Chapter II
Multi-Objective Particles Swarm Optimization Approaches

Konstantinos E. Parsopoulos
University of Patras, Greece

Michael N. Vrahatis
University of Patras, Greece

ABSTRACT

The multiple criteria nature of most real world problems has boosted research on multi-objective algorithms that can tackle such problems effectively, with the smallest possible computational burden. Particle Swarm Optimization has attracted the interest of researchers due to its simplicity, effectiveness and efficiency in solving numerous single-objective optimization problems. Up-to-date, there are a significant number of multi-objective Particle Swarm Optimization approaches and applications reported in the literature. This chapter aims at providing a review and discussion of the most established results on this field, as well as exposing the most active research topics that can give initiative for future research.

INTRODUCTION

Multi-objective optimization problems consist of several objectives that are necessary to be handled simultaneously. Such problems arise in many applications, where two or more, sometimes competing and/or incommensurable, objective functions have to be minimized concurrently. Due to the multicriteria nature of such problems, optimality of a solution has to be redefined, giving rise to the concept of Pareto optimality.

In contrast to the single-objective optimization case, multi-objective problems are characterized by trade-offs and, thus, there is a multitude of Pareto optimal solutions, which correspond to different settings of the investigated multi-objective problem. For example, in shape optimization, different Pareto optimal solutions correspond to different structure configurations of equal fitness but different properties. Thus, the necessity of finding the largest allowed number of such solutions, with adequate variety of their corresponding properties, is highly desirable.

Evolutionary algorithms seem to be particularly suited to multi-objective problems due to their ability to synchronously search for multiple Pareto optimal solutions and perform better global exploration of the search space (Coello, Van Veldhuizen, & Lamont, 2002; Deb, 1999; Schaffer, 1984). Up-to-date, a plethora of evolutionary algorithms have been proposed, implementing different concepts such as fitness sharing and niching (Fonseca & Fleming, 1993; Horn, Nafpliotis, & Goldberg, 1994; Srinivas & Deb, 1994), and elitism (Deb, Pratap, Agarwal, & Meyarivan, 2002; Erickson, Mayer, & Horn, 2001; Zitzler & Thiele, 1999). External archives have also been introduced as a means of memory for retaining Pareto optimal solutions. This addition enhanced significantly the performance of some algorithms, but it has also raised questions regarding the manipulation of the archive and its interaction with the actual population of search points.

Particle Swarm Optimization (PSO) is a swarm intelligence method that roughly models the social behavior of swarms (Kennedy & Eberhart, 2001). PSO shares many features with evolutionary algorithms that rendered its adaptation to the multi-objective context straightforward. Although several ideas can be adopted directly from evolutionary algorithms, the special characteristics that distinguish PSO from them, such as the directed mutation, population representation and operators must be taken into consideration in order to produce schemes that take full advantage of PSO's efficiency.

Up-to-date, several studies of PSO on multi-objective problems have appeared, and new, specialized variants of the method have been developed (Reyes-Sierra & Coello, 2006a). This chapter aims at providing a descriptive review of the state-of-the-art multi-objective PSO variants. Of course, it is not possible to include in the limited space of a book chapter the whole literature. For this reason, we selected to present the approaches that we considered most important and proper to sketch the most common features considered in the development of algorithms. Thus, we underline to the reader the fundamental issues in PSO-based multi-objective approaches, as well as the most active research directions and future trends. An additional reading section regarding applications and further developments is included at the end of the chapter, in order to provide a useful overview of this blossoming research field.

The rest of the chapter is organized as follows: Section 2 provides concise descriptions of the necessary background material, namely the basic multi-objective concepts and the PSO algorithm. Section 3 is devoted to the discussion of key concepts and issues that arise in the transition from single-objective to multi-objective cases. Section 4 exposes the established PSO approaches reported in the relative literature, and highlights their main features, while Section 5 discusses the most active research directions and future trends. The chapter concludes in Section 6.

BACKGROUND MATERIAL

Although the basic concepts of multi-objective optimization have been analyzed in another chapter of this book, we report the most essential for completeness purposes, along with a presentation of the PSO algorithm.

Basic Multi-Objective Optimization Concepts

Let $S \subset \mathbb{R}^n$ be an n-dimensional search space, and $f_i(x)$, $i=1,\ldots,k$, be k objective functions defined over S. Also, let $\mathbf{f}$ be a vector function defined as

$$\mathbf{f}(x) = [f_1(x), f_2(x), \ldots, f_k(x)], \qquad (1)$$

and

$$g_i(x) \leq 0, \qquad i = 1,\ldots, m, \qquad (2)$$

be *m* inequality constraints. Then, we are interested in finding a solution, $x^* = (x_1^*, x_2^*,\ldots, x_n^*)$, that minimizes $\mathbf{f}(x)$. The objective functions $f_i(x)$ may be conflicting with each other, thereby rendering the detection of a single global minimum at the same point in *S*, impossible. For this purpose, optimality of a solution in multi-objective problems needs to be redefined properly.

Let $u = (u_1,\ldots, u_k)$ and $v = (v_1,\ldots, v_k)$ be two vectors of the search space *S*. Then, *u dominates v*, if and only if, $u_i \leq v_i$ for all i=1, 2,…, *k*, and $u_i < v_i$ for at least one component. This property is known as *Pareto dominance*. A solution, *x*, of the multi-objective problem is said to be *Pareto optimal*, if and only if there is no other solution, *y*, in *S* such that $\mathbf{f}(y)$ dominates $\mathbf{f}(x)$. In this case, we also say that *x* is *nondominated* with respect to *S*. The set of all Pareto optimal solutions of a problem is called the *Pareto optimal set*, and it is usually denoted as $\mathcal{P}^*$. The set

$$\mathcal{PF}^* = \{ \mathbf{f}(x) : x \in \mathcal{P}^* \}, \qquad (3)$$

is called the *Pareto front*. A Pareto front is *convex* if and only if, for all $u, v \in \mathcal{PF}^*$ and for all $\lambda \in (0, 1)$, there exists a $w \in \mathcal{PF}^*$ such that

$$\lambda \|u\| + (1-\lambda) \|v\| \geq \|w\|,$$

while it is called *concave*, if and only if

$$\lambda \|u\| + (1-\lambda) \|v\| \leq \|w\|.$$

A Pareto front can also be partially convex and/or concave, as well as discontinuous. These cases are considered the most difficult for most multi-objective optimization algorithms.

The special nature of multi-objective problems makes necessary the determination of new goals for the optimization procedure, since the detection of a single solution, which is adequate in the single-objective case, is not valid in cases of many, possibly conflicting objective functions. Based on the definition of Pareto optimality, the detection of all Pareto optimal solutions is the main goal in multi-objective optimization problems. However, since the Pareto optimal set can be infinite and our computations adhere to time and space limitations, we are compelled to set more realistic goals. Thus, we can state as the main goal of the multi-objective optimization procedure, the *detection of the highest possible number of Pareto optimal solutions that correspond to an adequately spread Pareto front, with the smallest possible deviation from the true Pareto front*.

Particle Swarm Optimization

Eberhart and Kennedy (1995) developed PSO as an expansion of an animal social behavior simulation system that incorporated concepts such as nearest-neighbor velocity matching and acceleration by distance (Kennedy & Eberhart, 1995). Similarly to evolutionary algorithms, PSO exploits a population, called a *swarm*, of potential solutions, called *particles*, which are modified stochastically at each iteration of the algorithm. However, the manipulation of swarm differs significantly from that of evolutionary algorithms, promoting a cooperative rather than a competitive model.

More specifically, instead of using explicit mutation and selection operators in order to modify the population and favor the best performing individuals, PSO uses an adaptable velocity vector for each particle, which shifts its position at each iteration of the algorithm. The particles are moving towards promising regions of the search space by exploiting information springing from their own experience during the search, as well as the experience of other particles. For this purpose, a separate memory is used where each particle stores the best position it has ever visited in the search space.

Let us now put PSO more formally in the context of single-objective optimization. Let S be an n-dimensional search space, $f : S \to \mathbb{R}$ be the objective function, and N be the number of particles that comprise the swarm,

$$\mathbb{S} = \{x_1, x_2,\ldots, x_N\}.$$

Then, the i-th particle is a point in the search space,

$$x_i = (x_{i1,} x_{i2},\ldots, x_{in}) \in S,$$

as well as its best position,

$$p_i = (p_{i1,} p_{i2},\ldots, p_{in}) \in S,$$

which is the best position ever visited by x_i during the search. The velocity of x_i is also an n-dimensional vector,

$$v_i = (v_{i1,} v_{i2},\ldots, v_{in}).$$

The particles, their best positions, as well as their velocities, are randomly initialized in the search space.

Let $NG_i \subseteq \mathbb{S}$ be a set of particles that exchange information with x_i. This set is called the *neighborhood* of x_i, and it will be discussed later. Let also, g, be the index of the best particle in NG_i, that is,

$$f(p_g) \leq f(p_l), \qquad \text{for all } l \text{ with } x_l \in NG_i,$$

and t denote the iteration counter. Then, the swarm is manipulated according to the equations (Eberhart & Shi, 1998),

$$v_{ij}(t+1) = w\, v_{ij}(t) + c_1\, r_1\, (p_{ij}(t) - x_{ij}(t)) + c_2\, r_2\, (p_{gj}(t) - x_{ij}(t)), \quad (4)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \quad (5)$$

where $i = 1, 2,\ldots, N$; $j = 1, 2,\ldots, n$; w is a positive parameter called *inertia weight*; c_1 and c_2 are two positive constants called *cognitive* and *social* parameter, respectively; and r_1, r_2, are realizations of two independent random variables that assume the uniform distribution in the range (0, 1). The best position of each particle is updated at each iteration by setting

$$p_i(t+1) = x_i(t+1), \text{ if } f(x_i) < f(p_i),$$

otherwise it remains unchanged. Obviously, an update of the index g is also required at each iteration.

The inertia weight was not used in early PSO versions. However, experiments showed that the lack of mechanism for controlling the velocities could result in *swarm explosion*, that is, an unbounded increase in the magnitude of the velocities, which resulted in swarm divergence. For this purpose, a boundary, v_{max}, was imposed on the absolute value of the velocities, such that, if $v_{ij} > v_{max}$ then $v_{ij} = v_{max}$, and if $v_{ij} < -v_{max}$ then $v_{ij} = -v_{max}$. In later, more sophisticated versions, the new parameter was incorporated in the velocity update equation, in order to control the impact of the previous velocity on the current one, although the use of v_{max} was not abandoned.

Intelligent search algorithms, such as PSO, must demonstrate an ability to combine *exploration*, that is, visiting new regions of the search space, and *exploitation*, that is, performing more refined local search, in a balanced way in order to solve problems effectively (Parsopoulos & Vrahatis, 2004; Parsopoulos & Vrahatis, 2007). Since larger values of w promote exploration, while smaller values promote exploitation, it was proposed and experimentally verified that declining values of the inertia weight can provide better results than fixed values. Thus, an initial value of w around 1.0 and a gradually decline towards 0.0 are considered a good choice. On the other hand, the parameters c_1 and c_2 are usually set to fixed and equal values such that the particle is equally influenced by its own best position, p_i, as well as the best position of its neighborhood, p_g, unless the problem at hand implies the use of a different setting.

An alternative velocity update equation was proposed by Clerc and Kennedy (2002),

$$v_{ij}(t+1) = \chi \left[v_{ij}(t) + c_1 r_1 (p_{ij}(t) - x_{ij}(t)) + c_2 r_2 (p_{gj}(t) - x_{ij}(t))\right], \quad (6)$$

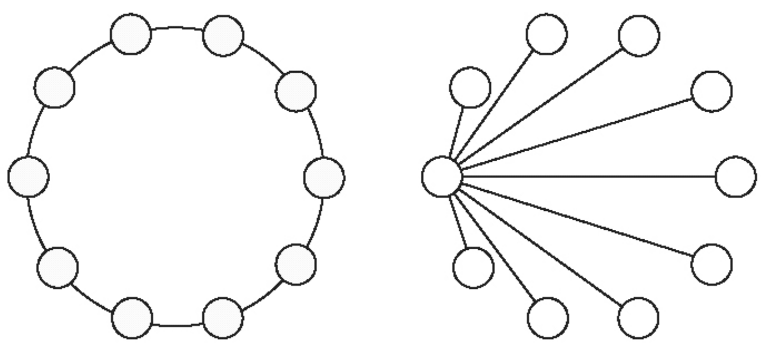
where χ is a parameter called *constriction factor*. This version is algebraically equivalent with the inertia weight version of Equation (4). However, the parameter selection in this case is based on the stability analysis due to Clerc and Kennedy (2002), which expresses χ as a function of c_1 and c_2. Different promising models were derived through the analysis of the algorithm, with the setting $\chi = 0.729$, $c_1 = c_2 = 2.05$, providing the most promising results and robust behavior, rendering it the default PSO parameter setting.

Regardless of the PSO version used, it is clear that its performance is heavily dependent on the information provided by the best positions, p_i and p_g, since they determine the region of the search space that will be visited by the particle. Therefore, their selection, especially for p_g, which is related to information exchange, plays a central role in the development of effective and efficient PSO variants.

Moreover, the concept of neighborhood mentioned earlier in this section, raises efficiency issues. A neighborhood has been already defined as a subset of the swarm. The most straightforward choice would be to consider as neighbors of the particle x_i, all particles enclosed in a sphere with center x_i and a user-defined radius in the search space. Despite its simplicity, this approach increases significantly the computational burden of the algorithm, since it requires the computation of all distances among particles at each iteration. This deficiency has been addressed by defining neighborhoods in the space of particles' indices instead of the actual search space.

Thus, the neighbors of x_i are determined based solely on the indices of the particles, assuming different *neighborhood topologies*, that is, orderings of the particles' indices. The most common neighborhood is the *ring topology*, depicted in Fig. 1 (left), where the particles are arranged on a ring, with x_{i-1} and x_{i+1} being the immediate neighbors of x_i, and x_1 following immediately after x_N. Based on this topology, a neighborhood of radius r of x_i is defined as

Figure 1. The ring (left) and star (right) neighborhood topologies of PSO



$$NG_i(r) = \{x_{i-r}, x_{i-r+1}, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_{i+r-1}, x_{i+r}\},$$

and the search is influenced by the particle's own best position, p_i, as well as the best position of its neighborhood. This topology promotes exploration, since the information carried by the best positions is communicated slowly through the neighbors of each particle. A different topology is the *star topology*, depicted in Figure 1 (right) where all particles communicate only with a single particle, which is the overall best position, p_g, of the swarm, that is, $NG_i \equiv \mathbb{S}$. This topology promotes exploitation, since all particles share the same information. This is also called the *global variant* of PSO, denoted as *gbest* in the relative literature, while all other topologies with $NG_i \subset \mathbb{S}$, define *local variants*, usually denoted as *lbest*. Different topologies have also been investigated with promising results (Janson & Middendorf, 2004; Kennedy, 1999).

KEY CONCEPTS OF MULTI-OBJECTIVE PSO ALGORITHMS

In the previous section, it was made clear that the search ability of a particle is heavily dependent on the best positions, p_i and p_g, involved in its velocity update equation (Equation (4) or (6)). These best positions attract the particle, biasing its movement towards the search space regions they lie, with p_i representing the inherent knowledge accumulated by the particle during its search, while p_g is the socially communicated information of its neighborhood, determined through the adopted neighborhood topology.

In multi-objective problems, we can distinguish two fundamental approaches for designing PSO algorithms (Reyes-Sierra & Coello, 2006a). The first approach consists of algorithms that consider each objective function separately. In these approaches, each particle is evaluated only for one objective function at a time, and the determination of the best positions is performed similarly to the single-objective optimization case. The main challenge in such cases is the proper manipulation of the information coming from each objective function in order to guide the particles towards Pareto optimal solutions.

The second approach consists of algorithms that evaluate all objective functions for each particle, and, based on the concept of Pareto optimality, they produce nondominated best positions (often called *leaders*) that are used to guide the particles. In these approaches, the determination of leaders is not straighforward, since there can be many nondominated solutions in the neighborhood of a particle, but only one is usually selected to participate in the velocity update.

In the aforementioned approaches, the problem of maintaining the detected Pareto optimal solutions must be addressed. The most trivial solution would be to store nondominated solutions as the particles' best positions. However, this choice is not always valid, since the desirable size of the Pareto front may exceed the swarm size. Moreover, two nondominated solutions are equally good, arising questions regarding the selection of the one that will be used as the best position of a particle. The size problem can be addressed by using an additional set, called the *external archive*, for storing the nondominated solutions discovered during search, while the problem of selection of the most proper archive member depends on the approach. Nevertheless, an external archive has also bounded size, thereby making unavoidable the imposition of rules regarding the replacement of existing solutions with new ones.

The general multi-objective PSO scheme can be described with the following pseudocode:

```
Begin
    Initialize swarm, velocities and best positions
    Initialize external archive (initially empty)
    While (stopping criterion not satisfied) Do
```

```
        For each particle
                Select a member of the external
                archive (if needed)
                Update velocity and position
                Evaluate new position
                Update best position and external
                archive
        End For
    End While
End
```

It is clear that selection of a member of the external archive, as well as the update of archive and best positions, constitute key concepts in the development of multi-objective PSO approaches, albeit not the only ones. Diversity also affects significantly the performance of the algorithm, since its loss can result in convergence of the swarm to a single solution.

The problem of selecting members from the external archive has been addressed through the determination of measures that assess the quality of each archive member, based on *density estimators*. Using such measures, archive members that promote diversity can be selected. The most commonly used density estimators are the *Nearest Neighbor Density Estimator* (Deb et al., 2002) and the *Kernel Density Estimator* (Deb & Goldberg, 1989). Both measures provide estimations regarding the proximity and number of neighbors for a given point.

The problem of updating the archive is more complex. A new solution is included in the archive if it is nondominated by all its members. If some members are dominated by the new solution, then they are usually deleted from the archive. The necessity for a bounded archive size originates from its tendency to increase significantly within a small number of algorithm iterations, rendering domination check computationally expensive.

Also, the user must decide for the action taken in the case of a candidate new solution that is nondominated by all members of a full archive. Obviously, this solution must compete all members of the archive in order to replace an existing member. Diversity is again the fundamental criterion, that is, the decision between an existing and a new solution is taken such that the archive retains the maximum possible diversity. For this purpose, different clustering techniques have been proposed (Knowles & Corne, 2000; Zitzler & Thiele, 1999). A similar approach uses the concept of ε-dominance to separate the Pareto front in boxes and retain one solution for each box. This approach has been shown to be more efficient than simple clustering techniques (Mostaghim & Teich, 2003b).

The update of each particle's own best position, is more straightforward. Thus, in approaches based on distinct evaluation of each objective function, it is performed as in standard PSO for single-objective optimization. On the other hand, in Pareto-based approaches, the best position of a particle is replaced only by a new one that dominates it. If the candidate and the existing best position are nondominated, then the old one is usually replaced in order to promote swarm diversity. At this point, we must also mention the effect of the employed neighborhood topology and PSO variant, on the performance of the algorithm. However, there are no extensive investigations to support the superiority of specific variants and topologies in multi-objective cases.

ESTABLISHED MULTI-OBJECTIVE PSO APPROACHES

In this section we review the state-of-the-art literature on multi-objective PSO algorithms. We will distinguish two fundamental categories of algorithms, based on the two approaches mentioned in the previous section, namely, approaches that exploit each objective function separately, and Pareto-based schemes. The distinction is made mainly for presentation purposes, and it is not strict, since there are algorithms that combine features from both approaches. The exposition of the methods for each category is based on a chronological ordering.

Algorithms that Exploit Each Objective Function Separately

This category consists of approaches that either combine all objective functions to a single one or consider each objective function in turn for the evaluation of the swarm, in an attempt to exploit the efficiency of PSO in solving single-objective problems. This approach has the advantage of straightforward update of the swarm and best positions, with an external archive usually employed for storing nondominated solutions. The main drawback of these methods is the lack of *a priori* information regarding the most proper manipulation of the distinct objective values, in order to converge to the actual Pareto front (Jin, Olhofer, & Sendhoff, 2001).

Objective Function Aggregation Approaches

These approaches aggregate, through a weighted combination, all objective functions in a single one,

$$F(x) = \sum_{i=1}^{k} w_i f_i(x),$$

where w_i are nonnegative weights such that

$$\sum_{i=1}^{k} w_i = 1,$$

and the optimization is performed on $F(x)$, similarly to the single-objective case. If the weights remain fixed during the run of the algorithm, we have the case of *Conventional Weighted Aggregation* (CWA). This approach is characterized by simplicity but it has also some crucial disadvantages. For only a single solution can be attained through the application of PSO for a specific weight setting, the algorithm must be applied repeatedly with different weight settings in order to detect a desirable number of nondominated solutions. Moreover, the CWA approach is unable to detect solutions in concave regions of the Pareto front (Jin et al., 2001).

The aforementioned limitations of CWA were addressed by using dynamically adjusted weights during optimization. Such approaches are the *Bang-Bang Weighted Aggregation* (BWA), which is defined for the case of bi-objective problems as (Jin et al., 2001),

$$w_1(t) = \text{sign}(\sin(2\pi t/a)), \quad w_2(t) = 1 - w_1(t),$$

as well as the *Dynamic Weighted Aggregation* (DWA), which is defined as,

$$w_1(t) = |\sin(2\pi t/a)|, \qquad w_2(t) = 1 - w_1(t),$$

where a is a user-defined adaptation frequency and t is the iteration number. The use of the sign function in BWA results in abrupt changes of the weights that force the algorithm to keep moving towards the Pareto front. The same effect is achieved with DWA, although the change in the weights is milder than BWA. Experiments with Genetic Algorithms have shown that DWA approaches perform better than BWA in convex Pareto fronts, while their performance is almost identical in concave Pareto fronts.

Parsopoulos and Vrahatis (2002a, 2002b) proposed the first multi-objective PSO weighted aggregation approach. They considered bi-objective problems with CWA, BWA and DWA approaches. Preliminary results on widely used benchmark problems were promising, and graphical representations showed that the two schemes could provide Pareto fronts with satisfactory spreading. As expected, the dynamically modified scheme outperformed the fixed weights approach. Although the simplicity and straightforward applicability render weighted aggregation schemes very attractive in combination with PSO, their efficiency on problems with more than two objectives has not been investigated extensively.

Baumgartner, Magele and Renhart (2004) considered a similar approach, where subswarms that use a different weight setting each, are used in combination with a gradient-based scheme for the detection of Pareto optimal solutions. More specifically, the swarm is divided in subswarms and each one uses a specific weight setting. The best particle of each subswarm serves as a leader only for itself. Also, a *preliminary pareto decision* is made in order to further investigate points that are candidate Pareto optimal solutions. This decision is made for each particle, x, based on the relation

$$\frac{1}{k}\left|\sum_{j=1}^{k}\operatorname{sgn}\Big(f_j\left(x(t+1)\right)-f_j\left(x(t)\right)\Big)\right|\neq 1,$$

where t stands for the iteration counter. If it holds, then x could be a Pareto optimal point, and the gradients of the objective functions $f_1,\ldots,f_k$, are computed on a perturbed point $x+\Delta x$. If none objective function improves at the perturbed point, then it is considered as a Pareto optimal point and it is removed from the swarm. Although results on a limited set of test problems are promising, the algorithm has not been fully evaluated and compared with other PSO approaches.

Mahfouf, Chen and Linkens (2004) proposed a dynamically modified weights approach. However, in this approach, the standard PSO scheme with linearly decreasing inertia weight was modified, by incorporating a mutation operator in order to alleviate swarm stagnation, as well as an acceleration term that accelerates convergence at later stages of the algorithm. More specifically, Equation (4) was modified to

$$v_{ij}(t+1) = w\, v_{ij}(t) + a\,\big(r_1\,(p_{ij}(t) - x_{ij}(t)) + r_2\,(p_{gj}(t) - x_{ij}(t))\big),$$

where a is an acceleration factor that depends on the current iteration number,

$$a = a_0 + t\,/\,\text{MIT},$$

where MIT is the maximum number of iterations and a_0 lies within the range [0.5, 1]. After the computation of the new positions of the particles, both new and old positions are entered in a list. The *Non-Dominated Sorting* technique (Li, 2003) is applied for this list, and the nondominated particles (that approximate the Pareto front) are selected. These particles suffer a mutation procedure in an attempt to further improve them. The resulting set of particles constitutes the swarm in the next iteration of the algorithm. Results from the application of this scheme on a problem from steel industry are reported with promising results. The algorithm combines characteristics of different approaches that have been shown to enhance the performance of multi-objective methods. Its competitive performance to both PSO and other evolutionary approaches, such as NSGA-II and SPEA2, can be attributed to the mutation operator that preserves swarm diversity, as well as to the Nondominated Sorting technique that allows the direct exploitation and evolution of points approximating the Pareto front, instead of using an external archive.

Objective Function Ordering Approaches

These approaches require the determination of a ranking of the objective functions. Then, minimization is performed for each function independently, starting from the most important one. Hu and Eberhart (2002) proposed a scheme based on such an ordering. Since Pareto front constitutes a boundary of the fitness values space, the algorithm retains the simplest objective function fixed and minimizes the rest of the objective functions. In their scheme, a local PSO variant with dynamic neighborhoods was used, with neighborhoods been defined rather in the fitness values space than the standard index-based scheme described in Section 2.2. Nondominated solutions are stored as particles' best positions and no external archive is used. The approach was applied successfully on problems with two objective functions but the function ordering procedure, which can be crucial for its performance especially in problems with more than two objectives, lacks justification.

An extension of the previous approach was proposed one year later (Hu, Eberhart, & Shi, 2003), incorporating an external archive in the form of external memory for storing nondominated solutions and reducing the computational cost. Albeit the reported preliminary results on problems with two objective functions were promising, further investigation is needed to reveal the algorithm's potential under more demanding situations, as well as its sensitivity to the parameter setting. Also, the authors mentioned the limited applicability of this approach, which was unable to address the binary string problem.

Non-Pareto, Vector Evaluated Approaches

Parsopoulos and Vrahatis (2002a; 2002b) proposed the Vector Evaluated PSO (VEPSO) scheme. This is a multiswarm approach based on the idea of Vector Evaluated Genetic Algorithm (VEGA) (Schaffer, 1985). In VEPSO, there is one swarm devoted to each objective function, and evaluated only for this objective function. However, in the swarm update (the algorithm employed the global variant of PSO), the best positions of one swarm are used for the velocity update of another swarm that corresponds to a different objective function.

Thus, if the problem consists of k objectives, then k swarms are used. If $v_i^{[s]}$ denotes the velocity of the i-th particle in the s-th swarm, s=1,…, k, then it is updated based on the relation

$$v_{ij}^{[s]}(t+1) = w\, v_{ij}^{[s]}(t) + c_1 r_1 (p_{ij}^{[s]}(t) - x_{ij}^{[s]}(t)) + c_2 r_2 (p_{gj}^{[q]}(t) - x_{ij}^{[s]}(t)),$$

where $p_g^{[q]}$ is the best position of the q-th swarm (which is evaluated with the q-th objective function). In this way, the information regarding the promising regions for one objective is inoculated to a swarm that already possesses information for a different objective. Experimental results imply that the algorithm is capable of moving towards the Pareto front, always in combination with the external archive approach of Jin et al. (2001).

A parallel implementation of VEPSO has also been investigated (Parsopoulos, Tasoulis & Vrahatis, 2004) with promising results. In this implementation, each swarm is assigned to a processor and the number of swarms is not necessarily equal to the number of objective functions. The communication among swarms is performed through an island migration scheme, similar to the ring topology used in PSO's ring neighborhood topology. VEPSO has been successfully used in two real-life problems, namely, the optimization of a radiometer array antenna (Gies & Rahmat-Samii, 2004), as well as for determining generator contributions to transmission systems (Vlachogiannis & Lee, 2005).

An approach similar to VEPSO, was proposed by Chow and Tsui (2004). The algorithm, called Multi-Species PSO, was introduced within a generalized autonomous agent response-learning framework, related to robotics. It uses subswarms that form species, one for each objective function. Each subswarm is then evaluated only with its own objective function, and information of best particles is communicated to neighboring subswarms with the form of an extra term in the velocity update of the particles. Thus, the velocity of the i-th particle in the s-th swarm is updated as follows:

$$v_{ij}^{[s]}(t+1) = v_{ij}^{[s]}(t) + a_1 (p_{ij}^{[s]}(t) - x_{ij}^{[s]}(t)) + a_2 (p_{gj}^{[s]}(t) - x_{ij}^{[s]}(t))) + A,$$

where

$$A = \sum_{l=1}^{H_s} \left(p_{gj}^{[l]}(t) - x_{ij}^{[s]}(t) \right),$$

with H_s being the number of swarms that communicate with the s-th swarm, and $p_g^{[l]}$ the best position of the l-th swarm, $l = 1,\ldots, H_s$. The algorithm was shown to be competitive to other established multi-objective PSO approaches, although in limited number of experiments. Also, questions arise regarding the velocity update that does not include any constriction factor or inertia weight, as well as on the scheme for defining neighboring swarms, since the schemes employed in the investigated problems are not analyzed.

Algorithms Based on Pareto Dominance

These approaches use the concept of Pareto dominance to determine the best positions (leaders) that will guide the swarm during search. As we mentioned in Section 3, several questions arise regarding the underlying schemes and rules for the selection of these positions among equally good solutions. For the imposition of additional criteria that take into consideration further issues (such as swarm diversity, Pareto front spread, etc.) is inevitable, the development of Pareto-based PSO approaches became a blossoming research area, with a significant number of different approaches reported in the literature. In the following paragraphs we review the most significant developments.

Coello and Salazar Lechuga (2002) proposed the Multi-objective PSO (MOPSO), one of the first Pareto-based PSO approaches (Coello, Toscano Pulido, & Salazar Lechuga, 2004). In MOPSO, the nondominated solutions detected by the particles are stored in a repository. Also, the search space is divided in hypercubes. Each hypercube is assigned a fitness value that is inversely proportional to the number of particles it contains. Then, the classical roulette wheel selection is used to select a hypercube and a leader from it. Thus, the velocity update for the i-th particle becomes

$$v_{ij}(t+1) = w\, v_{ij}(t) + c_1\, r_1\, (p_{ij}(t) - x_{ij}(t)) + c_2\, r_2\, (R_h(t) - x_{ij}(t)),$$

where p_i is its best position and R_h is the selected leader from the repository. The best position p_i is updated at each iteration, based on the domination relation between the existing best position of the particle and its new position.

Also, the repository has limited size and, if it is full, new solutions are inserted based on the *retention* criterion, that is, giving priority to solutions located in less crowded areas of the objective space. MOPSO was competitive against NSGA-II and PAES on typical benchmark problems, under common performance metrics, and it is currently considered one of the most typical multi-objective PSO approaches. A sensitivity analysis on the parameters of the algorithm, including the number of hypercubes used, can provide further useful information on this simple though efficient approach.

Fieldsend and Singh (2002) proposed a multi-objective PSO scheme that addresses the inefficiencies caused by the truncation of limited archives of nondominated solutions. For this purpose, a complex tree-like structure for unconstrained archiving maintenance, called the *dominated tree*, is used (Fieldsend, Everson, & Singh, 2003). The algorithm works similarly to MOPSO, except the repository, which is maintained through the aforementioned structures. An additional feature that works beneficially is the use of mutation, called *craziness*, on the particle velocity, in order to preserve diversity. The algorithm has shown to be competitive with PAES, although the authors underline the general deficiency of such approaches in cases where closeness in the objective space is loosely related to closeness in the parameter space.

Ray and Liew (2002) proposed an approach that employs the nearest neighbor density estimator in combination with a roulette wheel scheme for the selection of leaders. More specifically, leaders are generated through a multilevel sieve procedure that ranks individuals. Initially, all nondominated particles are assigned a rank of 1 and removed from swarm. The nondominated solutions from the remaining particles are assigned a rank of 2, and the procedure continues until all particles have been assigned a rank. If at most half of the swarm has been assigned a rank of 1, then all particles with rank smaller than the average rank are assigned to the set of leaders. Otherwise, only particles with a rank of 1 are assigned to the set of leaders. For the rest of the particles, a leader is selected and used for updating their position.

The selection of leader is based on the computation of the crowding radius for each leader and a roulette wheel selection mechanism that uses these values. Leaders with higher crowding radius have higher selection probability and, therefore, promote the uniform spread of solutions on the Pareto front. Special care is taken in constrained problems, where ranking takes into consideration both the objective and constraint values. The algorithm was tested on benchmark as well as engineering design problems and results were represented graphically. However, neither numerical results nor comparisons with any other multi-objective algorithm are reported to convince the reader regarding its efficiency.

Bartz-Beielstein, Limbourg, Mehnen, Schmitt, Parsopoulos, & Vrahatis (2003) proposed DOPS, a method based on an elitist archiving scheme. Their analysis considered different schemes for updating the archive and selecting the most proper solutions for the particles' update, using functions that assess the performance and contribution of each particle to the Pareto front spreading. More specifically, two functions, F_{sel} and F_{del}, are used to assign a *selection* and a *deletion* fitness value to each particle, respectively. The selection fitness value is a measure of the particle's influence to the spreading of the Pareto front, and increases with its distance to its nearest neighbors. Thus, every time a personal or a globally best position is needed, a member is chosen from the archive based on a roulette wheel selection over F_{sel}. If the number of available nondominated solutions surpasses the archive size, then a member of the archive is selected for deletion based on F_{del}. Different selection and deletion functions are proposed and evaluated. The method was supported by sensitivity analysis on its parameters, providing useful hints on the effect of archiving on the performance of multi-objective PSO.

Srinivasan and Seow (2003) introduced the Particle Swarm inspired Evolutionary Algorithm (PS-EA). This algorithm can only roughly be characterized as a PSO-based approach, since the update of particles is completely different than any PSO algorithm. More specifically, the particle update equations (Equations (4) and (5)) are substituted by a *probability inheritance tree*. Thus, instead of moving in the search space with an adaptable velocity, the particle rather inherits the parameters of its new position. Therefore, it can inherit parameters from an *elite particle*, that is, its own or the overall best position, or inherit parameters from a randomly selected neighboring particle. Further choices are pure mutation and the retainment of the existing parameter in the new position.

All these choices are made probabilistically, based on a *dynamic inheritance probability adjuster* (DIPA). This mechanism controls the probabilities based on feedback from the convergence status of the algorithm, and more specifically on the fitness of the overall best particle. If the overall best seems to stagnate or does not change positions frequently, DIPA adjusts the probabilities. Unfortunately, the authors do not provide details regarding the exact operation of DIPA even in their experiments. Thus, although the algorithm is shown to be competitive with a GA approach, there are no indications regarding the complexity of setting the DIPA mechanism properly in order to achieve acceptable performance.

Mostaghim and Teich proposed several algorithms based on MOPSO, incorporating special schemes for the selection of archive members that participate in the update of the particles' velocity. In Mostaghim and Teich (2003a), a MOPSO approach is proposed in combination with the *sigma method* that assigns a numerical value to each particle and member of the archive. For example, in a bi-objective problem, if the i-th particle has objective values (f_1, f_2), then it is assigned a sigma value,

$$\sigma = \frac{(K_2 f_1)^2 - (K_1 f_2)^2}{(K_2 f_1)^2 + (K_1 f_2)^2},$$

where K_1, K_2, are the maximum objective values of the particles for f_1 and f_2, respectively. Then, a particle uses as leader the archive member with the closest sigma value to its own. Also, a turbulence (mutation) factor is used for the position update of the particle, to maintain swarm diversity. The algorithm outperformed SPEA2 in typical bi-objective problems but the opposite happened for problems with three objectives. Also, the authors underline the necessity for large swarm sizes, since an adequate number of distributed solutions are required in the objective space.

Mostaghim and Teich (2003b) studied further the performance of MOPSO using also the concept of ε-dominance, and compared it to clustering-based approaches, with promising results. The investigation in this work focused mostly on the archiving methodology rather than the search algorithm itself, indicating the superiority of the ε-dominance approach with respect to the quality of the obtained Pareto fronts through MOPSO, in terms of convergence speed and diversity.

Furthermore, an algorithm for covering the Pareto front by using subswarms and an unbounded external archive, after the detection of an initial approximation of the Pareto front through MOPSO, was proposed by Mostaghim and Teich (2004). In this approach, an initial approximation of the Pareto front is detected through MOPSO, and subswarms are initialized around each nondominated solution in order to search the neighborhood around it. The algorithm outperformed an evolutionary approach (Hybrid MOEA) that incorporates a space subdivision scheme, on an antenna design problem. The applicability of the proposed MOPSO scheme on problems of any dimension and number of objectives, constitutes an additional advantage of the algorithm, as claimed by the authors.

Li (2004) proposed an approach called the MaximinPSO that exploits the maximin fitness function (Balling, 2003). For a given decision vector x, this fitness function is defined as,

$$\max_{j=1,\dots,N;\, x\neq y}\ \min_{i=1,\dots,k} \{ f_i(x) - f_i(y) \},$$

where k is the number of objective functions and N is the swarm size. Obviously, only decision vectors with a maximin function value less than zero can be nondominated solutions with respect to the current population. The maximin function promotes diversity of the swarm, since it penalizes particles that cluster in groups. Also, it has been argued that it favors the middle solutions in convex fronts and the extreme solutions in concave fronts (Balling, 2003). However, Li (2004) has shown that this effect can be addressed through the use of adequately large swarms. The particles in the proposed algorithm are evaluated with the maximin function and nondominated solutions are stored in an archive to serve as leaders (randomly selected by the particles). MaximinPSO outperformed NSGA-II on typical benchmark problems. However, experiments were restricted in bi-objective unconstrained problems, thus, no sound conclusions can be derived regarding its efficiency in more demanding cases.

Toscano Pulido and Coello (2004) proposed Another MOPSO (AMOPSO), an approach similar to VEPSO, where subswarms are used to probe different regions of the search space. Each subswarm has its own group of leaders. These groups are formed from a large set of nondominated solutions through a clustering technique. Then, each subswarm is assigned a group of leaders and select randomly those that will serve as its guides towards the Pareto front. This approach can alleviate problems related to disconnected search spaces, where a particle may be assigned a leader that lies in a disconnected region, wasting a lot of search effort. However, at some points, information exchange is allowed among subswarms. The authors show that AMOPSO is competitive to NSGA-II, and could be considered as a viable alternative.

AMOPSO does not use an external archive (nondominated solutions are stored as best positions of the particles), in contrast to OMOPSO, the approach of Reyes-Sierra and Coello (2005), which employs two external archives. This approach uses the nearest neighbor estimator and stores the selected best positions for the current iteration of PSO in the one archive and the overall nondominated solutions (final solutions) in the other archive. Established concepts such as turbulence (mutation) and ε-dominance are also used for diversity and archive maintenance purposes, respectively, increasing the complexity of the algorithm significantly, when compared to AMOPSO. The special feature of the algorithm is the incorporation of a mechanism for removing leaders, when their number exceeds a threshold. The aforementioned features result in an increased efficiency and effectiveness of the OMOPSO, which is shown to outperform previously presented MOPSO approaches as well as NSGA-II and SPEA2, rendering it a highly efficient method.

A different idea has been introduced in (Villalobos-Arias et al., 2005), where stripes are used on the search space and they are assigned particles that can exploit a unique leader that corresponds to a specific stripe. The core of this work is the stripes-based technique and its ability to maintain diversity in the employed optimizer. Its combination with MOPSO exhibits promising results, although it is not clear if this is independent of the search algorithm or the specific technique is beneficial specifically for MOPSO (the authors mention it as a future work direction).

Ho, Yang, Ni, Lo & Wong (2005) proposed a multi-objective PSO-based algorithm for design optimization. However, they introduced a plethora of unjustified modifications to the PSO algorithm regarding its parameter configuration and velocity update. Similarly to AMOPSO, the resulting scheme uses several external archives, one for the overall solutions and one for each particle, where it stores the most recent Pareto optimal solutions it has discovered. For the velocity update of the particle x_i, its best position, p_i, is selected from the latter archive, while p_g is selected from the overall archive, through a roulette wheel selection procedure. Aging of the leaders in the repositories is also introduced, as a means of biasing the selection scheme towards these leaders that have not been selected frequently. The algorithm is tested only on two problems and no comparisons with other methods are provided (the authors just mention its superiority against a simulated annealing approach), thus, no clear conclusions can be derived regarding its usefulness.

Raquel and Naval (2005) proposed MOPSO-CD, an approach that incorporates a *crowding distance* mechanism for the selection of the global best particle, as well as for the deletion of nondominated solutions from the external archive. Mutation is also employed to maintain diversity of the nondominated solutions in the external archive. Crowding distance is computed for each nondominated solution separately. If $f_1,\ldots,f_k$, are the objective functions and R is the external archive, then for the computation of the crowding distance of p in R, with respect to f_j, $j=1,\ldots,k$, we sort all points in R with respect to their f_j objective value and take

$$CD\_f_j = f_j(q) - f_j(r),$$

where q is the point of R that follows immediately after p in the sorting with respect to the f_j objective values, and r is the point that precedes p in the same ordering. Thus, the total crowding distance of p is given by

$$\sum_{j=1}^{k} CD\_f_j .$$

A proportion of the nondominated points of R with the highest crowding distances serve as leaders of the swarm (selected randomly). Also, mutation applied on the particles at randomly selected iterations promotes swarm diversity. Typical constraint-handling techniques adopted from the NSGA-II algorithm (Deb et al., 2002) are incorporated for addressing constrained problems. MOPSO-CD was compared to MOPSO, with results implying its viability as an alternative.

Alvarez-Benitez, Everson & Fieldsend (2005) proposed the *Rounds*, *Random*, and *Prob* techniques, which are based solely on the concept of Pareto dominance, for selecting leaders from the external archive. Each technique promotes different features in the algorithm. Rounds promotes as global guide of a particle x_i the nondominated solution that dominates the fewest particles of the swarm, including x_i. This solution is then excluded from selection for the rest of the particles. The procedure can be computationally expensive for large archives, however it is shown that promotes diversity. On the other hand, Random uses as global guide of a particle x_i a probabilistically selected nondominated solution that dominates x_i, with each nondominated solution having the same probability of selection. Prob constitutes an extension of Random that favors the archive members that dominate the smallest number of points. Mutation is also employed, while constraint-handling techniques are proposed and discussed, deriving the conclusion that careful handling of exploration near the boundaries of the search space can be beneficial for all multi-objective optimization approaches. However, this concept needs further experimentation to be confirmed.

As described earlier, MOPSO has an implicit fitness sharing mechanism for the selection of hypercubes. Salazar Lechuga and Rowe (2005) introduced MOPSO-*fs*, a MOPSO variant with explicit fitness sharing. According to this approach, each particle p_i in the repository of nondominated solutions, is assigned a fitness

$$F\_sh_i = 10 \Big/ \sum_{j=1}^{n} s_i^j ,$$

where

$$s_i^j = \begin{cases} 1 - \left(\dfrac{d_i^j}{\sigma_{\text{share}}} \right)^2, & \text{if } d_i^j < \sigma_{\text{share}}, \\ 0, & \text{otherwise}, \end{cases}$$

with σ_{share} being a user-defined distance and d_i^j be a distance measure between nondominated solutions p_i and p_j. This fitness-sharing scheme assigns higher fitness values to solutions with small number of other solutions around them. Then, the leaders of the swarm are selected through a roulette wheel selection technique that uses the assigned fitness values. MOPSO-*fs* has shown to be competitive with MOPSO, as well as with NSGA-II and PAES, although the analysis of choosing the fitness sharing parameters is under further investigation.

Mostaghim and Teich (2006) proposed recently a new idea, similar to that of Ho *et al.* (2005) described above. More specifically, each particle retains all nondominated solutions it has encountered in a personal archive. Naturally, a question arises regarding the final selection of the leader from the personal archive. The authors propose different techniques, ranging from pure random selection to the use of weights and diversity-preserving techniques. Experiments with the sigma-MOPSO (Mostaghim & Teich, 2003a) provided promising results on typical benchmark problems.

Huo, Shen Zhu (2006) proposed an interesting idea for the selection of leaders. More specifically, they evaluated each particle according to each objective function separately, and then, they assumed the mean of the best particles per function, as the global best position for the swarm update. Diversity of the swarm is preserved through a distance measure that biases the leader selection towards nondominated solutions that promote the alleviation of particle gathering in clusters. The resulting SMOPSO algorithm was tested on a limited number of test problems, and no comparisons were provided with other methods, to fully evaluate its efficiency.

Reyes-Sierra and Coello (2006b) conducted an interesting investigation on a hot research topic of both single- and multi-objective optimization, namely the on-line parameter adaptation of multi-objective algorithms. More specifically, the inertia weight, w, acceleration coefficients, c_1 and c_2, and selection method (dominance or crowding values) probability, P_s, of the MOPSO approach described earlier, were investigated using Analysis of Variance (ANOVA). The analysis has shown that large values of P_s, w, and c_2 provide better results, while c_1 seems to have a mild effect on MOPSO's performance. After identifying the most crucial parameters, different adaptation techniques, based on a reward system, were proposed. Thus, the parameter level selection could be *proportional*, *greedy*, or based on the *soft max strategy* that employs Gibbs distribution. The results are very promising, opening the way towards more efficient self-adaptive multi-objective approaches.

FUTURE RESEARCH DIRECTIONS

The non-Pareto algorithms describe in Section 4.1, share some characteristics that have concentrated the interest of the research community. With simplicity and straightforward applicability being their main advantage, while increased computational cost being their common drawback in some cases, these algorithms can be considered as significant alternatives that provide satisfactory solutions without complex implementation requirements.

However, there are still fundamental questions unanswered. More specifically, for the weighted aggregation approaches, the most efficient schedule for changing weights remains an open question. In most cases, the problem is addressed on a problem-dependent base, since there are no extensive investigations that can imply specific choices based on possible special characteristics of the problem at hand.

The same holds for the function ordering approaches. If the problem at hand implies a specific significance ordering for the objective functions, then these algorithms can be proved valuable. On the other hand, if there are no such indications, it is very difficult to make proper orderings and hold the overall computational cost at an acceptable level.

The non-Pareto vector evaluated approaches are the most popular in this category of algorithms, due to their straightforward applicability and use of the fundamental element of swarm intelligence, that is, the exchange of information among swarms. Still, there are features of these algorithms that need further investigation, such as the frequency and direction of information exchange among swarms. The size and number of swarms used, as well as the incorporation of external archives, constitute further interesting research issues.

It has been made obvious that the category of Pareto-based approaches is significantly wider than that of non-Pareto approaches. This can be partially attributed to the direct attack to the multi-objective problem through algorithms that incorporate in their criteria the key-property of

Pareto dominance. In this manner, many non-dominated solutions are considered in a single run of the algorithm, and stored as the resulting approximation of the Pareto front.

Naturally, there are crucial issues that need to be addressed prior to the design of efficient Pareto-based algorithms. In PSO Pareto-based approaches, we can distinguish three fundamental issues:

1. Selection of leaders,
2. Promotion of diversity, and
3. Archive maintenance.

The first two issues depend only on the swarm dynamics, while the latter can be considered as a more general issue that arises in all multi-objective algorithms that use archives. However, since the specific workings of an algorithm can mutually interact with the archiving procedures, it is possible that some archiving schemes fit better the multi-objective PSO approaches, resulting in more efficient algorithms.

Unfortunately, although there is a plethora of approaches for tackling the aforementioned issues, most of them are based on recombinations of established ideas from the field of evolutionary multi-objective algorithms. Also, the vast majority of experiments is conducted on a narrow set of test problems of small dimensionality, and perhaps this is the most proper point for underlining the necessity for extensive investigations of the algorithms, since this is the only way to reveal their advantages and deficiencies. The assumption of widely acceptable performance metrics from the multi-objective optimization community would also help towards this direction. It is not rare for two algorithms to compete completely different under two different metrics, but only favorable metrics are reported in most papers, hindering the user from detecting and interfering to the weak aspects of the algorithms.

Parallel implementations of multi-objective PSO approaches constitute also an active research area. Although PSO fits perfectly the framework for parallel implementations that can save significant computational effort in demanding problems, the development of such schemes as well as the interaction of the algorithms' modules (multiple swarms, archives, etc.) under this framework has not been studied extensively.

Furthermore, self-adaptation is considered a very challenging topic in almost all application areas where evolutionary algorithms are involved. The development of self-adaptive PSO schemes that can tackle multiple objectives will disengage the user from the necessity of providing proper parameter values, and it will render the algorithm applicable to any environment and problem, since it will be able to adapt its dynamic in order to fit the problem at hand.

The aforementioned topics can be extended to the field of dynamic multi-objective optimization, where the problem changes over time, along with its constraints. The dynamic case is far harder than the static one, since the algorithm shall be able to both approximate the Pareto front and track it through time. The literature in this field is limited and the development of PSO-based approaches for such problems is an open (although not very active yet) research area.

Finally, as time passes, the necessity for novel ideas that can tackle the aforementioned issues, while retaining the highest possible simplicity and efficiency for the algorithms, becomes more vivid. It is the authors' opinion that, besides the aforementioned topics, special emphasis should be given to it in future research.

CONCLUSIONS

This chapter provided a descriptive review of the state-of-the-art multi-objective PSO variants. Issues related to the operation of PSO in multi-objective environments have been pointed out and a plethora of approaches with various characteristics have been exposed. Naturally, the collection of algorithms described in the previous sections is far from complete, since the number of research works published on the field has been significantly increased in the late years. However, we provided the most significant results from our perspective, in order to sketch the up-to-date state of research in multi-objective PSO algorithms.

Since multi-objective optimization is intimately related to real-life applications, efficiency must be the key issue in the development of new multi-objective PSO approaches. The plethora of established approaches provides a wide variety of ideas and combinations of existing techniques for better manipulation of the algorithms, but only a minority of the existing methods have shown their potential in real-life problems. Thus, further work is needed to verify the nice properties of existing approaches in practice. Also, theoretical analyses that will provide further information on multi-objective PSO dynamics are expected to encourage the development of less complex and easily parametrized algorithms. Nevertheless, the development of multi-objective PSO approaches is currently and will remain a very active and exciting research field.

REFERENCES

Alvarez-Benitez, J. E., Everson, R. M., & Fieldsend, J. E. (2005). A MOPSO algorithm based exclusively on Pareto dominance concepts. *Lecture notes in computer science* (Vol. 3410, pp. 459-473). Springer-Verlag.

Balling, R. (2003). The maximin fitness function; Multi-objective city and regional planning. *Lecture notes in computer science* (Vol. 2632, pp. 1-15). Springer-Verlag.

Bartz-Beielstein, T., Limbourg, P., Mehnen, J., Schmitt, K., Parsopoulos, K. E., & Vrahatis, M. N. (2003). Particle swarm optimizers for Pareto optimization with enhanced archiving techniques. In *Proceedings of the IEEE 2003 Congress on Evolutionary Computation* (pp. 1780-1787). IEEE Press.

Baumgartner, U., Magele, C., & Renhart, W. (2004). Pareto optimality and particle swarm optimization. *IEEE Transactions on Magnetics, 40*(2), 1172-1175.

Chow, C.-K. & Tsui, H.-T. (2004). Autonomous agent response learning by a multi-species particle swarm optimization. In *Proceedings of the 2004 IEEE Congress on Evolutionary Computation* (pp. 778-785). IEEE Service Center.

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.

Coello, C. A. & Salazar Lechuga, M. (2002). MOPSO: A proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 IEEE Congress of Evolutionary Compututation* (pp. 1051-1056). IEEE Service Center.

Coello, C. A., Toscano Pulido, G., & Salazar Lechuga, M. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Trans. Evol. Comput., 8*(3), 256-279.

Coello, C. A., Van Veldhuizen, D. A., & Lamont, G. B. (2002). *Evolutionary algorithms for solving multi-objective problems*. New York: Kluwer.

Deb, K. (1999). Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation, 7*(3), 205-230.

Deb, K., & Goldberg, D. E. (1989). An investigation of niche and species formation in genetic function optimization. In *Proceedings of the 3rd International Conference on Genetic Algorithms* (pp. 42-50). Morgan Kaufmann Publishing.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput., 6*(2), 182-197.

Eberhart, R. C. & Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Proceedings of the Sixth Symposium on Micro Machine and Human Science* (pp. 39-43). Piscataway, NJ: IEEE Service Center.

Eberhart, R. C. & Shi, Y. (1998). Comparison between genetic algorithms and particle swarm optimization. In V. W. Porto et al. (Eds.), *Evolutionary programming: Vol. VII* (pp. 611-616). Springer.

Erickson, M., Mayer, A., & Horn, J. (2001). The niched Pareto genetic algorithm 2 applied to the design of groundwater remediation systems. *Lecture notes in computer science* (Vol. 1993, pp. 681-695). Springer-Verlag.

Fieldsend, J. E., Everson, R. M., & Singh, S. (2003). Using unconstrained elite archives for multi-objective optimization. *IEEE Trans. Evol. Comp., 7*(3), 305-323.

Fieldsend, J. E. & Singh, S. (2002). A multi-objective algorithm based upon particle swarm optimisation, An efficient data structure and turbulence. In *Proceedings of the 2002 UK Workshop on Computational Intelligence* (pp. 34-44). Birmingham, UK.

Fonseca, C. M. & Fleming, P. J. (1993). Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization. In *Proceedings of the 5th International Conference on Genetic Algorithms* (pp. 416-423).

Gies, D. & Rahmat-Samii, Y. (2004). Vector evaluated particle swarm optimization (VEPSO): Optimization of a radiometer array antenna. In *Proceedings of the AP-S IEEE International Symposium (Digest) of Antennas and Propagation Society, Vol. 3* (pp. 2297-2300).

Ho, S. L., Yang, S., Ni, G., Lo, E. W. C., & Wong, H. C. (2005). A particle swarm optimization-based method for multi-objective design optimizations. *IEEE Trans. Magnetics, 41*(5), 1756-1759.

Horn, J., Nafpliotis, N., & Goldberg, D. E. (1994). A niched Pareto genetic algorithm for multi-objective optimization. In *Proceedings of the 1st IEEE Conference on Evolutionary Computation* (pp. 82-87).

Hu, X. & Eberhart, R. (2002). Multi-objective optimization using dynamic neighborhood particle swarm optimization. In *Proceedings of the 2002 IEEE Congress Evolutionary Compututation* (pp. 1677-1681). IEEE Service Center.

Hu, X., Eberhart, R. C., & Shi, Y. (2003). Particle swarm with extended memory for multi-objective optimization. In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium* (pp. 193-197). IEEE Service Center.

Huo, X. H., Shen, L. C., Zhu, H. Y. (2006). A smart particle swarm optimization algorithm for multi-objective problems. *Lecture notes in computer science* (Vol. 4115, pp. 72-80). Springer-Verlag.

Janson, S. & Middendorf, M. (2004). A hierarchical particle swarm optimizer for dynamic optimization problems. *Lecture notes in computer science* (Vol. 3005, pp. 513-524). Springer-Verlag.

Jin, Y., Olhofer, M., & Sendhoff, B. (2001). Evolutionary dynamic weighted aggregation for multi-objective optimization: Why does it work and how? In *Proceedings of the GECCO 2001 Conference* (pp. 1042-1049), San Francisco, CA.

Kennedy, J. (1999). Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In *Proceedings of the IEEE Congress Evolutionary Computation* (pp. 1931-1938). IEEE Press.

Kennedy, J. & Eberhart, R. C. (1995). Particle swarm optimization. In *Proceedings of the IEEE International Conference Neural Networks, Vol. IV* (pp. 1942-1948). Piscataway, NJ: IEEE Service Center.

Kennedy, J. & Eberhart, R. C. (2001). *Swarm intelligence*. Morgan Kaufmann Publishers.

Knowles, J D. & Corne, D. W. (2000). Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary computation, 8*(2), 149-172.

Li, X. (2003). A non-dominated sorting particle swarm optimizer for multi-objective optimization. *Lecture notes in computer science, Vol. 2723* (pp. 37-48). Springer-Verlag.

Li, X. (2004). Better spread and convergence: Particle swarm multi-objective optimization using the maximin fitness function. *Lecture notes in computer science, Vol. 3102* (pp. 117-128). Springer-Verlag.

Mahfouf, M., Chen, M.-Y., & Linkens, D. A. (2004). Adaptive weighted particle swarm optimisation for multi-objective optimal design of alloy steels. *Lecture notes in computer science* (Vol. 3242, pp. 762-771). Springer.

Mostaghim, S. & Teich, J. (2003a). Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium* (pp. 26-33). IEEE Service Center.

Mostaghim, S. & Teich, J. (2003b). The role of ε-dominance in multi objective particle swarm optimization methods. In *Proceedings of the IEEE 2003 Congress on Evolutionary Computation* (pp. 1764-1771). IEEE Press.

Mostaghim, S. & Teich, J. (2004). Covering Pareto-optimal fronts by subswarms in multi-objective particle swarm optimization. In *Proceedings of the IEEE 2004 Congress on Evolutionary Computation* (pp. 1404-1411). IEEE Press.

Mostaghim, S. & Teich, J. (2006). About selecting the personal best in multi-objective particle swarm optimization. *Lecture notes in computer science* (Vol. 4193, pp. 523-532). Springer.

Parsopoulos, K. E., Tasoulis, D. K., & Vrahatis, M. N. (2004). Multi-objective optimization using parallel vector evaluated particle swarm optimization. In *Proceedings of the IASTED 2004 International Conference on Artificial Intelligence and Applications* (pp. 823-828). IASTED/ACTA Press.

Parsopoulos, K. E. & Vrahatis, M. N. (2002a). Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing, 1*(2-3), 235-306.

Parsopoulos, K. E. & Vrahatis, M. N. (2002b). Particle swarm optimization method in multiobjective problems. In *Proceedings of the ACM 2002 Symposium on Applied Computing* (pp. 603-607). ACM Press.

Parsopoulos, K. E. & Vrahatis, M. N. (2004). On the computation of all global minimizers through particle swarm optimization. *IEEE Transactions on Evolutionary Computation, 8*(3), 211-224.

Parsopoulos, K. E. & Vrahatis, M. N. (2007). Parameter selection and adaptation in unified particle swarm optimization. *Mathematical and Computer Modelling, 46*(1-2), 198-213.

Raquel, C. R. & Naval, P. C., Jr. (2005). An effecive use of crowding distance in multi-objective particle swarm optimization. In *Proceedings of the GECCO 2005* (pp. 257-264). ACM Press.

Ray, T. & Liew, K. M. (2002). A swarm metaphor for multi-objective design optimization. *Engineering Optimization, 34*(2), 141-153.

Reyes-Sierra, M. & Coello, C. A. (2005). Improving PSO-based multi-objective optimisation using crowding, mutation and ε-dominance. *Lecture notes in computer science* (Vol. 3410, pp. 505-519). Springer-Verlag.

Reyes-Sierra, M. & Coello, C. A. (2006a). Multiobjective particle swarm optimizers: A survey of the state-of-the-art. *International Journal* *of Computational Intelligence Research, 2*(3), 287-308.

Reyes-Sierra, M. & Coello, C. A. (2006b). Online adaptation in multi-objective particle swarm optimization. In *Proceedings of the 2006 IEEE Swarm Intelligence Symposium* (pp. 61-68). IEEE Press.

Salazar Lechuga, M. & Rowe, J. E. (2005). Particle swarm optimization and fitness sharing to solve multi-objective optimization problems. In *Proceedings of the 2005 IEEE Congress on Evolutionary Computation* (pp. 1204-1211). IEEE Service Center.

Schaffer, J. D. (1984). *Multiple objective optimization with vector evaluated genetic algorithms*. Unpublished doctoral thesis, Vanderbilt University, Nashville, TN.

Schaffer, J. D. (1985). Multiple objective optimisation with vector evaluated genetic algorithm. In *Proceedings of the 1st International Conference on Genetic Algorithms* (pp. 93-100). Morgan Kaufmann Publishers.

Srinivas, N. & Deb, K. (1994). Multi-objective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation, 2*(3), 221-248.

Srinivasan, D. & Seow, T. H. (2003). Particle swarm inspired evolutionary algorithm (PS-EA) for multi-objective optimization problem. In *Proceedings of the IEEE 2003 Congress on Evolutionary Computation* (pp. 2292-2297). IEEE Press.

Toscano Pulido, G., & Coello, C. A. (2004). Using clustering techniques to improve the performance of a particle swarm optimizer. *Lecture notes in computer science* (Vol. 3102, pp. 225-237). Springer.

Villalobos-Aria, M. A., Toscano Pulido, G., & Coello, C. A. (2005). A proposal to use stripes to maintain diversity in a multi-objective particle swarm optimizer. In *Proceedings of the 2005 IEEE Swarm Intelligence Symposium* (pp. 22-29). IEEE Service Center.

Vlachogiannis, J. G. & Lee, K. Y. (2005). Determining generator contributions to transmission system using parallel vector evaluated particle swarm optimization. *IEEE Transactions on Power Systems, 20*(4), 1765-1774.

Zitzler, E. & Thiele, L. (1999). Multi-objective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput., 3*(4), 257-271.

ADDITIONAL READING

The reader is strongly encouraged to visit the *Evolutionary Multi-objective Optimization Repository*, which is maintained by Carlos A. Coello Coello at the Web address:

http://delta.cs.cinvestav.mx/~ccoello/EMOO/

This excellent and up-to-date source of literature and software provides links and online copies of a plethora of papers published in the field. We selected some papers for the interested reader, with an emphasis in further developments and applications that were not discussed in the chapter due to space limitations:

Book Chapters

Hernandez Luna, E. & Coello Coello, C. A. (2004). Using a particle swarm optimizer with a multi-objective selection scheme to design combinational logic circuits. In Carlos A. Coello Coello & Gary B. Lamont (Eds.), *Applications of multi-objective evolutionary algorithms* (pp. 101-124). Singapore: World Scientific.

Reyes-Sierra, M. & Coello Coello, C. A. (2007). A study of techniques to improve the efficiency of a multi-objective particle swarm optimiser. In Shengxiang Yang, Yew Soon Ong, & Yaochu Jin (Eds.), *Evolutionary computation in dynamic and uncertain environments* (pp. 269-296). Springer.

Santana-Quintero, L. V., Ramirez-Santiago, N., Coello Coello, C. A., Molina Luque, J., & Hernandez-Diaz, A. G. (2006). A new proposal for multi-objective optimization using particle swarm optimization and rough sets theory. *Lecture notes in computer science* (Vol. 4193, pp. 483-492). Springer.

Santana-Quintero, L. V., Ramirez-Santiago, N., & Coello Coello, C. A. (2006). A multi-objective particle swarm optimizer hybridized with scatter search. *Lecture notes in artificial intelligence* (Vol. 4293, pp. 294-304). Springer.

Journal Papers

Gill, M. K., Kaheil, Y. H., Khalil, A., Mckee, M., & Bastidas, L. (2006). Multi-objective particle swarm optimization for parameter estimation in hydrology. *Water Resources Research, 42*(7), 2006.

Goudos, S. K. & Sahalos, J. N. (2006). Microwave absorber optimal design using Multi-objective particle swarm optimization. *Microwave and Optical Technology Letters, 48*(8), 1553-1558.

Huang, V. L., Suganthan, P. N., & Liang, J. J. (2006). Comprehensive learning particle swarm optimizer for solving multi-objective optimization problems. *International Journal of Intelligent Systems, 21*(2), 209-226.

Rahimi-Vahed, A. R. & Mirghorbani, S. M. (2007). A multi-objective particle swarm for a flow shop scheduling problem. *Journal of Combinatorial Optimization, 13*(1), 79-102.

Zhao, B. & Cao, Y.-J. (2005). Multiple objective particle swarm optimization technique for economic load dispatch. *Journal of Zhejiang University SCIENCE, 6A*(5), 420-427.

Conference Papers

Baltar, A. M. & Fontane, D. G. (2006, March). A generalized multi-objective particle swarm optimization solver for spreadsheet models: Application to water quality. In *Proceedings of Hydrology Days 2006*, Fort Collins, Colorado.

Brits, R., Engelbrecht, A. P., & van den Bergh, F. (2002). A niching particle swarm optimizer. In *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL'02)* (pp. 692-696).

Goldbarg, E. F. G., de Souza, G. R., & Goldbarg, M. C. (2006). Particle swarm optimization for the bi-objective degree-constrained minimum spanning tree. In *Proceedings of the 2006 IEEE Congress on Evolutionary Computation (CEC'2006)* (pp. 1527-1534). IEEE.

Gong, D. W., Zhang, Y., & Zhang, J. H. (2005). Multi-objective particle swarm optimization based on minimal particle angle. *Lecture notes in computer science* (Vol. 3644, pp. 571-580). Springer-Verlag.

Halter, W. & Mostaghim, S. (2006). Bilevel optimization of multi-component chemical systems using particle swarm optimization. In *Proceedings of the 2006 IEEE Congress on Evolutionary Computation (CEC'2006)* (pp. 4383-4390). IEEE.

Hendtlass, T. (2005). WoSP: A multi-optima particle swarm algorithm. In *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC'2005)* (pp. 727-734). IEEE Service Center.

Ma, M., Zhang, L. B., Ma, J., & Zhou, C. G. (2006). Fuzzy neural network optimization by a particle swarm optimization algorithm. *Lecture notes in computer science* (Vol. 3971, pp. 752-761). Springer.

Meng, H. Y., Zhang, X. H., & Liu, S. Y. (2005). Intelligent multi-objective particle swarm optimization based on AER model. *Lecture notes in artificial intelligence* (Vol. 3808, pp. 178-189). Springer.

Salazar-Lechugam, M. & Rowe, J. E. (2006). Particle swarm optimization and auto-fitness sharing to solve multi-objective optimization problems. In *Proceedings of the 2006 Swarm Intelligence Symposium (SIS'06)* (pp. 90-97). IEEE Press.

Wang, L. & Singh, C. (2006). Multi-objective stochastic power dispatch through a modified particle swarm optimization algorithm. In *Proceedings of the 2006 Swarm Intelligence Symposium (SIS'06)* (pp. 128-135). IEEE Press.

Yapicioglu, H., Dozier, G., & Smith, A. E. (2004). Bi-criteria model for locating a semi-desirable facility on a plane using particle swarm optimization. In *Proceedings of the 2004 Congress on Evolutionary Computation (CEC'2004)* (pp. 2328-2334). IEEE Service Center.

Munoz Zavala, A. E., Villa Diharce, E. R., & Hernandez Aguirre, A. (2005). Particle evolutionary swarm for design reliability optimization. *Lecture notes in computer science* (Vol. 3410, pp. 856-869). Springer.

Theses

Reyes-Sierra, M. (2006). *Use of coevolution and fitness inheritance for multi-objective particle swarm optimization.* Unpublished doctoral Thesis, Computer Science Section, Department of Electrical Engineering, CINVESTAV-IPN, Mexico.

Toscano Pulido, G. (2005). *On the use of self-adaptation and elitism for multi-objective particle swarm optimization.* Unpublished doctoral thesis, Computer Science Section, Department of Electrical Engineering, CINVESTAV-IPN, Mexico.

Tayal, M. (2003). *Particle swarm optimization for mechanical design.* Unpublished master's thesis, The University of Texas at Arlington, Arlington, Texas.