 37/42, 33/42, 25/42, and 42/42, respectively. On the other side, the proposed approach based on HV indicator is better than IDMOPSO, CCMaPSO, FDEAII, VMEF and DEAGNG, which are 41/42, 39/42, 26/42, 23/42, and 37/42, respectively. According to the two indicators' performance, it is shown that the MaOPSO-CA has superiority over competitors in solving MaF problems.

Table 6
Summary of the results on MaF test problems, where seven compared algorithms are better than (+), worse than (-), and equal (=) to MaOPSO-CA according to the Mann-Whitney U test

MaOPSO-CA vs	IGD (+,=,-)	HV (+,=,-)
IDMOPSO [25]	42/0/0	41/1/0
CCMaPSO [10]	37/3/2	39/0/3
FDEAII [29]	33/1/8	26/2/14
VMEF [28]	25/3/14	23/3/16
DEAGNG [21]	42/0/0	37/2/3

Table 7
Number and proportion of the best and second-best instances obtained by algorithms on MaF problem

Metric	Instance	IDMOPSO [25]	CCMaPSO [10]	FDEAII [29]	VMEF [28]	DEAGNG [21]	MaOPSO-CA
IGD	1_{st} and 2_{st}	0/0	0/5	2/12	15/17	0/0	25/8
HV	1_{st} and 2_{st}	0/0	0/5	4/18	16/8	4/1	18/10

In Figure 5, each algorithm is associated with two bars, where the first and second bars present HV and IGD, respectively. For each bar in the histogram, its lower and upper columns denote the number of the best instance and the second-best instances, respectively. Considering the highest bar, it is clear that the proposed MaOPSO-CA has the best performance.

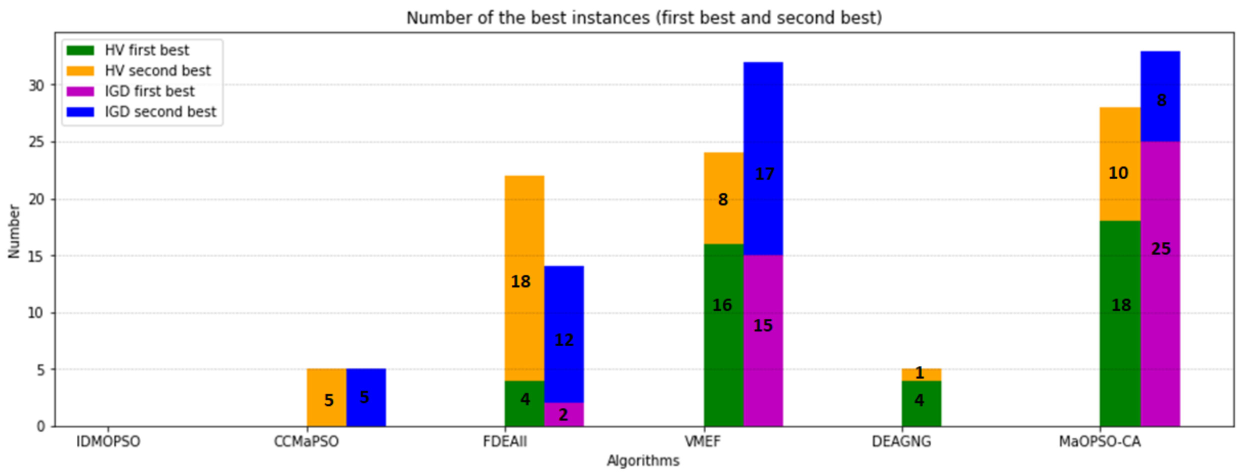
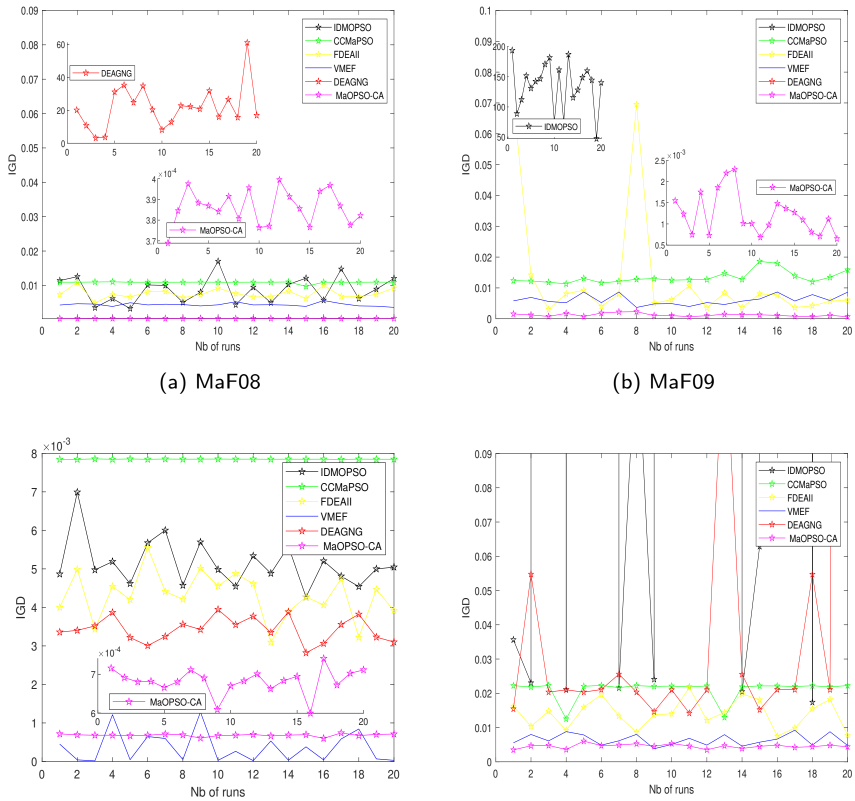


Figure 5: The best and second-best number of MaF instances obtained by six compared algorithms is shown. The bottom of each bar indicates the number of best instances, and the top of each bar indicates the number of second-best instances.

To visually demonstrate the performance and the stability of different algorithms, IGD values are presented in Figure 6 where the x-axis and y-axis present the number of independent runs and the IGD value obtained for each run, respectively. It can be observed that IGD values of MaOPSO-CA among 20 independent runs are better than its peer competitors over MaF08, MaF09, MaF11, and MaF13.

The solutions obtained by IDMOPSO, CCMaPSO, FDEAII, VMEF, DEAGNG, and MaOPSO-CA are plotted in Figure 7 and Figure 8 to visualize the convergence and diversity, where the example is MaF14 and MaF15 with 15 objectives. The geometric MaF 14 is linear, partially separable, and large-scale. For MaF 15, the geometric feature of the true PF is convex, partially separable, and large-scale. Specifically, the approximated PF is plotted by the blue color. As



(a) MaF08
(b) MaF09
(c) MaF11
(d) MaF13

Figure 6: The IGD curves of IDMOPSO, CCMaPSO, FDEAII, VMEF, DEAGNG, and MaOPSO-CA for MaF08, MaF09, MaF11, and MaF13 with 15 objectives

Figure 7 (f), the MaOPSO-CA archives a good balance between diversity and convergence. In other words, MaOPSO-CA is able to provide solutions that cover the true PF as much as possible while approximating the Pareto-optimal solutions. Figure 7 (a) and 7 (b) present the PF of IDMOPSO and CCMaOPSO, which show the bad performance of these two algorithms. For solutions found by FDEAII, VMEF, and DEAGNG (Figure 7 (c), Figure 7 (d), and Figure 7 (e)) cannot achieve a good distribution among the True PF.

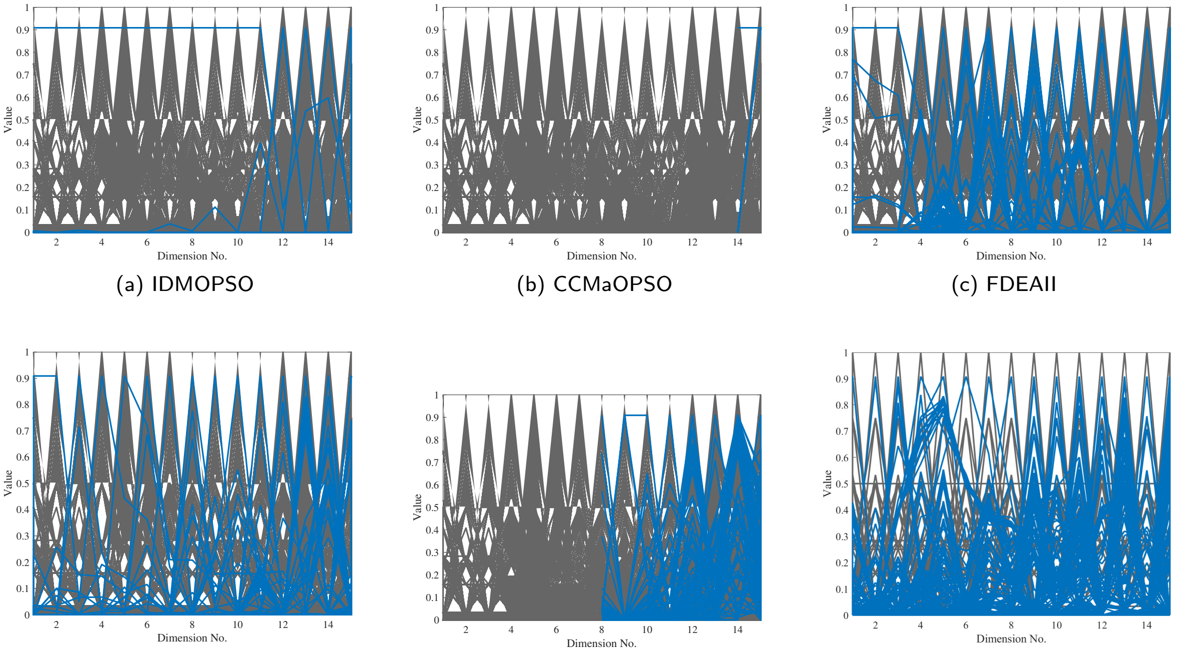
It can be concluded from Figure 8 that the solution set obtained by MaOPSO-CA is the most similar to the true Pareto of MaF15. In general, the proposed MaOPSO-CA can achieve a good balance between convergence and diversity.

4.6. Ablation study

To get a better understanding of how the incorporation of IGD and MAS improves the performance of our algorithm, we conduct an experiment of four variants with the different combinations of components:

- Multi-swarm MaPSO (MaPSO-MS),
- Multi-swarm MaPSO combined with MAS (MaPSO-MAS),
- MaPSO-MAS with IGD-based leader selection (MaPSO-LS),
- MaPSO-MAS with IGD-based updating archive (MaPSO-UA).

The comparison results of MaOPSO-CA and its four variants are illustrated in Table 8 in terms of IGD metric on 4 MaF problems: MaF01, MaF08, MaF10, and MaF13. The mean value of IGD is outlined above the standard deviation of over



(a) IDMOPSO (b) CCMaOPSO (c) FDEAII

(d) VMEF (e) DEAGNG (f) MaOPSO-CA

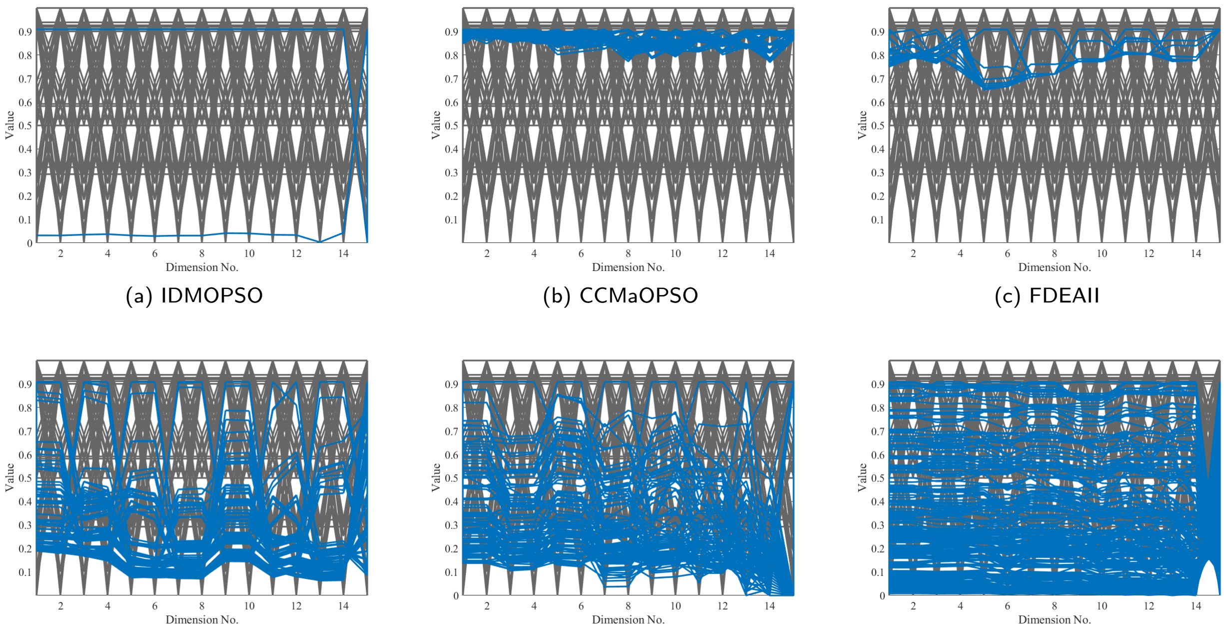
Figure 7: PF obtained by IDMOPSO, CCMaPSO, FDEAII, VMEF, DEAGNG, and MaOPSO-CA for the 15-objectives MaF14

Table 8
IGD Mean and standard deviation for contribution of each component in MaOPSO-CA

M	Problem	MaPSO-MS		MaPSO-MAS		MaPSO-LS		MaPSO-UA		MaOPSO-CA
5	MaF01	5.98E-3(1.0E-3)	(+)	5.80E-3(5.87E-4)	(+)	6.60e-3(7.63E-4)	(+)	6.70E-3(4.16E-4)	(+)	1.65E-3(1.15E-4)
	MaF08	5.93E-3(2.60E-4)	(+)	8.49E-4(3.84E-5)	(+)	8.19E-4(3.18E-5)	(+)	5.65E-3(8.95E-4)	(+)	1.73E-4(5.99E-6)
	MaF10	8.44E-3(4.71E-4)	(+)	7.91E-3(1.31E-4)	(+)	8.14E-3(2.07E-4)	(+)	7.99E-3(1.45E-4)	(+)	7.12E-3(2.86E-4)
	MaF13	7.44E-3(9.9E-4)	+	3.64E-3(2.6E-4)	(+)	3.62E-3(3.1E-4)	(+)	3.56E-3(3.2E-4)	(+)	3.17E-3(2.3E-4)
	Rank	4.5		2.75		3.25		3.5		1.0
10	MaF01	8.56E-3(6.81E-4)	(+)	7.73E-3(4.8E-4)	(+)	9.16E-3(5.66E-4)	(+)	8.97E-3(9.19E-4)	(+)	4.20E-3(3.61E-4)
	MaF08	7.77E-3(3.9E-4)	(+)	9.19E-4(2.1E-5)	(+)	8.97E-4(1.9E-5)	(+)	7.59E-3(5.1E-4)	(+)	2.68E-4(8.3E-5)
	MaF10	8.86E-3(3.68E-4)	(+)	8.46E-3(2.46E-4)	(+)	8.77E-3(3.58E-4)	(+)	8.66E-3(2.66E-4)	(+)	7.76E-3(4.10E-4)
	MaF13	1.02E-2(2.1E-3)	(+)	4.17E-3(3.6E-4)	(+)	4.15E-3(5.3E-4)	+	4.06E-3(4.6E-4)	(+)	4.00E-3(5.5E-4)
	Rank	4.5		2.75		3.5		3.25		1.0

20 independent runs. In addition, the row denoted as "Rank" summarize the average ranking of algorithms generated by the Friedman Test. In general, the effectiveness of a combination of different components is experimentally justified, as it enhanced our MaOPSO-CA performance. The superiority of MaOPSO-CA is also demonstrated statistically through the Friedman test.

Firstly, it is observed MaPSO-MAS outperforms MaPSO-MS on different test problems. Recall that the difference between MaPSO-MS and MaPSO-MAS lies in using MAS as a modeling method. Therefore, it can be confirmed that modeling multi-swarm as MAS can achieve a better result than multi-swarm. Based on these results, it can be concluded that the exchange of information based on automated negotiation has an advantage in improving exploration ability. This leads the MaPSO-MAS to outperform the multi-swarm based MaPSO over MaF01, MaF08, MaF10, and MaF13 with linear, concave, and mixed PF shape, respectively. Secondly, when comparing the MaPSO-LS with MaPSO-MAS, MaPSO-LS is the best performer on different problems, showing the impact of IGD-based leader selection in ameliorating the evolutionary process. Thirdly, by further comparing the MaPSO-UA with MaPSO-MAS, the results show also the advantage of IGD to keep the best solutions in terms of both convergence and diversity in the archive.



(a) IDMOPSO (b) CCMaOPSO (c) FDEAII

(d) VMEF (e) DEAGNG (f) MaOPSO-CA

Figure 8: PF obtained by IDMOPSO, CCMaPSO, FDEAII, VMEF, DEAGNG, and MaOPSO-CA for the 15-objectives MaF15

In fact, combining two effective components: MAS and archive updating based on IGD indicator have an advantage in not only improving the exploration but also maintaining a good balance between convergence and diversity.

In summary, the above experimental results have justified that MAS, IGD-based selection, and IGD-based archiving contribute to enhancing the performance of MaOPSO-CA.

5. Application of MaOPSO-CA on the Constrained Water Resource Management Problem

In this section, the water resource management (WRM) problem with 5 objectives is used to prove the ability of the proposed algorithm to deal with real-world problems. The problem description is provided in **Section 4 of the supplementary material**. The reference sets for this problem are provided in [34], where the numbers of reference points are 6725. The proposed algorithm MaOPSO-CA is compared with MaOEA-MS that was recently proposed in [35] and also with NSGA-III, DCNSGA-III, and RVEA algorithms, respectively.

5.1. Results analysis

The comparison results of MaOPSO-CA and its peer competitors, including NSGA-III, DCNSGA-III, RVEA, MaOEA-MS are presented in Table 9 in terms of IGD and HV metrics over 20 independent runs. The obtained solution set by five algorithms is plotted in Figure 9 to visualize the diversity and convergence, in which the x-axis presents the objective number (Dimension No), and the y-axis presents their corresponding objective value.

Specifically, the best IGD and HV values are highlighted in bold face with a gray background. In fact, the results of compared algorithms are originated from [35].

The results indicate that the MaOPSO-CA is the best performer with respect to IGD results. Where the MaOEA-MS is the best algorithm according to HV value. As shown in Figure 9, NSGA-III, DCNSGA-III and MaOEA-MS have comparable performance with respect to the convergence and diversity of their PFs. While the RVEA algorithm has a poor convergence to the true PF. Moreover, the proposed MaOPSO-CA is able to produce approximation sets converging to the true PF with well-diversity solutions. To this end, convergence and diversity are better than the other algorithms.

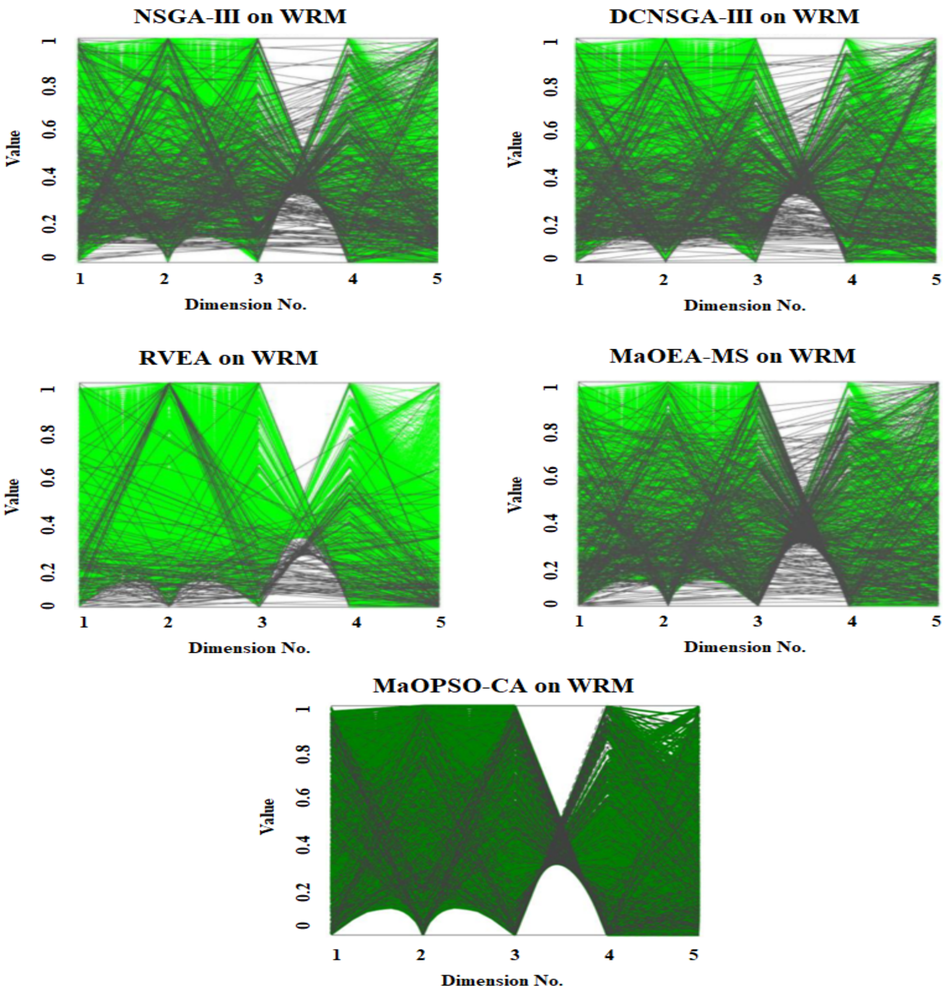


Figure 9: Solutions obtained by NSGA-III, DCNSGA-III, RVEA, MaOEA-MS [35] and the proposed MaOPSO-CA on WRM problem, where the green lines are the reference solutions.

Table 9
Mean values and standard deviation of IGD and HV obtained by compared algorithms on WRM problem

Metric	State-of-the art MaOEAs [35]				Proposed
	NSGA-III	DCNSGA-III	RVEA	MaOEA-MS	MaOPSO-CA
IGD	3.0600e+4 (3.68e + 3)	3.0933e+4 (4.77e+3)	1.6350e+5 (2.60e+4)	2.2383e+4 (1.53e+3)	1.2102e-4 (1.89e-05)
HV	1.0740e+26 (2.43e+23)	1.0731e+26 (3.24e+23)	9.1083e+25 (2.95e+24)	1.0816e+26 (1.63e+23)	7.6784e-01(7.21e-03)

6. Time complexity analysis

In this section, we compare the CPU time and memory of different algorithms. The two large-scale MaOPs: MaF14 and MaF15 are selected. The number of decision variables is 100, 200, and 300 for 5, 10, and 15 objectives respectively. The population size is set to 210 for 5 objectives, 275 for 10 objectives, and 135 for 15 objectives. Each algorithm is run 20 times independently for each test function. Experiments are carried out on a PC Intel(R) Core(TM) i5-9300HF CPU @ 2.40 GHz and 16 GB RAM. The summarized CPU times (seconds) are presented in Table 10.

The proposed algorithm takes more time than CCMaPSO, NSGAIII, and MOEA/D algorithms, and is slightly higher than MaOPSOs as depicted in Table 10. Actually, thanks to the high evolution of processing technology (GPU, TPU), it is possible nowadays to solve large-scale problems such as the optimization of deep learning model trainable parameters for computer vision task [40] [5] or medical applications [36] [1]. The complexity of algorithms is easy to deal with, but there is a need for high-quality performance. Based on our findings, we can conclude that the proposed MaOPSO-CA is suitable for problems sensitive to performance where time is not an important issue.

Table 10
CPU time (seconds) of the proposed MaOPSO-CA and compared algorithms.

M	IDMOPSO [25]	CCMaPSO [10]	FDEAII [29]	VMEF [28]	DEAGNG [21]	MaOPSO-CA
			MaF14			
5	1.1268e+6	1.8977e+1	6.9771e+0	1.1989e+1	1.7719e+1	1.2819e+6
10	1.0775e+5	6.2900e+1	9.4845e+0	1.3950e+1	1.6872e+1	1.5421e+6
15	1.3986e+6	1.1690e+2	5.6025e+0	2.0290e+1	3.1328e+1	8.7859e+5
			MaF15			
5	1.1774e+6	1.4926e+1	4.8893e+0	1.1025e+1	6.3019e+0	8.4447e+5
10	1.0775e+5	5.6850e+1	5.5890e+0	1.3317e+1	1.0136e+1	2.3186e+6
15	3.7941e+4	1.2361e+2	3.3856e+0	1.9862e+1	4.8817e+1	1.0226e+6

While our novel algorithm achieves good performance in different problems which are DTLZ, WFG, and MaF as well as a real-world problem, it is limed in terms of runtime. Each component of the proposed MaOPSO-CA contributes to the improvement of the global performance, but it takes more time for the final response in comparison with peer competitors. Therefore, we advocate the use of our algorithm in the context where performance is the main objective and time is not a requirement at a high level.

7. Conclusion

In this paper, the MaOPSO-CA algorithm has been proposed to deal with MaOPs. Specifically, the IGD indicator is investigated for leader selection in order to lead the entire swarm toward the PF by achieving a trade-off between convergence and diversity. Further, the IGD is used as an archiving method to save the best optimal solutions found in the search. Two key innovative ingredients characterize our proposed MaOPSO-CA. First, the multi-swarm is used to maintain better diversity in the search and decrease the complexity required for solving MaOPs. Second, a multi-agent system is introduced to ensure cooperation and improve knowledge sharing among sub-swarms through a newly proposed KS-MNA strategy. Our findings showed that this enhanced exploitation within each sub-swarm, and exploration among sub-swarms. In summary, the proposed IGD-based selection and MAS design strategies have effectively improved the performance of the new algorithm. To validate our proposed MaOPSO-CA, a series of well-designed experiments are performed over three widely used benchmark test suites with up to twenty objectives. The obtained results in comparison with recent works indicate that the new algorithm is competitive in dealing with many-objective optimization problems. Finally, the application of MaOPSO-CA to a challenging real-world MaOPs problem, notably water resource management, achieved very encouraging results.

For future work, we will focus mainly on three key aspects in order to address the current limitations of our approach: (1) consider more complex optimization problems with different PF shapes, in particular a mixed one, (2) realize a detailed theoretical analysis of our proposed algorithm's stability and convergence behavior in order to theoretically analyze its attainable performance, and (3) to investigate community detection as a challenging real-world problem, which is gaining growing attention for analysis of social networks. Novel integration of MaOPs with guided policy search approaches will also be explored in this context.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The research leading to these results has received funding from the Ministry of Higher Education and Scientific Research of Tunisia under grant agreement number LR11ES48.

References

[1] Albadr, M.A.A., Tiun, S., Ayob, M., Al-Dhief, F.T., 2022. Particle swarm optimization-based extreme learning machine for covid-19 detection. Cognitive Computation , 1–16.

[2] Barbati, M., Bruno, G., Genovese, A., 2012. Applications of agent-based models for optimization problems: A literature review. Expert Systems with Applications 39, 6020–6028. doi:https://doi.org/10.1016/j.eswa.2011.12.015.

[3] Brockhoff, D., Zitzler, E., 2007. Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods, in: 2007 IEEE Congress on Evolutionary Computation, pp. 2086–2093.

[4] Chen, H., Tian, Y., Pedrycz, W., Wu, G., Wang, R., Wang, L., 2020. Hyperplane assisted evolutionary algorithm for many-objective optimization problems. IEEE Transactions on Cybernetics 50, 3367–3380. doi:10.1109/TCYB.2019.2899225.

[5] Cheng, H., Wang, Z., Ma, L., Liu, X., Wei, Z., 2021. Multi-task pruning via filter index sharing: A many-objective optimization approach. Cognitive Computation 13, 1070–1084.

[6] Denysiuk, R., Costa, L., Espírito-Santo, I., 2013. Many-objective optimization using differential evolution with variable-wise mutation restriction, in: Proceedings of the 15th annual conference on Genetic and evolutionary computation, pp. 591–598. doi:10.1145/2463372.2463445.

[7] Falcón-Cardona, J.G., Coello Coello, C.A., 2018. Towards a more general many-objective evolutionary optimizer, in: Auger, A., Fonseca, C.M., Lourenço, N., Machado, P., Paquete, L., Whitley, D. (Eds.), Parallel Problem Solving from Nature – PPSN XV, Springer International Publishing, Cham. pp. 335–346.

[8] Falcón-Cardona, J.G., Emmerich, M.T.M., Coello, C.A.C., 2018. On the Combination of Quality Indicators for Multi-Objective Optimization. Technical Report. IEEE CIS.

[9] de Farias, L.R.C., Araujo, A.F.R., 2022. A decomposition-based many-objective evolutionary algorithm updating weights when required. Swarm Evol. Comput. 68, 100980.

[10] Gu, Q., Liu, Y., Chen, L., Xiong, N., 2022. An improved competitive particle swarm optimization for many-objective optimization problems. Expert Systems with Applications 189, 116118. doi:https://doi.org/10.1016/j.eswa.2021.116118.

[11] Jennings, N., Faratin, P., Lomuscio, A., Parsons, S., Wooldridge, M., Sierra, C., 2001. Automated negotiation: Prospects, methods and challenges. Group Decision and Negotiation 10, 199–215. doi:10.1023/A:1008746126376.

[12] Kouka, N., Fdhila, R., Alimi, A.M., 2017. Multi objective particle swarm optimization based cooperative agents with automated negotiation, in: Liu, D., Xie, S., Li, Y., Zhao, D., El-Alfy, E.S.M. (Eds.), Neural Information Processing, Springer International Publishing, Cham. pp. 269–278. doi:https://doi.org/10.1007/978-3-319-70093-9_28.

[13] Kouka, N., Fdhila, R., Hussain, A., Alimi, A.M., 2020. Dynamic multi objective particle swarm optimization with cooperative agents, in: 2020 IEEE Congress on Evolutionary Computation (CEC), IEEE. pp. 1–8.

[14] Li, F., Liu, J., Tan, S., Yu, X., 2015. R2-mopso: A multi-objective particle swarm optimizer based on r2-indicator and decomposition, in: Proc. CEC, Sendai, Japan. pp. 3148–3155. doi:10.1109/CEC.2015.7257282.

[15] Li, K., Liu, Q., Yang, S., Cao, J., Lu, G., 2020. Cooperative optimization of dual multiagent system for optimal resource allocation. IEEE Transactions on Systems, Man, and Cybernetics: Systems 50, 4676–4687. doi:10.1109/TSMC.2018.2859364.

[16] Li, L., Yen, G.G., Sahoo, A., Chang, L., Gu, T., 2021. On the estimation of pareto front and dimensional similarity in many-objective evolutionary algorithm. Information Sciences 563, 375–400. doi:https://doi.org/10.1016/j.ins.2021.03.008.

[17] Liang, Z., Luo, T., Hu, K., Ma, X., Zhu, Z., 2020. An indicator-based many-objective evolutionary algorithm with boundary protection. IEEE Transactions on Cybernetics 51, 4553–4566.

[18] Lin, Q., Liu, S., Zhu, Q., Tang, C., Song, R., Chen, J., Coello, C.A.C., Wong, K., Zhang, J., 2018. Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems. IEEE Transactions on Evolutionary Computation 22, 32–46. doi:10.1109/TEVC.2016.2631279.

[19] Liu, Q., Yang, S., Hong, Y., 2017. Constrained consensus algorithms with fixed step size for distributed convex optimization over multiagent networks. IEEE Transactions on Automatic Control 62, 4259–4265. doi:10.1109/TAC.2017.2681200.

[20] Liu, X., Zhan, Z., Gao, Y., Zhang, J., Kwong, S., Zhang, J., 2019. Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization. IEEE Transactions on Evolutionary Computation 23, 587–602.

[21] Liu, Y., Ishibuchi, H., Masuyama, N., Nojima, Y., 2020. Adapting reference vectors and scalarizing functions by growing neural gas to handle irregular pareto fronts. IEEE Transactions on Evolutionary Computation 24, 439–453. doi:10.1109/TEVC.2019.2926151.

[22] Lopez, E., Coello, C., 2018. An improved version of a reference-based multi-objective evolutionary algorithm based on igd +, in: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 713–720. doi:10.1145/3205455.3205530.

[23] Lopez, E.M., Coello, C.A.C., 2016. Igd+-emoa: A multi-objective evolutionary algorithm based on igd+. 2016 IEEE Congress on Evolutionary Computation (CEC) , 999–1006.

[24] Lopez, E.M., Coello, C.A.C., 2016. Igd+-emoa: A multi-objective evolutionary algorithm based on igd+, in: 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 999–1006.

[25] Luo, J., Huang, X., Yang, Y., Li, X., Wang, Z., Feng, J., 2020. A many-objective particle swarm optimizer based on indicator and direction vectors for many-objective optimization. Information Sciences 514, 166–202. doi:https://doi.org/10.1016/j.ins.2019.11.047.

[26] Martí, L., García, J., Berlanga, A., Molina, J.M., 2016. A stopping criterion for multi-objective optimization evolutionary algorithms. Information Sciences 367-368, 700–718. doi:https://doi.org/10.1016/j.ins.2016.07.025.

[27] Pamulapati, T., Mallipeddi, R., Suganthan, P.N., 2019. i_{sde} +—an indicator for multi and many-objective optimization. IEEE Transactions on Evolutionary Computation 23, 346–352.

[28] Qiu, W., Zhu, J., Wu, G., Chen, H., Pedrycz, W., Suganthan, P.N., 2022. Ensemble many-objective optimization algorithm based on voting mechanism. IEEE Transactions on Systems, Man, and Cybernetics: Systems 52, 1716–1730. doi:10.1109/TSMC.2020.3034180.

[29] Qiu, W., Zhu, J., Wu, G., Fan, M., Suganthan, P.N., 2021. Evolutionary many-objective algorithm based on fractional dominance relation and improved objective space decomposition strategy. Swarm and Evolutionary Computation 60, 100776. doi:https://doi.org/10.1016/j.

swevo.2020.100776.
[30] Ratajczak-Ropel, E., 2018. Agent-Based Optimization. Springer International Publishing, Cham. doi:10.1007/978-3-319-62893-6_2.
[31] Rodríguez Villalobos, C.A., Coello Coello, C.A., 2012. A new multi-objective evolutionary algorithm based on a performance assessment indicator, in: Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation, ACM. pp. 505–512.
[32] Sghir, I., Hao, J.K., Jaafar, I.B., Ghédira, K., 2015. A multi-agent based optimization method applied to the quadratic assignment problem. Expert Systems with Applications 42, 9252–9262. doi:https://doi.org/10.1016/j.eswa.2015.07.070.
[33] Shang, Y., 2013. L1 group consensus of multi-agent systems with switching topologies and stochastic inputs. Physics Letters A 377, 1582–1586. doi:https://doi.org/10.1016/j.physleta.2013.04.054.
[34] Shankar Bhattacharjee, K., Kumar Singh, H., Ray, T., 2017. A Novel Decomposition-Based Evolutionary Algorithm for Engineering Design Optimization. Journal of Mechanical Design 139. doi:10.1115/1.4035862. 041403.
[35] Shen, J., Wang, P., Dong, H., Li, J., Wang, W., 2022. A multistage evolutionary algorithm for many-objective optimization. Information Sciences 589, 531–549. doi:https://doi.org/10.1016/j.ins.2021.12.096.
[36] Singh, A.K., Kumar, A., Mahmud, M., Kaiser, M.S., Kishore, A., 2021. Covid-19 infection detection from chest x-ray images using hybrid social group optimization and support vector classifier. Cognitive Computation , 1–13.
[37] Sun, Y., Yen, G.G., Yi, Z., 2019. Igd indicator-based evolutionary algorithm for many-objective optimization problems. IEEE Transactions on Evolutionary Computation 23, 173–187.
[38] Tian, Y., Cheng, R., Zhang, X., Cheng, F., Jin, Y., 2018. An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility. IEEE Transactions on Evolutionary Computation 22, 609–622.
[39] Tian, Y., Cheng, R., Zhang, X., Su, Y., Jin, Y., 2019. A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization. IEEE Transactions on Evolutionary Computation 23, 331–345. doi:10.1109/TEVC.2018.2866854.
[40] Tian, Y., Lu, C., Zhang, X., Tan, K.C., Jin, Y., 2020. Solving large-scale multiobjective optimization problems with sparse optimal solutions via unsupervised neural networks. IEEE transactions on cybernetics 51, 3115–3128.
[41] Tian, Y., Zhang, X., Cheng, R., Jin, Y., 2016. A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric, in: 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 5222–5229.
[42] Wang, H., Sun, C., Zhang, G., Fieldsend, J.E., Jin, Y., 2021. Non-dominated sorting on performance indicators for evolutionary many-objective optimization. Information Sciences 551, 23–38. doi:https://doi.org/10.1016/j.ins.2020.11.008.
[43] Wooldridge, M., 2009. An Introduction to MultiAgent Systems. 2nd ed., Wiley Publishing.
[44] Wu, B., Hu, W., He, Z., Jiang, M., Yen, G.G., 2018. A many-objective particle swarm optimization based on virtual pareto front, in: 2018 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8.
[45] Xu, Q., Xu, Z., Ma, T., 2020. A survey of multiobjective evolutionary algorithms based on decomposition: Variants, challenges and future directions. IEEE Access 8, 41588–41614. doi:10.1109/ACCESS.2020.2973670.
[46] Yilun, S., 2017. Consensus in averager-copier-voter networks of moving dynamical agents. Chaos 27. doi:https://doi.org/10.1016/j.ins.2019.11.047.
[47] Zhang, M., Wang, L., Li, W., Hu, B., Li, D., Wu, Q., 2021. Many-objective evolutionary algorithm with adaptive reference vector. Information Sciences 563, 70–90. doi:https://doi.org/10.1016/j.ins.2021.01.015.
[48] long Zheng, X., Wang, L., 2015. A multi-agent optimization algorithm for resource constrained project scheduling problem. Expert Systems with Applications 42, 6039–6049. doi:https://doi.org/10.1016/j.eswa.2015.04.009.
[49] Zhou, Y., Li, S., Pedrycz, W., Feng, G., 2022. Acdb-ea: Adaptive convergence-diversity balanced evolutionary algorithm for many-objective optimization. Swarm and Evolutionary Computation 75, 101145. doi:https://doi.org/10.1016/j.swevo.2022.101145.
[50] Zhu, S., Xu, L., Goodman, E.D., Lu, Z., 2022. A new many-objective evolutionary algorithm based on generalized pareto dominance. IEEE Transactions on Cybernetics 52, 7776–7790. doi:10.1109/TCYB.2021.3051078.