Influence of the number of connections between particles in the performance of a multi-objective particle swarm optimizer

Diana Cristina Valencia-Rodríguez*, Carlos A. Coello Coello

CINVESTAV-IPN (Evolutionary Computation Group), Av. IPN 2508, San Pedro Zacatenco, 07360, Ciudad de México, México

---

Abstract

Particle Swarm Optimization (PSO) is a bio-inspired metaheuristic that operates on a set of potential solutions (called particles). In PSO, each particle moves throughout the search space using the information collected by itself and its neighbors. Experimental studies have shown that the way each particle is connected (the swarm topology) impacts the performance of PSO both for single- and multi-objective problems. Several experimental analyses have shown that the number of connections among particles directly relates to the behavior of single-objective PSO. However, few studies exist about this relationship in Multi-Objective Particle Swarm Optimizers (MOPSOs). Furthermore, the existing studies are limited to two-objective problems or do not use specific topologies to control the number of connections among particles. This work analyzes the influence on the number of connections among particles in a MOPSO using random regular graphs as the swarm topology in many-objective problems. In order to undertake this analysis, we modified a variation of the Speed-constrained Multi-objective Particle Swarm Optimizer that can handle swarm topologies to make it more sensitive to its topology. Then, we analyzed its performance using regular graphs of different degrees. Our experimental results show that, in various problems, a higher connection degree produces instability in the algorithms. Moreover, our analysis also indicates that MOPSOs have a similar behavior if they have a swarm topology with the same connection degree.

Keywords: Swarm topology, Particle Swarm Optimization, Multi-objective Particle Swarm Optimization, Multi-objective optimization

---

*Corresponding author

Email addresses: `diana.valencia@cinvestav.mx` (Diana Cristina Valencia-Rodríguez), `ccoello@cs.cinvestav.mx` (Carlos A. Coello Coello)

1. Introduction

Particle Swarm Optimization (PSO) is a bio-inspired metaheuristic that was originally proposed by James Kennedy and Russell Eberhart in 1995 [1]. PSO operates with a set of particles (called *swarm*) that exchange information with their neighbors, intending to find an optimal solution. Experimental studies have shown that the way in which connections among particles are made (i.e., the *swarm topology*) affects the behavior and performance of PSO [2, 3, 4, 5]. For example, a fully connected topology improves the convergence speed; in contrast, a topology with a lower number of connections slows down convergence but avoids getting trapped in local optima (i.e., it favors diversity) [4].

Multi-objective Particle Swarm Optimizers (MOPSOs) have been very successful in solving real-world many-objective problems, e.g., feature selection for medical diagnosis [6] and for other applications [7], optimization of process parameters in stamping [8], optimization indoor CO_2 and PM2.5, the carpooling problem [9], concentrations [10], and optimization of machining operations [11]. Therefore, it is essential to understand the MOPSOs' behavior.

In particular, some experiments have shown that including a swarm topology in a MOPSO improves its performance [12, 13]. Accordingly, some attempts have been made to analyze the influence of swarm topologies in multi-objective problems [12, 14, 15, 13]. These studies differ in the topologies adopted and how such topologies are implemented in the corresponding MOPSO (i.e., the topology handling scheme adopted). Nonetheless, they use problems with restricted characteristics, use swarm topologies where we cannot easily identify if a graph property influences the swarm, or use expensive topology handling schemes.

In this work, we study the effect of the number of connections between particles in a MOPSO's performance using many-objective problems with a wide range of characteristics. Our contributions are the following:

1. We propose a new MOPSO sensitive to swarm topologies called SMPSO-M. This algorithm is based on a version of the Speed-constrained Multi-objective Particle Swarm Optimizer that employs a topology handling scheme (SMPSO-E2).
2. We performed an experimental comparison between the SMPSO-M, SMPSO-E2, and their original version. Our experimental analysis showed that our proposed approach outperforms the other algorithms and is more sensitive to swarm topologies.
3. We analyzed the influence of the number of connections between particles in the SMPSO-M performance. In this study, we employed random regular graphs as topologies and many-objective problems with 3, 5, 7, and 10 objectives. Our experimental analysis showed that, in most problems, a higher number of connections produces instability in the MOPSO's performance. Moreover, the MOPSOs with a topology having the same number of connections act similarly.

The remainder of this paper is organized as follows. Section 2 provides the necessary background for understanding the rest of this paper. Then, in

Section 3, we present some related work. After that, in Section 4, we introduce a MOPSO that is sensitive to swarm topologies and evaluate its performance. Using this algorithm, we present in Section 5 a study on the influence of the number of connections among particles using random regular graphs. Finally, in Section 6, we present our conclusions and some possible paths for future work.

2. Background

2.1. Multi-objective optimization

In multi-objective optimization, the aim is to solve problems of the type[1]:

$$\text{minimize } \boldsymbol{f}(\boldsymbol{x}) := [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})] \tag{1}$$

subject to:

$$g_i(\boldsymbol{x}) \leq 0 \quad i = 1, 2, \ldots, m \tag{2}$$

$$h_j(\boldsymbol{x}) = 0 \quad j = 1, 2, \ldots, p \tag{3}$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$ is the vector of decision variables, $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, ..., k$ are the objective functions and $g_i, h_j : \mathbb{R}^n \to \mathbb{R}$, $i = 1, ..., m$, $j = 1, ..., p$ are the constraint functions of the problem.

In a multi-objective optimization problem, we aim to find the best possible trade-offs among objectives defined in terms of Pareto optimality. In order to describe this concept, we need to introduce first the following definitions:

Definition 1. We say that a vector $\boldsymbol{x} \in \mathbb{R}^n$ **dominates** vector $\boldsymbol{y} \in \mathbb{R}^n$ (denoted by $\boldsymbol{x} \prec \boldsymbol{y}$), if and only if $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{y})$ for all $i = 1, \ldots, k$, and $f_j(\boldsymbol{x}) < f_j(\boldsymbol{y})$ in at least one $j \in \{1, \ldots, k\}$.

Definition 2. A vector $x^* \in \Omega \subset \mathbb{R}^n$ (where Ω is the feasible region) is **Pareto optimal** if there does not exist an $x \in \Omega$ such that $x \prec x^*$.

Definition 3. The **Pareto Optimal Set** $\mathcal{P}^*$ is defined by: $\mathcal{P}^* = \{\boldsymbol{x} \in \Omega \mid \boldsymbol{x} \text{ is Pareto optimal}\}$.

Definition 4. The **Pareto Front** $\mathcal{PF}^*$ is defined by: $\mathcal{PF}^* = \{\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^k \mid \boldsymbol{x} \in \mathcal{P}^*\}$.

Therefore, our aim is to obtain the Pareto optimal set from the set Ω of all the decision variable vectors that satisfy (2) and (3). Note however that in practice, not all the Pareto optimal set is normally desirable or achievable, and decision makers tend to prefer certain types of solutions or regions of the Pareto front [16].

---

[1] Without loss of generality, we will assume only minimization problems.

2.2. Multi-objective particle swarm optimizers

The main idea of PSO is that of flying a set of potential solutions throughout the search space, accelerating towards the best solutions in their neighborhood. PSO updates a particle's position $\boldsymbol{x_i}(t)$ at generation t using the following expression:

$$\boldsymbol{x_i}(t+1) = \boldsymbol{x_i}(t) + \boldsymbol{v_i}(t+1) \tag{4}$$

where $\boldsymbol{v_i}(t+1)$ is known as *velocity* and is defined by:

$$\boldsymbol{v_i}(t+1) = w\boldsymbol{v_i}(t) + C_1 r_1(\boldsymbol{x_{p_i}} - \boldsymbol{x_i}(t)) + C_2 r_2(\boldsymbol{x_{l_i}} - \boldsymbol{x_i}(t)) \tag{5}$$

The term $\boldsymbol{v_i}(t)$ is the previous velocity of the particle; w is a positive constant known as *inertia weight*; $r_1, r_2 \in U(0,1)$; C_1 and C_2 are two positive constants known as *cognitive* and *social* factors, respectively; $\boldsymbol{x_{p_i}}$ is the best position that the particle has found so far, and $\boldsymbol{x_{l_i}}$ is the best position found by the particles' neighbors (called *leader*). The connections among particles (the swarm topology) determine the particles' neighborhood. Experimental studies have shown that swarm topologies do influence PSO's behavior [2, 3, 4, 5]. We will discuss this topic more in-depth in the following section.

In a single-objective PSO, the particle's leader is determined by evaluating the objective function in each neighbor's best position and selecting the particle with the best value. However, in a multi-objective problem, each particle may have more than one leader due to the problem's nature (let's keep in mind that, in multi-objective optimization, we aim to obtain a set of solutions representing the best possible trade-offs among the objectives). For this reason, many MOPSOs store the best positions found so far in an independent set called *external archive* and take the leaders from it [17, 18, 19, 20]. The external archive is a set that stores the non-dominated solutions found during the search process. Its size is usually bounded; therefore, an additional criterion is used to decide which solutions to retain when the archive is full.

In our experimental analysis, we adopted a standard Pareto-based MOPSO that operates with an external archive: the Speed-constrained Multi-objective Particle Swarm Optimizer (SMPSO) [21]. The SMPSO incorporates Clerc and Kennedy's constriction factor χ [22] to control the particles' velocity. This coefficient contracts the velocity in order to avoid large values. Furthermore, it guarantees convergence of the swarm under certain assumptions. The constriction factor is defined as follows:

$$\chi = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}} \tag{6}$$

where

$$\varphi = \begin{cases} C_1 + C_2 & \text{if } C_1 + C_2 > 4 \\ 0 & \text{if } C_1 + C_2 \leq 4 \end{cases} \tag{7}$$

In addition, the SMPSO sets a bound on the accumulated velocity of each particle i in the dimension j using the equation:

$$v_{i,j}(t) = \begin{cases} \delta_j & \text{if } v_{i,j}(t) > \delta_j \\ -\delta_j & \text{if } v_{i,j}(t) \leq -\delta_j \\ v_{i,j}(t) & \text{otherwise} \end{cases} \tag{8}$$

where $\delta_j = (x_{max,j} - x_{min,j})/2$, and the j^{th} decision variable is in the range $[x_{min,j}, x_{max,j}]$.

Moreover, the SMPSO uses the *Crowding Distance* (CD) of NSGA-II [23] as the pruning criterion of its external archive. The CD of a solution is an approximation of the perimeter of the cuboid whose vertices are the solution's nearest neighbors. A solution with a smaller CD implies a solution in a more crowded region of objective function space. Therefore, solutions with higher values are preferred.

In summary, the SMPSO works as follows. First, the swarm is initialized, and the external archive is created with the non-dominated solutions from the swarm. Then, the main loop of the algorithm is executed during a pre-defined number of generations. In the first step of this loop, the velocity of each particle is computed using equation (5). The leader of a particle is selected by randomly taking two elements from the external archive and selecting the one with the best CD. The resulting velocity is multiplied by the constriction coefficient defined in equation (6), and the final value is bounded using expression (8). After that, the new positions of the particles are computed employing expression (4), and the polynomial-based mutation operator [24] is applied with a probability p_m. Finally, the resulting particles are evaluated, and the particles' best position and the external archive are updated. Algorithm 1 summarizes the behavior of SMPSO.

2.3. Swarm topologies

A swarm topology defines which neighbors a particle has to examine when selecting its leader. In other words, it defines how the particles will influence each other. A topology is represented by a graph whose vertices symbolize the particles, and there is an edge between two particles if they influence each other [2]. The formal definition is the following [25]:

Definition 5. A **swarm topology** at generation i is a graph $T_i = (P_i, E_i)$ where the vertex set $P_i = \{p_0, p_1, ..., p_{n-1}\}$ is a set of particles.

A topology can remain fixed during the execution (i.e., a static topology) or change during the generations (i.e., a dynamic topology). We limit this work to static topologies; therefore, we describe the most representative static topologies next:

- **Star** (*gbest*). It represents a fully connected graph in which there is an edge between every pair of vertices so that each particle will be influenced by the whole swarm (see Fig. 1a). This topology resembles a small community where the decisions are taken involving everyone [3]. Furthermore,

Algorithm 1 SMPSO

```
Require: Polynomial-based mutation parameters ($p_m$ and $\eta_m$), Maximum number of generations ($maxGenerations$), Swarm size, Archive size
Ensure: Pareto set approximation
Initialize the swarm
Initialize the external archive
$generation = 0$
while $generation < maxGenerations$ do
    Compute the CD of the archive's elements
    for each particle $x_i$ in the swarm do
        $x1, x2 \leftarrow$ Randomly take two solutions from the external archive
        if $x1.CD > x2.CD$ then
            $\boldsymbol{x}_{l_i} \leftarrow x1$
        else
            $\boldsymbol{x}_{l_i} \leftarrow x2$
        end if
        Compute the velocity using equation (5) and multiply it by equation (6)
        Set a bound on the resulting velocity using equation (8)
        Compute the new position using equation (4)
        Apply polynomial-based mutation
        Evaluate the new particle
    end for
    Update the particle's memory and the external archive
    $generation = generation + 1$
end while
return External archive
```

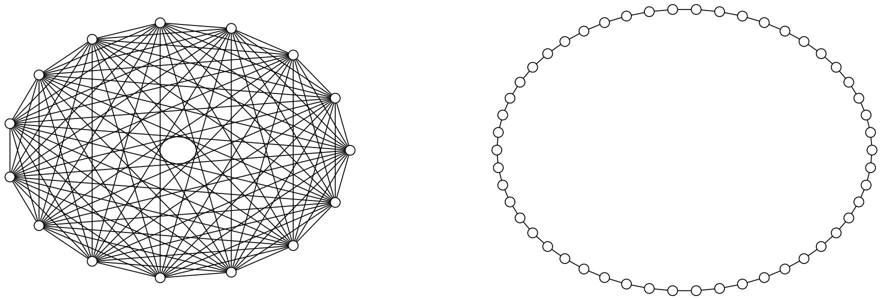
it has been shown that PSO converges faster with this topology. However, when using this topology, PSO can get easily trapped in local optima [5].

- **Ring** (*lbest*). In the graph representing this topology, each vertex is connected to its k nearest neighbors (usually $k = 2$). The information travels slowly through this topology. Hence, particles can explore different regions of the search space, and the best information will eventually spread in the graph [4, 3]. The convergence speed of PSO handling this topology is slow [3]. Figure 1b shows an example of a ring topology.
- **Lattice**. This topology is also known as square or von Neumann [5, 3]. It is represented by a rectangular lattice where each vertex has four neighbors, wrapping the edges like a torus (see Fig. 1c). The lattice topology is commonly used in other areas such as in cellular automata [3]. Moreover, an experimental analysis showed that this topology performs consistently well in many global optimization problems, so its use has been highly recommended [5].
- **Tree**. In this topology, the particles are ordered hierarchically, resembling a tree. Initially, this topology was dynamic. A particle was influenced by itself and its parent. Besides, if the personal best position of the particle was better than the parent, both particles were exchanged [26]. However, in order to make this topology static, we eliminated the exchange of particles and included the children nodes in the neighborhood. Fig. 1d shows an example of a tree topology.
- **Wheel**. It consists of a graph with a vertex connected to all the other vertices in the graph and viceversa (see Fig. 1e). Therefore, each particle is isolated from the others in the swarm, and they can only communicate through a focal particle. This focal particle compares the performance of all particles and adjusts its trajectory to the best position. If the adjustment is good enough, this information is communicated to the other particles. Therefore, the focal particle serves as a filter of the information [4]. Kennedy suggested in [4] that this topology could perform well in multimodal single-objective problems because the focal particle slows down the swarm's attraction to the current best position. The graph used to represent this topology is called *Star* in the graph theory community.

3. Previous Work

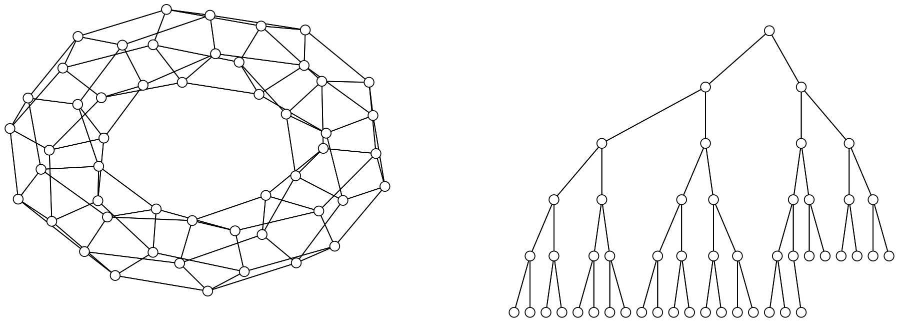
In this section, we will describe some previous studies of the influence of swarm topologies in MOPSOs. These studies differ in the topologies adopted, how they handle them, and the multi-objective problems they use.

Yamamoto et al. in 2012 [15] proposed a topology handling scheme where (in addition to an external archive) each i^{th} particle has a sub-archive A^i without limited size. Each A^i stores the non-dominated solutions found by the i^{th}



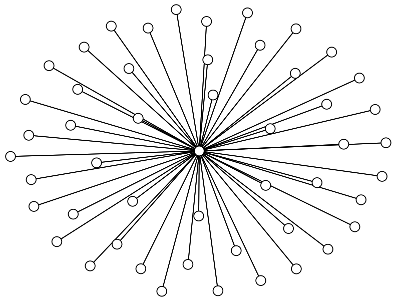
(a) Star topology. Each particle is connected to every other particle in the swarm.

(b) Ring topology. Each particle is connected to its two nearest neighbors.



(c) Lattice topology. The particles are connected to the neighbors above, below, and two on each side.

(d) Tree topology. The particles are arranged in hierarchical order, resembling a tree.



(e) Wheel topology. A focal particle is connected to every other particle in the swarm, and they are connected to the focal particle.

Figure 1: Most representative static topologies

particle and its neighbors. Moreover, the i^{th} particle will select its leader considering its sub-archive and the neighbors' sub-archives. This scheme improves the diversity of the swarm because some sub-archives may contain solutions dominated by members of other sub-archives. Nevertheless, the sub-archives also increase the computational complexity of the algorithm because they must be updated and inspected at every generation. Moreover, the computational cost may increase with the number of objectives because of the associated growth in the number of non-dominated solutions.

The authors applied the topology selection scheme to σ-MOPSO [27] and performed an experimental analysis using random-regular graphs. They adopted the ZDT1, ZDT2, and ZDT3 problems from the Zitzler-Deb-Thiele (ZDT) [28] test suite with two objective functions. The experimental results showed that the performance of the MOPSO improves when the number of connections increases. However, we can see that this analysis does not consider problems with different characteristics and with more than three objectives.

On the other hand, Yue et al. proposed in 2018 [13] a topology handling scheme that also adopts sub-archives. In this case, each particle has a Personal Best Archive (PBA), which contains a set of non-dominated solutions found by the particle. Moreover, each neighborhood has a Neighborhood Best Archive (NBA), which contains the non-dominated solutions of the neighbors' PBAs. Therefore, the leader of each particle is chosen considering the PBA and NBA. In particular, they use the non-dominated sorting and Special Crowing Distance [13] as the criteria to choose the leader. This scheme allows the information recollected by a particle to be preserved without external interaction, while the NBAs allow recollecting the best solutions from the neighbors. Therefore, we can fully exploit cognitive and social information. Nevertheless, this topology handling scheme also increases the computational cost of the algorithms because of the archives' maintenance.

The authors performed an experimental analysis using a MOPSO (called the MO_Ring_PSO_SCD) with the latter scheme and the ring topology. These studies are limited to multi-modal multi-objective problems. The experimental results showed that the MO_Ring_PSO_SCD outperforms a simple multi-objective PSO without a topology. Moreover, it showed that the algorithm outperforms state-of-the-art Multi-Objective Evolutionary Algorithms in the solution space but does not perform well in objective function space.

Agarwal et al. [29] inserted two new mechanisms (Levy's flight and the gamma parameter) into MO_Ring_PSO_SCD to improve its performance in multi-modal multi-objective problems. The resulting algorithm is called Enhanced Multi-Objective Particle Swarm Optimization (EMOPSO). The experimental analysis showed that the EMOPSO performs better in solution space than MO_Ring_PSO_SCD and some state-of-the-art algorithms. However, it did not obtain a good performance in objective space. In addition, the authors mentioned that the computational time of EMOPSO needs to be improved.

Finally, Valencia-Rodríguez and Coello Coello [12] proposed two new topology handling schemes that try to avoid the computational burden of the sub-archives. Scheme 1 simulates the behavior of a single-objective PSO. Namely, it analyzes the personal best positions of the particle's neighbors to select the leader and does not consider any external archive information. In contrast, Scheme 2 exploits the global information in the external archive. For this purpose, it associates each element of the archive to each particle of the swarm. If the archive size is less than the swarm size, the archive's elements are assigned anew. Then, a particle will select its leader by examining the external archive's elements assigned to the particle's neighbors.

The authors included Scheme 1 and Scheme 2 in SMPSO. The resulting MOPSOs were called SMPSO-E1 and SMPSO-E2, respectively. In SMPSO-E1, the leader is the first non-dominated personal best position in the neighborhood. While in SMPSO-E2, the leader is chosen by randomly taking two assigned elements from the neighborhood and picking the one with the best CD. The experimental analysis showed that SMPSO-E2 outperformed SMPSO-E1 regardless of the topology, suggesting that incorporating the external archive elements in the topology promotes the particles moving in better directions [12]. Moreover, further experiments revealed that SMPSO-E2 with the wheel topology outperforms the canonical SMPSO in multi- and many-objective problems [12, 14]. However, we can see in the experiments that the difference between the performance of the topologies is statistically significant, but the gap between the indicator values are insignificant.

In summary, the work from Yamamoto et al. [15] and Yue et al. [13] only studied the performance of swarm topologies using problems with limited characteristics. Moreover, their topology handling schemes are computationally expensive because of the sub-archives maintenance. Regarding the Valencia-Rodríguez and Coello Coello work [12, 14], their SMPSO-E2 is not so sensitive to swarm topologies. Furthermore, they used state-of-the-art topologies with which we cannot easily identify an attribute that most influences the MOPSO's performance.

In the following sections, we propose a more sensitive version of the SMPSO-E2 to swarm topologies. In addition, we perform an experimental analysis of the influence of the number of connections between particles in the MOPSO's performance using random regular graphs as topologies.

4. A multi-objective particle swarm optimizer which is more sentitive to its topology

Despite the good results of SMPSO-E2, it is worth noticing that the difference between the performance of topologies in SMPSO-E2 is statistically significant but not remarkable [12, 14]. One reason for this could be that the algorithm only considers two random elements from the neighborhood. Consequently, part of the topology information is omitted. In this work, we want to examine the influence of the number of connections between particles in the MOPSO's performance. Therefore, it is required a MOPSO that could detect modifications to its topology. Hence, SMPSO-E2 needs to be more sensitive to its topology.

In order to amplify the effect of the topology in SMPSO-E2, we modified its leader selection scheme. Instead of comparing only two random neighbors, we evaluated the entire neighborhood and selected the element with the best CD. The computational complexity of the SMPSO-M increases compared to SMPSO-E2 because each neighbor has to be examined instead of only two. However, this situation could make the MOPSO more sensitive to its topology. The final version of SMPSO-E2 with the new modification (called SMPSO-M) is described in Algorithm 2.

The source code of SMPSO-M and SMPSO-E2 can be downloaded from the following link:
`https://computacion.cs.cinvestav.mx/~dvalencia/Implementations/jMetal_SMPSOM_SMPSOE2.zip`.

Algorithm 2 SMPSO-M

```
Require: Polynomial-based mutation parameters ($p_m$ and $\eta_m$), Maximum number of generations ($maxGenerations$),
    Swarm size, Archive size
Ensure: Pareto set approximation
Initialize the swarm
Initialize the external archive
$generation = 0$
while $generation < maxGenerations$ do
    Compute the CD of the archive's elements
    Assign the archive's elements to the particles
    for each particle $x_i$ in the swarm do
        $N_i \leftarrow$ get particles in the neighborhood of $x_i$        ▷ Starts a new selection scheme
        $best \leftarrow N_i[1].assigned\_element$
        for $j \leftarrow 2$ to $N_i.size$ do
            if $N_i[j].assigned\_element.CD > best.CD$ then
                $best \leftarrow N_i[j].assigned\_element$
            end if
        end for
        $\boldsymbol{x}_{l_i} \leftarrow best$        ▷ Ends new selection scheme
        Compute the velocity using equation (5) and multiply it by equation (6)
        Set a bound on the resulting velocity using equation (8)
        Compute the new position using equation (4)
        Apply polynomial-based mutation
        Evaluate the new particle
    end for
    Update the particle's memory and the external archive
    $generation = generation + 1$
end while
return External archive
```

4.1. Experimental analysis

In this section, we compare the performance of SMPSO-M with respect to SMPSO-E2 and SMPSO. In addition, we verify if SMPSO-M is more sensitive to its swarm topology than SMPSO-E2.

4.1.1. Experimental setup

We performed 30 independent runs of SMPSO, SMPSO-M, and SMPSO-E2 using five state-of-the-art topologies: star, tree, wheel, lattice, and ring.

We adopted the DTLZ1-DTLZ7 problems from the Deb-Thiele-Laumanns-Zitzler (DTLZ) test suite [30] and the WFG1-WFG9 problems from the Walking Fish Group (WFG) test suite [31]. We selected these test suites for all the experiments because they offer problems with a wide range of different characteristics. For example, they contain Pareto Front shapes like convex, concave, linear, and disconnected. Moreover, they include multi-modal, unimodal, and deceptive problems.

The number of objectives was set to $m = 3$. In the DTLZ problems, we set the number of variables (as suggested in [30]) to $n = m + k - 1$ where $k = 5$ for DTLZ1, $k = 10$ for DTLZ2-DTLZ6, and $k = 20$ for DTLZ7. Regarding the WFG test problems, the position-related parameters were set to $k = 2 \times (m-1)$ since they must be divisible by $m-1$, the distance-related parameters to $l = 20$, and the number of variables to $n = k + l$.

We set the mutation parameters as in the original SMPSO [21], *i.e.*, $p_m = 1/n$ and $\eta_m = 20$. Moreover, we set the inertia weight to $w = 0.1$. In addition, the number of generations was set to 500, the swarm size to 100, and the archive size to 100.

We evaluated the algorithms' performance by computing the following ratio:

$$I_{NHV} = \frac{I_{HV}(A)}{I_{HV}(PF)} \tag{9}$$

where A is the approximated set, PF is the real Pareto Front, and I_{HV} is the hypervolume indicator which measures the size of the objective space covered by the given set [32]. I_{NHV} measures the relation of the approximated set's hypervolume over the real Pareto Front's hypervolume. Therefore, values close to 1.0 are preferred. We chose the hypervolume indicator because it assesses both the convergence and the maximum spread of the Pareto Front. Furthermore, the hypervolume is the only known indicator that is fully Pareto compliant [33], i.e., this indicator is strictly monotonic with respect to Pareto dominance.

4.1.2. Experimental results and discussion

Table 1 displays the mean and the standard deviation (shown in brackets) of the 30 independent runs of the I_{NHV} indicator. The best mean is highlighted in dark gray, and the second-best is highlighted in light gray. Furthermore, in the last row of each cell, we include the identifiers of the algorithms that have a worse performance than the current algorithm according to the Wilcoxon rank-sum test with a significance level of 5%. The identifier of each algorithm is located next to its name in the table header.

In Table 1, we can observe that SMPSO-M has the best mean values in 10 out of 16 problems. Moreover, we can see that most of the first and second places are on the SMPSO-M side.

We can also see that both algorithms (SMPSO-M and SMPSO-E2) are adequate since the original SMPSO has the best mean only in the WFG9 problem. Furthermore, in the remaining problems, SMPSO is statistically superior to no more than three algorithms.

It is worth noticing that the SMPSO-M with the lattice topology has the most consistent performance, having five best and four second best means. Moreover, when comparing the statistical significance, the SMPSO-M with the lattice topology outperforms many MOPSOs. This result is similar to the one found for the single-objective PSO, where the lattice topology is recommended for unknown problems.

Finally, we analyze the sensitivity of the MOPSOs to their topology. Accordingly, for each MOPSO and problem, we found the best and the worst mean among the five topologies and computed their difference. For example, considering SMPSO-M, the best mean in the DTLZ1 problem is 9.266×10^{-1}, and the worst is 8.306×10^{-1}. Hence, their difference is 0.096. On the other hand, the best mean in the SMPSO-E2 is 9.3319×10^{-1}, and the worst is 9.210×10^{-1}. Therefore, their difference is 0.01219. We can observe that the difference of SMPSO-M is more significant than the one of SMPSO-E2. This may suggest more variation in performance between the best and the worst topology of the SMPSO-M. Hence, we can infer that SMPSO-M is more sensitive to its topology in the DTLZ1 problem.

We continued this process for the remaining problems, and the results are displayed in Fig. 2. The squares correspond to the differences of SMPSO-M and the circles to the differences of SMPSO-E2. We can see that in 13 out of 16 problems, the difference of SMPSO-M is more significant or at least bigger than SMPSO-E2. Therefore, SMPSO-M shows a greater sensitivity to its topology than SMPSO-E2.

In conclusion, the experimental results have shown that SMPSO-M had a better performance than both SMPSO-E2 and SMPSO. In addition, the experiments have shown that SMPSO-M is more sensitive to its topology than SMPSO-E2, making SMPSO-M an excellent option for our experimental analysis.

5. Influence of random regular graphs in the performance of a multi-objective particle swarm optimizer

This work aims to grasp the influence of swarm topologies in MOPSOs. Therefore, we present in this section an experimental analysis of the effect of the number of connections among particles in the SMPSO-M's performance. This section is organized as follows. We first define the swarm topologies that we adopted to make this analysis: the random regular graphs. Then, we describe the experimental setup. Finally, we present the experimental results and the discussion.

Table 1: Mean and standard deviation of the I_{NHV} indicator values over 30 independent runs of SMPSO-E2, SMPSO-M, and SMPSO. The best means are highlighted in dark gray, and the second-best in light gray. The last row of each cell shows the algorithms that had worse performance according to the Wilcoxon rank-sum test with a significance level of 5%.

	SMPSO (0)	SMPSO-M					SMPSO-E2				
		Lattice (1)	Ring (2)	Star (3)	Tree (4)	Wheel (5)	Lattice (6)	Ring (7)	Star (8)	Tree (9)	Wheel (10)
DTLZ1	9.218e-01 (4.29e-03) 3	9.209e-01 (3.56e-02) 0,2,3,6,8,9	9.101e-01 (4.59e-02) 3	8.306e-01 (8.90e-02)	9.266e-01 (5.37e-03) 0,2,3,6,8,9	9.054e-01 (6.48e-02) 0,2,3,8	9.233e-01 (6.39e-03) 3	9.2661e-01 (4.15e-03) 0,2,3,6,8,9	9.210e-01 (5.84e-03) 3	9.241e-01 (5.00e-03) 3,8	9.3319e-01 (6.20e-03) 0,1,2,3,4,5,6,7,8,9
DTLZ2	8.546e-01 (8.36e-03) 3	8.687e-01 (9.33e-03) 0,3,6,7,8,9	8.656e-01 (1.09e-02) 0,3,6,7,8,9	8.237e-01 (3.23e-02)	8.667e-01 (7.38e-03) 0,3,6,7,8,9	8.7115e-01 (6.60e-03) 0,2,3,4,6,7,8,9	8.572e-01 (6.99e-03) 0,3,8	8.590e-01 (7.65e-03) 0,3,8	8.538e-01 (7.72e-03) 3	8.561e-01 (8.89e-03) 3	8.6981e-01 (8.15e-03) 0,3,6,7,8,9
DTLZ3	8.2458e-01 (1.21e-01) 3	7.003e-01 (2.43e-01) 3	7.682e-01 (2.05e-01) 0,1,3,6,7,8,9	4.420e-01 (1.56e-01)	8.232e-01 (1.31e-01) 0,1,3,6,7,8,9	7.710e-01 (1.91e-01) 3	8.058e-01 (1.59e-01) 3	7.887e-01 (1.66e-01) 3	8.147e-01 (1.37e-01) 3	8.4280e-01 (9.31e-02) 3	7.572e-01 (2.06e-01) 0,3
DTLZ4	8.705e-01 (7.78e-03) 3	8.7498e-01 (8.06e-03) 0,2,3,6,7,8,9	8.723e-01 (8.64e-03) 3	8.491e-01 (2.54e-02)	8.739e-01 (7.88e-03) 3,6,7,8,9	8.702e-01 (6.60e-02) 0,1,2,3,4,6,7,8,9	8.703e-01 (4.89e-03) 3	8.698e-01 (5.77e-03) 3	8.688e-01 (8.44e-03) 3	8.700e-01 (7.50e-03) 3	8.8151e-01 (9.02e-03) 0,1,2,3,4,6,7,8,9
DTLZ5	9.875e-01 (2.61e-04) 3	9.8807e-01 (1.76e-04) 0,3,4,5,6,7,8,9,10	9.8806e-01 (1.93e-04) 0,3,4,5,6,7,8,9,10	9.783e-01 (6.71e-03)	9.879e-01 (1.92e-04) 0,3,6,8,9	9.877e-01 (6.66e-04) 0,3,8	9.877e-01 (2.29e-04) 0,3,8	9.879e-01 (1.67e-04) 0,3,6,8,9	9.875e-01 (2.51e-04) 3	9.878e-01 (2.19e-04) 0,3,8	9.879e-01 (3.32e-04) 0,3,6,8,9
DTLZ6	9.882e-01 (1.18e-04) 3,5	9.883e-01 (1.19e-04) 0,3,5,8	9.8837e-01 (1.16e-04) 0,1,3,4,5,6,8,9	9.746e-01 (4.02e-02) 5	9.883e-01 (1.37e-04) 0,3,5,8	9.049e-01 (1.02e-01)	9.883e-01 (1.29e-04) 0,3,5,8	9.883e-01 (9.58e-05) 0,1,3,5,6,8,9	9.881e-01 (1.47e-04) 3,5	9.883e-01 (1.36e-04) 0,3,5,8	9.8834e-01 (1.37e-04) 0,1,3,4,5,6,8
DTLZ7	8.981e-01 (1.10e-02)	9.0213e-01 (8.34e-03) 4,5,9,10	9.013e-01 (1.01e-02) 4,5,9	9.1699e-01 (1.54e-02) 0,1,2,4,5,6,7,8,9,10	8.960e-01 (8.68e-03)	8.854e-01 (3.01e-02)	8.974e-01 (1.08e-02)	8.985e-01 (8.82e-03) 5,9	8.983e-01 (7.72e-03) 5,9	8.952e-01 (9.71e-03)	8.970e-01 (1.53e-02)
WFG1	3.129e-01 (2.15e-03)	3.209e-01 (2.93e-03) 0,6,8,9	3.226e-01 (2.62e-03) 0,1,6,8,9	3.2953e-01 (1.05e-02) 0,1,2,4,5,6,7,8,9,10	3.225e-01 (3.23e-03) 0,1,6,8,9	3.2311e-01 (2.76e-03) 0,1,6,7,8,9	3.183e-01 (2.77e-03) 0,8	3.215e-01 (2.88e-03) 0,6,8,9	3.128e-01 (1.93e-03)	3.190e-01 (2.56e-03) 0,8	3.214e-01 (3.41e-03) 0,6,8,9
WFG2	8.987e-01 (1.40e-02) 5,8,10	9.2003e-01 (1.25e-02) 0,2,3,4,5,6,7,8,9,10	9.057e-01 (1.03e-02) 0,5,6,7,8,9,10	9.088e-01 (2.05e-02) 0,5,6,7,8,9,10	9.1037e-01 (1.02e-02) 0,5,6,7,8,9,10	8.885e-01 (1.20e-02) 10	8.971e-01 (1.12e-02) 5,10	8.954e-01 (1.42e-02) 5,10	8.926e-01 (1.13e-02) 10	8.963e-01 (1.31e-02) 5,10	8.799e-01 (1.21e-02)
WFG3	7.300e-01 (4.55e-02) 3	7.4725e-01 (4.80e-02) 3,5,8,10	7.447e-01 (5.04e-02) 3,8,10	5.984e-01 (1.23e-01)	7.332e-01 (4.22e-02) 3	7.287e-01 (3.68e-02) 3	7.434e-01 (3.09e-02) 3,5,8,10	7.427e-01 (5.72e-02) 3,10	7.215e-01 (3.74e-02) 3	7.4495e-01 (3.64e-02) 3,5,8,10	7.160e-01 (4.70e-02) 3
WFG4	7.351e-01 (1.06e-02)	7.4770e-01 (1.16e-02) 0,2,6,7,8,9,10	7.413e-01 (1.11e-02) 0,6,7,8	7.377e-01 (6.17e-02) 0,6,7,8,9	7.449e-01 (1.10e-02) 0,6,7,8,9	7.4736e-01 (1.02e-02) 0,2,6,7,8,9,10	7.337e-01 (5.89e-03)	7.342e-01 (1.21e-02)	7.326e-01 (9.78e-03)	7.361e-01 (1.05e-02)	7.406e-01 (1.40e-02) 0,6,7,8
WFG5	7.165e-01 (2.13e-02) 3,5	7.3303e-01 (2.72e-02) 0,3,5,6,7,8,9	7.319e-01 (2.98e-02) 0,3,5,6,7,8,9	6.629e-01 (5.27e-02)	7.274e-01 (2.00e-02) 0,3,5,6,8,9	6.874e-01 (2.68e-02) 3	7.166e-01 (2.11e-02) 3,5	7.201e-01 (2.45e-02) 3,5	7.098e-01 (2.66e-02) 3,5	7.113e-01 (1.58e-02) 3,5	7.5579e-01 (2.40e-02) 0,1,2,3,4,5,6,7,8,9
WFG6	8.155e-01 (1.29e-02) 3,5,10	8.219e-01 (1.85e-02) 0,3,5,8,10	8.226e-01 (1.12e-02) 0,3,5,8,10	7.201e-01 (2.29e-02)	8.2577e-01 (1.38e-02) 0,3,5,8,10	7.434e-01 (1.84e-02) 3	8.228e-01 (1.28e-02) 0,3,5,8,10	8.216e-01 (1.54e-02) 0,3,5,10	8.157e-01 (1.47e-02) 3,5,10	8.2792e-01 (1.30e-02) 0,3,5,8,10	8.060e-01 (1.85e-02) 3,5
WFG7	6.675e-01 (1.85e-02)	7.4392e-01 (2.02e-02) 0,2,4,5,6,7,8,9,10	7.205e-01 (2.16e-02) 0,6,7,8,9,10	7.6234e-01 (3.40e-02) 0,1,2,4,5,6,7,8,9,10	7.236e-01 (1.46e-02) 0,5,6,7,8,9,10	7.158e-01 (2.79e-02) 0,6,7,8,9,10	6.811e-01 (1.76e-02) 0,8	6.874e-01 (2.09e-02) 0,8	6.645e-01 (1.99e-02)	6.864e-01 (2.27e-02) 0,8	6.991e-01 (2.18e-02) 0,6,7,8,9
WFG8	5.352e-01 (1.37e-02)	5.6896e-01 (1.64e-02) 0,2,3,4,5,6,7,8,9,10	5.524e-01 (1.55e-02) 0,6,8,9	5.497e-01 (5.91e-02)	5.518e-01 (2.08e-02) 0,6,8	5.5371e-01 (1.73e-02) 0,6,8	5.392e-01 (1.95e-02)	5.504e-01 (1.72e-02) 0,6,8	5.317e-01 (1.94e-02)	5.462e-01 (1.36e-02) 0,6,8	5.529e-01 (1.85e-02) 0,6,8
WFG9	7.6529e-01 (2.74e-02) 1,2,3,4,5,8,9,10	7.375e-01 (3.19e-02) 3,5	7.516e-01 (3.15e-02) 1,3,5	6.906e-01 (3.18e-02)	7.513e-01 (3.13e-02) 3,5	7.167e-01 (2.50e-02) 3	7.5900e-01 (2.90e-02) 1,3,5	7.515e-01 (2.93e-02) 1,3,5	7.524e-01 (2.68e-02) 1,3,5	7.460e-01 (3.13e-02) 3,5	7.528e-01 (2.90e-02) 1,3,5

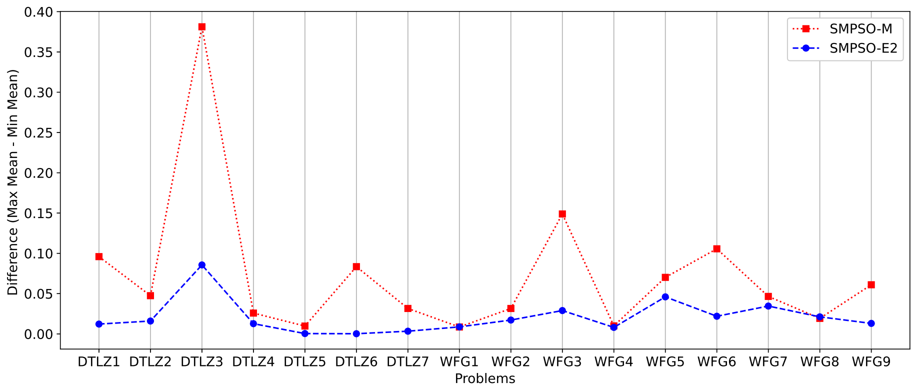


Figure 2: Difference between the maximum mean of I_{NHV} minus the minimum mean of I_{NHV} among all the topologies for each problem. The squares represent SMPSO-M's values, and the circles represent the SMPSO-E2's values.

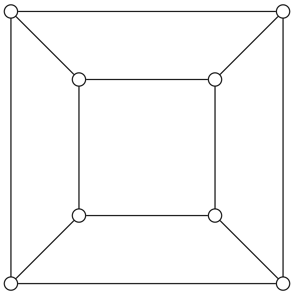


Figure 3: Example of a 3-regular graph

5.1. *Random regular graphs as topologies*

A regular graph is a graph where each vertex has the same number of incident edges (i.e., it has the same degree) [34]. A graph where each vertex has degree k is called k-regular. Fig. 3 displays an example of a 3-regular graph. Due to its properties, the regular graph allows us to control the number of connections in each vertex. Therefore, they are suitable for our experiments.

There are some methods to create regular graphs with a fixed pattern (see, for example, [15]); however, we decided to create them randomly to avoid skewing the experiments to a fixed shape. For simplicity, we generate these random regular graphs using the Steger and Wormald algorithm [35] included in the NetworkX Python package [36].

5.2. *Experimental setup*

We adopted the DTLZ1-DTLZ4 and DTLZ7 problems from the DTLZ test suite and the WFG1-WFG9 problems from the WFG test suite. We omitted the DTLZ5 and DTLZ6 problems because their behavior is unknown in more than three dimensions. As we mentioned, the DTLZ and WFG test suites offer problems with many characteristics. Moreover, they are scalable in the number of objectives. Therefore, we selected them for our analysis. We used the same configuration as in Section 4.1.1, but in this case, we set the number of objectives to $m = 3, 5, 7, 10$.

Regarding SMPSO-M, we used the same inertia weight and mutation parameters from Section 4.1.1. For 3 and 5 objectives, the swarm size and archive size were set to 100. Moreover, for 7 and 10 objectives, these parameters were set to 200. We increased the archive and swarm's size on seven and ten objectives because there is more space to cover on higher dimensionality. In all cases, we used 500 generations as the stopping criterion.

We used regular graphs with different degrees to analyze their influence on the MOPSOs' performance. For a swarm size of 100, we used graphs with degrees 20, 40, 60, and 80. Moreover, for 200, we used graphs with degrees 40, 80, 120, and 160. We generated four random regular graphs for each degree and swarm size. Hence, we performed 30 independent runs of SMPSO-M with 32 different regular graphs. For the performance assessment, we used the I_{NHV} indicator as in Section 4.1.1.

5.3. Experimental results and discussion

We display the experimental results of the I_{NHV} indicator in boxplots graphs of Appendix A (see Figures A.4 to A.17). Furthermore, the mean and standard deviation of I_{NHV} are presented in Tables B.2, B.3, B.4, and B.5 of Appendix B.

In 10 out of 14 problems, we observe that a higher graph degree produces a higher dispersion of the I_{NHV} indicator values in at least one of the four tested objectives. In particular, this happens in all the tested objectives of WFG1 and WFG4 (see Figures A.9 and A.12). Higher dispersion of the I_{NHV} values implies that the algorithm becomes unstable. Therefore, the algorithm could obtain a good approximation in one run and a poor one in another. Furthermore, we observe that in problems DTLZ3 and WFG3, this instability produces outliers with good values when the performance of the rest of the algorithms decays completely (see Figures A.6 and A.11). Thus, the instability can be beneficial in some cases.

We believe that a higher number of connections increases the instability of the algorithms because a topology with many connections can improve the exploitation of the areas, which provokes that the algorithm gets easily trapped in local fronts. Therefore, if the algorithm is close to the real Pareto front, it will obtain a good approximation. On the other hand, it will obtain a poor result.

It is also worth mentioning that the medians of the topologies with the same degree are close to each other regardless of the number of objectives. In contrast, if we compare two topologies with a different degree, they are likely to have distant medians. This situation suggests that regular graph topologies with the same degree have a similar behavior. Therefore, the number of connections impacts the behavior of SMPSO-M.

We do not observe a consistent behavior regarding the degree and the MOPSO's performance. In the following problems, the topologies with the highest degree have the best performance: DTLZ1, DTLZ2, WFG5, WFG6, and WFG9 with three objectives; DTLZ2, DTLZ4, DTLZ7, and WFG5 with five objectives; DTLZ2, DTLZ4, and DTLZ7 with seven objectives; DTLZ4 and DTLZ7 with ten objectives. Moreover, the topologies with the lowest degree have the best performance in the following problems: DTLZ3 with five objectives; WFG1 and WFG6 with five, seven, and ten objectives. Remarkably, in problem WFG6, the MOPSO's behavior changes depending on the number of objectives. For three objectives, a higher degree is beneficial. However, for the remaining number of objectives, the opposite happens (see Figure A.14). We did not find any common characteristics in the problems with similar behavior. We think this may happen because other graph aspects could influence MOPSO's behavior more. Therefore, we can only say that the influence of the graph degree on the MOPSOs' performance depends on the problem and the number of objectives.

Finally, in the following three problems, we could not distinguish a variation in the performance of the topologies with different degrees: DTLZ1 with ten objectives, DTLZ3 with three objectives, and WFG2 with two objectives. Consequently, in the majority of the problems, the topologies change the behavior of SMPSO-M. This fact strengthens the idea that topologies influence the performance of a MOPSO.

6. Conclusions and future work

This work analyzed the influence of the number of connections among particles in the performance of a MOPSO. For this sake, we first modified SMPSO-E2 (a MOPSO that implements a topology handling scheme) to be more sensitive to its swarm topology. The resulting MOPSO was called SMPSO-M and was tested using a variety of problems. The experimental results showed that SMPSO-M is more sensitive to its topology than SMPSO and had a better performance (using a lattice topology) than SMPSO and SMPSO-E2.

Due to these results, we continued the analysis using SMPSO-M. We tested 32 random regular graphs using many-objective problems. We observed that a higher degree in many problems makes the algorithm's performance unstable. This phenomenon could be because topologies with higher degrees exploit areas well, being trapped in local Fronts. Therefore, if the algorithm is close to the real Pareto Front, it will perform satisfactorily. On the contrary, if it is far, it will have a poor performance.

Furthermore, our experiments showed that random regular graphs with different degrees have similar behavior. Moreover, we concluded that the influence of the number of connections among particles on the MOPSO's performance depends on the problem and the number of objectives. However, we could not identify a specific number of connections that influenced the same type of problem. We think this may happen because other graph aspects could impact the MOPSO's behavior more.

Finally, our experiments confirm that swarm topologies impact the MOPSO's performance since the difference was not significant only in three of the problems adopted.

As part of our future work, we will examine other properties of the topologies that could change the performance of a MOPSO. For example, the distribution degree or the diameter of a graph.

Acknowledgements

The first author acknowledges support from CINVESTAV-IPN and CONACyT to pursue graduate studies in computer science. The second author acknowledges support from CONACyT grant no. 1920 and from SEP-Cinvestav grant (application no. 4).

Competing Interests

The authors have no competing interests to declare that are relevant to the content of this article.

Appendix A. Boxplots

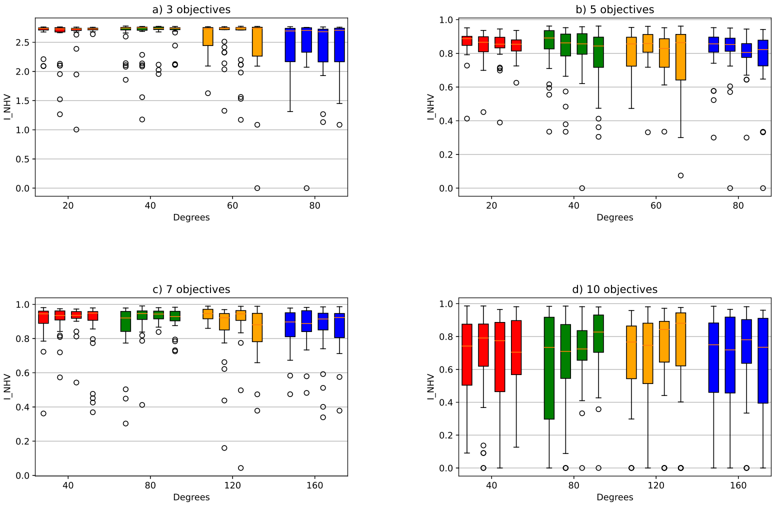


Figure A.4: I_{NHV} indicator values over 30 independent runs of the SMPSO-M using the DTLZ1 problem in 3, 5, 7, and 10 objectives. In the figures found at the top, the regular graph degrees used are 20, 40, 60, and 80. Moreover, the regular graph degrees used in the figures found at the bottom are 40, 80, 120, and 160.

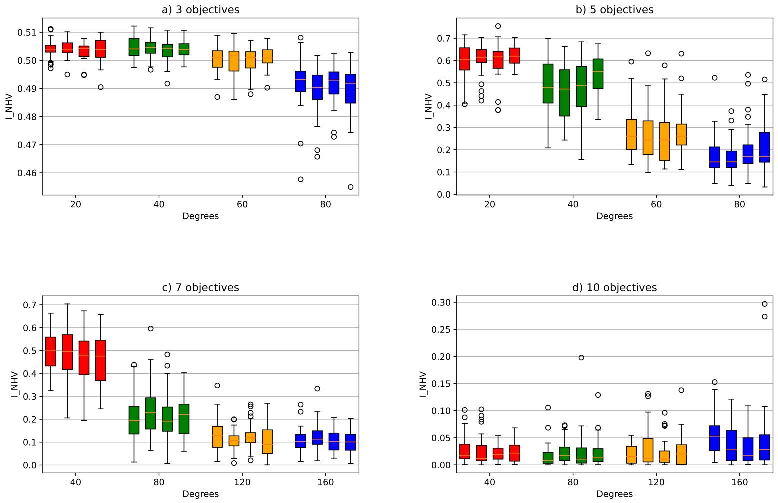


Figure A.5: I_{NHV} indicator values over 30 independent runs of the SMPSO-M using the DTLZ2 problem in 3, 5, 7, and 10 objectives. In the figures found at the top, the regular graph degrees used are 20, 40, 60, and 80. Moreover, the regular graph degrees used in the figures found at the bottom are 40, 80, 120, and 160.

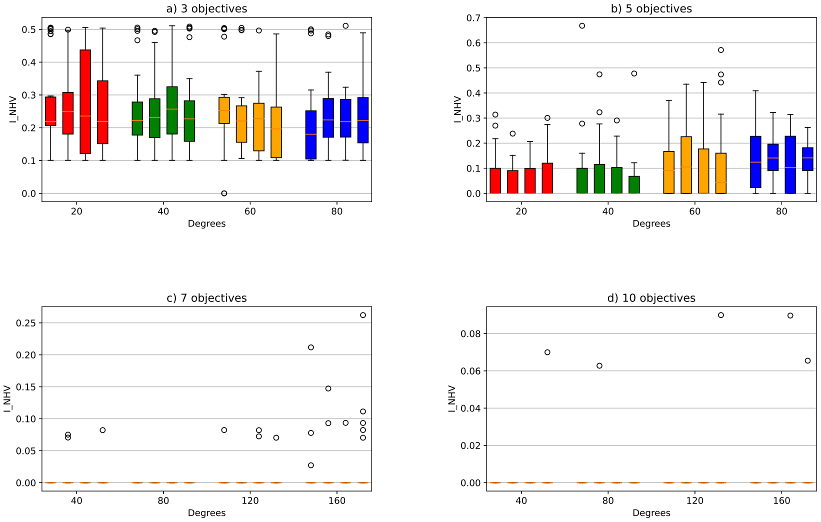
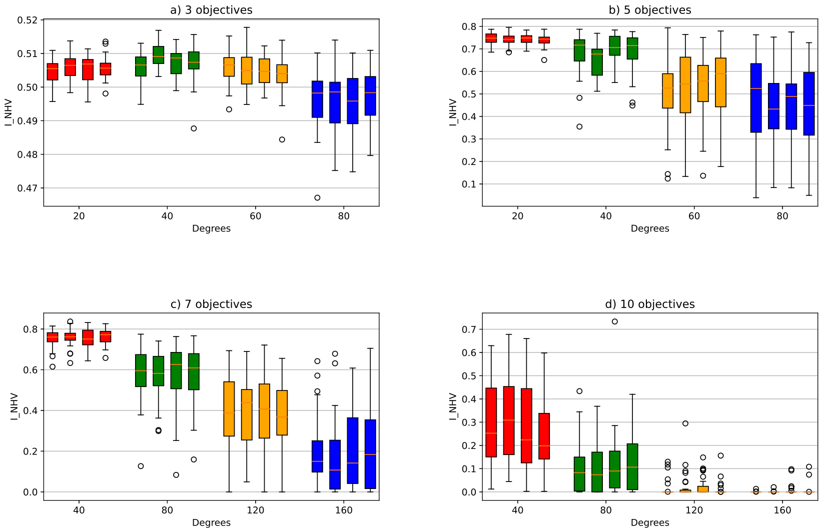


Figure A.6: I_{NHV} indicator values over 30 independent runs of the SMPSO-M using the DTLZ3 problem in 3, 5, 7, and 10 objectives. In the figures found at the top, the regular graph degrees used are 20, 40, 60, and 80. Moreover, the regular graph degrees used in the figures found at the bottom are 40, 80, 120, and 160.

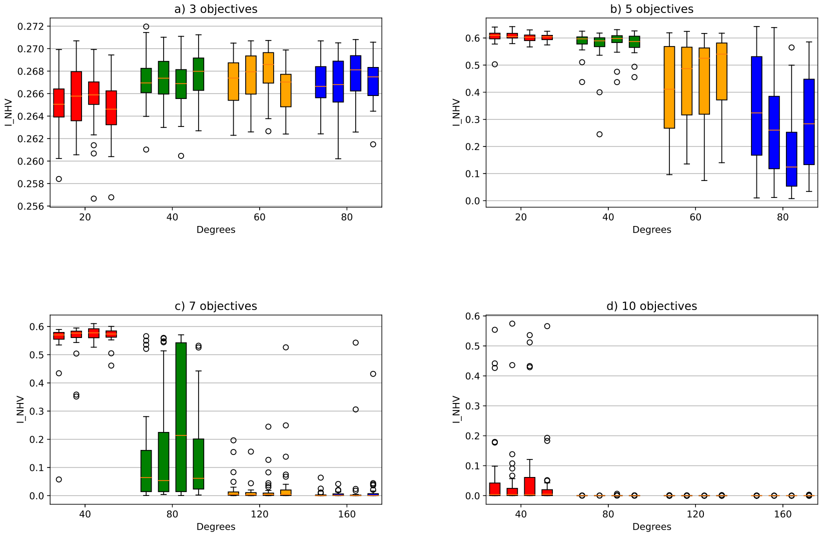

a) 3 objectives

b) 5 objectives

c) 7 objectives

d) 10 objectives

Figure A.7: I_{NHV} indicator values over 30 independent runs of the SMPSO-M using the DTLZ4 problem in 3, 5, 7, and 10 objectives. In the figures found at the top, the regular graph degrees used are 20, 40, 60, and 80. Moreover, the regular graph degrees used in the figures found at the bottom are 40, 80, 120, and 160.


a) 3 objectives

b) 5 objectives

c) 7 objectives

d) 10 objectives

Figure A.8: I_{NHV} indicator values over 30 independent runs of the SMPSO-M using the DTLZ7 problem in 3, 5, 7, and 10 objectives. In the figures found at the top, the regular graph degrees used are 20, 40, 60, and 80. Moreover, the regular graph degrees used in the figures found at the bottom are 40, 80, 120, and 160.

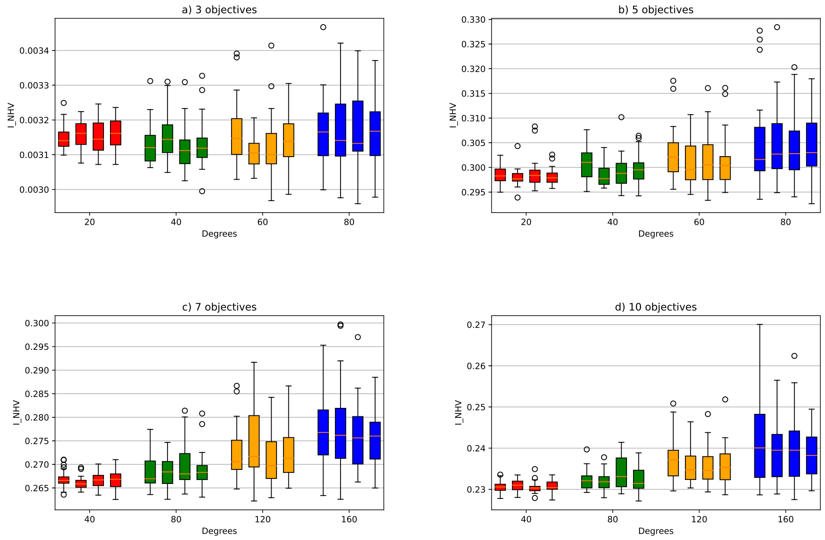


Figure A.9: I_{NHV} indicator values over 30 independent runs of the SMPSO-M using the WFG1 problem in 3, 5, 7, and 10 objectives. In the figures found at the top, the regular graph degrees used are 20, 40, 60, and 80. Moreover, the regular graph degrees used in the figures found at the bottom are 40, 80, 120, and 160.

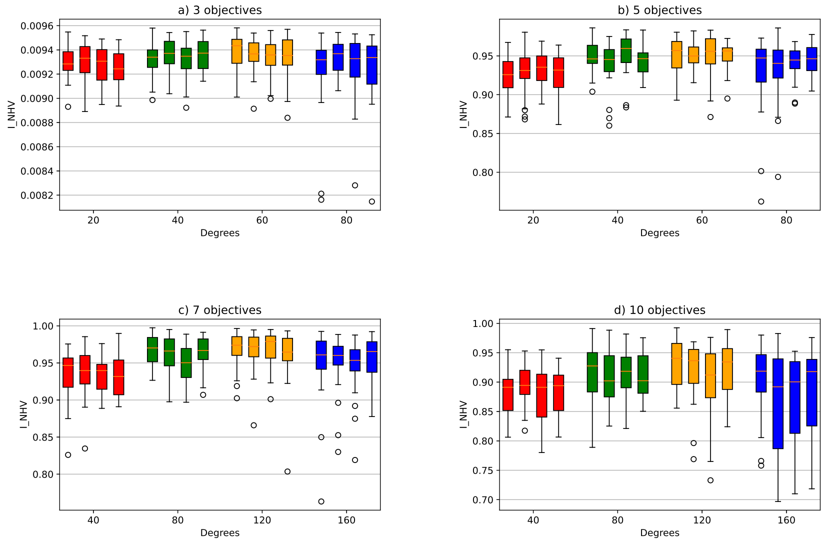


Figure A.10: I_{NHV} indicator values over 30 independent runs of the SMPSO-M using the WFG2 problem in 3, 5, 7, and 10 objectives. In the figures found at the top, the regular graph degrees used are 20, 40, 60, and 80. Moreover, the regular graph degrees used in the figures found at the bottom are 40, 80, 120, and 160.

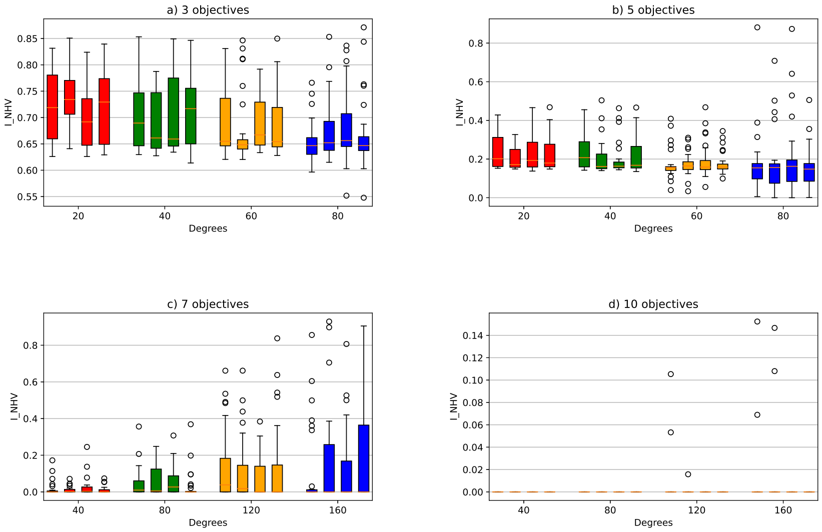


Figure A.11: I_{NHV} indicator values over 30 independent runs of the SMPSO-M using the WFG3 problem in 3, 5, 7, and 10 objectives. In the figures found at the top, the regular graph degrees used are 20, 40, 60, and 80. Moreover, the regular graph degrees used in the figures found at the bottom are 40, 80, 120, and 160.

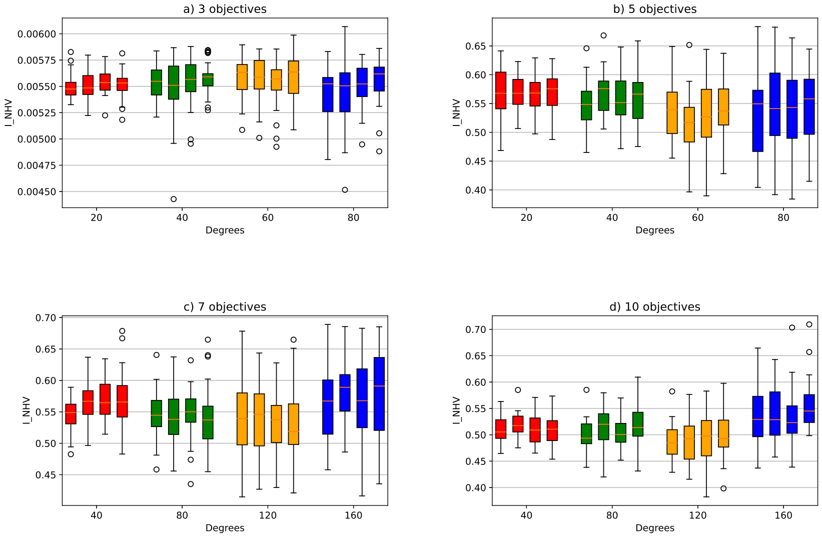


Figure A.12: I_{NHV} indicator values over 30 independent runs of the SMPSO-M using the WFG4 problem in 3, 5, 7, and 10 objectives. In the figures found at the top, the regular graph degrees used are 20, 40, 60, and 80. Moreover, the regular graph degrees used in the figures found at the bottom are 40, 80, 120, and 160.

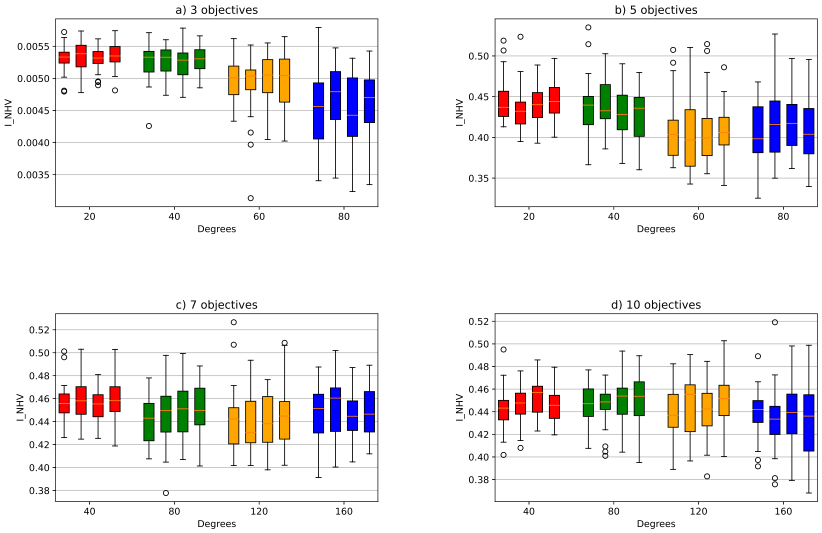


Figure A.13: I_{NHV} indicator values over 30 independent runs of the SMPSO-M using the WFG5 problem in 3, 5, 7, and 10 objectives. In the figures found at the top, the regular graph degrees used are 20, 40, 60, and 80. Moreover, the regular graph degrees used in the figures found at the bottom are 40, 80, 120, and 160.

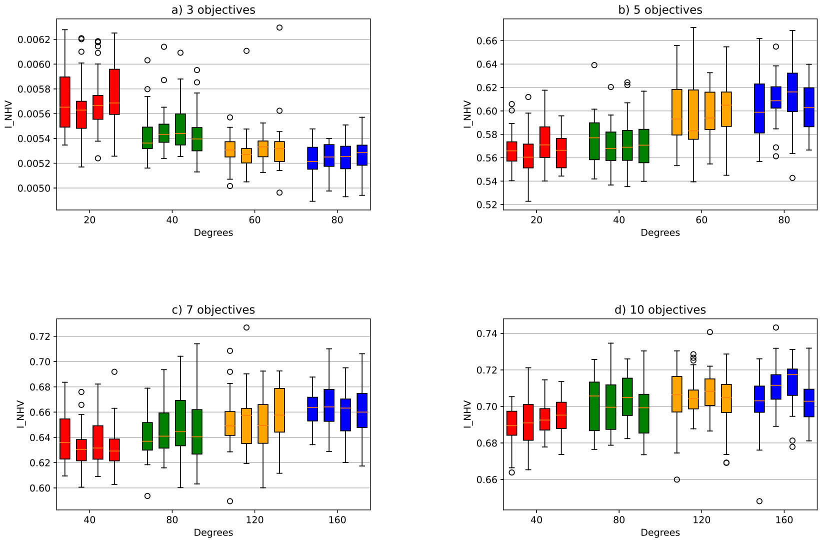


Figure A.14: I_{NHV} indicator values over 30 independent runs of the SMPSO-M using the WFG6 problem in 3, 5, 7, and 10 objectives. In the figures found at the top, the regular graph degrees used are 20, 40, 60, and 80. Moreover, the regular graph degrees used in the figures found at the bottom are 40, 80, 120, and 160.

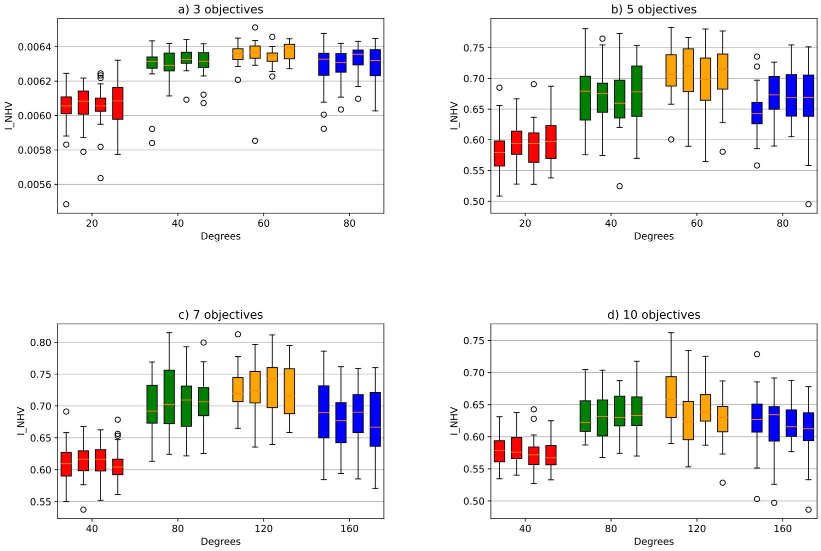


Figure A.15: I_{NHV} indicator values over 30 independent runs of the SMPSO-M using the WFG7 problem in 3, 5, 7, and 10 objectives. In the figures found at the top, the regular graph degrees used are 20, 40, 60, and 80. Moreover, the regular graph degrees used in the figures found at the bottom are 40, 80, 120, and 160.

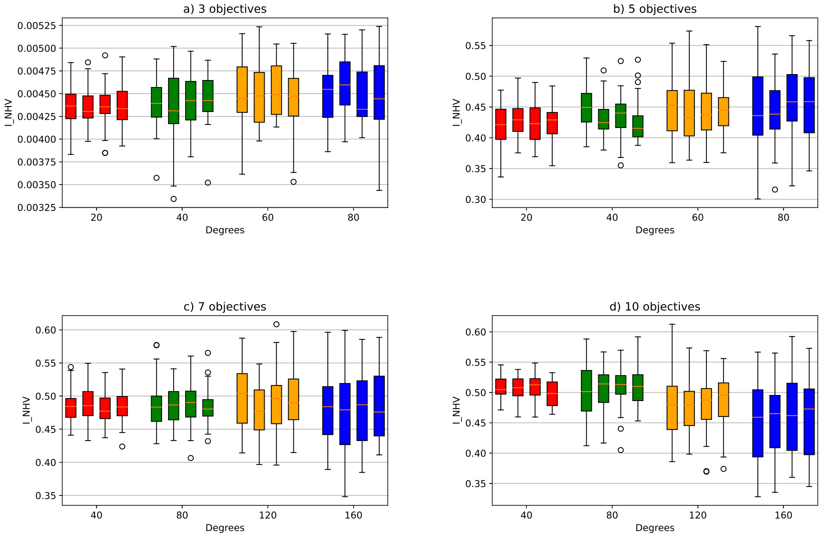


Figure A.16: I_{NHV} indicator values over 30 independent runs of the SMPSO-M using the WFG8 problem in 3, 5, 7, and 10 objectives. In the figures found at the top, the regular graph degrees used are 20, 40, 60, and 80. Moreover, the regular graph degrees used in the figures found at the bottom are 40, 80, 120, and 160.

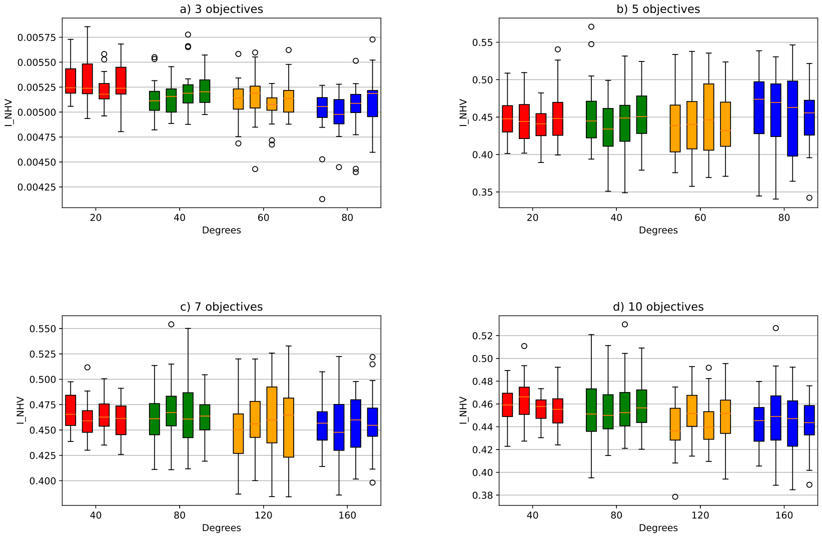


Figure A.17: I_{NHV} indicator values over 30 independent runs of the SMPSO-M using the WFG9 problem in 3, 5, 7, and 10 objectives. In the figures found at the top, the regular graph degrees used are 20, 40, 60, and 80. Moreover, the regular graph degrees used in the figures found at the bottom are 40, 80, 120, and 160.

Appendix B. Tables

References

[1] J. Kennedy, R. Eberhart, Particle Swarm Optimization, in: Proceedings of the 1995 IEEE International Conference on Neural Networks (ICNN 1995), Vol. 4, 1995, pp. 1942–1948.

[2] R. Mendes, Population topologies and their influence in particle swarm performance, Ph.D. thesis, Departamento de Informática, Escola de Engenharia, Universidade do Minho (Apr. 2004).

[3] R. Mendes, J. Kennedy, J. Neves, The Fully Informed Particle Swarm: Simpler, Maybe Better, IEEE Transactions on Evolutionary Computation 8 (3) (2004) 204–210. doi:10.1109/TEVC.2004.826074.

[4] J. Kennedy, Small Worlds and Mega-Minds: Effects of Neighborhood Topology on Particle Swarm Performance, in: Proceedings of the 1999 Congress on Evolutionary Computation (CEC 1999), Vol. 3, 1999, pp. 1931–1938. doi:10.1109/CEC.1999.785509.

[5] J. Kennedy, R. Mendes, Population Structure and Particle Swarm Performance, in: Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002), Vol. 2, 2002, pp. 1671–1676. doi:10.1109/CEC.2002.1004493.

Table B.2: Mean and standard deviation of the I_{NHV} indicator values over 30 independent runs of SMPSO-M. The best means are highlighted in dark gray, and the second-best in light gray. The first four columns display the algorithms that use 20-regular graphs, and the rest use 40-regular graphs.

		20-regular graphs				40-regular graphs			
	m	Graph 1	Graph 2	Graph 3	Graph 4	Graph 1	Graph 2	Graph 3	Graph 4
DTLZ1	3	2.672e+0 (1.8e-1)	2.554e+0 (3.8e-1)	2.630e+0 (3.4e-1)	2.725e+0 (2.5e-2)	2.644e+0 (2.4e-1)	2.574e+0 (3.8e-1)	2.669e+0 (2.1e-1)	2.667e+0 (1.9e-1)
	5	8.616e-1 (9.6e-2)	8.408e-1 (9.2e-2)	8.388e-1 (1.0e-1)	8.382e-1 (6.6e-2)	8.421e-1 (1.4e-1)	8.041e-1 (1.6e-1)	8.161e-1 (1.8e-1)	7.805e-1 (1.7e-1)
DTLZ2	3	5.041e-1 (3.4e-3)	5.041e-1 (2.8e-3)	5.032e-1 (2.9e-3)	5.04e-1 (4.2e-3)	5.043e-1 (3.9e-3)	5.044e-1 (3.2e-3)	5.034e-1 (4.1e-3)	5.041e-1 (3.4e-3)
	5	5.925e-1 (8.6e-2)	6.032e-1 (7.0e-2)	5.954e-1 (8.3e-2)	6.200e-1 (4.4e-2)	4.887e-1 (1.2e-1)	4.660e-1 (1.2e-1)	4.736e-1 (1.3e-1)	5.344e-1 (9.9e-2)
DTLZ3	3	2.737e-1 (1.3e-1)	2.687e-1 (1.3e-1)	2.716e-1 (1.6e-1)	2.553e-1 (1.3e-1)	2.471e-1 (1.1e-1)	2.547e-1 (1.2e-1)	2.724e-1 (1.3e-1)	2.441e-1 (1.2e-1)
	5	5.257e-2 (9.e-2)	4.009e-2 (6.2e-2)	5.509e-2 (6.4e-2)	6.463e-2 (1.0e-1)	6.466e-2 (1.3e-1)	7.852e-2 (1.2e-1)	6.298e-2 (8.2e-2)	3.843e-2 (9.1e-2)
DTLZ4	3	5.045e-1 (3.9e-3)	5.059e-1 (3.7e-3)	5.052e-1 (4.3e-3)	5.056e-1 (3.4e-3)	5.06e-1 (4.5e-3)	5.094e-1 (3.4e-3)	5.074e-1 (3.9e-3)	5.070e-1 (5.3e-3)
	5	7.455e-1 (2.4e-2)	7.429e-1 (2.7e-2)	7.425e-1 (2.4e-2)	7.375e-1 (2.8e-2)	6.813e-1 (9.1e-2)	*6.503e-1 (7.0e-2)	7.013e-1 (6.7e-2)	6.810e-1 (9.1e-2)
DTLZ7	3	2.65e-1 (2.5e-3)	2.658e-1 (2.8e-3)	2.656e-1 (2.7e-3)	2.645e-1 (2.5e-3)	2.672e-1 (2.3e-3)	2.676e-1 (2.e-3)	2.667e-1 (2.2e-3)	2.674e-1 (2.2e-3)
	5	6.054e-1 (2.4e-2)	6.080e-1 (1.4e-2)	6.016e-1 (1.5e-2)	6.029e-1 (1.2e-2)	5.862e-1 (3.6e-2)	5.680e-1 (7.2e-2)	5.870e-1 (4.e-2)	5.815e-1 (3.6e-2)
WFG1	3	3.148e-3 (3.5e-5)	3.159e-3 (3.7e-5)	3.154e-3 (4.7e-5)	3.159e-3 (4.1e-5)	3.127e-3 (5.3e-5)	3.154e-3 (6.3e-5)	3.116e-3 (6.1e-5)	3.13e-3 (6.5e-5)
	5	2.985e-1 (1.8e-3)	2.98e-1 (1.7e-3)	2.988e-1 (2.8e-3)	2.981e-1 (1.5e-3)	3.005e-1 (3.3e-3)	2.985e-1 (2.3e-3)	2.991e-1 (3.2e-3)	2.997e-1 (3.1e-3)
WFG2	3	9.291e-3 (1.2e-4)	9.305e-3 (1.6e-4)	9.279e-3 (1.5e-4)	9.253e-3 (1.4e-4)	9.31e-3 (1.4e-4)	9.371e-3 (1.1e-4)	9.321e-3 (1.5e-4)	9.349e-3 (1.3e-4)
	5	9.237e-1 (2.5e-2)	9.283e-1 (2.8e-2)	9.328e-1 (2.1e-2)	9.250e-1 (2.7e-2)	9.496e-1 (2.e-2)	9.408e-1 (2.9e-2)	9.538e-1 (2.5e-2)	9.44e-1 (2.0e-2)
WFG3	3	7.215e-1 (6.6e-2)	7.372e-1 (4.9e-2)	7.013e-1 (5.8e-2)	7.220e-1 (6.9e-2)	7.044e-1 (6.4e-2)	6.89e-1 (5.6e-2)	7.063e-1 (7.5e-2)	7.141e-1 (6.3e-2)
	5	2.373e-1 (8.8e-2)	2.042e-1 (5.8e-2)	2.303e-1 (8.4e-2)	2.272e-1 (8.6e-2)	2.414e-1 (9.9e-2)	2.018e-1 (8.5e-2)	1.958e-1 (8.2e-2)	2.113e-1 (8.6e-2)
WFG4	3	5.501e-3 (1.2e-4)	5.504e-3 (1.3e-4)	5.544e-3 (1.2e-4)	5.512e-3 (1.3e-4)	5.542e-3 (1.7e-4)	5.471e-3 (3.1e-4)	5.529e-3 (2.2e-4)	5.59e-3 (1.6e-4)
	5	5.695e-1 (4.2e-2)	5.659e-1 (3.2e-2)	5.657e-1 (3.6e-2)	5.681e-1 (3.7e-2)	5.483e-1 (4.2e-2)	5.685e-1 (3.8e-2)	5.582e-1 (4.7e-2)	5.599e-1 (4.4e-2)
WFG5	3	5.31e-3 (2.0e-4)	5.335e-3 (2.2e-4)	5.307e-3 (1.8e-4)	5.36e-3 (2.1e-4)	5.255e-3 (2.8e-4)	5.260e-3 (2.4e-4)	5.235e-3 (2.6e-4)	5.284e-3 (2.2e-4)
	5	4.440e-1 (2.6e-2)	4.342e-1 (2.6e-2)	4.403e-1 (2.3e-2)	4.467e-1 (2.2e-2)	4.351e-1 (3.7e-2)	4.386e-1 (2.9e-2)	4.311e-1 (3.2e-2)	4.264e-1 (3.3e-2)
WFG6	3	5.728e-3 (2.8e-4)	5.631e-3 (2.4e-4)	5.699e-3 (2.4e-4)	5.728e-3 (2.6e-4)	5.431e-3 (1.9e-4)	5.474e-3 (1.8e-4)	5.507e-3 (2.1e-4)	5.434e-3 (1.9e-4)
	5	5.658e-1 (1.6e-2)	5.627e-1 (2.e-2)	5.726e-1 (1.8e-2)	5.656e-1 (1.4e-2)	5.769e-1 (2.1e-2)	5.693e-1 (1.8e-2)	5.726e-1 (2.2e-2)	5.711e-1 (2.0e-2)
WFG7	3	6.04e-3 (1.4e-4)	6.061e-3 (1.0e-4)	6.056e-3 (1.2e-4)	6.064e-3 (1.5e-4)	6.286e-3 (1.2e-4)	6.296e-3 (7.9e-5)	6.328e-3 (6.3e-5)	6.312e-3 (7.3e-5)
	5	5.826e-1 (3.8e-2)	5.933e-1 (3.3e-2)	5.898e-1 (3.7e-2)	5.998e-1 (3.8e-2)	6.694e-1 (5.0e-2)	6.708e-1 (4.4e-2)	6.674e-1 (5.e-2)	6.775e-1 (4.9e-2)
WFG8	3	4.367e-3 (2.3e-4)	4.355e-3 (2.0e-4)	4.352e-3 (2.4e-4)	4.350e-3 (2.3e-4)	4.394e-3 (2.7e-4)	4.377e-3 (4.0e-4)	4.443e-3 (2.9e-4)	4.454e-3 (2.7e-4)
	5	4.189e-1 (3.1e-2)	4.308e-1 (2.9e-2)	4.251e-1 (3.1e-2)	4.249e-1 (2.9e-2)	4.479e-1 (3.6e-2)	4.308e-1 (3.1e-2)	4.357e-1 (3.3e-2)	4.254e-1 (3.4e-2)
WFG9	3	5.301e-3 (1.8e-4)	5.325e-3 (2.4e-4)	5.214e-3 (1.4e-4)	5.300e-3 (1.9e-4)	5.133e-3 (1.7e-4)	5.127e-3 (1.4e-4)	5.218e-3 (1.9e-4)	5.231e-3 (1.6e-4)
	5	4.487e-1 (2.6e-2)	4.468e-1 (2.9e-2)	4.399e-1 (2.1e-2)	4.532e-1 (3.7e-2)	4.508e-1 (4.1e-2)	4.365e-1 (3.6e-2)	4.459e-1 (3.8e-2)	4.508e-1 (3.3e-2)

Table B.3: Mean and standard deviation of the I_{NHV} indicator values over 30 independent runs of SMPSO-M. The best means are highlighted in dark gray, and the second-best in light gray. The first four columns display the algorithms that use 60-regular graphs, and the rest use 80-regular graphs.

		60-regular graphs				80-regular graphs			
	m	Graph 1	Graph 2	Graph 3	Graph 4	Graph 1	Graph 2	Graph 3	Graph 4
DTLZ1	3	2.566e+0 (3.0e-1)	2.607e+0 (3.1e-1)	2.529e+0 (4.3e-1)	2.468e+0 (5.9e-1)	2.487e+0 (3.6e-1)	2.498e+0 (5.2e-1)	2.404e+0 (4.2e-1)	2.423e+0 (4.4e-1)
	5	8.117e-1 (1.2e-1)	8.419e-1 (1.2e-1)	7.961e-1 (1.3e-1)	7.688e-1 (2.1e-1)	8.157e-1 (1.4e-1)	8.119e-1 (1.7e-1)	7.925e-1 (1.2e-1)	7.517e-1 (2.2e-1)
DTLZ2	3	5.003e-1 (4.7e-3)	5.003e-1 (4.8e-3)	4.997e-1 (4.9e-3)	5.009e-1 (3.7e-3)	4.920e-1 (9.5e-3)	4.894e-1 (8.6e-3)	4.916e-1 (6.7e-3)	4.892e-1 (8.9e-3)
	5	2.865e-1 (1.1e-1)	2.684e-1 (1.2e-1)	2.686e-1 (1.3e-1)	2.847e-1 (1.1e-1)	1.755e-1 (9.4e-2)	1.606e-1 (7.2e-2)	2.006e-1 (1.2e-1)	2.036e-1 (1.1e-1)
DTLZ3	3	2.596e-1 (1.3e-1)	2.276e-1 (1.1e-1)	2.217e-1 (9.2e-2)	2.036e-1 (1.0e-1)	2.116e-1 (1.2e-1)	2.307e-1 (9.8e-2)	2.275e-1 (8.4e-2)	2.455e-1 (1.1e-1)
	5	9.053e-2 (1.0e-1)	1.376e-1 (1.3e-1)	1.01e-1 (1.4e-1)	1.141e-1 (1.6e-1)	1.448e-1 (1.2e-1)	1.480e-1 (8.e-2)	1.225e-1 (1.1e-1)	1.285e-1 (8.2e-2)
DTLZ4	3	5.06e-1 (4.6e-3)	5.053e-1 (5.9e-3)	5.048e-1 (4.2e-3)	5.037e-1 (6.1e-3)	4.961e-1 (8.6e-3)	4.960e-1 (9.0e-3)	4.947e-1 (9.5e-3)	4.972e-1 (8.3e-3)
	5	4.977e-1 (1.5e-1)	5.236e-1 (1.7e-1)	5.333e-1 (1.4e-1)	5.433e-1 (1.5e-1)	4.704e-1 (2.e-1)	4.515e-1 (1.7e-1)	4.499e-1 (1.5e-1)	4.394e-1 (1.7e-1)
DTLZ7	3	2.669e-1 (2.3e-3)	2.676e-1 (2.2e-3)	2.679e-1 (2.3e-3)	2.664e-1 (2.1e-3)	2.669e-1 (1.9e-3)	2.667e-1 (2.6e-3)	2.676e-1 (2.0e-3)	2.671e-1 (1.8e-3)
	5	4.016e-1 (1.6e-1)	4.394e-1 (1.5e-1)	4.392e-1 (1.6e-1)	4.637e-1 (1.5e-1)	3.396e-1 (2.1e-1)	2.696e-1 (1.8e-1)	*1.789e-1 (1.6e-1)	2.902e-1 (1.8e-1)
WFG1	3	3.160e-3 (8.5e-5)	3.108e-3 (4.4e-5)	3.126e-3 (8.6e-5)	3.143e-3 (6.9e-5)	3.157e-3 (1.0e-4)	3.167e-3 (1.1e-4)	3.179e-3 (1.1e-4)	3.166e-3 (1.0e-4)
	5	3.025e-1 (5.2e-3)	3.009e-1 (4.4e-3)	3.015e-1 (5.2e-3)	3.010e-1 (4.9e-3)	3.043e-1 (8.6e-3)	3.051e-1 (7.5e-3)	3.042e-1 (7.e-3)	3.046e-1 (5.7e-3)
WFG2	3	9.377e-3 (1.6e-4)	9.356e-3 (1.4e-4)	9.339e-3 (1.6e-4)	9.330e-3 (1.9e-4)	9.238e-3 (3.2e-4)	9.338e-3 (1.4e-4)	9.268e-3 (2.6e-4)	9.250e-3 (2.6e-4)
	5	9.522e-1 (2.1e-2)	9.505e-1 (1.8e-2)	9.495e-1 (2.7e-2)	9.508e-1 (1.7e-2)	9.289e-1 (4.8e-2)	9.348e-1 (3.8e-2)	9.413e-1 (2.0e-2)	9.443e-1 (2.1e-2)
WFG3	3	6.935e-1 (6.2e-2)	6.749e-1 (6.5e-2)	6.899e-1 (5.2e-2)	6.862e-1 (6.1e-2)	6.557e-1 (3.7e-2)	6.761e-1 (5.7e-2)	6.827e-1 (6.9e-2)	6.661e-1 (6.5e-2)
	5	1.714e-1 (7.5e-2)	1.718e-1 (5.6e-2)	1.884e-1 (8.7e-2)	1.754e-1 (5.5e-2)	1.735e-1 (1.5e-1)	1.702e-1 (1.5e-1)	1.955e-1 (1.9e-1)	1.569e-1 (1.0e-1)
WFG4	3	5.587e-3 (1.9e-4)	5.566e-3 (2.2e-4)	5.519e-3 (2.1e-4)	5.592e-3 (2.2e-4)	5.431e-3 (2.4e-4)	5.440e-3 (2.9e-4)	5.502e-3 (1.9e-4)	5.551e-3 (2.2e-4)
	5	5.367e-1 (4.9e-2)	5.146e-1 (5.2e-2)	5.3e-1 (6.6e-2)	5.376e-1 (5.6e-2)	5.275e-1 (6.8e-2)	5.456e-1 (7.8e-2)	5.423e-1 (7.1e-2)	5.46e-1 (6.7e-2)
WFG5	3	4.976e-3 (3.1e-4)	4.903e-3 (4.7e-4)	4.986e-3 (3.9e-4)	4.957e-3 (4.2e-4)	4.542e-3 (5.9e-4)	4.658e-3 (5.3e-4)	4.475e-3 (5.8e-4)	4.577e-3 (5.6e-4)
	5	4.090e-1 (3.9e-2)	4.054e-1 (4.9e-2)	4.07e-1 (4.e-2)	4.064e-1 (3.e-2)	4.056e-1 (3.6e-2)	4.165e-1 (4.2e-2)	4.159e-1 (3.4e-2)	4.049e-1 (3.8e-2)
WFG6	3	5.306e-3 (1.2e-4)	5.290e-3 (1.8e-4)	5.326e-3 (9.5e-5)	5.329e-3 (2.2e-4)	5.216e-3 (1.4e-4)	5.246e-3 (1.2e-4)	5.232e-3 (1.4e-4)	5.272e-3 (1.3e-4)
	5	5.979e-1 (2.5e-2)	5.972e-1 (3.5e-2)	5.97e-1 (2.e-2)	6.04e-1 (2.5e-2)	6.03e-1 (2.9e-2)	6.108e-1 (1.9e-2)	6.135e-1 (2.7e-2)	6.049e-1 (2.0e-2)
WFG7	3	6.359e-3 (5.3e-5)	6.352e-3 (1.0e-4)	6.341e-3 (4.5e-5)	6.374e-3 (5.e-5)	6.291e-3 (1.3e-4)	6.291e-3 (9.1e-5)	6.33e-3 (8.2e-5)	6.305e-3 (9.8e-5)
	5	7.089e-1 (3.8e-2)	7.119e-1 (4.5e-2)	6.927e-1 (5.0e-2)	7.101e-1 (4.1e-2)	*6.456e-1 (3.8e-2)	6.737e-1 (3.5e-2)	6.726e-1 (4.1e-2)	6.646e-1 (5.6e-2)
WFG8	3	4.512e-3 (3.6e-4)	4.493e-3 (3.2e-4)	4.550e-3 (2.8e-4)	4.434e-3 (3.7e-4)	4.521e-3 (3.3e-4)	4.601e-3 (3.1e-4)	4.479e-3 (3.4e-4)	4.448e-3 (4.1e-4)
	5	4.499e-1 (4.9e-2)	4.428e-1 (5.1e-2)	4.413e-1 (4.5e-2)	4.45e-1 (3.8e-2)	4.454e-1 (6.4e-2)	4.434e-1 (5.1e-2)	4.618e-1 (5.6e-2)	4.547e-1 (5.3e-2)
WFG9	3	5.129e-3 (1.8e-4)	5.161e-3 (2.3e-4)	5.058e-3 (1.4e-4)	5.146e-3 (1.7e-4)	5.011e-3 (2.2e-4)	4.989e-3 (1.8e-4)	5.052e-3 (2.2e-4)	5.112e-3 (2.2e-4)
	5	4.405e-1 (4.0e-2)	4.421e-1 (4.4e-2)	4.488e-1 (5.0e-2)	4.402e-1 (4.0e-2)	4.62e-1 (5.e-2)	4.597e-1 (4.6e-2)	4.565e-1 (5.7e-2)	4.512e-1 (3.8e-2)

Table B.4: Mean and standard deviation of the I_{NHV} indicator values over 30 independent runs of SMPSO-M. The best means are highlighted in dark gray, and the second-best in light gray. The first four columns display the algorithms that use 40-regular graphs, and the rest use 80-regular graphs.

		40-regular graphs				80-regular graphs			
	m	Graph 1	Graph 2	Graph 3	Graph 4	Graph 1	Graph 2	Graph 3	Graph 4
DTLZ1	7	8.991e-1 (1.2e-1)	9.117e-1 (8.5e-2)	9.199e-1 (7.9e-2)	8.677e-1 (1.8e-1)	8.578e-1 (1.6e-1)	9.146e-1 (1.1e-1)	9.354e-1 (3.3e-2)	9.146e-1 (6.8e-2)
	10	6.767e-1 (2.7e-1)	6.628e-1 (3.1e-1)	6.583e-1 (3.0e-1)	6.744e-1 (2.5e-1)	5.962e-1 (3.6e-1)	6.332e-1 (3.0e-1)	7.148e-1 (2.0e-1)	7.603e-1 (2.3e-1)
DTLZ2	7	4.951e-1 (8.4e-2)	4.876e-1 (1.1e-1)	4.707e-1 (1.1e-1)	4.607e-1 (1.2e-1)	2.125e-1 (1.0e-1)	2.363e-1 (1.2e-1)	2.056e-1 (1.1e-1)	2.179e-1 (9.4e-2)
	10	2.847e-2 (2.7e-2)	2.606e-2 (2.9e-2)	2.086e-2 (1.4e-2)	2.393e-2 (1.9e-2)	1.675e-2 (2.3e-2)	2.347e-2 (2.1e-2)	2.447e-2 (3.7e-2)	2.218e-2 (2.7e-2)
DTLZ3	7	0.e+0 (0.e+0)	4.857e-3 (1.8e-2)	0.e+0 (0.e+0)	2.74e-3 (1.5e-2)	0.e+0 (0.e+0)	0.e+0 (0.e+0)	0.e+0 (0.e+0)	0.e+0 (0.e+0)
	10	0.e+0 (0.e+0)	0.e+0 (0.e+0)	0.e+0 (0.e+0)	2.333e-3 (1.3e-2)	0.e+0 (0.e+0)	2.092e-3 (1.1e-2)	0.e+0 (0.e+0)	0.e+0 (0.e+0)
DTLZ4	7	7.532e-1 (4.5e-2)	7.586e-1 (4.2e-2)	7.536e-1 (4.6e-2)	7.64e-1 (3.9e-2)	5.823e-1 (1.3e-1)	5.669e-1 (1.2e-1)	5.839e-1 (1.6e-1)	5.706e-1 (1.5e-1)
	10	2.837e-1 (1.8e-1)	3.109e-1 (1.7e-1)	2.745e-1 (1.8e-1)	2.317e-1 (1.5e-1)	1.092e-1 (1.2e-1)	1.064e-1 (1.1e-1)	1.249e-1 (1.5e-1)	1.182e-1 (1.2e-1)
DTLZ7	7	5.469e-1 (9.5e-2)	5.585e-1 (5.7e-2)	5.757e-1 (2.1e-2)	5.693e-1 (2.7e-2)	1.360e-1 (1.8e-1)	1.513e-1 (2.e-1)	2.743e-1 (2.5e-1)	1.405e-1 (1.6e-1)
	10	6.89e-2 (1.4e-1)	5.281e-2 (1.3e-1)	7.874e-2 (1.6e-1)	4.074e-2 (1.1e-1)	6.49e-5 (1.2e-4)	7.203e-5 (1.1e-4)	3.823e-4 (1.2e-3)	1.093e-4 (2.3e-4)
WFG1	7	2.67e-1 (1.7e-3)	2.660e-1 (1.2e-3)	2.664e-1 (1.5e-3)	2.668e-1 (2.1e-3)	2.688e-1 (3.8e-3)	2.687e-1 (3.3e-3)	2.697e-1 (4.6e-3)	2.686e-1 (3.9e-3)
	10	2.306e-1 (1.4e-3)	2.309e-1 (1.3e-3)	2.304e-1 (1.3e-3)	2.307e-1 (1.4e-3)	2.323e-1 (2.4e-3)	2.32e-1 (2.2e-3)	2.339e-1 (3.7e-3)	2.323e-1 (3.e-3)
WFG2	7	9.366e-1 (3.2e-2)	9.368e-1 (3.0e-2)	9.335e-1 (2.2e-2)	9.333e-1 (2.8e-2)	9.678e-1 (1.9e-2)	9.611e-1 (2.6e-2)	9.491e-1 (2.4e-2)	9.64e-1 (2.1e-2)
	10	8.786e-1 (3.9e-2)	8.954e-1 (3.3e-2)	8.782e-1 (4.5e-2)	8.831e-1 (3.8e-2)	9.117e-1 (5.7e-2)	9.056e-1 (4.4e-2)	9.118e-1 (4.1e-2)	9.098e-1 (3.5e-2)
WFG3	7	1.522e-2 (3.8e-2)	1.006e-2 (1.7e-2)	2.385e-2 (5.0e-2)	9.718e-3 (1.8e-2)	4.719e-2 (7.9e-2)	6.254e-2 (8.1e-2)	5.321e-2 (7.1e-2)	2.858e-2 (7.6e-2)
	10	0.e+0 (0.e+0)	0.e+0 (0.e+0)	0.e+0 (0.e+0)	0.e+0 (0.e+0)	0.e+0 (0.e+0)	0.e+0 (0.e+0)	0.e+0 (0.e+0)	0.e+0 (0.e+0)
WFG4	7	5.455e-1 (2.7e-2)	5.67e-1 (3.3e-2)	5.694e-1 (3.3e-2)	5.698e-1 (4.3e-2)	5.424e-1 (3.9e-2)	5.418e-1 (4.6e-2)	5.5e-1 (3.9e-2)	5.408e-1 (4.7e-2)
	10	5.126e-1 (2.6e-2)	5.185e-1 (2.3e-2)	5.115e-1 (2.7e-2)	5.106e-1 (2.8e-2)	4.978e-1 (3.e-2)	5.158e-1 (3.4e-2)	5.012e-1 (3.e-2)	5.16e-1 (3.9e-2)
WFG5	7	4.573e-1 (1.5e-2)	4.58e-1 (1.9e-2)	4.545e-1 (1.4e-2)	4.591e-1 (1.9e-2)	4.425e-1 (2.1e-2)	4.466e-1 (2.3e-2)	4.507e-1 (2.3e-2)	4.483e-1 (2.4e-2)
	10	4.427e-1 (1.8e-2)	4.461e-1 (1.5e-2)	4.528e-1 (1.7e-2)	4.453e-1 (1.5e-2)	4.467e-1 (1.8e-2)	4.460e-1 (1.8e-2)	4.497e-1 (2.0e-2)	4.498e-1 (2.3e-2)
WFG6	7	6.387e-1 (2.e-2)	6.322e-1 (1.7e-2)	6.366e-1 (1.8e-2)	6.321e-1 (1.7e-2)	6.404e-1 (1.8e-2)	6.459e-1 (2.3e-2)	6.5e-1 (2.4e-2)	6.448e-1 (2.5e-2)
	10	6.896e-1 (1.0e-2)	6.919e-1 (1.4e-2)	6.926e-1 (8.8e-3)	6.950e-1 (1.1e-2)	7.016e-1 (1.5e-2)	6.998e-1 (1.4e-2)	7.042e-1 (1.2e-2)	6.979e-1 (1.5e-2)
WFG7	7	6.092e-1 (3.e-2)	6.137e-1 (2.6e-2)	6.128e-1 (2.7e-2)	6.083e-1 (2.6e-2)	6.993e-1 (4.e-2)	7.121e-1 (4.9e-2)	7.047e-1 (4.3e-2)	7.051e-1 (3.9e-2)
	10	5.791e-1 (2.1e-2)	5.812e-1 (2.3e-2)	5.731e-1 (2.5e-2)	5.733e-1 (2.3e-2)	6.309e-1 (3.e-2)	6.322e-1 (3.5e-2)	6.376e-1 (3.0e-2)	6.406e-1 (3.7e-2)
WFG8	7	4.837e-1 (2.4e-2)	4.876e-1 (2.4e-2)	4.807e-1 (2.4e-2)	4.834e-1 (2.6e-2)	4.885e-1 (3.8e-2)	4.848e-1 (2.8e-2)	4.9e-1 (3.2e-2)	4.846e-1 (2.7e-2)
	10	5.074e-1 (2.0e-2)	5.068e-1 (1.9e-2)	5.103e-1 (2.e-2)	4.990e-1 (2.1e-2)	5.018e-1 (4.3e-2)	5.072e-1 (3.3e-2)	5.077e-1 (3.3e-2)	5.126e-1 (3.4e-2)
WFG9	7	4.677e-1 (1.8e-2)	4.596e-1 (1.9e-2)	4.651e-1 (1.5e-2)	4.597e-1 (1.8e-2)	4.586e-1 (2.4e-2)	4.692e-1 (2.8e-2)	4.638e-1 (3.3e-2)	4.633e-1 (2.1e-2)
	10	4.584e-1 (1.5e-2)	4.635e-1 (1.9e-2)	4.558e-1 (1.0e-2)	4.559e-1 (1.8e-2)	4.549e-1 (2.7e-2)	4.53e-1 (2.2e-2)	4.588e-1 (2.3e-2)	4.579e-1 (2.0e-2)

Table B.5: Mean and standard deviation of the I_{NHV} indicator values over 30 independent runs of SMPSO-M. The best means are highlighted in dark gray, and the second-best in light gray. The first four columns display the algorithms that use 120-regular graphs, and the rest use 160-regular graphs.

		120-regular graphs				160-regular graphs			
	m	Graph 1	Graph 2	Graph 3	Graph 4	Graph 1	Graph 2	Graph 3	Graph 4
DTLZ1	7	9.401e-1 (3.5e-2)	8.544e-1 (1.7e-1)	8.894e-1 (1.8e-1)	8.428e-1 (1.4e-1)	8.574e-1 (1.2e-1)	8.772e-1 (1.1e-1)	8.476e-1 (1.7e-1)	8.618e-1 (1.3e-1)
	10	6.717e-1 (2.8e-1)	6.919e-1 (2.4e-1)	6.99e-1 (3.0e-1)	6.948e-1 (3.7e-1)	6.718e-1 (2.8e-1)	6.444e-1 (2.9e-1)	6.891e-1 (2.9e-1)	6.254e-1 (3.2e-1)
DTLZ2	7	1.333e-1 (7.1e-2)	1.041e-1 (4.7e-2)	1.267e-1 (5.7e-2)	1.04e-1 (7.e-2)	1.07e-1 (5.7e-2)	1.283e-1 (6.5e-2)	1.042e-1 (4.7e-2)	9.845e-2 (5.2e-2)
	10	2.052e-2 (1.8e-2)	3.158e-2 (3.8e-2)	2.140e-2 (2.5e-2)	2.626e-2 (3.1e-2)	5.546e-2 (3.9e-2)	3.824e-2 (3.5e-2)	3.351e-2 (3.5e-2)	5.087e-2 (7.0e-2)
DTLZ3	7	2.744e-3 (1.5e-2)	0.e+0 (0.e+0)	5.151e-3 (1.9e-2)	2.344e-3 (1.3e-2)	1.056e-2 (4.0e-2)	8.013e-3 (3.1e-2)	3.119e-3 (1.7e-2)	2.065e-2 (5.4e-2)
	10	0.e+0 (0.e+0)	0.e+0 (0.e+0)	0.e+0 (0.e+0)	2.997e-3 (1.6e-2)	0.e+0 (0.e+0)	0.e+0 (0.e+0)	2.989e-3 (1.6e-2)	2.182e-3 (1.2e-2)
DTLZ4	7	3.855e-1 (1.9e-1)	3.910e-1 (1.8e-1)	3.898e-1 (1.9e-1)	3.595e-1 (1.7e-1)	2.037e-1 (1.7e-1)	1.68e-1 (1.8e-1)	2.084e-1 (1.9e-1)	2.083e-1 (1.9e-1)
	10	1.478e-2 (3.6e-2)	2.328e-2 (5.9e-2)	2.271e-2 (4.1e-2)	1.010e-2 (3.1e-2)	5.177e-4 (2.4e-3)	8.085e-4 (3.7e-3)	8.592e-3 (2.4e-2)	6.094e-3 (2.3e-2)
DTLZ7	7	2.199e-2 (4.5e-2)	1.18e-2 (2.8e-2)	2.201e-2 (4.9e-2)	4.074e-2 (1.0e-1)	4.609e-3 (1.2e-2)	5.486e-3 (9.e-3)	3.041e-2 (1.1e-1)	2.074e-2 (7.7e-2)
	10	5.537e-5 (2.7e-4)	4.937e-5 (2.1e-4)	5.733e-6 (1.3e-5)	1.15e-4 (4.e-4)	1.713e-5 (7.1e-5)	8.000e-7 (3.5e-6)	8.667e-7 (2.7e-6)	1.418e-4 (5.9e-4)
WFG1	7	2.724e-1 (5.3e-3)	2.746e-1 (6.9e-3)	2.718e-1 (6.0e-3)	2.726e-1 (5.5e-3)	2.77e-1 (8.1e-3)	2.772e-1 (9.4e-3)	2.76e-1 (7.e-3)	2.754e-1 (5.8e-3)
	10	2.372e-1 (4.9e-3)	2.356e-1 (4.1e-3)	2.357e-1 (4.6e-3)	2.358e-1 (4.7e-3)	2.417e-1 (9.6e-3)	2.394e-1 (7.3e-3)	2.399e-1 (8.9e-3)	2.388e-1 (5.3e-3)
WFG2	7	9.701e-1 (2.2e-2)	9.67e-1 (2.5e-2)	9.685e-1 (2.5e-2)	9.607e-1 (3.5e-2)	9.519e-1 (4.5e-2)	9.525e-1 (3.6e-2)	9.473e-1 (3.6e-2)	9.558e-1 (3.0e-2)
	10	9.308e-1 (4.1e-2)	9.224e-1 (4.9e-2)	8.999e-1 (6.1e-2)	9.198e-1 (4.5e-2)	9.052e-1 (5.9e-2)	8.683e-1 (8.1e-2)	8.748e-1 (7.e-2)	8.838e-1 (7.4e-2)
WFG3	7	1.319e-1 (1.9e-1)	1.118e-1 (1.7e-1)	7.648e-2 (1.2e-1)	1.286e-1 (2.2e-1)	1.032e-1 (2.2e-1)	1.536e-1 (2.6e-1)	1.221e-1 (2.1e-1)	1.999e-1 (3.2e-1)
	10	5.286e-3 (2.1e-2)	5.252e-4 (2.8e-3)	0.e+0 (0.e+0)	0.e+0 (0.e+0)	7.376e-3 (3.e-2)	8.488e-3 (3.2e-2)	0.e+0 (0.e+0)	0.e+0 (0.e+0)
WFG4	7	5.420e-1 (6.4e-2)	5.394e-1 (5.5e-2)	5.344e-1 (4.6e-2)	5.319e-1 (5.6e-2)	5.626e-1 (5.7e-2)	5.792e-1 (4.7e-2)	5.712e-1 (6.3e-2)	5.81e-1 (7.0e-2)
	10	4.856e-1 (3.2e-2)	4.875e-1 (4.2e-2)	4.939e-1 (4.7e-2)	4.998e-1 (4.2e-2)	5.344e-1 (5.4e-2)	5.421e-1 (5.1e-2)	5.362e-1 (5.4e-2)	5.576e-1 (4.6e-2)
WFG5	7	4.411e-1 (2.8e-2)	4.420e-1 (2.7e-2)	4.411e-1 (2.2e-2)	4.472e-1 (2.8e-2)	4.468e-1 (2.4e-2)	4.516e-1 (2.5e-2)	4.459e-1 (2.0e-2)	4.48e-1 (2.1e-2)
	10	4.381e-1 (2.4e-2)	4.478e-1 (2.6e-2)	4.408e-1 (2.3e-2)	4.507e-1 (2.4e-2)	4.383e-1 (2.2e-2)	4.339e-1 (2.7e-2)	4.37e-1 (2.9e-2)	4.329e-1 (3.4e-2)
WFG6	7	6.514e-1 (2.1e-2)	6.54e-1 (2.2e-2)	6.511e-1 (2.2e-2)	6.587e-1 (2.2e-2)	6.622e-1 (1.4e-2)	6.640e-1 (1.8e-2)	6.589e-1 (2.0e-2)	6.597e-1 (2.1e-2)
	10	7.053e-1 (1.6e-2)	7.054e-1 (1.1e-2)	7.082e-1 (1.1e-2)	7.023e-1 (1.5e-2)	7.023e-1 (1.5e-2)	7.118e-1 (1.3e-2)	7.129e-1 (1.3e-2)	7.043e-1 (1.3e-2)
WFG7	7	7.249e-1 (3.7e-2)	7.269e-1 (4.2e-2)	7.285e-1 (4.4e-2)	7.234e-1 (4.0e-2)	6.903e-1 (5.3e-2)	6.753e-1 (3.8e-2)	6.86e-1 (4.1e-2)	6.693e-1 (5.0e-2)
	10	6.607e-1 (4.3e-2)	6.307e-1 (4.7e-2)	6.453e-1 (3.5e-2)	6.278e-1 (3.2e-2)	6.268e-1 (4.4e-2)	6.204e-1 (4.4e-2)	6.214e-1 (3.e-2)	6.113e-1 (4.1e-2)
WFG8	7	4.971e-1 (4.7e-2)	4.799e-1 (4.2e-2)	4.938e-1 (4.7e-2)	4.994e-1 (4.7e-2)	4.790e-1 (5.1e-2)	4.724e-1 (6.7e-2)	4.822e-1 (5.6e-2)	4.855e-1 (5.1e-2)
	10	4.748e-1 (5.2e-2)	4.755e-1 (4.4e-2)	4.780e-1 (4.4e-2)	4.863e-1 (4.4e-2)	4.54e-1 (6.2e-2)	4.552e-1 (6.1e-2)	4.65e-1 (6.6e-2)	4.582e-1 (6.4e-2)
WFG9	7	4.48e-1 (3.e-2)	4.565e-1 (2.7e-2)	4.590e-1 (3.8e-2)	4.602e-1 (3.8e-2)	4.575e-1 (2.4e-2)	4.522e-1 (3.1e-2)	4.559e-1 (2.7e-2)	4.569e-1 (2.8e-2)
	10	4.398e-1 (2.1e-2)	4.539e-1 (2.e-2)	4.43e-1 (2.1e-2)	4.503e-1 (2.4e-2)	4.428e-1 (2.e-2)	4.487e-1 (2.9e-2)	4.446e-1 (2.8e-2)	4.427e-1 (2.0e-2)

[6] M. Habib, I. Aljarah, H. Faris, S. Mirjalili, Multi-objective Particle Swarm Optimization: Theory, Literature Review, and Application in Feature Selection for Medical Diagnosis, in: S. Mirjalili, H. Faris, I. Aljarah (Eds.), Evolutionary Machine Learning Techniques: Algorithms and Applications, Springer, Singapore, 2020, pp. 175–201. doi:10.1007/978-981-32-9990-0_9.

[7] F. Han, W.-T. Chen, Q.-H. Ling, H. Han, Multi-objective particle swarm optimization with adaptive strategies for feature selection, Swarm and Evolutionary Computation 62, article Number: 100847 (July 2021).

[8] Y. Xie, L. Du, J. Zhao, C. Liu, W. Li, Multi-objective optimization of process parameters in stamping based on an improved RBM–BPNN network and MOPSO algorithm, Structural and Multidisciplinary Optimization 64 (2021) 4209–4235.

[9] S. Su, D. Xiong, H. Yu, X. Dong, A multiple leaders particle swarm optimization algorithm with variable neighborhood search for multiobjective fixed crowd carpooling problem, Swarm and Evolutionary Computation 72, article Number: 101103 (July 2022).

[10] L. Li, Y. Fu, J. C. Fung, K. T. Tse, A. K. Lau, Development of a back-propagation neural network combined with an adaptive multi-objective particle swarm optimizer algorithm for predicting and optimizing indoor CO2 and PM2.5 concentrations, Journal of Building Engineering 54 (2022) 104600. doi:10.1016/j.jobe.2022.104600.

[11] L. Song, J. Shi, A. Pan, J. Yang, J. Xie, A Dynamic Multi-Swarm Particle Swarm Optimizer for Multi-Objective Optimization of Machining Operations Considering Efficiency and Energy Consumption, Energies 13 (10) (2020). doi:10.3390/en13102616.

[12] D. C. Valencia-Rodríguez, C. A. Coello Coello, A Study of Swarm Topologies and Their Influence on the Performance of Multi-Objective Particle Swarm Optimizers, in: T. Bäck, M. Preuss, A. Deutz, H. Wang, C. Doerr, M. Emmerich, H. Trautmann (Eds.), Parallel Problem Solving from Nature – PPSN XVI, Springer International Publishing, 2020, pp. 285–298. doi:10.1007/978-3-030-58115-2_20.

[13] C. Yue, B. Qu, J. Liang, A Multiobjective Particle Swarm Optimizer Using Ring Topology for Solving Multimodal Multiobjective Problems, IEEE Transactions on Evolutionary Computation 22 (5) (2018) 805–817. doi:10.1109/TEVC.2017.2754271.

[14] D. C. Valencia-Rodríguez, C. A. Coello Coello, The Influence of Swarm Topologies in Many-Objective Optimization Problems, in: H. Ishibuchi, Q. Zhang, R. Cheng, K. Li, H. Li, H. Wang, A. Zhou (Eds.), 11th International Conference on Evolutionary Multi-Criterion Optimization

(EMO 2021), Springer International Publishing, 2021, pp. 387–398. doi:10.1007/978-3-030-72062-9_31.

[15] M. Yamamoto, T. Uchitane, T. Hatanaka, An Experimental Study for Multi-objective Optimization by Particle Swarm with Graph Based Archive, in: Proceedings of SICE Annual Conference (SICE 2012), 2012, pp. 89–94, iSBN: 978-1-4673-2259-1.

[16] J. Branke, K. Deb, Integrating User Preferences into Evolutionary Multi-Objective Optimization, in: Y. Jin (Ed.), Knowledge Incorporation in Evolutionary Computation, Springer, Berlin Heidelberg, 2005, pp. 461–477, iSBN 3-540-22902-7.

[17] K. E. Parsopoulos, M. N. Vrahatis, Multi-objective Particles Swarm Optimization Approaches, in: Multi-objective Optimization in Computational Intelligence: Theory and practice, IGI Global, 2008, pp. 20–42. doi:10.4018/978-1-59904-498-9.ch002.

[18] J. Luo, X. Huang, X. Li, K. Gao, A novel particle swarm optimizer for many-objective optimization, in: 2019 IEEE Congress on Evolutionary Computation (CEC), 2019, pp. 958–965. doi:10.1109/CEC.2019.8790343.

[19] M. Leung, C. A. Coello Coello, C. Cheung, S. Ng, A. K. Lui, A hybrid leader selection strategy for many-objective particle swarm optimization, IEEE Access 8 (2020) 189527–189545. doi:10.1109/ACCESS.2020.3031002.

[20] R. Nshimirimana, A. Abraham, G. Nothnagel, A multi-objective particle swarm for constraint and unconstrained problems, Neural Computing and Applications 33 (2021) 11355–11385. doi:10.1007/s00521-020-05555-6.

[21] A. J. Nebro, J. J. Durillo, J. García-Nieto, C. A. Coello Coello, F. Luna, E. Alba, SMPSO: A new PSO-based Metaheuristic for Multi-objective Optimization, in: 2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM 2009), 2009, pp. 66–73. doi:10.1109/MCDM.2009.4938830.

[22] M. Clerc, J. Kennedy, The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space, IEEE Transactions on Evolutionary Computation 6 (1) (2002) 58–73. doi:10.1109/4235.985692.

[23] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197. doi:10.1109/4235.996017.

[24] K. Deb, R. B. Agrawal, Simulated Binary Crossover for Continuous Search Space, Complex Systems 9 (2) (1995) 115–148.

[25] A. McNabb, M. Gardner, K. Seppi, An Exploration of topologies and communication in large particle swarms, in: 2009 IEEE Congress on Evolutionary Computation (CEC 2009), 2009, pp. 712–719. doi:10.1109/CEC.2009.4983015.

[26] M. Reyes-Sierra, C. A. Coello Coello, Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art, International Journal of Computational Intelligence Research 2 (3) (2006) 287–308.

[27] S. Mostaghim, J. Teich, Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO), in: Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No.03EX706), 2003, pp. 26–33. doi:10.1109/SIS.2003.1202243.

[28] E. Zitzler, K. Deb, L. Thiele, Comparison of Multiobjective Evolutionary Algorithms: Empirical Results, Evolutionary Computation 8 (2) (2000) 173–195.

[29] P. Agarwal, R. Agrawal, B. Kaur, Multi-objective particle swarm optimization with guided exploration for multimodal problems, Applied Soft Computing 120 (2022) 108684. doi:10.1016/j.asoc.2022.108684.

[30] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable Test Problems for Evolutionary Multiobjective Optimization, in: A. Abraham, L. Jain, R. Goldberg (Eds.), Evolutionary Multiobjective Optimization: Theoretical Advances and Applications, Springer London, London, 2005, pp. 105–145. doi:10.1007/1-84628-137-7_6.

[31] S. Huband, L. Barone, L. While, P. Hingston, A Scalable Multi-objective Test Problem Toolkit, in: C. A. Coello Coello, A. Hernández Aguirre, E. Zitzler (Eds.), Evolutionary Multi-Criterion Optimization (EMO 2005), Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 280–295. doi:10.1007/978-3-540-31880-4_20.

[32] E. Zitzler, Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications, Ph.D. thesis, Swiss Federal Institute of Technology (ETH), Zurich, Suiza (Nov. 1999).

[33] J. Knowles, L. Thiele, E. Zitzler, A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers, 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, revised version (feb 2006).

[34] R. J. Wilson, Introduction to Graph Theory, 4th Edition, Longman, 1996.

[35] A. Steger, N. C. Wormald, Generating Random Regular Graphs Quickly, Combinatorics, Probability and Computing 8 (4) (1999) 377–396. doi:10.1017/S0963548399003867.

[36] A. A. Hagberg, D. A. Schult, P. J. Swart, Exploring Network Structure, Dynamics, and Function using NetworkX, in: G. Varoquaux, T. Vaught, J. Millman (Eds.), Proceedings of the 7th Python in Science Conference (SciPy2008), Pasadena, CA USA, 2008, pp. 11 – 15.