Int. J. Mach. Learn. & Cyber.
DOI 10.1007/s13042-014-0308-3

ORIGINAL ARTICLE

Particle swarm optimization with neighborhood-based budget allocation

D. Souravlias · K. E. Parsopoulos

Received: 13 December 2013 / Accepted: 16 October 2014

Abstract The standard particle swarm optimization (PSO) algorithm allocates the total available budget of function evaluations equally and concurrently among the particles of the swarm. In the present work, we propose a new variant of PSO where each particle is dynamically assigned different computational budget based on the quality of its neighborhood. The main goal is to favor particles with high-quality neighborhoods by asynchronously providing them with more function evaluations than the rest. For this purpose, we define quality criteria to assess a neighborhood with respect to the information it possesses in terms of solutions' quality and diversity. Established stochastic techniques are employed for the final selection among the particles. Different variants are proposed by combining various quality criteria in a single- or multi-objective manner. The proposed approach is assessed on widely used test suites as well as on a set of real-world problems. Experimental evidence reveals the efficiency of the proposed approach and its competitiveness against other PSO-based variants as well as different established algorithms.

Keywords Particle swarm optimization · Computational budget allocation · Neighborhood quality

D. Souravlias · K. E. Parsopoulos (✉)
Department of Computer Science and Engineering, University of Ioannina, 45110 Ioannina, Greece
e-mail: kostasp@cs.uoi.gr

D. Souravlias
e-mail: dsouravl@cs.uoi.gr

1 Introduction

Particle Swarm Optimization (PSO) is a population-based algorithm that models social behavior to effectively solve global optimization problems by guiding swarms of particles towards the most promising regions of the search space. It was originally introduced by Eberhart and Kennedy [12] in 1995 and, since then, it has gained increasing popularity. This can be ascribed to its efficiency and effectiveness in solving hard optimization problems with minor programming effort. Up-to-date, there is a considerable amount of works on PSO-based applications in various scientific and technological fields [15, 19, 20, 22, 25, 29].

The standard PSO algorithm considers all particles of the swarm to be equally important. Thus, it synchronously allocates the same fraction of function evaluations to each one. On the other hand, it would be reasonable to promote the search in the most promising regions of the search space by favoring the particles that probe such regions. Due to the inherent collective dynamics of PSO, these particles communicate their experience also to their neighbors, thereby offering them an opportunity to enhance their performance. For this reason, the idea of using neighborhood characteristics and qualities to identify and favor some of the particles in the budget allocation procedure is appealing.

In this framework, we recently introduced PSO with budget allocation through neighborhood ranking (PSO-BANR) [23]. This algorithm irregularly allocates the available computational budget among the particles. Specifically, a rank-based scoring scheme is used to assess each particle based on the information carried by its neighborhood in terms of objective values rather than the particle's value solely. The neighborhoods' scores are used to assign selection probabilities to the particles. Finally, a stochastic selection scheme (fitness proportionate selection) is used to select the particle that will be awarded the next function evaluation. Two alternative scoring schemes were considered in [23] and preliminary experiments revealed the potential of PSO-BANR to be efficient and effective, triggering further our interest on neighborhood-based budget allocation.

The present work constitutes a radical extension of the ideas exposed in our previous work. We generalize the PSO-BANR algorithm to a new approach called particle swarm optimization with neighborhood-based budget allocation, henceforth denoted as PSO-NBA. The new algorithm employs two essential budget allocation strategies to assess the quality of neighborhoods. The two strategies are based on single-objective and multi-objective scoring modes, respectively. The single-objective approach is based on the total or, alternatively, on the best information carried by the neighborhood in terms of objective values. The multi-objective approach takes into consideration also another aspect of quality, namely the diversity of the neighborhood. In this case, each neighborhood is assessed on the basis of a 2-dimensional scoring vector. The first component of the vector is identical to the solution quality criterion of the single-objective approach. The second component of the vector depends on the diversity of the best positions of the particles that comprise the neighborhood. Then, a scheme that is based on the concept of Pareto dominance is used for selection among neighborhoods.

The proposed PSO-NBA approach individually handles each function evaluation and stochastically assigns it to a particle according to its selection probability. This probability is determined through the aforementioned neighborhood scoring schemes. Evidently, this procedure implies an asynchronous update of the particles. Besides that, particles that are associated with the most promising neighborhoods are highly probable to gain more function evaluations than the rest. However, due to the stochastic nature of the selection process, also particles that belong to less promising neighborhoods have the potential of gaining function evaluations. Intuitively, this property results in deliberate promotion of the exploitation dynamic of the algorithm, yet without neglecting its exploration capability.

The concept of budget allocation was previously considered in PSO although in different frameworks. In [4, 17, 21, 33] PSO was equipped with the optimal computational budget allocation (OCBA) method in order to cope with optimization problems contaminated by noise. These works differ from the present study since we cope with noiseless problems and propose a more general budget allocation scheme. Also, we use both solution quality and diversity of the neighborhood as assessment criteria. There are also works that propose rank-based PSO variants in the literature. In [1] the presented algorithm uses only a fraction of the particles to update velocity. This approach is solely based on the global (gbest) PSO model, neglecting the neighborhoods. In [28] the proposed approach uses ranking to replace low-fitness particles with better ones. A relevant (although not rank-based) asynchronous PSO variant is PSO-DLI [27], which employs a special scheme to allocate function evaluations to some of the particles while the rest remain idle. Our approach differs also from these approaches, since we use neighborhood ranking schemes to allocate the available computational budget in a sophisticated manner. To the best of our knowledge, this work is the first study that uses rank-based criteria to assess the quality of neighborhoods and dynamically distribute the available computational budget among the corresponding particles.

The rest of the paper is organized as follows: Section 2 provides a brief description of the necessary background in PSO. In Sect. 3, the proposed PSO-NBA algorithm is presented. Experimental results are reported and discussed in Sect. 4. The paper concludes in Sect. 5.

2 Background information

In this section, we provide a brief presentation of the original (synchronous) PSO algorithm as well as its asynchronous variant. Without loss of generality, we consider the continuous bound constrained minimization problem,

$$\min_{\mathbf{x}\in\mathbf{X}\subset\mathbb{R}^n} f(\mathbf{x}), \qquad (1)$$

where X is the search space under consideration defined as an n-dimensional hyperbox.

2.1 Particle swarm optimization

Consider the sets,

$$I = \{1, 2, \ldots, N\}, \qquad D = \{1, 2, \ldots, n\},$$

which denote the indices of the search agents and the indices of the direction components, respectively. PSO employs a set of search points,

$$S = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}\},$$

which is called a swarm, to iteratively probe the search space X. Each search point is an n-dimensional vector,

$$\mathbf{x_i} = (\mathbf{x_{i1}}, \mathbf{x_{i2}}, \ldots, \mathbf{x_{in}})^\top \in \mathbf{X}, \quad \mathbf{i} \in \mathbf{I},$$

called a particle. Each particle explores the search space by moving to new positions (candidate solutions) in X and adjusts its exploratory behavior according to its own findings as well as the findings of the other particles.

During its quest for better solutions, each particle records in memory the best position it has encountered. In minimization problems, this position has the lowest objective value among all positions visited by the particle. If t denotes the iteration counter and $\mathbf{x_i(t)}$ are the subsequent positions of the ith particle, then its best position is denoted as

$$\mathbf{p_i} = (\mathbf{p_{i1}}, \mathbf{p_{i2}}, \ldots, \mathbf{p_{in}})^{\top} \in \mathbf{X}, \quad \mathbf{i} \in \mathbf{I},$$

and defined as

$$\mathbf{p_i(t)} = \arg \min_{\mathbf{t=0,1,2,\ldots}} \{\mathbf{f(x_i(t))}\}.$$

The particle moves in the search space by using an adaptable position shift, called velocity,

$$\mathbf{v_i} = (\mathbf{v_{i1}}, \mathbf{v_{i2}}, \ldots, \mathbf{v_{in}})^{\top}, \quad \mathbf{i} \in \mathbf{I},$$

which is added to its current position to produce a new one.

The velocity of each particle is updated at each iteration by taking into consideration its own best position as well as the best position among a set of adjacent particles, which constitute its neighborhood [11, 24]. The adjacency between particles is determined according to arbitrary interconnection schemes that allow groups of particles to exchange information among them. These schemes are called neighborhood topologies and they are usually illustrated as undirected graphs where nodes denote the particles and edges denote communication channels. Figure 1 illustrates two common neighborhood topologies.

Various neighborhood topologies have been proposed in the literature. The most common one is the ring topology, illustrated in the left part of Fig. 1, where each particle assumes as neighbors the particles with adjacent indices. The size of the neighborhood is determined by a parameter r that is called the neighborhood's radius. Formally, a ring neighborhood of radius r of the ith particle is defined by the set of indices:

$$NB_{i,r} = \{i-r, \ldots, i-1, i, i+1, \ldots, i+r\}. \tag{2}$$

This means that the best position among the ones with indices from $i-r$ up to $i+r$ is used for the ith particle's velocity update. The indices are assumed to recycle at their limits, i.e., the particle with index 1 follows immediately after the one with index N.

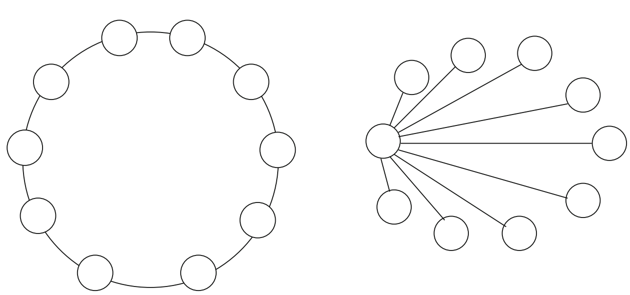

Fig. 1 Neighborhood topologies: ring (*left*) and star (*right*)

Based on the neighborhood size, two prevailing PSO models have been established. The first one, called the global PSO model (denoted as gbest), assumes the whole swarm as neighborhood of each particle. Thus, the overall best position of the swarm is used to update all particles' velocities. This approach was mainly used in early PSO variants and exhibited rapid convergence (exploitation) properties. However, rapid convergence was habitually accompanied by loss of diversity, leading to premature convergence in undesirable suboptimal solutions.

On the other hand, using significantly smaller neighborhoods can enhance the exploration properties of the swarm. This is attributed to the limited connectivity among the particles, which restricts the rapid diffusion of the detected best positions to the rest of the swarm. This approach defines the local PSO model (denoted as lbest).

Let $\mathbf{p_{g_{(i,t)}}}$ denote the best position in the neighborhood of the ith particle at iteration t, i.e.,

$$g_{(i,t)} = \arg \min_{j \in NB_{i,r}} \left\{ f\left(\mathbf{p_j(t)}\right) \right\}.$$

Then, based on the definitions above, the update equations of PSO are given as follows [5]:

$$\begin{aligned} \mathbf{v_i(t+1)} = & \chi \Big[\mathbf{v_i(t)} + \mathbf{c_1} \mathcal{R}_1 \otimes \Big(\mathbf{p_i(t)} - \mathbf{x_i(t)} \Big) + \\ & c_2 \mathcal{R}_2 \otimes \Big(\mathbf{p_{g_{(i,t)}}(t)} - \mathbf{x_i(t)} \Big) \Big], \end{aligned} \tag{3}$$

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

where $i \in I$, and $\otimes$ denotes componentwise multiplication of vectors. The parameter χ is called the constriction coefficient and it is used to clamp the velocities to avoid the swarm explosion effect [5]. The scalars c_1 and c_2 are called the cognitive and social parameter, respectively, and they are used to bias velocity towards either the particle's own best position or the neighborhood's best position. The parameters $\mathcal{R}_1$ and $\mathcal{R}_2$ are random vectors that induce stochasticity in the algorithm. Their components are drawn from the uniform distribution $\mathcal{U}([0,1])$.

After updating all particles, their new positions compete against their best positions. Thus, the best position of each particle is updated as follows:

$$\mathbf{p_i(t+1)} = \begin{cases} \mathbf{x_i(t+1)}, & \text{if } \mathrm{f}(\mathbf{x_i(t+1)}) < \mathrm{f}\,(\mathbf{p_i(t)}), \\ \mathbf{p_i(t)}, & \text{otherwise}, \end{cases} \tag{5}$$

where $i \in I$.

The presented variant of PSO is supported by thorough stability and convergence analysis [5], which suggested the general-purpose parameter setting,

$$\chi = 0.729, \quad c_1 = c_2 = 2.05.$$

This is considered to be a satisfactory setting that produces balanced convergence speed for the algorithm. Nevertheless, alternative successful settings have also been proposed in the literature [26].

The standard PSO algorithm allocates one function evaluation per particle per iteration. Hence, at the end of its execution, all particles have spent equal portions of the available computational budget. Moreover, the update of Eqs. (3), (4), and (5), is synchronous, i.e., the new best positions are determined only after the position update of all particles. Alternatively, asynchronous PSO variants have been developed. These approaches are briefly sketched in the following section.

2.2 Asynchronous particle swarm optimization

Asynchronous PSO variants have been developed as alternatives to the standard (synchronous) approach. Contrary to synchronous PSO, in the asynchronous model each particle updates and communicates at once its new best position to its neighbors, without waiting for the rest of the particles to update their memory at a given iteration. This immediate exposition of the particles to new findings has significant impact on their convergence speed. Also, it can radically reduce the algorithm's runtime in parallel implementations on inhomogeneous systems or problems with high diversity of function evaluation time.

On the other hand, rapid convergence of asynchronous PSO can lead the swarm to deceitful positions more frequently than the synchronous approach. Thus, it increases the probability of getting trapped in low-quality solutions. Therefore, special attention is required when selecting between the synchronous and the asynchronous model.

3 Proposed approach

In this section, we thoroughly describe the essential parts of the proposed particle swarm optimization with neighborhood-based budget allocation (PSO-NBA) approach. First, we present the considered neighborhood quality criteria, followed by the neighborhood selection schemes. Finally, we present the single- and multi-objective budget allocation strategies that are integrated in PSO-NBA.

3.1 Neighborhood quality criteria

The two essential properties that define the dynamics of any population-based optimization algorithm are exploration (diversification) and exploitation (intensification). The first one is the ability of the algorithm to explore diverse parts of the search space, while the second one is the ability to perform more refined search around the discovered good solutions. Proper balancing between these properties has been associated with highly competitive optimization algorithms.

It is easily inferred (and experimentally verified) that these two properties are intimately related with two performance indices of an algorithm, namely solution quality and diversity. The most successful approaches are expected to retain adequate diversity in the swarm such that search stagnation is eluded, while concurrently improving solution quality within reasonable time limits.

Transferring these concepts from swarm level to the neighborhood level, we consider two types of neighborhood quality criteria. The first type refers to solution quality and consists of two alternative schemes, while the second type refers to diversity. Specifically, the first solution quality criterion, denoted as *SumBest* (SB), is based on the total solution information carried by the neighborhood in terms of objective values. Thus, each neighborhood is assessed according to the collective achievements of its members.

The second solution quality criterion, denoted as *LocalBest* (LB), takes into consideration only the best position attained by the neighborhood's members. Thus, it clearly promotes elitism. Regarding diversity, we consider a criterion denoted as *AvgDev* (AD), which assesses each neighborhood in terms of diversity of the best positions that comprise it.

The three criteria are summarized in Table 1, and they are formally defined below.

3.1.1 SumBest (SB)

Let $NB_{i,r}$ be the neighborhood of the ith particle as defined in Eq. (2). Then, its SB ranking score at iteration t is defined as

$$SBR_i = \sum_{k \in NB_{i,r}} f\left(\mathbf{p_k}(\mathbf{t})\right), \quad \mathbf{i} \in \mathbf{I}. \tag{6}$$

Thus, the SB score assesses the neighborhood's quality in terms of the sum of the objective values of all best positions that comprise it. In order to facilitate the use of SB

Table 1 Neighborhood quality criteria of PSO-NBA

Type	Criterion	Abbreviation	Description
Solution quality	SumBest	SB	Sum of all objective values in neighborhood
	LocalBest	LB	Best objective value in neighborhood
Diversity	AvgDev	AD	Average standard deviation of direction components of the best positions that comprise the neighborhood

scores for the computation of the neighborhoods' selection probabilities, a normalization step takes place,

$$SBR_i^* = \frac{SBR_i}{\sum_{m \in I} SBR_m}, \quad i \in I. \tag{7}$$

Lower values of the SB ranking score correspond to neighborhoods that possess lower cumulative information in terms of their objective values. These neighborhoods are considered to be superior to the ones with higher scores and, hence, their corresponding particles shall be assigned higher selection probabilities in subsequent steps of the algorithm.

3.1.2 *LocalBest (LB)*

Let again $NB_{i,r}$ be the neighborhood of the ith particle. Then, the LB ranking score for this neighborhood is defined as

$$LBR_i = \min_{k \in NB_{i,r}} f(\mathbf{p_k}(\mathbf{t})), \quad \mathbf{i} \in \mathbf{I}. \tag{8}$$

The LB score promotes elitism by assessing the neighborhood's quality only in terms of the best position involved in it. Normalization takes place also in this case,

$$LBR_i^* = \frac{LBR_i}{\sum_{m \in I} LBR_m}, \quad i \in I. \tag{9}$$

Similarly to SB, particles with neighborhoods with lower LB ranking scores shall be assigned higher selection probabilities.

3.1.3 *AvgDev (AD)*

This diversity measure is based on the average standard deviation of the direction components of the best positions that comprise the neighborhood $NB_{i,r}$. Thus, if $NB_{i,r} = \{k_1, \ldots, k_{2r+1}\}$ and $D = \{1, 2, \ldots, n\}$, we first compute the standard deviation per direction component $j \in D$,

$$\sigma_j^{[i]} = \text{standard deviation of vector} \begin{pmatrix} \mathrm{p_{k_1 j}(t)} \\ \vdots \\ \mathrm{p_{k_{2r+1} j}(t),} \end{pmatrix}. \tag{10}$$

where p_{kj} stands for the jth component of the best position $\mathbf{p_k}$, $k \in NB_{i,r}$.

Then, the AD ranking score for the neighborhood is obtained by averaging the standard deviations over all dimensions,

$$AD_i = \frac{1}{n} \sum_{j=1}^{n} \sigma_j^{[i]}, \quad i \in I. \tag{11}$$

The obtained values are normalized as follows:

$$AD_i^* = \frac{AD_i}{\sum_{m \in I} AD_m}, \quad i \in I. \tag{12}$$

Obviously, neighborhoods with higher AD scores contain more dispersed best positions. Thus, they are preferable against neighborhoods with lower scores to promote exploration.

Also, contrary to the solution quality scores SB and LB, which are based on objective values, the AD scores are based on the actual positions of the neighborhood's members in the search space.

3.2 Selection probability

After the computation of the neighborhoods' ranking scores, each particle is assigned a selection probability based on the score of its neighborhood. We considered two alternative selection probability schemes. Let xBR denote the selected ranking scheme (SB or LB), i.e.,

$$xBR_i^* = SBR_i^* \text{ or } \mathrm{LBR_i^*}, \quad \forall \mathrm{i} \in \mathrm{I}.$$

Let also

$$Q = \left\{ xBR_{k_1}^*, xBR_{k_2}^*, \ldots, xBR_{k_N}^* \right\}, \quad k_i \in I,$$

be the ordering of the neighborhoods' ranking scores, sorted from the highest to the lowest value, and

$$q_i = \text{position of } i\text{th neighborhood's score } (\mathrm{xBR_i^*}) \text{ in Q}.$$

Then, the first selection probability scheme is the well-known linear ranking that is widely used in Genetic Algorithms (GAs) [3]. This scheme assigns selection probabilities that are linear with respect to xBR_i^* as follows:

$$LPR_i = 2 - s + 2(s-1)\frac{q_i - 1}{(N-1)}, \quad i \in I, \tag{13}$$

where the parameter $s \in [1, 2]$ is called the selection pressure, and N is the total number of neighborhoods (equal to swarm size). Note that intense elitism is promoted when $s = 2$, while equal selection probabilities are assigned to all neighborhoods when it is equal to $s = 1$.

The corresponding selection probability of the ith particle becomes

$$SP_i = \frac{LPR_i}{\sum_{m \in I} LPR_m}, \quad i \in I. \tag{14}$$

Henceforth, this scheme will be denoted as L (linear).

The second selection probability scheme comes again from the field of GAs and it is nonlinear, henceforth denoted as NL:

$$NLPR_i = \left(xBR_i^*\right)^{-\rho}, \quad i \in I, \tag{15}$$

where ρ is a positive integer. This scheme resembles the power selection operator in GAs [3]. The corresponding selection probabilities for this scheme are given as

$$SP_i = \frac{NLPR_i}{\sum\limits_{m \in I} NLPR_m}, \quad i \in I. \tag{16}$$

Clearly, higher values of the power weight ρ favor elitism since they result in higher selection probabilities for the neighborhoods with lower ranking scores xBR_i^*.

The neighborhoods' selection probabilities, computed either linearly through Eqs. (13) and (14) or nonlinearly through Eqs. (15) and (16), are used as input in a stochastic selection mechanism that determines the particle that will receive the next function evaluation. This mechanism can use either the selection probabilities solely or take into consideration also the AD diversity criterion defined in Sect. 3.1.3. We refer to the first case as the single-objective budget allocation strategy (SOBA), and to the second one as the multi-objective budget allocation Strategy (MOBA). Both strategies are analyzed in the following sections.

3.3 Single-objective budget allocation (SOBA) strategy

In the *Single-Objective Budget Allocation* (SOBA) strategy, the selection probabilities are fed as input in a stochastic selection mechanism, neglecting the AD diversity criterion. The employed selection mechanism is the *fitness proportionate selection* technique from GAs literature, also known as *roulette-wheel* [3]. This scheme makes a randomized decision among the competitors based on their selection probabilities.

Algorithm 1 SOBA strategy (for SB and LB) and MOBA strategy (for LW and DW).

Input: Strategy (S), computational budget ($FE_{\max}$), PSO parameters, $I = \{1, 2, \ldots, N\}$.
Output: Best detected solution.

initialize $\mathbf{x}_i$, $\mathbf{v}_i$, $\mathbf{p}_i$, $\forall i \in I$
compute selection probabilities SP_i, $\forall i \in I$
$t \leftarrow 0$
while ($t \leqslant FE_{max}$) **do**
  $k \leftarrow$ `RouletteWheel`($SP_1, \ldots, SP_N$)
  update $\mathbf{v}_k$ and $\mathbf{x}_k$ according to Eqs. (3) and (4)
  update $\mathbf{p}_k$ according to Eq. (5)
  if ($\mathbf{p}_k$ has changed) **then**
    if (S=SOBA with SB or LB) **then**
      $I_k = \{i \in I; k \in NB_{i,r}\}$
      update neighborhoods $NB_{j,r}$, $j \in I_k$
      compute SP_i, $\forall i \in I$
    else if (S=MOBA with LW or DW) **then**
      compute $w_1(t)$, $w_2(t)$, according to Eq. (19) (LW case) or Eq. (20) (DW case)
      update SP_i, $\forall i \in I$
    end if
  end if
  $t \leftarrow t + 1$
end while

Obviously, particles with neighborhoods of higher values SP_i have higher probability of being selected. However, it is still possible that particles with inferior selection probabilities are selected due to the stochastic nature of the selection scheme. A pseudocode for the application of the SOBA selection strategy is given in Algorithm 1.

3.4 Multi-objective budget allocation (MOBA) Strategy

The multi-objective budget allocation (MOBA) strategy takes into consideration both solution quality and diversity of the neighborhood. In general, there are two alternative multi-objective approaches to combine the two criteria. The first one is the weighted aggregation approach, which uses weighted combinations of the two criteria. The second one is the Pareto front approach, which is based on the concept of Pareto dominance. Both approaches are described in the following sections.

3.4.1 Weighted aggregation

The weighted aggregation is a popular technique for coping with multiple objectives [10, 18]. Its popularity lies in the transformation of the multi-objective problem to a single-objective one, which allows the use of a wide variety of optimization methods.

In our approach, the two objectives are solution quality, which is related to the exploitation property of the algorithm, and diversity, which is related to the exploration property. Nonnegative weights are used to balance their contribution in the aggregated score, which is defined as

$$F = w_1(t) \times \text{quality} + w_2(t) \times \text{diversity}, \tag{17}$$

where $w_2(t) = 1 - w_1(t)$, and t is the counter of function evaluations. The quality-based component is the selection probability SP_i as computed in Section 3.2. The diversity-based component is AD_i^*, as defined in Section 3.1.3. Therefore, Eq. (17) becomes

$$F_i = w_1(t)\, SP_i + w_2(t)\, AD_i^*, \quad i \in I, \tag{18}$$

with $w_2(t) = 1 - w_1(t)$. Notice that both SP_i and AD_i^* are better when they receive higher values. Thus, higher aggregated values F_i are better. The values F_i are normalized and fed as probabilities in a roulette-wheel selection scheme similarly to SOBA.

The weights can either remain fixed or be dynamically adjusted during the optimization process. In general, different phases of the optimization process require different exploration-exploitation trade-off of the algorithm. Since the case of fixed weights neglect this necessity, we adopted dynamically changing weights in our approach.

There are two widely used schemes for dynamically changing weighted aggregation. The first one is the linear weighted aggregation (henceforth denoted as LWA), where the weights are defined as follows:

$$w_1(t) = \frac{t}{FE_{\max}}, \qquad w_2(t) = 1 - w_1(t), \tag{19}$$

where $FE_{\max}$ is the maximum budget of function evaluations and t is their counter. Note that, at the early stages of the optimization process the diversity component is favored in order to promote better exploration of the search space, whereas the quality component is promoted at later stages in order to intensify the search near the most promising candidate solutions.

The second scheme is the dynamic weighted aggregation (denoted as DWA). The weights are modified as follows:

$$w_1(t) = |\sin(2\pi t/FR)|, \qquad w_2(t) = 1 - w_1(t), \tag{20}$$

where t is the counter of function evaluations and FR is the weights' change frequency. The use of the trigonometric function implies the interchange between exploration and exploitation, repeatedly.

Pseudocode for the LWA and DWA schemes is given in Algorithm 1. Note that, the sole difference between the SOBA and the weighted aggregation approach lies in the employed neighborhood scoring scheme.

Algorithm 2 MOBA strategy (PFA approach).

Input: Computational budget ($FE_{\max}$), PSO parameters, $I = \{1, 2, \ldots, N\}$.
Output: Best detected solution.

initialize $\mathbf{x}_i$, $\mathbf{v}_i$, $\mathbf{p}_i$, (xBR_i^*, AD_i^*), $\forall i \in I$
$t \leftarrow 0$
while $(t \leqslant FE_{max})$ **do**
  $I' \leftarrow$ `Tournament`$\{(xBR_1^*, AD_1^*), \ldots, (xBR_N^*, AD_N^*)\}$
  $I^* \leftarrow$ `Non-Dominated`(I')
  for all $i^* \in I^*$ **do**
    update $\mathbf{v}_{i^*}$, $\mathbf{x}_{i^*}$ according to Eqs. (3) and (4)
    update $\mathbf{p}_{i^*}$ according to Eq. (5)
  end for
  if (some $\mathbf{p}_{i^*}$ has changed) **then**
    for all $i^* \in I^*$ **do**
      update xBR_j^*, for all j with $i^* \in NB_{j,r}$, according to Eq. (6) or Eq. (8)
      update AD_j^*, for all j with $i^* \in NB_{j,r}$, according to Eq. (11)
    end for
  end if
  $t \leftarrow t + 1$
end while

3.4.2 *Pareto front approach*

In the *Pareto front* approach (henceforth denoted as PFA), we maintain the 2-dimensional scoring vector,

$$(xBR_i^*, AD_i^*), \quad i \in I,$$

for each neighborhood, where the xBR_i^* is related to solution quality (see Sect. 3.2), while AD_i^* is the diversity criterion (see Sect. 3.1.3). Alternatively, the selection probability SP_i can be used instead of xBR_i^* with minor modifications.

The core idea behind PFA is the promotion of the non-dominated neighborhoods with respect to the two criteria, in terms of the multi-objective optimization concepts of domination and Pareto optimality [6]. Thus, a neighborhood with scoring vector (xBR_i^*, AD_i^*) is dominated by another one with scoring vector (xBR_j^*, AD_j^*), if,

$$xBR_j^* < xBR_i^* \quad \text{and} \quad \mathrm{AD}_j^* \geqslant \mathrm{AD}_i^*$$

or

$$AD_j^* > AD_i^* \quad \text{and} \quad \mathrm{xBR}_j^* \leqslant \mathrm{xBR}_i^*.$$

Note that larger values of diversity and lower values of the solution quality score are preferable.

The non-dominated neighborhoods are candidates for gaining function evaluations through a tournament selection scheme. Specifically, at each iteration of the algorithm, a prespecified number (tournament size) of particles are selected from the swarm. The particles (among the selected) whose neighborhoods are non-dominated are awarded one function evaluation each. Obviously, the allocated number of function evaluations can differ from one iteration to another. The pseudocode of the PFA appoach is given in Algorithm 2.

The use of tournament selection instead of all non-dominated neighborhoods allows to address search stagnation. Specifically, we frequently observed that a few (usually one or two) neighborhoods could dominate all others at early stages of the algorithm's execution and, thus, collect almost all the allocated computational budget. This was proved to be detrimental for the algorithm's exploration ability, leading to search stagnation. The stochasticity of tournament selection provides the option of assigning function evaluations also to particles with neighborhoods of low quality and diversity, thereby amplifying the algorithm's exploration capability.

4 Experimental results

PSO-NBA was initially assessed over two test suites. The first one consists of five widely used test functions (TP0-

TP4), while the second one contains six problems (TP5-TP10) that come from real-world applications and they are modeled as systems of nonlinear equations. The descriptions of these test problems are given in "Appendix A", while their dimensions and ranges in our experimental setting are reported in Table 2. Further experimentation was also conducted on the test suite proposed at the special issue on "*Scalability of Evolutionary Algorithms and Other Metaheuristics for Large-Scale Optimization Problems*" of the *Soft Computing* journal [14]. This test suite consists of 19 problems that include problems from the CEC 2008 challenge, shifted problems, as well as hybrid composition functions.

We considered four variants of PSO-NBA, namely the SOBA strategy and the MOBA strategy with the LWA, DWA, and PFA schemes. These approaches were combined with the SB and LB scoring schemes under different parameter settings. The complete set of parameter values that were used in our experiments is reported in Table 3. In total, there were 36 individual PSO-NBA variants composed as different combinations of these schemes and parameter values.

The experimental evaluation consisted of two stages. In the first stage, we identified the most promising among the different PSO-NBA variants for all test problems. In the second stage, the distinguished variants were further assessed against different algorithms (PSO-based and not). The obtained results are presented in detail in the following sections.

Table 2 Dimensions and ranges of test problems

Problem	Dimension	Range
TP0	10, 50, 100	$[-100, 100]^n$
TP1	10, 50, 100	$[-30, 30]^n$
TP2	10, 50, 100	$[-5.12, 5.12]^n$
TP3	10, 50, 100	$[-600, 600]^n$
TP4	10, 50, 100	$[-20, 30]^n$
TP5	10	$[-2, 2]^{10}$
TP6	6	$[-10, 10]^6$
TP7	5	$[-10, 10]^5$
TP8	8	$[-10, 10]^8$
TP9	10	$[-10, 10]^{10}$
TP10	20	$[-10, 10]^{20}$

Table 3 Parameter values for the considered SOBA and MOBA strategies

Parameter	Value
PSO model	lbest
PSO parameters	$\chi = 0.729$, $c1 = c2 = 2.05$
Neighborhood topology	Ring
Neighborhood radius	1
Quality criteria	SumBest(SB), LocalBest(LB)
Diversity criterion	AvgDev(AD)
Selection scheme	Linear(L), Nonlinear(NL)
Selection pressure	$s \in \{1.0, 1.5, 2.0\}$
Nonlinear weight	$\rho \in \{1.0, 2.0\}$
Problem dimensions	$n = \{10, 50, 100\}$
Swarm size	$N = 10 \times n$
Function evaluations	$FE_{\max} = 1000 \times n$
Tournament size	$T \in \{N/2, N/3, N/5\}$
Weight's change frequency	$FR = 200$
Number of experiments	100 per approach

4.1 Assessment of SOBA strategy

As described in Sect. 3.3, the SOBA strategy quantifies the quality of each neighborhood according to a single rank-based score. The SB and LB scoring schemes of Sects. 3.1.1 and 3.1.2 were employed for this purpose and both were combined with the linear (L) and the nonlinear (NL) approaches described in Sect. 3.2, to compute the corresponding selection probabilities.

For the linear approach, we considered three different values of selection pressure, namely $s \in \{1.0, 1.5, 2.0\}$. In the nonlinear approach, we used two different values for the power weight, namely $\rho \in \{1.0, 2.0\}$. These combinations result in ten SOBA variants that are henceforth denoted as

$$\mathtt{X/Y/Z}$$

where $\mathtt{X} \in \{\text{SB}, \text{LB}\}$ and $\mathtt{Y} \in \{\text{L}, \text{NL}\}$. If $\mathtt{Y} = \text{L}$ then $\mathtt{Z}$ stands for the selection pressure and, thus, $\mathtt{Z} \in \{1.0, 1.5, 2.0\}$. If $\mathtt{Y} = \text{NL}$ then $\mathtt{Z} \in \{1.0, 2.0\}$ (power weight). For example, LB/NL/2.0 stands for the SOBA variant with LocalBest neighborhood scoring and nonlinear probability selection with selection pressure $s = 2.0$.

We performed 100 independent experiments for each problem instance and algorithm variant. In all experiments, the available computational budget was equal to $1{,}000 \times n$ function evaluations, where n stands for the problem's dimension. For each experiment, we recorded the best solution found by the algorithm as well as its objective value.

Table 4 reports the mean, standard deviation, minimum, and maximum of the 100 solutions' objective values per algorithm and problem instance of TP0-TP4. For the sake of presentation compactness, we report results only for the variants with the best values of selection pressure in the L schemes. The same holds also for the power weights in the NL schemes. Thus, four variants are reported per test

Table 4 Results for the SOBA approach for test problems TP0-TP4 (standard test suite)

Problem	Dimension	Algorithm	Mean	SD	Min	Max
TP0	10	SB/L/2.0	$3.535e-02$	$3.528e-02$	$2.919e-04$	$1.645e-01$
		LB/L/2.0	$2.131e-02$	$2.303e-02$	$1.404e-03$	$1.153e-01$
		SB/NL/1.0	$8.199e-02$	$3.127e-01$	$4.868e-06$	$2.889e+00$
		LB/NL/2.0	$\mathbf{9.406e-26}$	$8.806e-25$	$1.523e-35$	$8.807e-24$
	50	SB/L/2.0	$2.092e+03$	$4.243e+02$	$1.305e+03$	$3.343e+03$
		LB/L/2.0	$1.980e+03$	$4.118e+02$	$8.238e+02$	$3.322e+03$
		SB/NL/2.0	$5.378e+00$	$1.349e+01$	$3.603e-03$	$1.003e+02$
		LB/NL/2.0	$\mathbf{3.116e-08}$	$1.332e-07$	$1.762e-012$	$1.293e-06$
	100	SB/L/2.0	$1.758e+04$	$2.471e+03$	$1.218e+04$	$2.364e+04$
		LB/L/2.0	$1.680e+04$	$2.199e+03$	$1.093e+04$	$2.532e+04$
		SB/NL/2.0	$3.055e+02$	$1.728e+03$	$1.835e-04$	$1.015e+04$
		LB/NL/2.0	$\mathbf{1.025e+02}$	$1.021e+03$	$5.849e-05$	$1.021e+04$
TP1	10	SB/L/2.0	$2.096e+01$	$2.129e+01$	$3.641e+00$	$1.309e+02$
		LB/L/2.0	$\mathbf{1.944e+01}$	$2.379e+01$	$2.576e+00$	$1.359e+02$
		SB/NL/1.0	$6.709e+02$	$9.798e+02$	$3.819e+00$	$4.183e+03$
		LB/NL/1.0	$2.841e+03$	$1.542e+04$	$1.177e-01$	$9.001e+04$
	50	SB/L/2.0	$7.509e+05$	$2.821e+05$	$2.696e+05$	$1.721e+06$
		LB/L/2.0	$6.395e+05$	$2.642e+05$	$1.250e+05$	$1.365e+06$
		SB/NL/1.0	$3.279e+03$	$1.538e+04$	$6.913e+01$	$9.015e+04$
		LB/NL/2.0	$\mathbf{3.031e+03}$	$1.541e+04$	$1.832e+01$	$9.016e+04$
	100	SB/L/2.0	$1.411e+07$	$3.222e+06$	$6.315e+06$	$2.110e+07$
		LB/L/2.0	$1.311e+07$	$2.922e+06$	$7.330e+06$	$2.147e+07$
		SB/NL/1.0	$8.806e+04$	$3.708e+05$	$2.359e+02$	$3.069e+06$
		LB/NL/2.0	$\mathbf{1.442e+03}$	$9.031e+03$	$1.621e+02$	$9.060e+04$
TP2	10	SB/L/2.0	$1.025e+01$	$3.122e+00$	$2.479e+00$	$1.808e+01$
		LB/L/2.0	$9.866e+00$	$3.169e+00$	$4.150e+00$	$1.755e+01$
		SB/NL/2.0	$9.233e+00$	$3.253e+00$	$2.985e+00$	$1.845e+01$
		LB/NL/2.0	$\mathbf{7.302e+00}$	$3.347e+00$	$9.950e-01$	$1.792e+01$
	50	SB/L/2.0	$2.751e+02$	$2.650e+01$	$1.985e+02$	$3.284e+02$
		LB/L/2.0	$\mathbf{2.707e+02}$	$2.125e+01$	$2.093e+02$	$3.222e+02$
		SB/NL/2.0	$2.934e+02$	$3.642e+01$	$1.588e+02$	$3.530e+02$
		LB/NL/2.0	$2.793e+02$	$4.174e+01$	$1.668e+02$	$3.598e+02$
	100	**SB/L/2.0**	$\mathbf{7.746e+02}$	$3.730e+01$	$6.324e+02$	$8.545e+02$
		LB/L/2.0	$7.758e+02$	$3.902e+01$	$6.610e+02$	$8.473e+02$
		SB/NL/2.0	$8.544e+02$	$4.779e+01$	$6.957e+02$	$9.499e+02$
		LB/NL/2.0	$8.392e+02$	$5.525e+01$	$6.920e+02$	$9.258e+02$
TP3	10	SB/L/2.0	$3.166e-01$	$1.326e-01$	$9.189e-02$	$6.199e-01$
		LB/L/2.0	$2.896e-01$	$1.180e-01$	$7.101e-02$	$6.369e-01$
		SB/NL/1.0	$1.350e-01$	$1.060e-01$	$7.396e-03$	$5.944e-01$
		LB/NL/1.0	$\mathbf{7.808e-02}$	$5.022e-02$	$0.000e+00$	$2.753e-01$
	50	SB/L/2.0	$2.008e+01$	$4.572e+00$	$1.052e+01$	$4.020e+01$
		LB/L/2.0	$1.853e+01$	$4.383e+00$	$8.683e+00$	$3.427e+01$
		SB/NL/2.0	$5.325e-01$	$8.326e-01$	$3.273e-05$	$4.649e+00$
		LB/NL/2.0	$\mathbf{1.034e-02}$	$1.817e-02$	$2.463e-010$	$8.768e-02$
	100	SB/L/2.0	$1.575e+02$	$2.284e+01$	$9.160e+01$	$2.192e+02$
		LB/L/2.0	$1.566e+02$	$2.039e+01$	$1.150e+02$	$2.062e+02$
		SB/NL/2.0	$2.101e+00$	$1.315e+01$	$5.689e-05$	$9.397e+01$
		LB/NL/2.0	$\mathbf{3.826e-01}$	$4.391e-01$	$2.724e-03$	$3.173e+00$

Table 4 continued

Problem	Dimension	Algorithm	Mean	SD	Min	Max
TP4	10	SB/L/2.0	$1.037e-01$	$1.042e-01$	$2.589e-02$	$9.993e-01$
		LB/L/2.0	$7.962e-02$	$6.811e-02$	$6.170e-03$	$3.781e-01$
		SB/NL/2.0	$1.580e-01$	$3.890e-01$	$2.774e-04$	$1.646e+00$
		LB/NL/2.0	$\mathbf{1.176e-02}$	$1.155e-01$	$9.948e-014$	$1.155e+00$
	50	SB/L/2.0	$7.821e+00$	$6.142e-01$	$6.135e+00$	$9.182e+00$
		LB/L/2.0	$\mathbf{7.675e+00}$	$5.482e-01$	$6.279e+00$	$8.775e+00$
		SB/NL/2.0	$9.738e+00$	$8.693e-01$	$7.080e+00$	$1.122e+01$
		LB/NL/2.0	$9.513e+00$	$7.807e-01$	$7.493e+00$	$1.084e+01$
	100	SB/L/2.0	$1.224e+01$	$4.918e-01$	$1.089e+01$	$1.353e+01$
		LB/L/2.0	$\mathbf{1.214e+01}$	$4.772e-01$	$1.081e+01$	$1.311e+01$
		SB/NL/2.0	$1.416e+01$	$5.042e-01$	$1.309e+01$	$1.521e+01$
		LB/NL/2.0	$1.416e+01$	$4.283e-01$	$1.300e+01$	$1.491e+01$

problem and dimension. Also, the algorithm with the smallest mean is boldfaced per problem instance.

A quick inspection of Table 4 verifies that there is no single variant dominating all the rest. This was anticipated, since the combination of different schemes and parameter values can equip the algorithm with significantly different exploration/exploitation properties. However, we can clearly identify some variants that habitually exhibit good performance. Specifically, the best SB/L approach (Sum-Best with linear ranking) was the one with selection pressure $s = 2.0$ in all test problems. This value corresponds to a purely elitist linear ranking (see Sect. 3.2). The same holds also for the best LB/L approach (LocalBest with linear ranking), where $s = 2.0$ was again the dominant selection pressure value.

Thus, the experimental evidence suggests that the linear ranking variants of PSO-NBA perform better under high selection elitism. This is a consequence of the algorithm's exploration dynamic, which is increased due to the neighborhood-based budget allocation scheme. The increased exploration is counterbalanced with the intense exploitation imposed through selection elitism.

Elitism was proved to be beneficial also for the non-linear (NL) selection schemes. Indeed, power weight $\rho = 2.0$ was shown to be superior to $\rho = 1.0$ in 10 out of 15 cases for the SB/NL variants, and in 13 out of 15 cases for the LB/NL variants, as reported in Table 4. Obviously, the power selection with $\rho = 2.0$ in Eq. (15) provides a significant advantage to neighborhoods with better solution quality by assigning them exponentially higher selection probabilities. Therefore, selection elitism is promoted also in this case. Another interesting observation is that the superiority of $\rho = 1.0$ (observed only in NL-based variants), was restricted in the 10-dimensional instances of the problems, with the exception of TP1. This exception can be ascribed to the fact that TP1 becomes easier problem when its dimension increases [30].

Overall, the LB/NL/2.0 variant was the most successful one, outperforming the rest in 9 out of 15 cases. Also, the LB-based variants dominated the SB-based variants in all cases except one. Finally, the dominant variants were based on the NL scheme in 10 out of 15 cases.

A similar set of experiments was conducted also for TP5-TP10. All the ten SOBA-based variants of PSO-NBA were applied on these problems, using the same experimental setting and analysis with problems TP0-TP4. The results for this case are reported in Table 5. As we can see, the LB/NL approaches were dominant in half of the problems and, specifically, the ones of higher dimension (TP5, TP9, and TP10). In the rest, the variants that are based on linear ranking (SB/L and LB/L) exhibited the best performance. Thus, dimensionality was verified to play a crucial role on efficiency.

The aforementioned observations identify clear tendencies and indications regarding the superiority of some schemes. However, further statistical evidence (e.g., the reported standard deviations in the tables) suggested that some of the observed differences might be statistically insignificant. In order to gain more sound insight, we conducted pairwise statistical significance tests for all variants. Specifically, we conducted Wilcoxon rank-sum tests at significance level 99 % for each pair of the studied variants (including the ones that are not reported in Tables 4 and 5). Recall that there were 10 algorithmic variants and 21 different problem instances in total. Thus, each variant had 9 competitors over 21 problem instances, which results in $9 \times 21 = 189$ statistical tests in total per algorithmic variant. For each test where algorithm A was superior to B with statistical significance, we counted a win for A and a loss for B. If there was no statistical

Table 5 Results for the SOBA approach for test problems TP5-TP10 (nonlinear systems)

Problem	Dimension	Algorithm	Mean	SD	Min	Max
TP5	10	SB/L/2.0	$6.312e-03$	$3.557e-03$	$1.035e-03$	$1.855e-02$
		LB/L/2.0	$5.523e-03$	$3.189e-03$	$1.070e-03$	$1.625e-02$
		SB/NL/1.0	$5.155e-04$	$1.912e-03$	$3.978e-07$	$1.538e-02$
		LB/NL/2.0	$\mathbf{4.833e-10}$	$2.108e-09$	$6.556e-015$	$1.221e-08$
TP6	6	**SB/L/2.0**	$\mathbf{3.751e-03}$	$5.196e-03$	$4.176e-05$	$2.578e-02$
		LB/L/2.0	$4.020e-03$	$7.317e-03$	$7.693e-06$	$4.181e-02$
		SB/NL/1.0	$1.514e-01$	$2.579e-01$	$6.463e-09$	$9.859e-01$
		LB/NL/1.0	$9.961e-02$	$2.432e-01$	$0.000e+00$	$9.363e-01$
TP7	5	SB/L/2.0	$1.904e-01$	$1.360e-01$	$2.017e-02$	$6.531e-01$
		LB/L/1.5	$\mathbf{1.741e-01}$	$1.066e-01$	$1.610e-02$	$5.573e-01$
		SB/NL/1.0	$3.078e-01$	$2.387e-01$	$3.518e-02$	$1.179e+00$
		LB/NL/1.0	$2.201e-01$	$1.721e-01$	$6.671e-03$	$7.199e-01$
TP8	8	SB/L/2.0	$3.419e-01$	$2.033e-01$	$2.969e-02$	$1.176e+00$
		LB/L/2.0	$\mathbf{2.926e-01}$	$1.552e-01$	$7.430e-02$	$7.061e-01$
		SB/NL/1.0	$3.167e-01$	$2.444e-01$	$1.055e-02$	$1.085e+00$
		LB/NL/1.0	$3.208e-01$	$2.509e-01$	$5.308e-03$	$9.437e-01$
TP9	10	SB/L/2.0	$8.820e-02$	$6.530e-02$	$8.032e-04$	$3.955e-01$
		LB/L/2.0	$7.770e-02$	$5.849e-02$	$5.178e-03$	$2.526e-01$
		SB/NL/1.0	$4.683e-02$	$4.217e-02$	$6.689e-04$	$2.013e-01$
		LB/NL/2.0	$\mathbf{1.648e-02}$	$1.933e-02$	$1.856e-05$	$9.591e-02$
TP10	20	SB/L/2.0	$2.227e-04$	$2.215e-04$	$1.107e-06$	$8.203e-04$
		LB/L/2.0	$1.236e-04$	$2.127e-04$	$2.809e-07$	$1.199e-03$
		SB/NL/1.0	$2.099e-03$	$6.839e-03$	$1.053e-278$	$5.399e-02$
		LB/NL/1.0	$\mathbf{4.881e-07}$	$2.167e-06$	$2.220e-262$	$1.497e-05$

significance between them, we counted a draw for both algorithms. The results of these tests are illustrated in Figs. 2, 3, 4.

Figure 2 illustrates the number of wins, losses, and draws for all studied variants. The superiority of the LB/NL/2.0 variant is confirmed against the rest. On the other hand, SB/L/1.0 and LB/L/1.0 are evidently the worst combinations as they exhibit the highest number of losses. Besides each variant individually, we collectively considered the four main categories with respect to the combination of quality criterion and selection probability, namely SB/L, LB/L, SB/NL, and LB/NL. For each category, we computed the number of wins, draws, and losses as the sum of the corresponding values for all variants that comprise it. The results are reported in Fig. 3 where we can clearly see the tendency of LB/NL to produce more efficient variants. The combinations SB/L and LB/L are the worst (they have the highest number of losses), exhibiting similar behavior between them.

In order to further probe the influence of the selection pressure and the nonlinear weight, we performed Wilcoxon rank-sum tests between pairs of variants that use the same neighborhood scoring approach (SB or LB) and probability selection scheme (L or NL) but different values of selection pressure. These results are reported in Fig. 4 and denoted as L/2.0 vs L/1.0, L/2.0 vs L/1.5, and L/1.5 vs L/1.0, where L/s stands for all approaches with linear ranking (L) and selection probability s. Similar analysis was conducted also for the nonlinear approaches (NL) for different values of the power weight. This case is denoted as NL/2.0 vs NL/1.0 in Fig. 4. There is an apparently monotonic superiority for the selection pressure values, i.e., $s = 2.0$ prevails $s = 1.5$, which in turn prevails $s = 1.0$. Again, this verifies the benefits of increased elitism in the proposed PSO-NBA variants. The same can be inferred also for the nonlinear weight, since the elitistic choice $\rho = 2.0$ has almost twice as many wins as $\rho = 1.0$.

4.2 Assessment of MOBA strategy

The MOBA strategy assesses each neighborhood using two criteria instead of one, as described in Sect. 3.4. The first criterion is solution quality while the second one is diversity of the best positions involved in the neighborhood. We considered two different ways to cope with the multi-objective scoring, namely weighted aggregation (LWA and

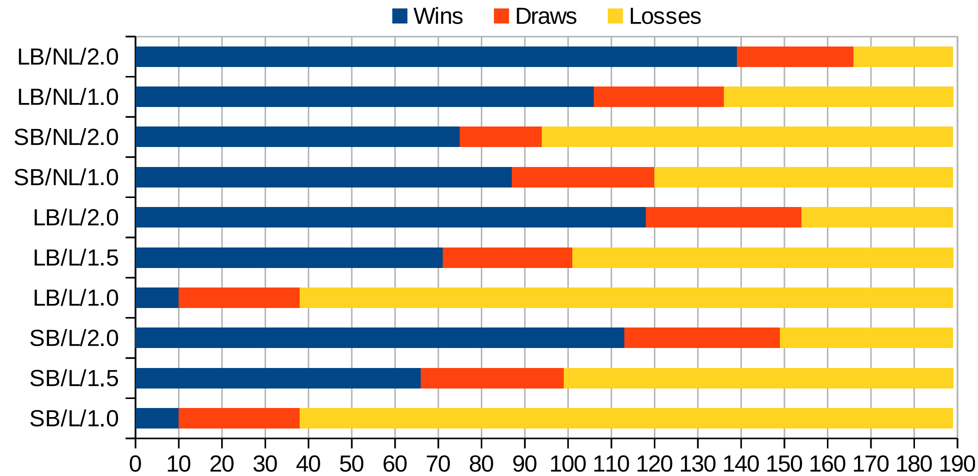

Fig. 2 Number of wins, draws, and losses for all SOBA variants

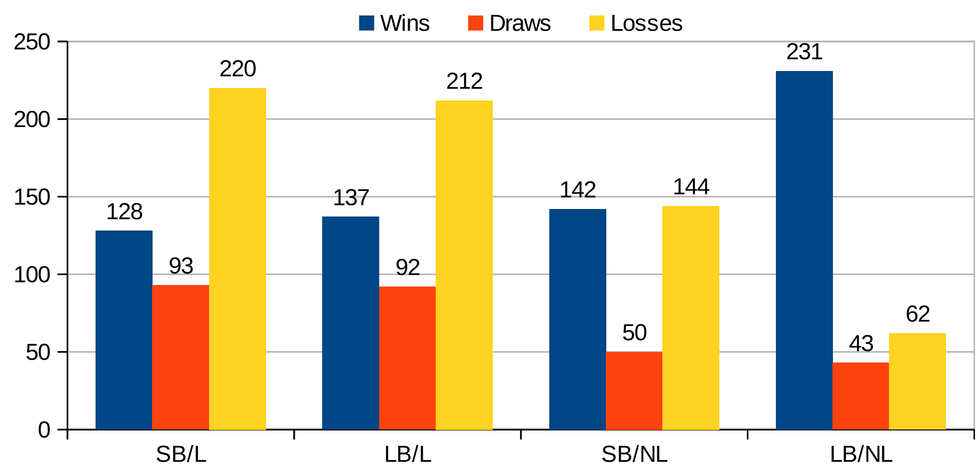

Fig. 3 Aggregate number of wins, draws, and losses for different combinations of quality criteria and selection probability in SOBA-based variants

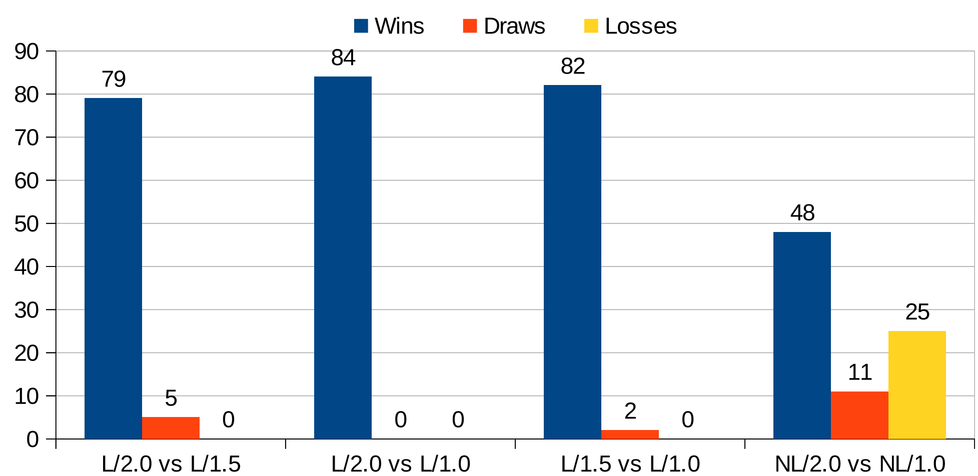

Fig. 4 Aggregate number of wins, draws, and losses for pairs of variants with the same selection probability scheme (L or NL) but different parameters in SOBA-based variants

DWA) and the Pareto front (PFA) approach (Sects. 3.4.1 and 3.4.2). For notation purposes, we extended the formalism of Sect. 4.1 as follows:

$$\mathtt{A} \;/\; \mathtt{X} \;/\; \mathtt{Y} \;/\; \mathtt{Z},$$

where $\mathtt{A}$ takes the values DW (for DWA), LW (for LWA), and PF (for PFA), while $\mathtt{X} \in \{\mathrm{SB}, \mathrm{LB}\}$ and $\mathtt{Y} \in \{\mathrm{L}, \mathrm{NL}\}$. Experimental results for all MOBA approaches are reported and analyzed in the following sections.

4.2.1 Results for weighted aggregation approaches

Initially, we studied the weighted aggregation approach, which is based on the conversion of the multi-objective scoring to a single-objective one as described in Sect. 3.4.1. We considered both the linear weighted aggregation (LWA) and the dynamic weighted aggregation (DWA) approaches. The neighborhoods' quality was determined based on the SB and LB schemes, while diversity was quantified through the AD scheme (see Sects. 3.1.1, 3.1.2, and 3.1.3, respectively). The L and NL selection probability approaches (see Sect. 3.2) were combined with the aforementioned schemes. The parameter setting of Table 3 was used also here, resulting in ten LWA and ten DWA variants.

All variants were applied on all instances of test problems TP0-TP10. The best variants were distinguished per problem instance on the basis of the average best solution value within the prespecified computational budget over 100 experiments. The results for TP0-TP4 are reported in Table 6 and for TP5-TP10 in Table 7.

Table 6 offers two interesting observations. First, the LB-based variants clearly dominate the SB-based variants in 12 out of 15 cases. Moreover, the NL/2.0 approaches performed better in 11 out of 15 cases, while L/2.0 approaches were the best in the rest 4 cases. These findings are aligned with the ones for the SOBA strategy in the previous section.

However, the picture becomes complicated when DWA is considered against LWA. In Table 6, there is no clear tendency for either of the two approaches. In fact, DWA was superior in 7 out of 15 cases, while LWA appeared as a better choice for the rest 8 problem instances. Therefore, no clear conclusion can be derived from these results. Yet, we ascertain that DWA performs better when combined with LB/NL approaches. On the other hand, the LWA approaches do not favor a single combination. Indeed, LW/LB/NL appears 4 times in Table 6, while LW/LB/L and LW/SB/L appear 2 times each.

The second set of test problems offers similar conclusions. As we can see in Table 7, DWA is distinguished in half of the cases and LWA in the rest. However, this time we can see that all variants are based on the LB/NL combination. Interestingly, DWA dominates in the three high-dimensional problems (TP5, TP9, and TP10), while LWA is distinguished in the lower-dimensional cases.

In order to facilitate comparisons between different variants, we conducted Wilcoxon rank-sum tests among all LWA and DWA variants at significance level 99 %, similarly to the SOBA approach. Figure 5 illustrates the number of wins, draws, and losses per algorithmic variant. As we can see, there is an indisputable predominance of the LB/NL/2.0 variants both for DWA and LWA, with the later exhibiting the highest number of wins. This is in line with our observations in Tables 6 and 7.

Similarly to the SOBA approaches, we also considered the four main categories SB/L, LB/L, SB/NL, and LB/NL, both for LWA and DWA. For each category, we computed the aggregate number of wins, draws, and losses. The results are reported in the net chart of Fig. 6, where we can clearly verify the previous findings. Finally, L- and NL-based approaches for both LWA and DWA were compared with different parameter values. The results are reported in Fig. 7, where we can verify the monotonic decline of performance as selection pressure decreases (suppressing elitism) as well as the superiority of higher power weight values in NL-based variants.

4.2.2 Results for Pareto front approach

The PFA approach uses a radically different mechanism for neighborhood scoring than the previous SOBA and MOBA approaches. Specifically, each neighborhood is assessed with two distinct criteria, namely solution quality and diversity. These criteria are not combined as in the weighted aggregation approaches. Instead, they are used for vectorial comparisons between neighborhoods in the sense of Pareto dominance. The comparisons are conducted through a tournament selection mechanism in order to avoid search stagnation.

The tournament size is usually an influential factor in tournament selection. For this reason, three different values were used, i.e., $T = N/2, N/3, N/5$, where N is the swarm size. The combinations of the neighborhood scoring schemes with the different tournament sizes resulted in six PFA variants. We denote each combination with the notation,

$$\texttt{PF} \;/\; \texttt{X} \;/\; \texttt{TS},$$

where $\texttt{X} \in \{\text{SB}, \text{LB}\}$ and $\texttt{TS} \in \{2, 3, 5\}$. For example, PF/LB/2 stands for the PFA variant with LB neighborhood scoring and tournament size $T = N/2$, whereas PF/SB/5 denotes the PFA variant with SB neighborhood scoring mode and tournament size $T = N/5$. The experimental setting was identical to the previous cases of MOBA and SOBA strategies. Table 8 reports the best solution values for each problem instance, averaged over 100 experiments. For presentation compactness reasons, we report only the best SB-based and LB-based variants per case.

In the upper part of Table 8 (problems TP0-TP4), the variant SB/2 is distinguished in 12 out of 15 problem instances. Also, the TS = 2 (i.e., $T = N/2$) case appeared as the most efficient in 10 out of 15 cases. The SB approach implies reduced elitism than LB. On the other hand, smaller values of TS correspond to higher tournament sizes

Table 6 Results for the MOBA weighted aggregation approaches (LWA and DWA) for test problems TP0-TP4 (standard test suite)

Problem	Dimension	Algorithm	Mean	SD	Min	Max
TP0	10	DW/SB/L/2.0	$3.630e-01$	$2.251e-01$	$5.058e-02$	$9.873e-01$
		LW/LB/L/2.0	$3.405e-01$	$2.452e-01$	$2.447e-02$	$1.493e+00$
		DW/SB/NL/1.0	$7.666e-03$	$1.835e-02$	$7.828e-06$	$1.430e-01$
		DW/LB/NL/2.0	$\mathbf{1.992e-15}$	$1.227e-14$	$1.236e-22$	$1.171e-13$
	50	LW/SB/L/2.0	$3.128e+03$	$4.198e+02$	$2.149e+03$	$4.294e+03$
		LW/LB/L/2.0	$3.052e+03$	$4.056e+02$	$2.008e+03$	$3.832e+03$
		DW/SB/NL/2.0	$\mathbf{6.121e+00}$	$2.768e+01$	$7.749e-03$	$2.363e+02$
		DW/LB/NL/2.0	$1.300e+01$	$1.174e+02$	$1.763e-05$	$1.174e+03$
	100	LW/SB/L/2.0	$2.163e+04$	$1.731e+03$	$1.725e+04$	$2.582e+04$
		DW/LB/L/2.0	$2.136e+04$	$1.947e+03$	$1.598e+04$	$2.582e+04$
		DW/SB/NL/2.0	$1.353e+04$	$7.436e+03$	$4.331e+01$	$2.525e+04$
		DW/LB/NL/2.0	$\mathbf{1.269e+04}$	$7.356e+03$	$1.838e+01$	$2.493e+04$
TP1	10	LW/SB/L/2.0	$3.417e+01$	$2.548e+01$	$6.264e+00$	$1.383e+02$
		LW/LB/L/2.0	$3.057e+01$	$2.338e+01$	$4.785e+00$	$1.446e+02$
		LW/SB/NL/1.0	$2.332e+01$	$3.548e+01$	$4.790e-01$	$2.089e+02$
		LW/LB/NL/1.0	$\mathbf{2.156e+01}$	$3.934e+01$	$1.631e-02$	$2.234e+02$
	50	LW/SB/L/2.0	$1.122e+06$	$2.704e+05$	$5.704e+05$	$1.795e+06$
		LW/LB/L/2.0	$1.109e+06$	$2.918e+05$	$4.950e+05$	$1.713e+06$
		LW/SB/NL/2.0	$7.607e+03$	$1.965e+04$	$1.950e+02$	$9.306e+04$
		DW/LB/NL/2.0	$\mathbf{4.047e+03}$	$1.770e+04$	$2.238e+01$	$9.027e+04$
	100	LW/SB/L/2.0	$1.717e+07$	$2.412e+06$	$1.075e+07$	$2.258e+07$
		LW/LB/L/2.0	$1.662e+07$	$2.122e+06$	$1.085e+07$	$2.109e+07$
		DW/SB/NL/2.0	$1.686e+04$	$2.952e+04$	$5.324e+02$	$1.459e+05$
		LW/LB/NL/2.0	$\mathbf{3.752e+03}$	$9.879e+03$	$3.929e+02$	$9.757e+04$
TP2	10	DW/SB/L/2.0	$1.155e+01$	$2.976e+00$	$4.488e+00$	$2.054e+01$
		LW/LB/L/2.0	$1.066e+01$	$2.611e+00$	$5.273e+00$	$1.881e+01$
		LW/SB/NL/2.0	$1.036e+01$	$3.514e+00$	$3.239e+00$	$2.040e+01$
		LW/LB/NL/2.0	$\mathbf{8.997e+00}$	$3.495e+00$	$2.252e+00$	$1.813e+01$
	50	**LW/SB/L/2.0**	$\mathbf{2.932e+02}$	$1.974e+01$	$2.295e+02$	$3.427e+02$
		LW/LB/L/2.0	$2.943e+02$	$1.777e+01$	$2.411e+02$	$3.426e+02$
		LW/SB/NL/2.0	$3.129e+02$	$2.673e+01$	$1.966e+02$	$3.572e+02$
		LW/LB/NL/2.0	$3.063e+02$	$3.032e+01$	$2.174e+02$	$3.672e+02$
	100	LW/SB/L/2.0	$8.127e+02$	$2.755e+01$	$7.487e+02$	$8.725e+02$
		LW/LB/L/2.0	$\mathbf{8.124e+02}$	$2.738e+01$	$7.312e+02$	$8.610e+02$
		LW/SB/NL/2.0	$8.830e+02$	$3.096e+01$	$7.892e+02$	$9.481e+02$
		LW/LB/NL/2.0	$8.665e+02$	$3.122e+01$	$7.624e+02$	$9.261e+02$
TP3	10	LW/SB/L/2.0	$4.752e-01$	$1.339e-01$	$9.608e-02$	$8.261e-01$
		DW/LB/L/2.0	$4.543e-01$	$1.362e-01$	$1.699e-01$	$8.738e-01$
		LW/SB/NL/2.0	$1.928e-01$	$1.322e-01$	$2.464e-02$	$7.160e-01$
		DW/LB/NL/2.0	$\mathbf{1.038e-01}$	$6.630e-02$	$7.396e-03$	$3.327e-01$
	50	LW/SB/L/2.0	$2.913e+01$	$4.014e+00$	$1.977e+01$	$3.863e+01$
		DW/LB/L/2.0	$2.811e+01$	$3.823e+00$	$1.760e+01$	$3.660e+01$
		DW/SB/NL/2.0	$6.145e-01$	$2.668e+00$	$1.637e-03$	$2.628e+01$
		DW/LB/NL/2.0	$\mathbf{3.970e-01}$	$4.026e-01$	$1.424e-03$	$1.613e+00$
	100	LW/SB/L/2.0	$1.966e+02$	$1.701e+01$	$1.535e+02$	$2.424e+02$
		LW/LB/L/2.0	$1.927e+02$	$1.824e+01$	$1.415e+02$	$2.443e+02$
		DW/SB/NL/2.0	$1.270e+02$	$6.175e+01$	$8.447e-01$	$2.356e+02$
		DW/LB/NL/2.0	$\mathbf{1.101e+02}$	$6.169e+01$	$3.474e+00$	$2.254e+02$

Table 6 continued

Problem	Dimension	Algorithm	Mean	SD	Min	Max
TP4	10	LW/SB/L/2.0	$4.714e-01$	$3.097e-01$	$6.339e-02$	$1.699e+00$
		DW/LB/L/2.0	$4.864e-01$	$3.246e-01$	$1.117e-01$	$1.627e+00$
		LW/SB/NL/2.0	$2.485e-01$	$4.314e-01$	$1.808e-03$	$1.597e+00$
		LW/LB/NL/2.0	**1.106e − 01**	$3.406e-01$	$1.189e-06$	$1.524e+00$
	50	LW/SB/L/2.0	$8.918e+00$	$4.258e-01$	$7.794e+00$	$9.861e+00$
		LW/LB/L/2.0	**8.863e + 00**	$3.804e-01$	$7.517e+00$	$9.573e+00$
		LW/SB/NL/2.0	$1.020e+01$	$4.392e-01$	$8.762e+00$	$1.108e+01$
		LW/LB/NL/2.0	$1.021e+01$	$4.662e-01$	$8.583e+00$	$1.097e+01$
	100	**LW/SB/L/2.0**	**1.306e + 01**	$3.285e-01$	$1.190e+01$	$1.375e+01$
		LW/LB/L/2.0	$1.312e+01$	$3.119e-01$	$1.217e+01$	$1.373e+01$
		LW/SB/NL/2.0	$1.433e+01$	$3.070e-01$	$1.328e+01$	$1.490e+01$
		LW/LB/NL/2.0	$1.428e+01$	$2.843e-01$	$1.352e+01$	$1.485e+01$

Table 7 Results for the MOBA weighted aggregation approaches (LWA and DWA) for test problems TP5-TP10 (nonlinear systems)

Problem	Dimension	Algorithm	Mean	SD	Min	Max
TP5	10	LW/SB/L/2.0	$2.405e-02$	$8.260e-03$	$6.498e-03$	$5.362e-02$
		LW/LB/L/2.0	$2.183e-02$	$8.869e-03$	$5.610e-03$	$5.908e-02$
		LW/SB/NL/2.0	$4.517e-03$	$5.577e-03$	$2.076e-04$	$3.597e-02$
		DW/LB/NL/2.0	**8.575e − 06**	$6.669e-05$	$1.835e-10$	$6.651e-04$
TP6	6	LW/SB/L/2.0	$9.637e-03$	$8.184e-03$	$5.393e-04$	$3.941e-02$
		LW/LB/L/2.0	$7.083e-03$	$5.662e-03$	$5.466e-04$	$2.818e-02$
		LW/SB/NL/1.0	$4.218e-03$	$5.612e-03$	$1.146e-05$	$2.426e-02$
		LW/LB/NL/2.0	**1.060e − 03**	$3.762e-03$	$2.321e-16$	$2.509e-02$
TP7	5	LW/SB/L/2.0	$1.753e-01$	$1.029e-01$	$2.367e-02$	$5.233e-01$
		LW/LB/L/2.0	$1.780e-01$	$9.777e-02$	$2.886e-02$	$4.870e-01$
		LW/SB/NL/1.0	$1.947e-01$	$1.303e-01$	$7.659e-03$	$7.236e-01$
		LW/LB/NL/1.0	**1.411e − 01**	$1.008e-01$	$6.790e-03$	$5.604e-01$
TP8	8	LW/SB/L/2.0	$4.043e-01$	$1.708e-01$	$7.351e-02$	$9.027e-01$
		LW/LB/L/2.0	$3.387e-01$	$1.687e-01$	$5.565e-02$	$8.826e-01$
		LW/SB/NL/2.0	$3.583e-01$	$1.632e-01$	$5.907e-02$	$8.943e-01$
		LW/LB/NL/2.0	**2.334e − 01**	$1.864e-01$	$7.806e-03$	$7.858e-01$
TP9	10	DW/SB/L/2.0	$1.566e-01$	$8.544e-02$	$1.537e-02$	$4.686e-01$
		LW/LB/L/2.0	$1.401e-01$	$8.279e-02$	$1.056e-02$	$4.241e-01$
		DW/SB/NL/1.0	$8.990e-02$	$7.890e-02$	$1.983e-03$	$5.038e-01$
		DW/LB/NL/2.0	**3.616e − 02**	$6.486e-02$	$4.262e-04$	$5.240e-01$
TP10	20	LW/SB/L/2.0	$1.235e-03$	$1.196e-03$	$1.832e-05$	$5.835e-03$
		DW/LB/L/2.0	$9.602e-04$	$1.107e-03$	$1.039e-05$	$7.384e-03$
		LW/SB/NL/1.0	$4.581e-04$	$1.152e-03$	$1.776e-15$	$9.323e-03$
		DW/LB/NL/1.0	**4.877e − 06**	$1.583e-05$	$4.878e-134$	$1.100e-04$

T, which promote elitism since the overall best individual has higher probability of being selected. Thus, it may be reasonable to assume that the lower values of TS counterbalance the selection of the SB approach in terms of elitism.

The picture changes in the lower part of Table 8 (TP5-TP10). In these cases, the LB-based approaches outperform the rest in all but one problem. Also, the tournament size seems to be problem-dependent. This evidence suggests that elitism plays significant role in TP5-TP10. This is in

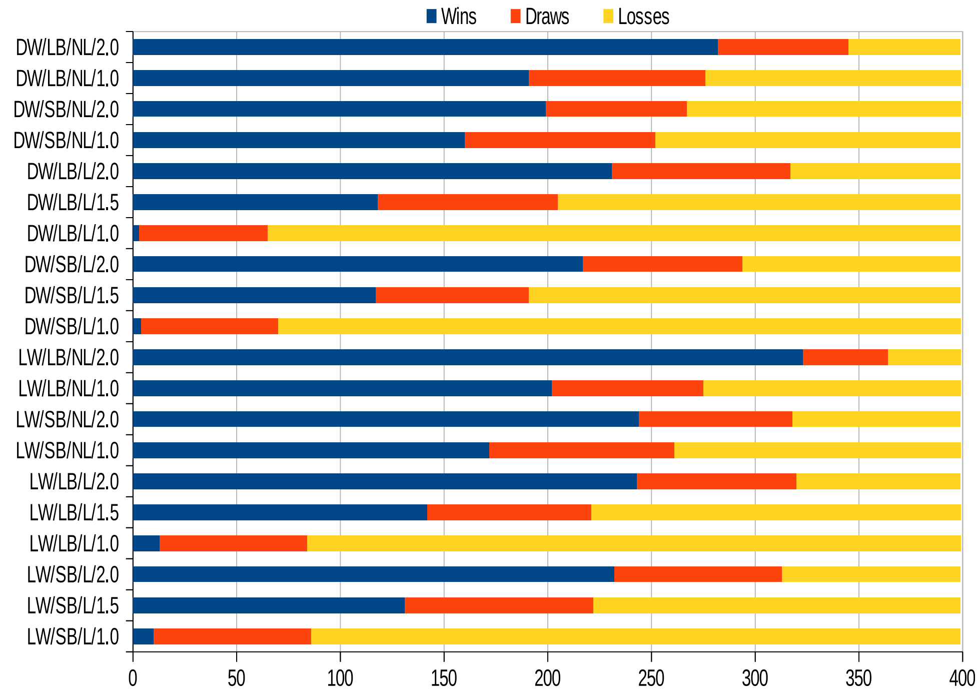

Fig. 5 Number of wins, draws, and losses for the MOBA-based variants LWA and DWA

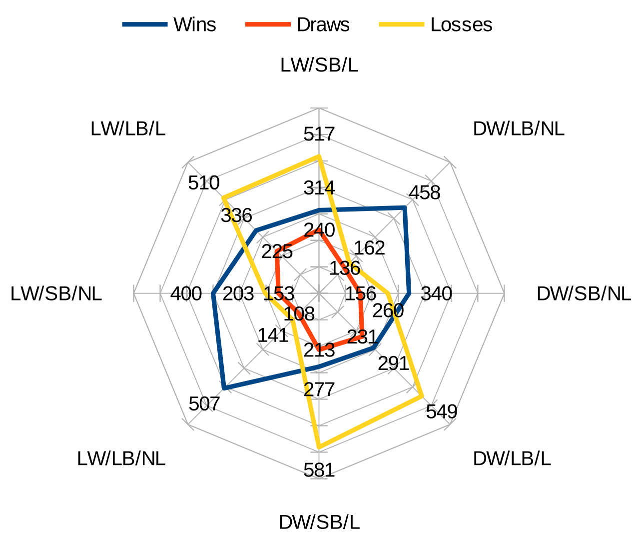

Fig. 6 Aggregate number of wins, draws, and losses for different combinations of quality criteria and selection probability in LWA and DWA approaches

accordance with previous findings for the rest of SOBA and MOBA variants.

Following the analysis of previous sections, we conducted Wilcoxon rank-sum tests among all PFA variants. Figure 8 illustrates the number of wins, draws, and losses per variant. The statistical evidence clearly shows a monotonic decline of performance as TS increases. Also, the SB-based variants seem to prevail especially for lower TS values.

In order to further explore the impact of tournament size, we conducted Wilcoxon rank-sum tests between variants that use identical scoring schemes but different tournament sizes. Then, for each tournament size we summed up the corresponding number of wins, draws, and losses. The results are reported in Fig. 9. Clearly, TS = 2 is the best choice, verifying the monotonic decline as its value increases. Therefore, smaller tournament sizes produce less efficient approaches evidently due to reduced elitism.

4.3 Comparative results

In the previous sections, we individually studied each strategy of the proposed PSO-NBA approach. In this section, we offer comparisons among all the presented variants. This includes the SOBA approaches, as well as all MOBA approaches (LWA, DWA, and PFA). The comparisons were all based on test problems TP0-TP10. Moreover, we report results from comparisons with other algorithms.

First, we compared all PSO-NBA approaches among them for all test problems. Each pairwise comparison was based on Wilcoxon rank-sum tests at significance level of 99 %. For each algorithm, we recorded its aggregate number of wins, draws, and losses. These results are reported in Table 9.

We can make two interesting observations in Table 9. First, we can easily notice that the LB/NL/2.0 approach prevails both in SOBA and MOBA strategies (boldfaced entries in Table 9). This was also pointed out in the previous

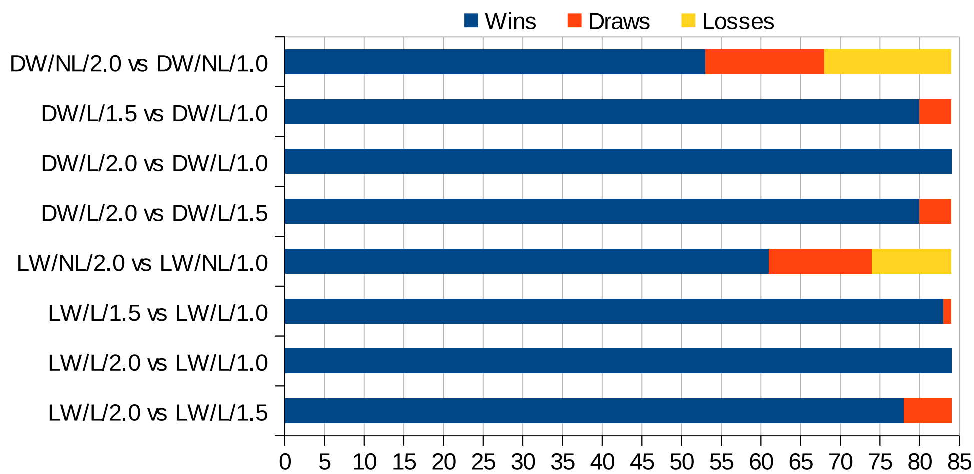

Fig. 7 Aggregate number of wins, draws, and losses for pairs of LWA and DWA variants with identical selection probability scheme (L or NL), but different parameters

Table 8 Results for the PFA approach of MOBA strategy for test problems TP0-TP10

Problem	Dimension	Algorithm	Mean	SD	Min	Max
TP0	10	**PF/SB/2**	**4.956e − 03**	9.674*e* − 03	3.582*e* − 05	7.846*e* − 02
		PF/LB/2	7.788*e* − 03	8.127*e* − 03	1.266*e* − 04	4.324*e* − 02
	50	**PF/SB/2**	**1.059e + 01**	5.525*e* + 00	1.802*e* + 00	2.796*e* + 01
		PF/LB/2	2.527*e* + 01	1.220*e* + 01	5.353*e* + 00	7.520*e* + 01
	100	**PF/SB/2**	**1.489e + 02**	6.499*e* + 01	5.378*e* + 01	3.799*e* + 02
		PF/LB/2	2.524*e* + 02	9.412*e* + 01	1.171*e* + 02	7.451*e* + 02
TP1	10	**PF/SB/5**	**1.346e + 01**	1.532*e* + 01	2.117*e* + 00	8.625*e* + 01
		PF/LB/3	1.350*e* + 01	2.095*e* + 01	2.389*e* − 01	1.138*e* + 02
	50	PF/SB/3	4.120*e* + 03	1.259*e* + 04	2.363*e* + 02	9.155*e* + 04
		PF/LB/3	**3.651e + 03**	9.885*e* + 03	6.327*e* + 02	9.083*e* + 04
	100	**PF/SB/2**	**3.076e + 04**	2.883*e* + 04	4.238*e* + 03	1.160*e* + 05
		PF/LB/2	4.517*e* + 04	3.261*e* + 04	1.270*e* + 04	1.586*e* + 05
TP2	10	PF/SB/3	8.552*e* + 00	3.468*e* + 00	9.954*e* − 01	1.994*e* + 01
		PF/LB/5	**8.224e + 00**	3.116*e* + 00	2.985*e* + 00	1.866*e* + 01
	50	**PF/SB/5**	**1.439e + 02**	2.614*e* + 01	9.102*e* + 01	2.098*e* + 02
		PF/LB/5	1.463*e* + 02	2.943*e* + 01	7.830*e* + 01	1.990*e* + 02
	100	**PF/SB/5**	**3.814e + 02**	4.841*e* + 01	2.734*e* + 02	5.077*e* + 02
		PF/LB/5	4.096*e* + 02	6.149*e* + 01	2.753*e* + 02	5.311*e* + 02
TP3	10	**PF/SB/2**	**2.142e − 01**	1.265*e* − 01	4.135*e* − 02	6.241*e* − 01
		PF/LB/3	2.288*e* − 01	1.266*e* − 01	3.214*e* − 02	5.560*e* − 01
	50	**PF/SB/2**	**1.092e + 00**	5.068*e* − 02	1.011*e* + 00	1.296*e* + 00
		PF/LB/2	1.205*e* + 00	7.270*e* − 02	1.070*e* + 00	1.388*e* + 00
	100	**PF/SB/2**	**2.224e + 00**	6.657*e* − 01	1.311*e* + 00	5.931*e* + 00
		PF/LB/2	3.280*e* + 00	8.334*e* − 01	1.718*e* + 00	6.719*e* + 00
TP4	10	**PF/SB/2**	**3.168e − 02**	6.641*e* − 02	1.923*e* − 03	6.358*e* − 01
		PF/LB/2	3.543*e* − 02	3.993*e* − 02	5.304*e* − 03	3.435*e* − 01
	50	**PF/SB/2**	**2.265e + 00**	5.141*e* − 01	7.843*e* − 01	3.532*e* + 00
		PF/LB/2	2.308*e* + 00	4.078*e* − 01	1.226*e* + 00	3.431*e* + 00
	100	PF/SB/2	3.806*e* + 00	5.387*e* − 01	2.812*e* + 00	5.817*e* + 00
		PF/LB/2	**3.761e + 00**	4.558*e* − 01	2.879*e* + 00	5.400*e* + 00
TP5	10	**PF/SB/2**	**2.100e − 03**	1.444*e* − 03	5.228*e* − 05	7.846*e* − 03
		PF/LB/2	3.202*e* − 03	2.261*e* − 03	7.199*e* − 05	1.527*e* − 02

Table 8 continued

Problem	Dimension	Algorithm	Mean	SD	Min	Max
TP6	6	PF/SB/5	$5.067e-03$	$1.053e-02$	$3.121e-06$	$6.372e-02$
		PF/LB/3	**1.329e − 03**	$2.891e-03$	$7.233e-07$	$1.680e-02$
TP7	5	PF/SB/3	$1.698e-01$	$1.113e-01$	$9.561e-03$	$5.112e-01$
		PF/LB/5	**1.395e − 01**	$9.246e-02$	$9.646e-03$	$4.269e-01$
TP8	8	PF/SB/3	$3.098e-01$	$2.272e-01$	$2.095e-02$	$1.268e+00$
		PF/LB/2	**2.539e − 01**	$1.872e-01$	$1.803e-02$	$8.879e-01$
TP9	10	PF/SB/2	$4.831e-02$	$5.041e-02$	$3.136e-04$	$2.945e-01$
		PF/LB/2	**4.070e − 02**	$3.635e-02$	$1.515e-03$	$2.136e-01$
TP10	20	PF/SB/5	$1.428e-04$	$3.736e-04$	$5.837e-15$	$3.225e-03$
		PF/LB/5	**1.364e − 04**	$2.927e-04$	$2.037e-11$	$1.477e-03$

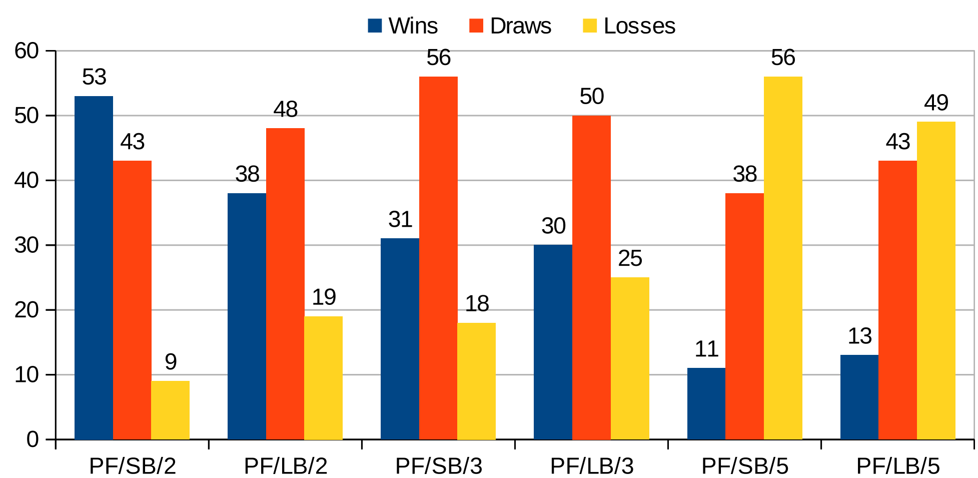

Fig. 8 Number of wins, draws, and losses for the PFA variant of MOBA strategy

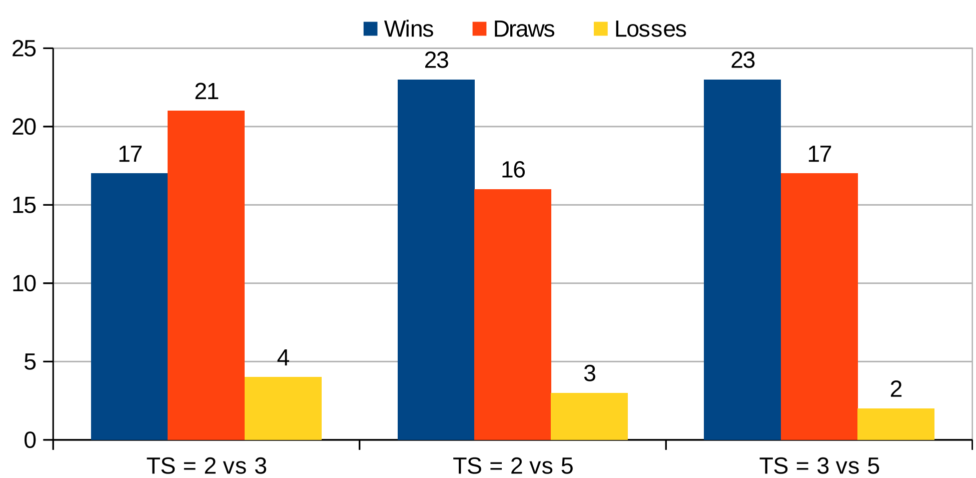

Fig. 9 Aggregate number of wins, draws, and losses for different tournament sizes in PFA variants

sections. Second, we can see that all PFA approaches exhibit a remarkably high number of wins and small number of losses. This is a new evidence, which indicates that using diversity along with solution quality for neighborhood rating can be beneficial for the algorithm.

According to Table 9, the variants LB/NL/2.0 and PF/LB/2 exhibited the highest number of wins for the SOBA and the MOBA strategies, respectively. These two variants were considered for comparisons with different PSO-based algorithms under the experimental setting of Table 3. More

Table 9 Aggregate numbers of wins, draws, and losses for all PSO-NBA variants for all test problems

Strategy	Algorithm	Wins	Draws	Losses
SOBA	SB/L/1.0	25	80	630
	SB/L/1.5	277	110	348
	SB/L/2.0	438	97	200
	SB/NL/1.0	329	149	257
	SB/NL/2.0	301	84	350
	LB/L/1.0	26	81	628
	LB/L/1.5	299	106	330
	LB/L/2.0	450	109	176
	LB/NL/1.0	398	128	209
	LB/NL/2.0	**544**	**87**	**104**
MOBA	LW/SB/L/1.0	43	102	590
	LW/SB/L/1.5	189	113	433
	LW/SB/L/2.0	327	105	303
	LW/SB/NL/1.0	252	148	335
	LW/SB/NL/2.0	357	137	241
	LW/LB/L/1.0	44	102	589
	LW/LB/L/1.5	203	102	430
	LW/LB/L/2.0	340	115	280
	LW/LB/NL/1.0	306	128	301
	LW/LB/NL/2.0	**498**	**105**	**132**
	DW/SB/L/1.0	32	90	613
	DW/SB/L/1.5	170	94	471
	DW/SB/L/2.0	303	106	326
	DW/SB/NL/1.0	240	160	335
	DW/SB/NL/2.0	307	123	305
	DW/LB/L/1.0	30	85	620
	DW/LB/L/1.5	171	112	452
	DW/LB/L/2.0	325	113	297
	DW/LB/NL/1.0	292	152	291
	DW/LB/NL/2.0	**454**	**120**	**161**
	PF/SB/2	555	114	66
	PF/SB/3	542	119	74
	PF/SB/5	508	105	122
	PF/LB/2	**575**	**88**	**72**
	PF/LB/3	566	95	74
	PF/LB/5	537	90	108

specifically, we compared them against the standard synchronous Particle Swarm Optimization (PSO) algorithm as well as its asychronous version (ASY) presented in [27]. The corresponding results are reported in Tables 10 and 11 for the standard test suite and the nonlinear systems, respectively.

The reported results reveal an apparent superiority of the PSO-NBA algorithm (by orders of magnitude) against PSO and ASY for all problems and dimensions. More specifically, the SOBA-based variant LB/NL/2.0 surmounts all

Table 10 Comparative results of PSO-NBA with PSO-based variants for test problems TP0-TP4 (standard test suite)

Problem	Dimension	Algorithm	Mean	SD
TP0	10	PSO	$3.608e+00$	$2.038e+00$
		ASY	$2.067e+00$	$1.091e+00$
		PF/LB/2	$7.788e-03$	$8.127e-03$
		LB/NL/2.0	$\mathbf{9.406e-26}$	$8.806e-25$
	50	PSO	$8.801e+03$	$9.596e+02$
		ASY	$7.162e+03$	$7.755e+02$
		PF/LB/2	$2.527e+01$	$1.220e+01$
		LB/NL/2.0	$\mathbf{3.116e-08}$	$1.332e-07$
	100	PSO	$4.808e+04$	$2.913e+03$
		ASY	$3.876e+04$	$2.494e+03$
		PF/LB/2	$2.524e+02$	$9.412e+01$
		LB/NL/2.0	$\mathbf{1.025e+02}$	$1.021e+03$
TP1	10	PSO	$2.369e+03$	$1.790e+03$
		ASY	$1.270e+03$	$8.705e+02$
		PF/LB/2	$\mathbf{2.035e+01}$	$3.011e+01$
		LB/NL/2.0	$5.330e+03$	$2.072e+04$
	50	PSO	$7.382e+08$	$1.569e+08$
		ASY	$5.187e+08$	$1.214e+08$
		PF/LB/2	$3.685e+03$	$1.269e+04$
		LB/NL/2.0	$\mathbf{3.031e+03}$	$1.541e+04$
	100	PSO	$7.760e+09$	$1.135e+09$
		ASY	$5.573e+09$	$7.742e+08$
		PF/LB/2	$4.517e+04$	$3.261e+04$
		LB/NL/2.0	$\mathbf{1.442e+03}$	$9.031e+03$
TP2	10	PSO	$1.587e+01$	$3.773e+00$
		ASY	$1.563e+01$	$3.977e+00$
		PF/LB/2	$8.306e+00$	$3.390e+00$
		LB/NL/2.0	$\mathbf{7.302e+00}$	$3.347e+00$
	50	PSO	$3.508e+02$	$2.098e+01$
		ASY	$3.330e+02$	$1.751e+01$
		PF/LB/2	$\mathbf{1.601e+02}$	$3.212e+01$
		LB/NL/2.0	$2.793e+02$	$4.174e+01$
	100	PSO	$9.289e+02$	$3.046e+01$
		ASY	$8.877e+02$	$3.119e+01$
		PF/LB/2	$\mathbf{4.273e+02}$	$6.944e+01$
		LB/NL/2.0	$8.392e+02$	$5.525e+01$
TP3	10	PSO	$8.536e-01$	$1.173e-01$
		ASY	$7.369e-01$	$1.598e-01$
		PF/LB/2	$2.375e-01$	$1.306e-01$
		LB/NL/2.0	$\mathbf{8.893e-02}$	$5.447e-02$
	50	PSO	$8.095e+01$	$9.016e+00$
		ASY	$6.425e+01$	$7.925e+00$
		PF/LB/2	$1.205e+00$	$7.270e-02$
		LB/NL/2.0	$\mathbf{1.034e-02}$	$1.817e-02$
	100	PSO	$4.331e+02$	$2.505e+01$
		ASY	$3.520e+02$	$2.312e+01$
		PF/LB/2	$3.280e+00$	$8.334e-01$
		LB/NL/2.0	$\mathbf{3.826e-01}$	$4.391e-01$

Table 10 continued

Problem	Dimension	Algorithm	Mean	SD
TP4	10	PSO	$2.059e+00$	$4.495e-01$
		ASY	$1.706e+00$	$5.198e-01$
		PF/LB/2	$3.543e-02$	$3.993e-02$
		LB/NL/2.0	$\mathbf{1.176e-02}$	$1.155e-01$
	50	PSO	$1.370e+01$	$4.042e-01$
		ASY	$1.284e+01$	$4.087e-01$
		PF/LB/2	$\mathbf{2.308e+00}$	$4.078e-01$
		LB/NL/2.0	$9.513e+00$	$7.807e-01$
	100	PSO	$1.730e+01$	$2.400e-01$
		ASY	$1.636e+01$	$2.488e-01$
		PF/LB/2	$\mathbf{3.761e+00}$	$4.558e-01$
		LB/NL/2.0	$1.416e+01$	$4.283e-01$

Table 11 Comparative results of PSO-NBA with PSO-based variants for test problems TP5-TP10 (nonlinear systems)

Problem	Dimension	Algorithm	Mean	SD
TP5	10	PSO	$6.921e-02$	$1.7539e-02$
		ASY	$6.214e-02$	$1.755e-02$
		PF/LB/2	$3.202e-03$	$2.261e-03$
		LB/NL/2.0	$\mathbf{4.833e-10}$	$2.108e-09$
TP6	6	PSO	$2.765e-02$	$2.482e-02$
		ASY	$2.081e-02$	$1.388e-02$
		PF/LB/2	$\mathbf{6.827e-03}$	$2.914e-02$
		LB/NL/2.0	$1.908e-01$	$3.177e-01$
TP7	5	PSO	$2.640e-01$	$1.273e-01$
		ASY	$2.192e-01$	$1.215e-01$
		PF/LB/2	$\mathbf{1.402e-01}$	$1.006e-01$
		LB/NL/2.0	$2.904e-01$	$2.285e-01$
TP8	8	PSO	$6.120e-01$	$2.050e-01$
		ASY	$5.396e-01$	$1.974e-01$
		PF/LB/2	$\mathbf{2.539e-01}$	$1.872e-01$
		LB/NL/2.0	$3.870e-01$	$4.366e-01$
TP9	10	PSO	$2.980e-01$	$1.565e-01$
		ASY	$2.391e-01$	$1.317e-01$
		PF/LB/2	$4.070e-02$	$3.635e-02$
		LB/NL/2.0	$\mathbf{1.648e-02}$	$1.933e-02$
TP10	20	PSO	$4.617e-03$	$4.544e-03$
		ASY	$3.377e-03$	$2.518e-03$
		PF/LB/2	$3.482e-04$	$1.243e-03$
		LB/NL/2.0	$\mathbf{1.576e-06}$	$6.868e-06$

other in 10 out of 15 problem instances of the standard test suite (Table 10), while the MOBA-based variant PF/L/2 is superior in the rest 5 cases. In Table 11, similar results for the nonlinear systems are reported, with the two variants being distinguished in 3 cases each. Finally, we can infer that the SOBA strategy dominates in the three high-dimensional problems (TP5, TP9, and TP10), while the MOBA one is better in the lower-dimensional cases.

Table 12 Computational budgets for GA, DE, MONS, PSO, and ASY [27], in test problems TP5-TP10

Problem	Comp. Budget
TP5	15×10^4
TP6	6×10^4
TP7	25×10^4
TP8	50×10^4
TP9	15×10^4
TP10	15×10^4

The four distinguished PSO-NBA approaches from Table 9 were used for further comparisons with different algorithms on TP5-TP10. For this purpose, we adopted the results for a steady-state Genetic Algorithm (GA), Differential Evolution (DE), and the multi-objective MONS approach that were reported in the recent study [27]. The experimental setting that was used in [27] assumed higher computational budgets than the one used in our study. Thus, we repeated the experiments on TP5-TP10 for the distinguished PSO-NBA approaches with the new computational budget, to obtain comparable results. The computational budgets adopted from [27] are reported in Table 12. For the PSO-NBA variants, the parameters were identical to the ones used in previous sections and reported in Table 3 without any further fine-tuning.

All results are reported in Table 13. The best performance among the other algorithms as well as among PSO-NBA variants is boldfaced. The reported experimental evidence offers some useful conclusions. First, we can see that the MOBA approaches of PSO-NBA outperformed the SOBA one with the exception of TP10. Secondly, the MOBA approaches performed better (by orders of magnitude) than MONS and GA in most of the problems.

Finally, we can see that PSO-NBA could outperform DE, which was the best algorithm among the rest, in half of the problems. We shall take into consideration that the results of PSO-NBA were received with the same parameters that were used in our default experimental setting without any further fine-tuning for the specific problems and computational budgets, while population sizes for the rest of the algorithms were fine-tuned per case.

4.4 Further experiments

We further assessed the PSO-NBA algorithm on a test suite of 19 problems that was proposed as a benchmark at the special issue on “Scalability of Evolutionary Algorithms and Other Metaheuristics for Large-Scale Optimization Problems” of the Soft Computing journal [14]. These problems will be henceforth denoted as SC-TP0–SC-TP18.

Table 13 Comparative results of PSO-NBA with different algorithms. The results of MONS, GA, DE are adopted from [27]

		TP5	TP6	TP7	TP8	TP9	TP10
MONS	Mean	$1.80e+00$	$1.00e-01$	$6.00e-01$	$1.10e+00$	$2.00e-01$	$2.00e-02$
	SD	–	–	–	–	–	–
GA	Mean	$1.01e-01$	$1.29e-02$	$9.57e-01$	$1.03e+00$	$4.53e-01$	$\mathbf{2.10e-06}$
	SD	$6.21e-02$	$2.29e-02$	$6.78e-01$	$5.50e-01$	$4.74e-01$	$1.00e-06$
DE	Mean	$\mathbf{1.44e-16}$	$\mathbf{1.29e-03}$	$\mathbf{1.01e-02}$	$\mathbf{1.29e-16}$	$\mathbf{5.20e-04}$	$6.37e-03$
	SD	$1.90e-18$	$2.47e-03$	$9.07e-04$	$5.87e-17$	$1.92e-04$	$3.74e-03$
LB/NL/2.0 (SOBA)	Mean	$1.44e-16$	$5.24e-02$	$1.46e-01$	$1.85e-01$	$2.57e-02$	$\mathbf{8.53e-10}$
	SD	$6.74e-19$	$1.66e-01$	$1.20e-01$	$1.97e-01$	$2.49e-01$	$6.84e-09$
LW/LB/NL/2.0 (MOBA)	Mean	$\mathbf{1.44e-16}$	$1.03e-05$	$\mathbf{3.31e-02}$	$\mathbf{2.09e-02}$	$4.33e-03$	$6.43e-07$
	SD	$0.00e+00$	$6.96e-05$	$1.69e-02$	$2.39e-02$	$5.13e-03$	$2.53e-06$
DW/LB/NL/2.0 (MOBA)	Mean	$1.44e-16$	$\mathbf{2.80e-06}$	$3.57e-02$	$3.40e-02$	$\mathbf{2.82e-03}$	$7.38e-09$
	SD	$5.20e-19$	$1.90e-05$	$1.94e-02$	$3.39e-02$	$4.25e-03$	$3.07e-08$
PF/LB/2 (MOBA)	Mean	$\mathbf{1.44e-16}$	$2.92e-06$	$3.79e-02$	$6.04e-02$	$3.48e-03$	$1.32e-05$
	SD	$0.00e+00$	$1.45e-05$	$2.21e-02$	$8.25e-02$	$5.77e-03$	$3.34e-05$

The problems SC-TP0–SC-TP5 belong to the CEC 2008 test suite, SC-TP6–SC-TP10 are shifted problems, and SC-TP11–SC-TP18 are hybrid composition functions. Their definitions as well as source codes can be obtained through online sources.[1] Comparative results for different algorithms are also publicly available.[2]

PSO-NBA was applied on the 50- and 100-dimensional instances of the test problems, adopting the experimental setting in [14]. In all cases, our algorithm assumed population size equal to $2 \times n$, where n stands for the problem's dimension. At each experiment, the solution error $|f_{\mathrm{PSO-NBA}} - f^*|$ was recorded, where f^* denotes the actual global minimum of the problem and $f_{\mathrm{PSO-NBA}}$ is the best solution value achieved by our approach.

The two best SOBA approaches and the two best MOBA approaches (in terms of number of wins) from Table 9 were considered for further experimentation. These approaches were also compared against six established algorithms, namely the CHC Genetic (Cross-generational elitist selection, Heterogeneous recombination and Cataclysmic mutation) algorithm [8], the G-CMA-ES (Restart Covariant Matrix Evolutionary Strategy) algorithm [2], the EvoPROpt (Evolutionary Path Relinking) algorithm [7], the SPSO2011 (Standard PSO 2011) algorithm [32], the ITHS (Intelligent Tuned Harmony Search) algorithm [16, 31] and the DBC (Directed Bee Colony) algorithm [13]. Note that G-CMA-ES was the dominant algorithm in the CEC 2005 challenge.

In Table 14, the obtained average errors for the two SOBA variants LB/NL/2.0 and LB/L/2.0 are reported. MOBA variants had slightly inferior performance, which was anticipated since in the previous experiments they were shown to perform better in lower dimensions. For this reason they are omitted from the current results. Also, in Table 14 we report the corresponding results for the rest of the algorithms. The results of the EvoPROpt, G-CMA-ES, and CHC algorithms are publicly available in the aforementioned online sources, while the results of the SPSO2011, ITHS, and DBC emerged from our implementations closely following the instructions, pseudocodes, and parameter settings provided in the original sources.

A first inspection of the results reveals that PSO-NBA is highly competitive to the other algorithms. More specifically, the linear PSO-NBA variant achieved zero or marginally deviant values in 12 problem instances, while the nonlinear variant had similar success in 8 out of 38 problem instances. The corresponding successes for EvoPROpt, SPSO2011, ITHS, DBC, G-CMA-ES, and CHC were 0, 2, 0, 0, 7, and 6, respectively, out of 38 problem instances.

In order to facilitate comparisons and provide further insight into the algorithm's effectiveness, we conducted pairwise comparisons of each algorithm with the rest. At each comparison, we recorded the number of *hits* (successes) over the accuracy levels,

$$10^{-4},\ 10^{-3},\ 10^{-2},\ 10^{-1},\ 10^{0},\ 10^{1},\ 10^{2},\ 10^{3},\ 10^{4}.$$

A hit is recorded for an algorithm when it outperforms another algorithm, i.e., it achieves a lower average error for the specific problem and dimension, and both their average errors are smaller than the particular accuracy level.

Figures 10 and 11 depict the distribution of the number of hits over the predefined accuracy levels for the linear and the nonlinear case, respectively. Evidently, PSO-NBA exhibits high numbers of hits for the majority of accuracy

[1] http://sci2s.ugr.es/eamhco/testfunctions-SOCO.

[2] http://sci2s.ugr.es/eamhco/SOCO-results.xls.

Table 14 Comparative results of PSO-NBA with different algorithms on test problems SC-TP0–SC-TP18

Dim.	Algorithm	SC-TP0	SC-TP1	SC-TP2	SC-TP3	SC-TP4	SC-TP5	SC-TP6	SC-TP7	SC-TP8	SC-TP9
50	EvoPROpt	1.22e − 02	3.71e − 01	1.12e + 02	4.96e − 02	5.13e − 02	6.85e − 03	2.63e − 02	2.08e − 02	8.02e + 00	4.80e − 02
	SPSO2011	0.00e + 00	4.43e + 01	6.36e + 01	1.53e + 02	5.32e − 03	2.18e + 00	1.18e + 01	5.05e − 03	1.40e + 02	2.30e + 01
	ITHS	2.76e + 01	9.69e + 00	1.07e + 04	1.17e + 01	1.15e + 00	1.42e + 00	1.08e + 00	1.36e + 03	4.82e + 01	4.97e + 00
	DBC	2.67e + 03	4.72e + 01	3.17e + 07	1.92e + 02	2.27e + 01	1.30e + 01	2.54e + 01	8.91e + 02	2.32e + 02	1.31e + 02
	CHC	0.00e + 00	6.19e + 01	1.25e + 06	7.43e + 01	1.67e − 03	0.00e + 00	0.00e + 00	2.24e + 02	3.10e + 02	7.30e + 00
	G-CMA-ES	0.00e + 00	0.00e + 00	7.97e − 01	1.05e + 02	2.96e − 04	2.09e + 01	0.00e + 00	0.00e + 00	1.66e + 01	6.81e + 00
	PSO-NBA (linear)	0.00e + 00	6.40e + 00	6.22e + 01	1.96e + 02	0.00e + 00	0.00e + 00	0.00e + 00	2.07e + 02	1.03e + 02	0.00e + 00
	PSO-NBA (nonlinear)	0.00e + 00	7.55e + 00	8.87e + 01	1.95e + 02	6.36e − 03	0.00e + 00	0.00e + 00	6.74e + 00	1.02e + 02	1.73e + 00
100	EvoPROpt	4.34e − 02	3.30e + 00	3.98e + 02	1.07e − 01	3.92e − 02	2.50e − 04	9.17e − 02	2.27e + 03	2.91e + 01	2.05e − 01
	SPSO2011	0.00e + 00	6.87e + 01	2.23e + 03	4.02e + 02	3.65e − 03	3.81e + 00	5.18e + 01	7.12e + 00	3.81e + 02	6.16e + 01
	ITHS	2.03e + 02	1.95e + 01	2.08e + 05	5.12e + 01	2.60e + 00	2.74e + 00	4.58e + 00	6.88e + 03	1.31e + 02	2.47e + 01
	DBC	1.35e + 04	6.52e + 01	3.15e + 08	4.35e + 02	1.14e + 02	1.47e + 01	6.54e + 01	4.25e + 03	4.97e + 02	4.06e + 02
	CHC	0.00e + 00	8.58e + 01	4.19e + 06	2.19e + 02	3.83e − 03	0.00e + 00	1.40e − 02	1.69e + 03	5.86e + 02	3.30e + 01
	G-CMA-ES	0.00e + 00	0.00e + 00	3.88e + 00	2.50e + 02	1.58e − 03	2.12e + 01	4.22e − 04	0.00e + 00	1.02e + 02	1.66e + 01
	PSO-NBA (linear)	0.00e + 00	2.39e + 01	3.09e + 02	5.10e + 02	0.00e + 00	7.86e − 01	0.00e + 00	9.67e + 03	4.37e + 02	8.90e + 00
	PSO-NBA (nonlinear)	0.00e + 00	2.60e + 01	3.77e + 02	5.44e + 02	0.00e + 00	8.93e − 01	0.00e + 00	1.71e + 03	4.61e + 02	8.00e + 00

Dim.	Algorithm	SC-TP10	SC-TP11	SC-TP12	SC-TP13	SC-TP14	SC-TP15	SC-TP16	SC-TP17	SC-TP18
50	EvoPROpt	9.68e + 00	2.27e + 00	4.22e + 01	9.97e − 01	6.38e − 02	5.63e + 00	6.77e + 01	1.62e + 00	5.03e − 02
	SPSO2011	1.34e + 02	7.84e + 01	1.20e + 02	1.07e + 02	1.64e + 01	1.62e + 02	3.45e + 02	6.37e + 01	1.89e + 01
	ITHS	4.66e + 01	2.12e + 01	1.98e + 03	8.70e + 00	1.20e + 00	9.09e + 00	1.19e + 02	1.93e + 00	1.91e + 00
	DBC	2.25e + 02	1.95e + 02	1.15e + 05	1.18e + 02	3.87e + 01	2.07e + 02	2.73e + 02	6.10e + 01	5.22e + 01
	CHC	2.16e + 00	9.57e − 01	2.08e + 06	6.17e + 01	3.98e − 01	0.00e + 00	2.26e + 04	1.58e + 01	3.59e + 02
	G-CMA-ES	3.01e + 01	1.88e + 02	1.97e + 02	1.09e + 02	9.79e − 04	4.27e + 02	6.89e + 02	1.31e + 02	4.76e + 00
	PSO-NBA(linear)	9.50e + 01	8.48e − 05	1.32e + 02	1.42e + 02	0.00e + 00	3.85e + 01	3.01e + 02	6.79e + 01	0.00e + 00
	PSO-NBA (nonlinear)	1.13e + 02	7.58e + 01	1.82e + 02	1.50e + 02	0.00e + 00	1.94e + 02	3.91e + 02	7.02e + 01	1.24e + 00
100	EvoPROpt	2.60e + 01	5.01e + 00	1.40e + 02	1.24e + 00	6.56e − 02	8.29e + 00	1.97e + 02	3.34e + 00	1.43e − 01
	SPSO2011	3.76e + 02	1.68e + 02	2.36e + 02	2.87e + 02	5.95e + 01	3.31e + 02	6.52e + 02	1.49e + 02	5.65e + 01
	ITHS	1.38e + 02	1.16e + 02	1.74e + 04	3.86e + 01	5.87e + 00	5.71e + 01	4.11e + 02	6.10e + 00	1.08e + 01
	DBC	5.07e + 02	7.82e + 02	2.95e + 06	2.83e + 02	1.08e + 02	4.61e + 02	6.01e + 02	1.32e + 02	1.42e + 02
	CHC	7.32e + 01	1.03e + 01	2.70e + 06	1.66e + 02	8.13e + 00	2.23e + 01	1.47e + 05	7.00e + 01	5.45e + 02
	G-CMA-ES	1.64e + 02	4.17e + 02	4.21e + 02	2.55e + 02	6.30e − 01	8.59e + 02	1.51e + 03	3.07e + 02	2.02e + 01
	PSO-NBA(linear)	4.46e + 02	7.88e + 01	3.53e + 02	3.59e + 02	0.00e + 00	2.96e + 02	6.60e + 02	1.73e + 02	0.00e + 00
	PSO-NBA (nonlinear)	4.65e + 02	1.81e + 02	3.89e + 02	3.99e + 02	0.00e + 00	3.96e + 02	8.21e + 02	1.73e + 02	2.58e + 00

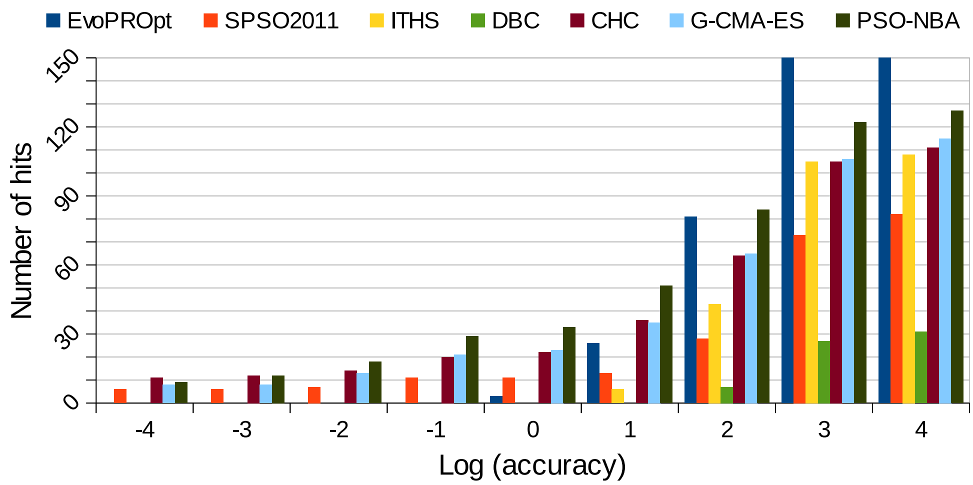

Fig. 10 Cumulative number of hits for different accuracy levels (linear case)

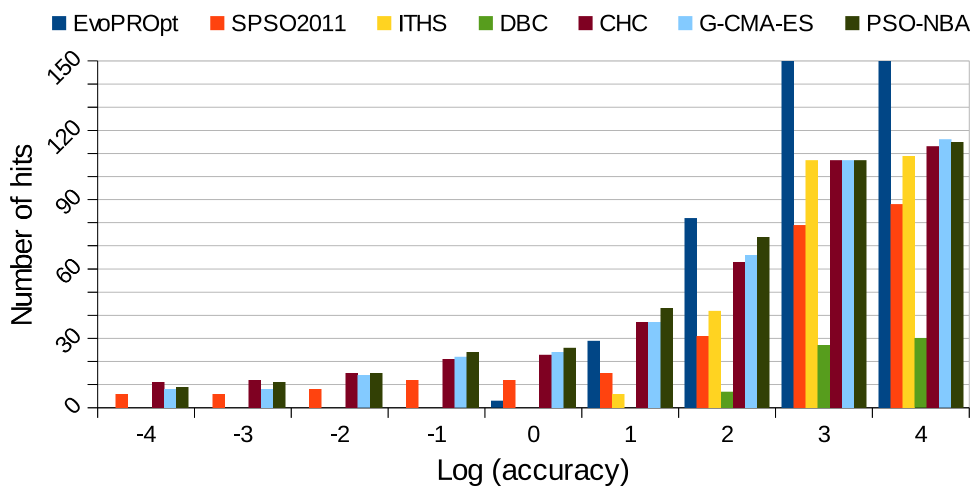

Fig. 11 Cumulative number of hits for different accuracy levels (nonlinear case)

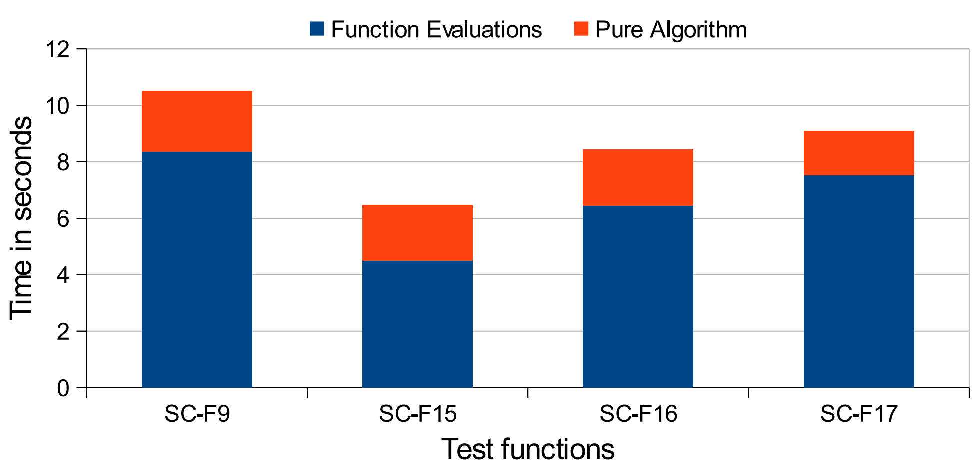

Fig. 12 Portion of time spent on algorithmic procedures vs function evaluations

levels, especially for the smallest ones, which are the most desirable in practice. EvoPROpt outperformed PSO-NBA only for the highest accuracy levels. Yet, it did not achieve any hit in almost half of the (smaller) accuracy levels.

The presented experimental evidence verifies that the proposed PSO-NBA approach can be very competitive also to other algorithms. Of course, the best choice among different PSO-NBA variants is always problem-dependent.

However, the observations that were pointed out in the previous sections can be helpful for the practitioner.

Finally, we considered the time complexity of the algorithm. Specifically, we investigated the fraction of the time spent to algorithmic procedures against the time spent purely for function evaluations per run. Figure 12 illustrates the required time for the 100-dimensional instances of 4 of the most demanding problems from the current test suite. The measured time is indicative for a single experiment. Despite the high dimensionality of the problems, the large population size, as well as the lack of any optimization in the source code of our implementation, we can clearly see that the function evaluation dominates the time required by the algorithm. This is an indication that smaller execution times can be achieved with further optimization of the algorithm's procedures and source code.

5 Conclusions

We introduced PSO-NBA, an asynchronous PSO variant that distributes the available computational budget of function evaluations in an irregular way among the particles of the swarm. In order to select the favored particles, the algorithm assesses their neighborhoods with respect to solution quality and diversity. Particles that possess highly ranked neighborhoods have higher probability of receiving function evaluations than the rest.

We studied two essential budget allocation strategies, namely a single- and a multi-objective one. For both strategies, a multitude of PSO-NBA variants were defined. All variants were tested on a standard suite of benchmark problems as well as on problems drawn from real-life applications. The most successful variants were distinguished after statistical analysis of the results. Further experiments were conducted on an established test suite. Comparisons with various algorithms were provided.

The acquired results suggested that PSO-NBA can be highly competitive. Overall, elitistic options were shown to be beneficial on performance. Both single- and multi-objective strategies exhibited efficiency and robustness.

There are many issues to be considered for future research. First, the experiments can be further extended in specific problem categories such as very large-scale problems, mixed integer optimization, and noisy environments (among others). Second, scalability and the impact of the problem's size on performance can be further investigated along with different neighborhood quality criteria. Finally, the budget allocation mechanisms of PSO-NBA can be used also in different metaheuristics or combined with other (PSO-based or not) algorithms to formulate hybrid schemes.

Acknowledgments The authors wish to thank the editor as well as the anonymous reviewers for their constructive comments and suggestions.

Appendix: Test problems

Standard test suite

The standard test suite consists of the following problems:

Test Problem 0 (TP0—Sphere) [19]. This is a separable n-dimensional problem, defined as

$$f(x) = \sum_{i=1}^{n} x_i^2, \tag{21}$$

and it has a single global minimizer, $x^* = (0, 0, \ldots, 0)^\top$, with $f(x^*) = 0$.

Test Problem 1 (TP1—Generalized Rosenbrock) [19]. This is a non-separable n-dimensional problem, defined as

$$f(x) = \sum_{i=1}^{n-1} \left(100\left(x_{i+1} - x_i^2\right)^2 + \left(x_{i-1}\right)^2\right), \tag{22}$$

and it has a global minimizer, $x^* = (1, 1, \ldots, 1)^\top$, with $f(x^*) = 0$.

Test Problem 2 (TP2—Rastrigin) [19]. This is a separable n-dimensional problem, defined as

$$f(x) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i)\right), \tag{23}$$

and it has a global minimizer, $x^* = (0, 0, \ldots, 0)^\top$, with $f(x^*) = 0$.

Test Problem 3 (TP3—Griewank) [19]. This is a non-separable n-dimensional problem, defined as

$$f(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \tag{24}$$

and it has a global minimizer, $x^* = (0, 0, \ldots, 0)^\top$, with $f(x^*) = 0$.

Test Problem 4 (TP4—Ackley) [19]. This is a non-separable n-dimensional problem, defined as

$$f(x) = 20 + \exp(1) - 20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right), \tag{25}$$

and it has a global minimizer, $x^* = (0, 0, \ldots, 0)^\top$, with $f(x^*) = 0$.

Nonlinear systems

This test set consists of six real-application problems, which are modeled as systems of nonlinear equations. Computing a solution of a nonlinear system is a very challenging task and it has received the ongoing attention of the scientific community. A common methodology for solving such systems is their transformation to an equivalent global optimization problem, which allows the use of a wide range of optimization tools. The transformation produces a single objective function by aggregating all the system's equations, such that the solutions of the original system are exactly the same with that of the derived optimization problem.

Consider the system of nonlinear equations: $\begin{cases} f_1(x) = 0, \\ f_2(x) = 0, \\ \vdots \\ f_m(x) = 0, \end{cases}$ with $x \in S \subset \mathbb{R}^n$. Then, the objective function,

$$f(x) = \sum_{i=1}^{m} |f_i(x)|, \tag{26}$$

defines an equivalent optimization problem. Obviously, if x^* with $f(x^*) = 0$ is a global minimizer of the objective function, then x^* is also a solution of the corresponding nonlinear system and vice versa.

In our experiments, we considered the following nonlinear systems, previously employed by Grosan and Abraham [9] to justify the usefulness of evolutionary approaches as efficient solvers of nonlinear systems:

`Test Problem` 5 (TP5—Interval Arithmetic Benchmark) [9]. This problem consists of the following system:

$$\begin{cases} x_1 - 0.25428722 - 0.18324757\; x_4x_3x_9 = 0, \\ x_2 - 0.37842197 - 0.16275449\; x_1x_{10}x_6 = 0, \\ x_3 - 0.27162577 - 0.16955071\; x_1x_2x_{10} = 0, \\ x_4 - 0.19807914 - 0.15585316\; x_7x_1x_6 = 0, \\ x_5 - 0.44166728 - 0.19950920\; x_7x_6x_3 = 0, \\ x_6 - 0.14654113 - 0.18922793\; x_8x_5x_{10} = 0, \\ x_7 - 0.42937161 - 0.21180486\; x_2x_5x_8 = 0, \\ x_8 - 0.07056438 - 0.17081208\; x_1x_7x_6 = 0, \\ x_9 - 0.34504906 - 0.19612740\; x_{10}x_6x_8 = 0, \\ x_{10} - 0.42651102 - 0.21466544\; x_4x_8x_1 = 0. \end{cases} \tag{27}$$

The resulting objective function defined by Eq. (26), is 10-dimensional with global minimum $f(x^*) = 0$.

`Test Problem` 6 (TP6—Neurophysiology Application) [9] This problem consists of the following system:

$$\begin{cases} x_1^2 + x_3^2 = 1, \\ x_2^2 + x_4^2 = 1, \\ x_5x_3^3 + x_6x_4^3 = c_1, \\ x_5x_1^3 + x_6x_2^3 = c_2, \\ x_5x_1x_3^2 + x_6x_4^2x_2 = c_3, \\ x_5x_1^2x_3 + x_6x_2^2x_4 = c_4, \end{cases} \tag{28}$$

where the constants, $c_i = 0$, $i = 1, 2, 3, 4$. The resulting objective function is 6-dimensional with global minimum $f(x^*) = 0$.

`Test Problem` 7 (TP7—Chemical Equilibrium Application) [9] This problem consists of the following system:

$$\begin{cases} x_1x_2 + x_1 - 3x_5 = 0, \\ 2x_1x_2 + x_1 + x_2x_3^2 + R_8x_2 - Rx_5 + 2R_{10}x_2^2 + R_7x_2x_3 + \\ R9x_2x_4 = 0, \\ 2x_2x_3^2 + 2R_5x_3^2 - 8x_5 + R_6x_3 + R_7x_2x_3 = 0, \\ R_9x_2x_4 + 2x_4^2 - 4Rx_5 = 0, \\ x_1(x_2+1) + R_{10}x_2^2 + x_2x_3^2 + R_8x_2 + R_5x_3^2 + x_4^2 - 1 \\ +R_6x_3 + R_7x_2x_3 + R_9x_2x_4 = 0, \end{cases} \tag{29}$$

where

$$R = 10, \quad R_5 = 0.193, \quad R_6 = \frac{0.002597}{\sqrt{40}}, \quad R_7 = \frac{0.003448}{\sqrt{40}},$$

$$R_8 = \frac{0.00001799}{40}, \quad R_9 = \frac{0.0002155}{\sqrt{40}}, \quad R_{10} = \frac{0.00003846}{40}.$$

The corresponding objective function is 5-dimensional with global minimum $f(x^*) = 0$.

`Test Problem` 8 (TP8—Kinematic Application) [9] This problem consists of the following system:

$$\begin{cases} x_i^2 + x_{i+1}^2 - 1 = 0, \\ a_{1i}x_1x_3 + a_{2i}x_1x_4 + a_{3i}x_2x_3 + a_{4i}x_2x_4 + a_{5i}x_2x_7 + \\ a_{6i}x_5x_8 + a_{7i}x_6x_7 + a_{8i}x_6x_8 + a_{9i}x_1 + a_{10i}x_2 + a_{11i}x_3 + \\ a_{12i}x_4 + a_{13i}x_5 + a_{14i}x_6 + a_{15i}x_7 + a_{16i}x_8 + a_{17i} = 0, \end{cases} \tag{30}$$

with a_{ki}, $1 \leqslant k \leqslant 17$, $1 \leqslant i \leqslant 4$, is the corresponding element of the k-th row and i-th column of the matrix:

$$A = \begin{bmatrix} -0.249150680 & 0.125016350 & -0.635550077 & 1.48947730 \\ 1.609135400 & -0.686607360 & -0.115719920 & 0.23062341 \\ 0.279423430 & -0.119228120 & -0.666404480 & 1.32810730 \\ 1.434801600 & -0.719940470 & 0.110362110 & -0.25864503 \\ 0.000000000 & -0.432419270 & 0.290702030 & 1.16517200 \\ 0.400263840 & 0.000000000 & 1.258776700 & -0.26908494 \\ -0.800527680 & 0.000000000 & -0.629388360 & 0.53816987 \\ 0.000000000 & -0.864838550 & 0.581404060 & 0.58258598 \\ 0.074052388 & -0.037157270 & 0.195946620 & -0.20816985 \\ -0.083050031 & 0.035436896 & -1.228034200 & 2.68683200 \\ -0.386159610 & 0.085383482 & 0.000000000 & -0.69910317 \\ -0.755266030 & 0.000000000 & -0.079034221 & 0.35744413 \\ 0.504201680 & -0.039251967 & 0.026387877 & 1.24991170 \\ -1.091628700 & 0.000000000 & -0.057131430 & 1.46773600 \\ 0.000000000 & -0.432419270 & -1.162808100 & 1.16517200 \\ 0.049207290 & 0.000000000 & 1.258776700 & 1.07633970 \\ 0.049207290 & 0.013873010 & 2.162575000 & -0.69686809 \end{bmatrix}$$

The corresponding objective function is 8-dimensional with global minimum $f(x^*) = 0$.

`Test Problem` 9 (TP9—Combustion Application) [9] This problem consists of the following system:

$$\begin{cases} x_2 + 2x_6 + x_9 + 2x_{10} = 10^{-5}, \\ x_3 + x_8 = 3 \times 10^{-5}, \\ x_1 + x_3 + 2x_5 + 2x_8 + x_9 + x_{10} = 5 \times 10^{-5}, \\ x_4 + 2x_7 = 10^{-5}, \\ 0.5140437 \times 10^{-7} x_5 = x_1^2, \\ 0.1006932 \times 10^{-6} x_6 = 2x_2^2, \\ 0.7816278 \times 10^{-15} x_7 = x_4^2, \\ 0.1496236 \times 10^{-6} x_8 = x_1 x_3, \\ 0.6194411 \times 10^{-7} x_9 = x_1 x_2, \\ 0.2089296 \times 10^{-14} x_{10} = x_1 x_2^2. \end{cases} \tag{31}$$

The corresponding objective function is 10-dimensional with global minimum $f(x^*) = 0$.

`Test Problem` 10 (TP10—Economics Modeling Application) [9] This problem consists of the following system:

$$\begin{cases} \left(x_k + \sum_{i=1}^{n-k-1} x_i x_{i+k} \right) x_n - c_k = 0, \\ \sum_{l=1}^{n-1} x_l + 1 = 0, \end{cases} \tag{32}$$

where $1 \leqslant k \leqslant n-1$, and $c_i = 0$, $i = 1, 2, \ldots, n$. The problem was considered in its 20-dimensional instance. Thus, the corresponding objective function was also 20-dimensional, with global minimum $f(x^*) = 0$.

References

1. Akbari R, Ziarati K (2011) A rank based particle swarm optimization algorithm with dynamic adaptation. J Comput Appl Math 235(8):2694–2714
2. Auger A, Hansen N (2005) A restart CMA evolution strategy with increasing population size. In Proc. IEEE Congress On Evolutionary Computation, pages 1769–1776, Edinburgh, UK
3. Bäck T, Fogel D, Michalewicz Z (1997) Handbook of evolutionary computation. IOP Publishing and Oxford University Press, New York
4. Bartz-Beielstein T, Blum D, Branke J (2007) Particle swarm optimization and sequential sampling in noisy environments. vol 39 of In Metaheuristics, Operations Research/Computer Science Interfaces Series, Springer, p 261–273
5. Clerc M, Kennedy J (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evol Comput 6(1):58–73
6. Coello Coello CA, Van Veldhuizen DA, Lamont GB (2002) Evolutionary algorithms for solving multi-objective problems. Kluwer, New York
7. Duarte A, Mart R, Gortazar F (2011) Path relinking for large-scale global optimization. Soft Comput 15(11):2257–2273
8. Eshelman J (1991) The chc adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. Foundations of Genetic Algorithms, p 265–283
9. Grosan C, Abraham A (2008) A new approach for solving nonlinear equations systems. IEEE Trans Syst Man Cybern Part A Syst Hum 38(3):698–714
10. Jin Y, Olhofer M, Sendhoff B (2001) Evolutionary dynamic weighted aggregation for multiobjective optimization: Why does it work and how? In: Proceedings of the GECCO 2001 Conference, San Francisco, CA, p 1042–1049
11. Kennedy J (1999) Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In: Proceedings of the 1999 congress on evolutionary computation, Washington, D.C., USA, IEEE Press, p 1931–1938
12. Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks, vol IV, Piscataway, NJ, IEEE Service Center p 1942–1948
13. Kumar R (2014) Directed bee colony optimization algorithm. Swarm Evol Comput 17:60–73
14. Lozano M, Molina D, Herrera F (2011) Editorial scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems. Soft Comput 15(11):2085–2087
15. Ma W, Wang M, Zhu X (2014) Improved particle swarm optimization based approach for bilevel programming problem-an application on supply chain model. Int J Mach Learn Cybern 5(2):281–292
16. Omran MGH, Mahdavi M (2008) Global-best harmony search. Appl Math Comput 198(2):643–656
17. Pan H, Wang L, Liu B (2006) Particle swarm optimization for function optimization in noisy environment. Appl Math Comput 181(2):908–919
18. Parsopoulos KE, Vrahatis MN (2002) Particle swarm optimization method in multiobjective problems. In: Proceedins of the ACM 2002 Symposium on Applied Computing (SAC 2002), p 603–607, Madrid, Spain. ACM Press
19. Parsopoulos KE, Vrahatis MN (2010) Particle swarm optimization and intelligence: advances and applications. Information Science Publishing (IGI Global)

20. Poli R (2007) An analysis of publications on particle swarm optimisation applications. Technical Report CSM-649, University of Essex, Department of Computer Science, UK
21. Rada-Vilela J, Zhang M, Johnston M (2013) Optimal computing budget allocation in particle swarm optimization. In Proc. 2013 Genetic and Evolutionary Computation Conference (GECCO'13), Amsterdam, Netherlands, p 81–88
22. Rana S, Jasola S, Kumar R (2013) A boundary restricted adaptive particle swarm optimization for data clustering. Int J Mach Learn Cybern 4(4):391–400
23. Souravlias D, Parsopoulos KE (2013) Particle swarm optimization with budget allocation through neighborhood ranking. In: Proceedings of the 2013 Genetic and Evolutionary Computation Conference (GECCO'13), p 105–112
24. Suganthan PN (1999) Particle swarm optimizer with neighborhood operator. In: Proceedings of the IEEE Congress on Evolutionary Computation, Washington, D.C., USA p 1958–1961
25. Tian N, Lai C-H (2014) Parallel quantum-behaved particle swarm optimization. Int J Mach Learn Cybern 5(2):309–318
26. Trelea IC (2003) The particle swarm optimization algorithm: convergence analysis and parameter selection. Inf Process Lett 85:317–325
27. Voglis C, Parsopoulos KE, Lagaris IE (2012) Particle swarm optimization with deliberate loss of information. Soft Comput 16(8):1373–1392
28. Wan L-Y, Li W (2008) An improved particle swarm optimization algorithm with rank-based selection. In: Proceedings of the IEEE international conference on machine learning and cybernetics, vol 7, pp 4090–4095
29. Wang X, He Y, Dong L, Zhao H (2011) Particle swarm optimization for determining fuzzy measures from data. Inf Sci 181(19):4230–4252
30. Whitley D, Lunacek M, Knight J (2004) Ruffled by ridges: How evolutionary algorithms can fail. In: Deb K et al. (ed.) Lecture Notes in Computer science (LNCS), vol 3103, p 294–306. Springer
31. Yadav P, Kumar R, Panda SK, Chang CS (2012) An intelligent tuned harmony search algorithm for optimisation. Inf Sci 196:47–72
32. Zambrano-Bigiarini M, Clerc M, Rojas R (2013) Standard particle swarm optimisation 2011 at cec-2013: a baseline for future pso improvements. In: Proceedings of the IEEE 2013 congress on evolutionary computation, Mexico, p 2337–2344
33. Zhang S, Chen P, Lee LH, Peng CE, Chen C-H (2011) Simulation optimization using the particle swarm optimization with optimal computing budget allocation. In: Proceedings of the 2011 winter simulation conference, p 4298–4309