Swarm Intelligence manuscript No.
(will be inserted by the editor)

Multi-guide Particle Swarm Optimization

A Multi-swarm Multi-objective Particle Swarm Optimizer

Christiaan Scheepers ·
Andries P. Engelbrecht

Received: date / Accepted: date

Abstract This article presents a proposal for a new particle swarm optimization (PSO) based multi-objective optimization (MOO) algorithm, named multi-guided particle swarm optimization (MGPSO). The exploration behavior of MGPSO is analyzed using a previously developed candidate solution visualization technique. A thorough performance analysis using the inverted generational distance (IGD) performance measure on the Zitzler, Deb, Thiele (ZDT) and Walking Fish Group (WFG) test sets covering both 2 and 3-objective problems is presented. A comparative analysis using the attainment surface inspired porcupine measure is also presented. The results show that MGPSO is highly competitive when compared against current PSO-based MOO algorithms as well as *state of the art* multi-objective evolutionary algorithms (MOEAs).

Keywords multi-guided particle swarm optimization · multi-objective optimization · particle swarm optimization · attainment surface

1 Introduction

Many real life optimization problems consist of multiple objectives, often in conflict with one another, that need to be optimized. These problems are called multi-objective optimization problems (MOPs). The solution for a MOP is not a single solution, but rather, a set of optimal trade-offs, also called Pareto-optimal solutions. This set of Pareto-optimal solutions is referred to as the Pareto-optimal front (POF). Multi-objective optimization (MOO) algorithms can be used to find POFs for MOPs.

Deb (2001) defined the goals for a MOO algorithm as follows:

C. Scheepers
Department of Computer Science, School for Information Technology, University of Pretoria, Pretoria, 0002, South Africa
E-mail: cscheepers@acm.org

A.P. Engelbrecht
Department of Computer Science, School for Information Technology, University of Pretoria, Pretoria, 0002, South Africa
E-mail: engel@cs.up.ac.za

1. find solutions as close to the true POF as possible, and
2. find solutions as diverse as possible along the obtained POF.

Since the introduction of the first MOO algorithm, *i.e.* the vector evaluated genetic algorithm (VEGA) by Schaffer (1985), various MOO algorithms have been developed. In this article, a multi-objective variant of particle swarm optimization (PSO) (Kennedy and Eberhart 1995), named *multi-guided particle swarm optimization* (MGPSO), is proposed.

MGPSO is a multi-swarm adaptation of the PSO algorithm. Each of the objectives is optimized using a subswarm. The MGPSO velocity update equation contains an additional, randomly weighted, archive guide to help guide the swarm towards the POF. The addition of the archive term is shown to improve convergence towards the POF while the random weighting helps to maintain diversity along the POF.

The MGPSO is compared to other PSO-based MOO algorithms as well as *state of the art* multi-objective evolutionary algorithms (MOEAs). The results indicate that MGPSO performs highly competitive and is well suited to solving MOPs.

The remainder of this article is organized as follows. Section 2 discusses multi-objective optimization and provides the definitions used throughout this article. Section 3 introduces the MGPSO. Section 4 describes the experimental procedure and benchmark functions used throughout this article. A performance analysis is presented in Section 5, followed by a comparative analysis against *state of the art* MOEAs in Section 6. Finally, the conclusions and future work are given in Section 7, followed by the acknowledgments.

2 Multi-objective Optimization

Without loss of generality, a multi-objective optimization problem with M objectives is of the form:

$$minimize(f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_M(\mathbf{x})) \tag{1}$$

with $\mathbf{x} \in \mathcal{F}$, $f_m : \mathbb{R}^n \to \mathbb{R}$ for all $m \in [1, M]$, and $\mathcal{F} \subset \mathbb{R}^n$ is the feasible space as determined by constraints.

The following definitions are used throughout this paper.

Definition 2.1 Domination A decision vector $\mathbf{x}_1 \in \mathcal{F}$ dominates a decision vector $\mathbf{x}_2 \in \mathcal{F}$ (denoted by $\mathbf{x}_1 \prec \mathbf{x}_2$) if and only if $f_m(\mathbf{x}_1) \leq f_m(\mathbf{x}_2)\ \forall\ m \in [1, M]$ and $\exists\ m \in [1, M]$ such that $f_m(\mathbf{x}_1) < f_m(\mathbf{x}_2)$.

Definition 2.2 Pareto-optimal A decision vector $\mathbf{x}_1 \in \mathcal{F}$ is said to be Pareto-optimal if no decision vector $\mathbf{x}_2 \in S$ exists such that $\mathbf{x}_2 \prec \mathbf{x}_1$.

Definition 2.3 Pareto-optimal Front A set $Q \subseteq \mathbb{R}^M$ is said to be the Pareto-optimal front if it contains only objective vectors for Pareto-optimal decision vectors. Q is formally defined as $Q = \{\mathbf{x}_1 \in \mathcal{F} \mid \nexists\ \mathbf{x}_2 \in \mathcal{F} : \mathbf{x}_2 \prec \mathbf{x}_1\}$.

The objectives of a MOO algorithm can be summarised as follows:

1. find solutions as close to the true POF as possible, and
2. find solutions as diverse as possible along the obtained POF.

Many MOO algorithms have been developed in the evolutionary computing and swarm intelligence paradigms (Deb et al 2000, 2002; Corne et al 2001; Zitzler et al 2001; Sierra and Coello Coello 2005; Zhang and Li 2007; Nebro et al 2009; Grobler 2009; Harrison et al 2013). The focus of this article is the development of a new PSO based MOO algorithm. For a more general treatment of MOO and MOEAs, the interested reader is referred to Deb (2001).

3 Multi-guide Particle Swarm Optimization

Parsopoulos and Vrahatis (2002b,a) first introduced vector evaluated particle swarm optimization (VEPSO), a multi-swarm multi-objective particle swarm optimization algorithm, inspired from VEGA (Schaffer 1985). Each objective is optimized by a separate subswarm. Particles in each subswarm are evaluated using the corresponding objective function. VEPSO is implemented by selecting the neighborhood guide using a knowledge transfer strategy (KTS) instead of from a neighborhood topology (Parsopoulos and Vrahatis 2002a; Parsopoulos et al 2004; Grobler and Engelbrecht 2009; Matthysen et al 2013; Harrison et al 2013). The VEPSO velocity update equation is formally defined as follows:

$$\mathbf{v}_i(t+1) = w\mathbf{v}_i(t) + c_1\mathbf{r}_1(\mathbf{y}_i(t) - \mathbf{x}_i(t)) + c_2\mathbf{r}_2(\hat{\mathbf{y}}_i(t) - \mathbf{x}_i(t)) \tag{2}$$

where $\mathbf{v}_i(t)$ is the velocity of particle i at iteration t, w is the inertia weight, c_1 and c_2 are the acceleration coefficients, $\mathbf{r}_1$ and $\mathbf{r}_2$ are random vectors with components sampled uniformly from $(0, 1)$, $\mathbf{x}_i(t)$ is the position of particle i at iteration t, $\mathbf{y}_i(t)$ is the personal best position of particle i at iteration t, $\mathbf{y}_i(t)$ is the KTS selected guide for particle i at iteration t.

At the end of each iteration, the non-dominated solutions are stored in an archive. Once the stopping conditions for the algorithm is reached, the POF will be the solutions that have been stored in the archive. VEPSO does not specify how the archive should be implemented. Scheepers and Engelbrecht (2016a, 2017b) found that the choice of the archive implementation has a notable impact on an algorithm's performance. Scheepers and Engelbrecht (2016a, 2017b) further found that, overall, the crowding distance based archive performed best.

Scheepers and Engelbrecht (2016c,b) further investigated the exploration behavior of VEPSO and found that VEPSO does not exploit the well-performing regions of the search space enough.

This article introduces the MGPSO algorithm. MGPSO is loosely based on VEPSO and attempts to address the aforementioned shortcomings that have been identified in the VEPSO algorithm. Similar to VEPSO, MGPSO is a multi-swarm algorithm where each objective is represented by a subswarm. Particles in each subswarm are evaluated using the corresponding objective function. The personal best and neighborhood best positions are also updated using the corresponding objective function fitness value.

In contrast to VEPSO, MGPSO does not make use of a KTS. Instead, MGPSO retains the neighborhood guide and adds an additional archive guide term to the velocity update equation to facilitate information exchange between the subswarms. The MGPSO velocity update equation is formally defined as follows:

$$\begin{aligned}\mathbf{v}_i(t+1) = w\mathbf{v}_i(t) + c_1\mathbf{r}_1(\mathbf{y}_i(t) - \mathbf{x}_i(t)) + \lambda_i c_2 \mathbf{r}_2(\hat{\mathbf{y}}_i(t) - \mathbf{x}_i(t)) \\ + (1-\lambda_i)c_3\mathbf{r}_3(\hat{\mathbf{a}}_i(t) - \mathbf{x}_i(t))\end{aligned} \tag{3}$$

where c_3 is an acceleration coefficient, $\mathbf{r}_3$ is a random vector with components sampled uniformly from $(0, 1)$, $\hat{\mathbf{a}}_i(t)$ is a randomly selected solution from the archive for particle i at iteration t, and λ_i is the exploitation trade-off coefficient for particle i. λ_i controls the amount of influence that the archive guide has on the particle's velocity and thus the amount of exploitation of the already found POF. Smaller λ_i values increase the influence of the archive guide while simultaneously decreasing the influence of the neighborhood guide.

Inspired by the findings of Scheepers and Engelbrecht (2016a, 2017b) on the archive's influence on VEPSOs performance, the MGPSO archive is a bounded archive using the crowding distance based archive implementation. For easier comparison with other algorithms, the default archive size is set equal to the total number of particles in the subswarms.

The archive guide, $\hat{\mathbf{a}}_i(t)$, is selected from a competition pool as the solution with the largest corresponding crowding distance in the archive. The competition pool is constructed by randomly selecting a predefined number of solutions from the archive. Empirical experimentation showed that tournament sizes of 2 and 3 yielded good results (Scheepers 2018). By using crowding distance as part of the archive guide selection process, the MGPSO is guided to focus more on sparsely populated areas of the objective space.

The MGPSO algorithm is formally defined as follows:

Algorithm 1 Multi-guided Particle Swarm Optimization (MGPSO)

for each objective $m = 1, ..., n_m$ **do**
  Create and initialize a swarm, S_m, of n_{s_m} particles uniformly within a predefined hypercube of dimension n_x;
  Let f_m be the objective function;
  Let $S_m.\mathbf{y}_i$ represent the personal best position of particle $S_m.\mathbf{x}_i$, initialized to $S_m.\mathbf{x}_i(0)$;
  Let $S_m.\hat{\mathbf{y}}_i$ represent the neighborhood best position of particle $S_m.\mathbf{x}_i$, initialized to $S_m.\mathbf{x}_i(0)$;
  Initialize $S_m.\mathbf{v}_i(0)$ to $\mathbf{0}$;
  Initialize $S_m.\lambda_i \sim U(0, 1)$;
end for
Let $t = 0$;
repeat
  for each objective $m = 1, ..., n_m$ **do**
    for each particle $i = 1, ..., S_m.n_s$ **do**
      if $f_m(S_m.\mathbf{x}_i) < f_m(S_m.\mathbf{y}_i)$ **then**
        $S_m.\mathbf{y}_i = S_m.\mathbf{x}_i(t)$;
      end if
      for particles $\hat{\imath}$ with particle i in their neighborhood **do**
        if $f_m(S_m.\mathbf{y}_i) < f_m(S_m.\hat{\mathbf{y}}_{\hat{\imath}})$ **then**
          $S_m.\hat{\mathbf{y}}_{\hat{\imath}} = S_m.\mathbf{y}_i$;
        end if
      end for
      Update the archive with the solution $S_m.\mathbf{x}_i$;
    end for
  end for

Algorithm 1 Multi-guided Particle Swarm Optimization (MGPSO)

for each objective $m = 1, ..., n_m$ **do**
  for each particle $i = 1, ..., S_m.n_s$ **do**
    Select a solution, $S_m.\hat{\mathbf{a}}_i(t)$, from the archive using tournament selection;
    $S_m.\mathbf{v}_i(t+1) = wS_m.\mathbf{v}_i(t) + c_1\mathbf{r}_1(S_m.\mathbf{y}_i(t) - S_m.\mathbf{x}_i(t))$
    $+S_m.\lambda_i c_2\mathbf{r}_2(S_m.\hat{\mathbf{y}}_i(t) - S_m.\mathbf{x}_i(t))$
    $+(1 - S_m.\lambda_i)c_3\mathbf{r}_3(S_m.\hat{\mathbf{a}}_i(t) - S_m.\mathbf{x}_i(t)))$;
    $S_m.\mathbf{x}_i(t+1) = S_m.\mathbf{x}_i(t) + S_m.\mathbf{v}_i(t+1)$;
  end for
end for
$t = t + 1$;
until stopping condition is true

4 Experimental Procedure

MGPSO was tested using the Zitzler, Deb and Thiele (ZDT) (Zitzler et al 2000) and the Walking Fish Group (WFG) test sets (Huband et al 2005, 2006). The test sets provide a mix of challenges to test MOO algorithms against. Table 1 present a summary of the properties of each of the problems in the aforementioned test sets.

MGPSO was executed with 50 particles, divided between the subswarms, for each of the ZDT, 2-objective WFG, and 3-objective WFG problems. Optimized values for the number of particles per subswarm, the inertia weight, w, the acceleration coefficients, c_1, c_2, and c_3, and the tournament size were used.

The parameters were optimized using a visual analysis technique proposed by Franken (2009). Franken proposed plotting the results for various control parameter value combinations on a parallel coordinate plot. Each control parameter value combination is represented by a single pattern on the parallel coordinate plot. The color for each pattern is determined according to how well the control parameter value combination being represented, performed. By adjusting the colors and highlighting the region of the well-performing patterns a researcher can vi-

Table 1 Properties of the ZDT and WFG Problems

Name	Separability	Modality	Geometry
ZDT1	separable	unimodal	convex
ZDT2	separable	unimodal	concave
ZDT3	separable	unimodal/multimodal	disconnected
ZDT4	separable	unimodal/multimodal	convex
ZDT6	separable	multimodal	concave
WFG1	separable	unimodal	convex, mixed
WFG2	non-separable	unimodal/multimodal	convex, disconnected
WFG3	non-separable	unimodal	linear, degenerate
WFG4	separable	multimodal	concave
WFG5	separable	multimodal	concave
WFG6	non-separable	unimodal	concave
WFG7	separable	unimodal	concave
WFG8	non-separable	unimodal	concave
WFG9	non-separable	multimodal, deceptive	concave

sually explore the parameter space. Table 2 lists the optimized MGPSO control parameter values for each of the problems (Scheepers 2018).

The MGPSO algorithm was implemented and executed using the CIlib framework[1] (Pampara et al 2008b,a). The results presented were taken over 30 independent runs of 2000 iterations for each algorithm for each problem.

5 Exploration and Performance Analysis

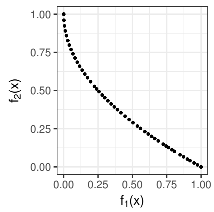
POFs obtained by the MGPSO are shown in figures 1(a) through 1(e) for the ZDT problems, figures 2(a) through 2(i) for the 2-objective WFG problems, and figures 3(a) through 3(i) for the 3-objective WFG problems. The figures show that MGPSO managed to obtain a close approximation of the true POFs for almost all of the test problems. The POF for ZDT4 shows that MGPSO was not able to obtain a close approximation and the problem was particularly challenging for MGPSO to solve. The ZDT4 result indicates that the MGPSO is susceptible to getting stuck in local optima under certain conditions. Further work is required to fully analyze MGPSO's susceptibility to local optima.

In order to study the exploration behavior of MGPSO, the candidate solutions represented by the particle positions were tracked and plotted. For each of the 2-objective problems, two candidate solution plots were generated, one for each of the sub-swarms. Figures 4(a) through 4(j) show the candidate solution plots for

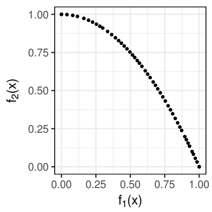
[1] https://cirg-up.github.io/cilib/

Table 2 Optimized MGPSO parameters

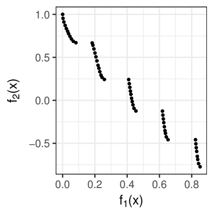
Problem	**Objectives**	$\lvert S_1 \rvert$	$\lvert S_2 \rvert$	$\lvert S_3 \rvert$	T	w	c_1	c_2	c_3
ZDT1	2	33	17		3	0.475	1.80	1.10	1.80
ZDT2	2	8	42		3	0.075	1.60	1.35	1.90
ZDT3	2	8	42		3	0.050	1.85	1.90	1.90
ZDT4	2	5	45		2	0.175	1.85	1.35	1.85
ZDT6	2	1	49		3	0.600	1.85	1.55	1.80
WFG1	2	45	5		3	0.275	1.65	1.80	1.75
WFG2	2	24	26		2	0.750	1.15	1.70	1.05
WFG3	2	31	19		2	0.600	1.60	1.85	0.95
WFG4	2	2	48		2	0.100	0.80	1.65	1.70
WFG5	2	50	0		2	0.600	0.80	1.60	1.85
WFG6	2	19	31		2	0.525	0.65	0.60	1.65
WFG7	2	29	21		2	0.450	1.20	1.85	1.55
WFG8	2	37	13		3	0.750	1.00	1.65	1.05
WFG9	2	13	37		2	0.275	1.00	0.50	1.70
WFG1	3	37	4	9	2	0.125	1.20	1.30	1.75
WFG2	3	24	25	1	2	0.275	1.25	1.40	1.70
WFG3	3	29	10	11	2	0.525	1.65	1.75	0.75
WFG4	3	29	21	0	2	0.275	1.75	0.50	1.05
WFG5	3	2	48	0	3	0.575	0.60	1.85	1.75
WFG6	3	5	30	15	3	0.300	0.90	0.90	1.90
WFG7	3	10	22	18	2	0.425	1.45	1.50	1.40
WFG8	3	4	23	23	3	0.425	0.95	1.75	1.85
WFG9	3	4	45	1	2	0.275	1.25	0.75	1.50



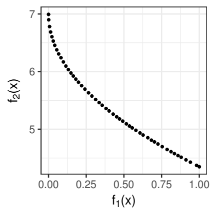
(a) ZDT1



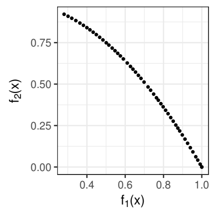
(b) ZDT2



(c) ZDT3



(d) ZDT4



(e) ZDT6

Fig. 1 MGPSO calculated POFs for ZDT problems

the ZDT problems and figures 5(a) through 5(r) show the candidate solution plots for the WFG problems.

The candidate solution plots for the majority of the problems show that at least one swarm tend to explore the region close to the true POF. The exploration pattern clearly aligns with the two goals for a MOO algorithm as defined by Deb (2001). The candidate solution plots show that after an initial exploration of the search space, the MGPSO focusses on exploitation of the well-performing regions of the search space. The candidate solution plots indicate that the MGPSO successfully address the shortcomings of the VEPSO algorithm.

While the candidate solution plots show the exploration behavior, no information regarding the quality of the found solutions are given. In order to quantify the quality of the solutions, the inverted generational distance (IGD) was calculated for each of the problems (Coello Coello and Sierra 2004; Sierra and Coello Coello 2004). The true POF sets from the jMetal framework (Nebro et al 2015) were used for the IGD calculations.

For comparison, the IGD values for VEPSO with the random knowledge transfer strategy (KTS) (Grobler 2009), VEPSO with parent centric crossover archive (PCXA) KTS (Harrison et al 2013), speed-constrained multi-objective PSO (SMPSO) (Nebro et al 2009) and optimized multi-objective PSO (OMOPSO) (Sierra and Coello Coello 2005) are also given. Figures 6(a) through 7(i) depict the mean IGD for each of the ZDT and 2-objective WFG problems. Figures 8(a) through 8(i) depict the mean IGD for each of the 3-objective WFG problems.

The figures indicate that MGPSO manages to successfully explore and exploit the well-performing regions of the search space. The figures additionally show that MGPSO outperformed VEPSO for the majority of the test problems, especially when considering the 3-objective test problems. Only SMPSO performed better than MGPSO in a few of the 2-objective problems.

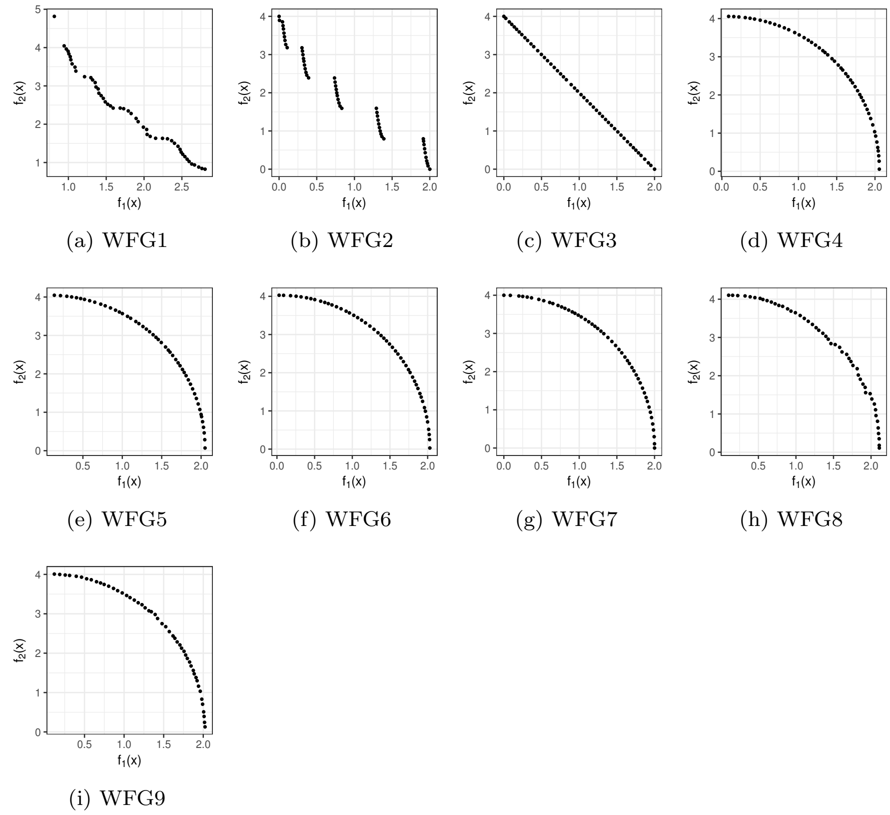


Fig. 2 MGPSO calculated POFs for 2-objective WFG problems

For the majority of the problems, the best-performing IGD measurement values were achieved in less than 500 iterations. WFG1 is a notable exception where the IGD measurement value continued to decrease, albeit very slowly, all the way up to iteration 2000.

For each of the test sets a win, loss, difference, and ranking table was calculated to statistically rank the performance of MGPSO in comparison with other algorithms. IGD measurement values for each algorithm were compared with all the other algorithms using the Mann-Whitney U (Gibbons and Chakraborti 2010) test with a confidence level of 95%. If a statistically significant difference was detected, a win was recorded for the algorithm and a loss was recorded for the other algorithm. The difference between the wins and losses were then computed and listed in the table.

Table 3 lists the ranking for the ZDT test set, table 4 lists the ranking for the 2-objective WFG test set, and table 5 lists the ranking for the 3-objective WFG test set.

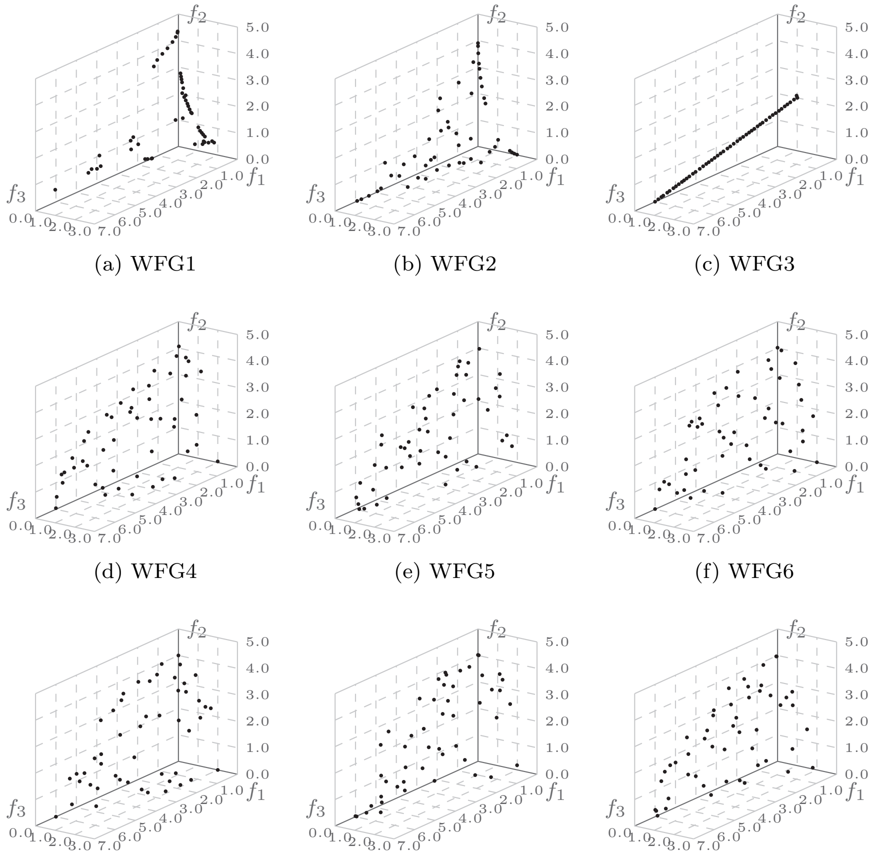
For the ZDT test set, MGPSO was ranked second with SMPSO ranked first. For each of the ZDT problems, MGPSO recorded only a single loss, indicating that only SMPSO, that recorded no losses, won against it. For the 2-objective WFG

Table 3 Inverted Generational Distance Ranking - Multi-objective PSO Algorithms

Algorithm	**Result**	**ZDT Function** 1	2	3	4	6	**Overall**
MGPSO	Wins	2	3	3	3	3	14
	Losses	1	1	1	1	1	5
	Difference	1	2	2	2	2	9
	Rank	2	2	2	2	2	2
SMPSO	Wins	4	4	4	4	4	20
	Losses	0	0	0	0	0	0
	Difference	4	4	4	4	4	20
	Rank	**1**	**1**	**1**	**1**	**1**	**1**
OMOPSO	Wins	1	2	1	2	0	6
	Losses	3	2	3	2	3	13
	Difference	-2	0	-2	0	-3	-7
	Rank	4	3	4	3	4	4
$\text{VEPSO}_{\text{Random}}$	Wins	0	1	0	0	0	1
	Losses	4	3	4	4	3	18
	Difference	-4	-2	-4	-4	-3	-17
	Rank	5	4	5	5	4	5
$\text{VEPSO}_{\text{PCXA}}$	Wins	2	0	2	1	2	7
	Losses	1	4	2	3	2	12
	Difference	1	-4	0	-2	0	-5
	Rank	2	5	3	4	3	3

Table 4 Inverted Generational Distance Ranking - Multi-objective PSO Algorithms

Algorithm	**Result**	**2-objective WFG Function** 1	2	3	4	5	6	7	8	9	**Overall**
MGPSO	Wins	4	4	4	2	3	3	4	3	3	30
	Losses	0	0	0	2	1	0	0	0	1	4
	Difference	4	4	4	0	2	3	4	3	2	26
	Rank	**1**	**1**	**1**	3	2	**1**	**1**	**1**	2	**1**
SMPSO	Wins	2	2	2	4	4	3	2	2	4	25
	Losses	2	1	2	0	0	0	2	2	0	9
	Difference	0	1	0	4	4	3	0	0	4	16
	Rank	3	2	3	**1**	**1**	**1**	3	3	**1**	2
OMOPSO	Wins	3	2	3	3	0	0	3	3	0	17
	Losses	1	1	1	1	4	3	1	0	4	16
	Difference	2	1	2	2	-4	-3	2	3	-4	1
	Rank	2	2	2	2	5	4	2	**1**	5	3
$\text{VEPSO}_{\text{Random}}$	Wins	0	0	0	0	1	2	0	0	2	5
	Losses	4	4	4	4	3	2	4	4	2	31
	Difference	-4	-4	-4	-4	-2	0	-4	-4	0	-26
	Rank	5	5	5	5	4	3	5	5	3	5
$\text{VEPSO}_{\text{PCXA}}$	Wins	1	1	1	1	2	0	1	1	1	9
	Losses	3	3	3	3	2	3	3	3	3	26
	Difference	-2	-2	-2	-2	0	-3	-2	-2	-2	-17
	Rank	4	4	4	4	3	4	4	4	4	4



(a) WFG1 (b) WFG2 (c) WFG3

(d) WFG4 (e) WFG5 (f) WFG6

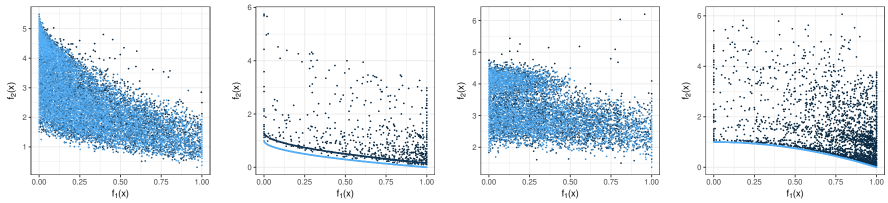
(g) WFG7 (h) WFG8 (i) WFG9

Fig. 3 MGPSO calculated POFs for 3-objective WFG problems

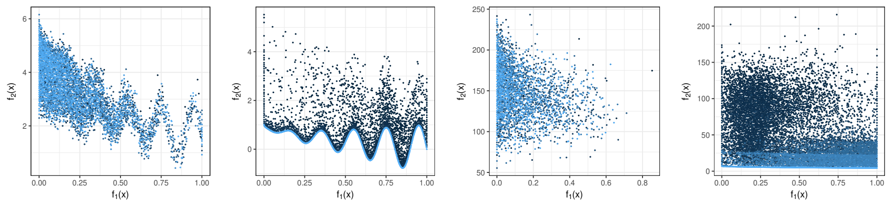
test set, MGPSO was ranked first with SMPSO ranked second. Finally, for the 3-objective WFG test set, MGPSO ranked not only first overall, but also ranked first for each individual problem. Only for the 3-objective WFG6 problem did SMPSO rank jointly first with MGPSO. Similar to the ZDT and 2-objective WFG results, SMPSO ranked second overall for the 3-objective WFG test set.

The results clearly show that MGPSO performed on par with leading PSO based multi-objective optimizers. MGPSO's lowest rank was 3 out of 5 for only one of the 23 problems. MGPSO was ranked first for 15 of the 23 tested problems. Overall, both of the tested VEPSO variants performed poorly, in terms of IGD, when compared to the other algorithms.

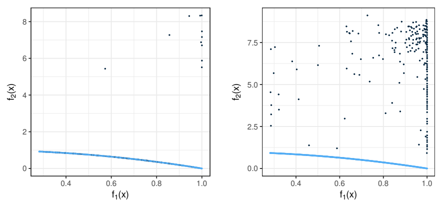
In addition to quantifying the performance of MGPSO using IGD, a comparative quantitative measure named the porcupine measure (Scheepers and Engelbrecht 2017a) was also used. The porcupine measure is based on the quantitative measure developed by Knowles and Corne (2000) to quantify attainment surface (Fonseca and Fleming 1995) comparisons. Results for comparing two algorithms



(a) ZDT1 S_1 (b) ZDT1 S_2 (c) ZDT2 S_1 (d) ZDT2 S_2



(e) ZDT3 S_1 (f) ZDT3 S_2 (g) ZDT4 S_1 (h) ZDT4 S_2



(i) ZDT6 S_1 (j) ZDT6 S_2

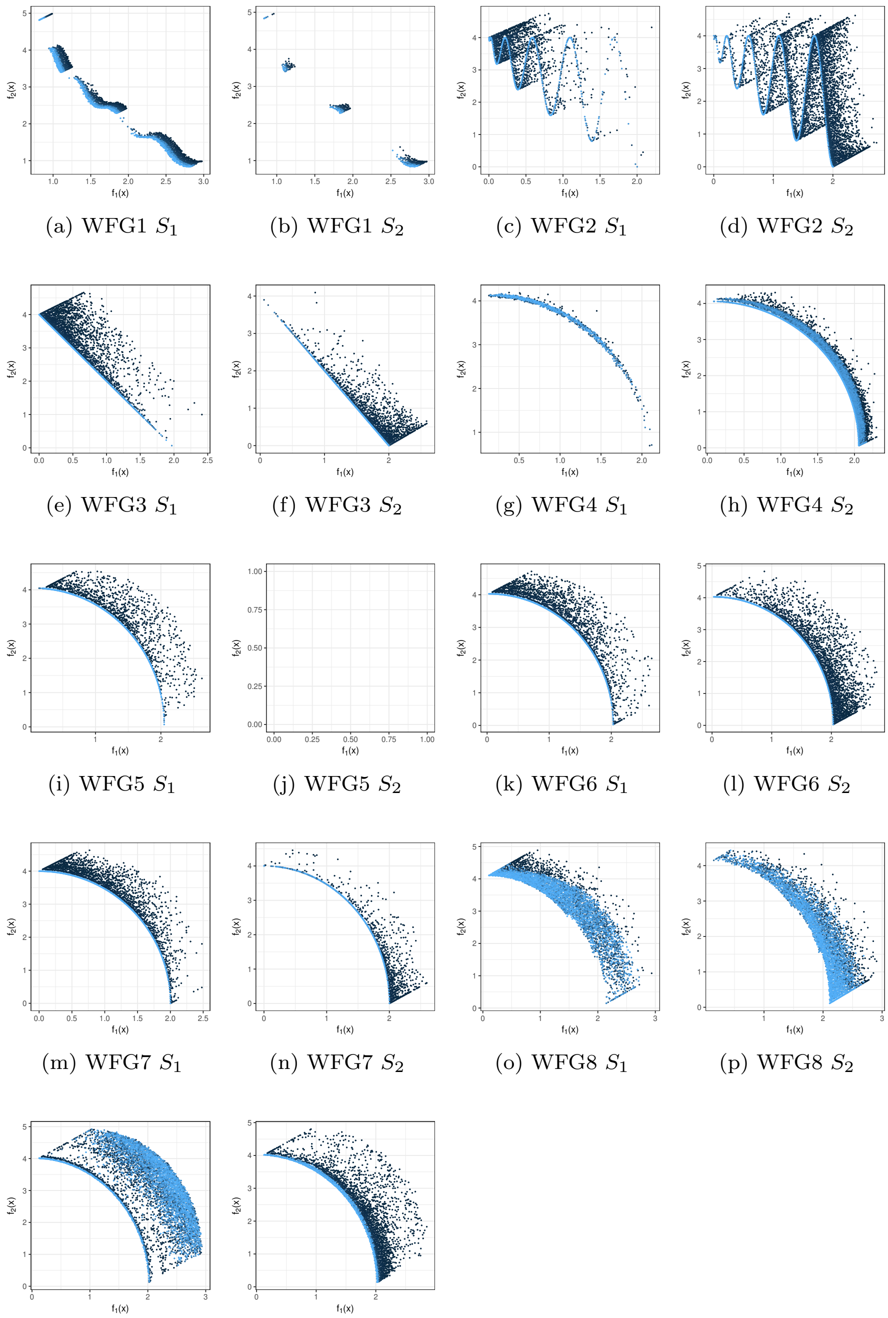
Fig. 4 MGPSO swarms S_1 and S_2 candidate solutions for iterations 1-2000 (darker color represents lower iterations, lighter color indicates higher iterations)

A and B are presented as a value pair $[a, b]$, indicating the percentage, a, of the attainment surface where algorithm A statistically outperformed algorithm B and the percentage, b, where algorithm B statistically outperformed algorithm A.

Table 6 presents the porcupine measure for comparisons between MGPSO and the other PSO-based MOO algorithms. Percentages that exceed those of the algorithm being compared against by more than 5% are shown in bold. The results reinforce what the IGD comparison has already shown: MGPSO compared favorably against the majority of the PSO-based MOO algorithms using the ZDT, 2-objective WFG, and 3-objective WFG test sets.

For 78 of the 92 comparisons, MGPSO performed statistically significantly better than the algorithm being compared against. For 42 of the comparisons, MGPSO performed statistically significantly better than the algorithm being compared against by more than 90%. MGPSO only performed worse by more than 5% for 10, or less than 11%, of the comparisons. For four of the comparisons, neither algorithm outperformed the other by more than 5%.

Overall, the results presented in this section showed that MGPSO is highly competitive when compared against PSO-based MOO algorithms for 2 and 3-



(a) WFG1 S_1 (b) WFG1 S_2 (c) WFG2 S_1 (d) WFG2 S_2

(e) WFG3 S_1 (f) WFG3 S_2 (g) WFG4 S_1 (h) WFG4 S_2

(i) WFG5 S_1 (j) WFG5 S_2 (k) WFG6 S_1 (l) WFG6 S_2

(m) WFG7 S_1 (n) WFG7 S_2 (o) WFG8 S_1 (p) WFG8 S_2

(q) WFG9 S_1 (r) WFG9 S_2

Fig. 5 MGPSO swarms S_1 and S_2 candidate solutions for iterations 1-2000 (darker color represents lower iterations, lighter color indicates higher iterations)

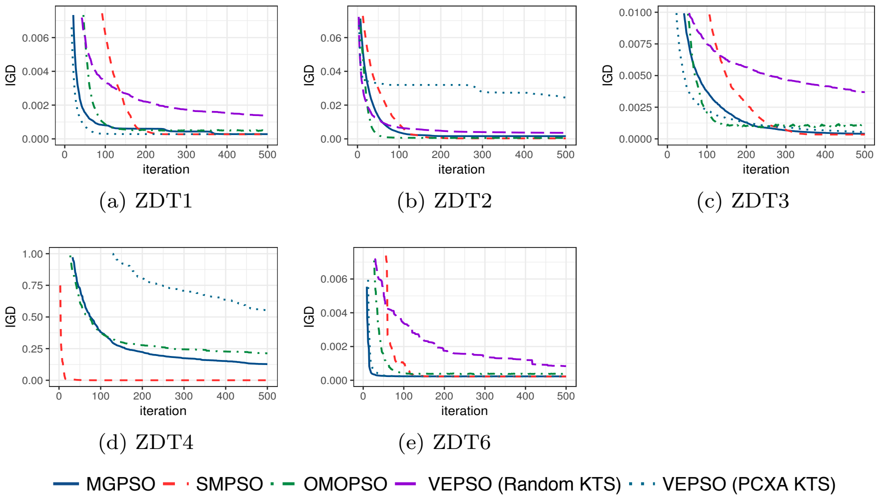


Fig. 6 PSO inverted generational distance (IGD) for ZDT problems

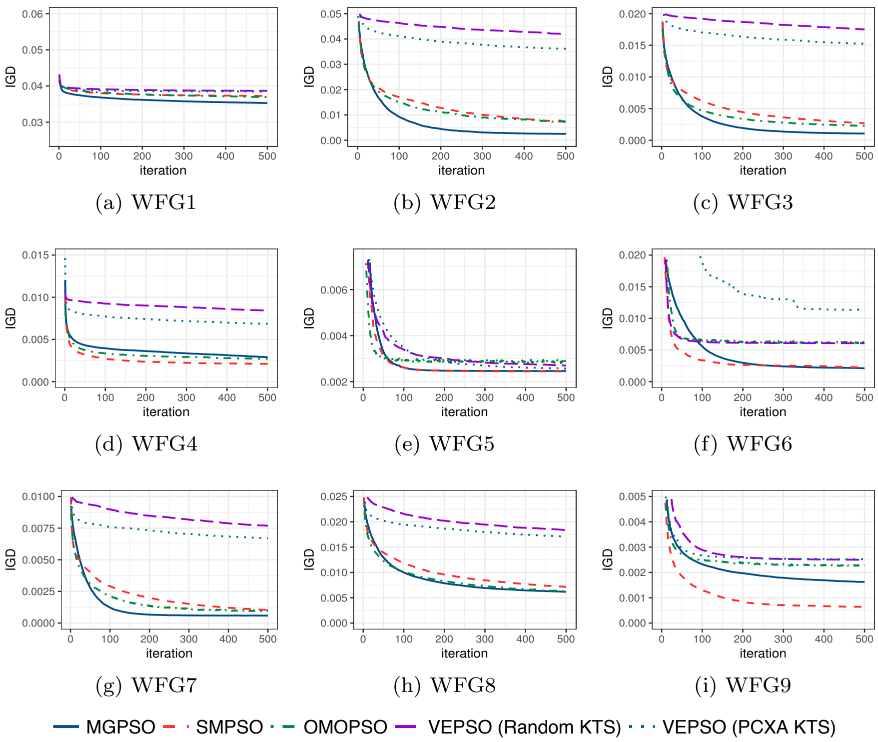


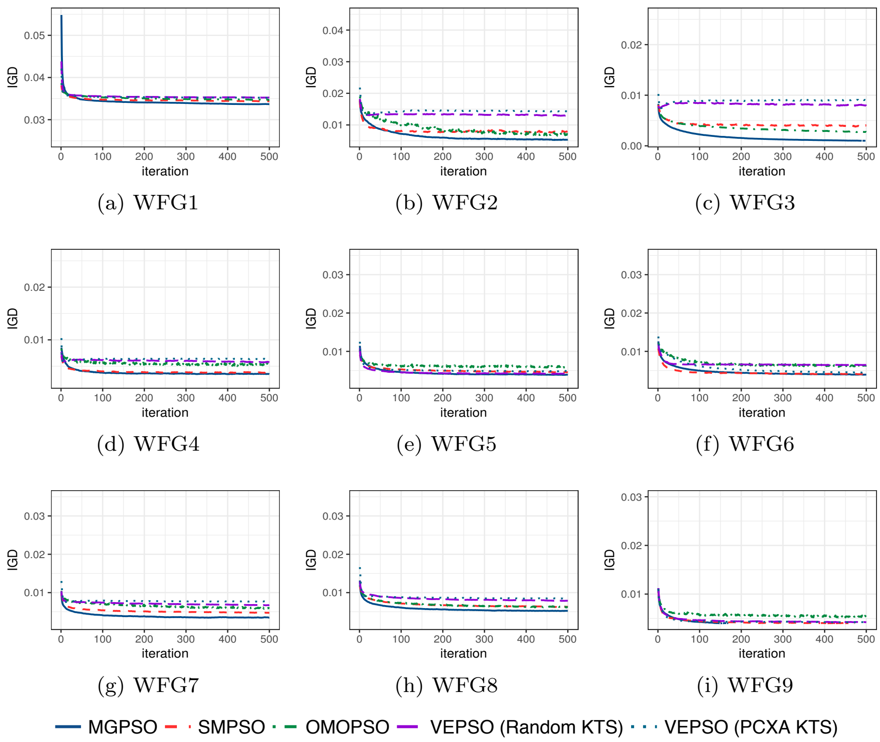
Fig. 7 PSO inverted generational distance (IGD) for 2-objective WFG problems

Table 5 Inverted Generational Distance Ranking - Multi-objective PSO Algorithms

Algorithm	Result	3-objective WFG Function 1	2	3	4	5	6	7	8	9	Overall
MGPSO	Wins	4	4	4	4	4	3	4	4	4	35
	Losses	0	0	0	0	0	0	0	0	0	0
	Difference	4	4	4	4	4	3	4	4	4	35
	Rank	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
SMPSO	Wins	3	2	2	3	2	3	3	2	3	23
	Losses	1	2	2	1	2	0	1	2	1	12
	Difference	2	0	0	2	0	3	2	0	2	11
	Rank	2	3	3	2	3	**1**	2	3	2	2
OMOPSO	Wins	2	3	3	2	0	0	2	3	0	15
	Losses	2	1	1	2	4	3	2	1	4	20
	Difference	0	2	2	0	-4	-3	0	2	-4	-5
	Rank	3	2	2	3	5	4	3	2	5	3
$\text{VEPSO}_{\text{Random}}$	Wins	0	1	1	1	3	0	1	1	1	9
	Losses	3	3	3	3	1	3	3	3	2	24
	Difference	-3	-2	-2	-2	2	-3	-2	-2	-1	-15
	Rank	4	4	4	4	2	4	4	4	3	4
$\text{VEPSO}_{\text{PCXA}}$	Wins	0	0	0	0	1	2	0	0	1	4
	Losses	3	4	4	4	3	2	4	4	2	30
	Difference	-3	-4	-4	-4	-2	0	-4	-4	-1	-26
	Rank	4	5	5	5	4	3	5	5	3	5

Table 6 Porcupine Measure - Multi-objective PSO Algorithms

Problem	Objectives	MGPSO vs SMPSO	OMOPSO	$\text{VEPSO}_{\text{RND}}$	$\text{VEPSO}_{\text{PCXA}}$
ZDT1	2	[8.3, 9.2]	[**71.4**, 3.5]	[**85.6**, 3.3]	[7.2, 8.9]
ZDT2	2	[4.0, **10.1**]	[**52.8**, 5.0]	[**92.9**, 1.1]	[**100.0**, 0.0]
ZDT3	2	[2.3, **85.9**]	[37.6, **56.7**]	[**96.3**, 0.0]	[**86.3**, 0.0]
ZDT4	2	[0.0, **87.7**]	[**67.9**, 0.0]	[**100.0**, 0.0]	[**100.0**, 0.0]
ZDT6	2	[**31.6**, 4.6]	[**48.7**, 15.1]	[**45.5**, 6.9]	[5.7, 6.9]
WFG1	2	[**99.9**, 0.0]	[**99.9**, 0.0]	[**99.9**, 0.0]	[**99.9**, 0.0]
WFG2	2	[**99.9**, 0.0]	[**99.9**, 0.0]	[**99.9**, 0.0]	[**99.9**, 0.0]
WFG3	2	[**99.9**, 0.0]	[**99.9**, 0.0]	[**99.9**, 0.0]	[**99.9**, 0.0]
WFG4	2	[0.0, **78.3**]	[28.7, **34.7**]	[**99.9**, 0.0]	[**99.9**, 0.0]
WFG5	2	[5.1, **11.0**]	[**45.7**, 4.2]	[**55.4**, 1.4]	[**37.0**, 0.1]
WFG6	2	[**30.1**, 2.4]	[**99.9**, 0.0]	[**99.9**, 0.0]	[**99.9**, 0.0]
WFG7	2	[**99.9**, 0.0]	[**99.8**, 0.0]	[**99.9**, 0.0]	[**99.9**, 0.0]
WFG8	2	[**77.2**, 0.6]	[**43.5**, 33.3]	[**99.9**, 0.0]	[**99.9**, 0.0]
WFG9	2	[0.0, **99.9**]	[**89.1**, 0.0]	[**99.9**, 0.0]	[**99.9**, 0.0]
WFG1	3	[**78.0**, 4.9]	[**83.0**, 2.7]	[**90.2**, 0.8]	[**80.6**, 0.9]
WFG2	3	[**81.2**, 0.2]	[16.7, **27.7**]	[**100.0**, 0.0]	[**100.0**, 0.0]
WFG3	3	[**89.1**, 8.5]	[**81.4**, 14.1]	[**92.4**, 6.5]	[**91.1**, 7.9]
WFG4	3	[**58.6**, 11.4]	[**62.7**, 15.1]	[**100.0**, 0.0]	[**100.0**, 0.0]
WFG5	3	[**71.4**, 11.1]	[41.5, 38.5]	[**44.3**, 12.4]	[**83.1**, 0.0]
WFG6	3	[**19.4**, 11.3]	[**59.7**, 13.9]	[**84.8**, 0.0]	[**72.9**, 0.0]
WFG7	3	[**99.9**, 0.0]	[**81.7**, 1.3]	[**100.0**, 0.0]	[**100.0**, 0.0]
WFG8	3	[**86.4**, 0.0]	[**65.7**, 15.2]	[**99.9**, 0.0]	[**100.0**, 0.0]
WFG9	3	[4.9, **17.8**]	[**39.6**, 31.8]	[**49.9**, 7.9]	[**60.7**, 3.9]



(a) WFG1 (b) WFG2 (c) WFG3 (d) WFG4 (e) WFG5 (f) WFG6 (g) WFG7 (h) WFG8 (i) WFG9

Fig. 8 PSO inverted generational distance (IGD) for 3-objective WFG problems

objective problems. MGPSO consistently outperformed OMOPSO, VEPSO with random KTS, and VEPSO with a PCXA KTS.

6 Comparative Analysis

In order to rank MGPSO's performance for solving MOO problems, a comparison against *state of the art* MOEAs was carried out. The non-dominated sorting genetic algorithm (NSGA-II) by Deb et al (2000, 2002), multi-objective evolutionary algorithm based on decomposition (MOEA/D) by Zhang and Li (2007), strength pareto evolutionary algorithm by Zitzler et al (2001), and pareto envelope based selection II (PESA-II) by Corne et al (2001) algorithms were selected to compare against.

Figures 9(a) through 9(e), figures 10(a) through 10(i), and figures 11(a) through 11(i) depicts the IGD for the ZDT, 2-objective WFG, and 3-objective WFG test sets. The figures show that MGPSO performed favorably against the leading MOO algorithms. For ZDT1, ZDT2, and ZDT6, the IGD for MGPSO decreased significantly faster than that of the other algorithms. However, for ZDT2, the IGD leveled out significantly higher than that of the other algorithms. For the 2-objective WFG2, MGPSO significantly outperformed all the other algorithms.

For ZDT4, MGPSO performed significantly worse than the other algorithms; the obtained POF also shows the weak performance.

For both the 2-objective and 3-objective WFG1 problems, NSGA II performed notably better than any of the other MOO algorithms, including MGPSO. For the 2-objective WFG2 and WFG6 problems, MGPSO outperformed all the other MOO algorithms.

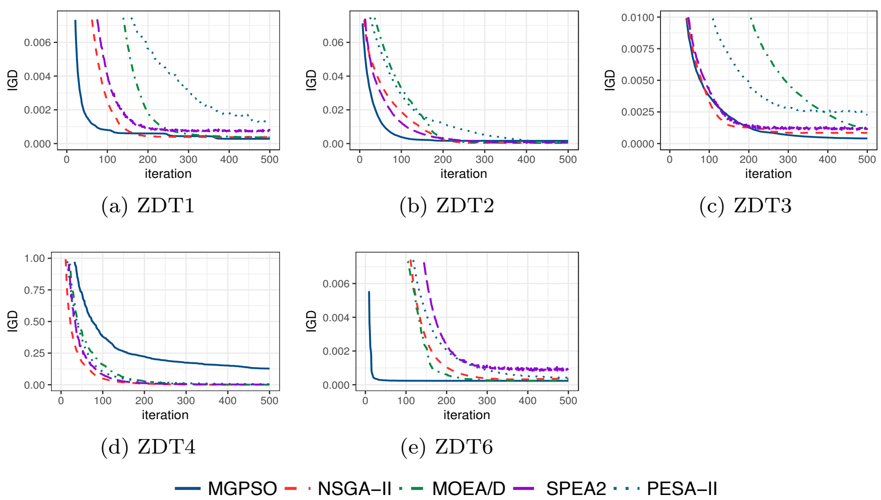
Tables 7, 8, and 9 list the win, loss, difference, and ranking for the ZDT, 2-objective WFG, and 3-objective WFG test sets respectively. In all three cases, MGPSO ranked first overall.

For the ZDT test set, MGPSO ranked first for all the problems except ZDT4 where MGPSO ranked last and NSGA II first. MOEA/D ranked jointly first for ZDT4 and ZDT6. MOEA/D and NSGA II both achieved 12 wins behind MGPSO's 15 wins. MGPSO recorded losses for only ZDT4.

For the 2-objective WFG test set, MGPSO ranked first for six of the nine problems. MGPSO ranked second for WFG1 and WFG8, and fourth for WFG4. NSGA II ranked second overall and ranked jointly first for WFG1, WFG4, and WFG8.

For the 3-objective WFG2 test set, MGPSO performed even better than on the 3-objective problems. MGPSO ranked first for all but two of the problems. For the remaining two problems, WFG1 and WFG8, MGPSO ranked second. MGPSO recorded 34 wins and only two losses, followed by NSGA II with 29 wins and seven losses. NSGA II ranked jointly first for WFG1 and WFG8 and second for the remaining seven problems.

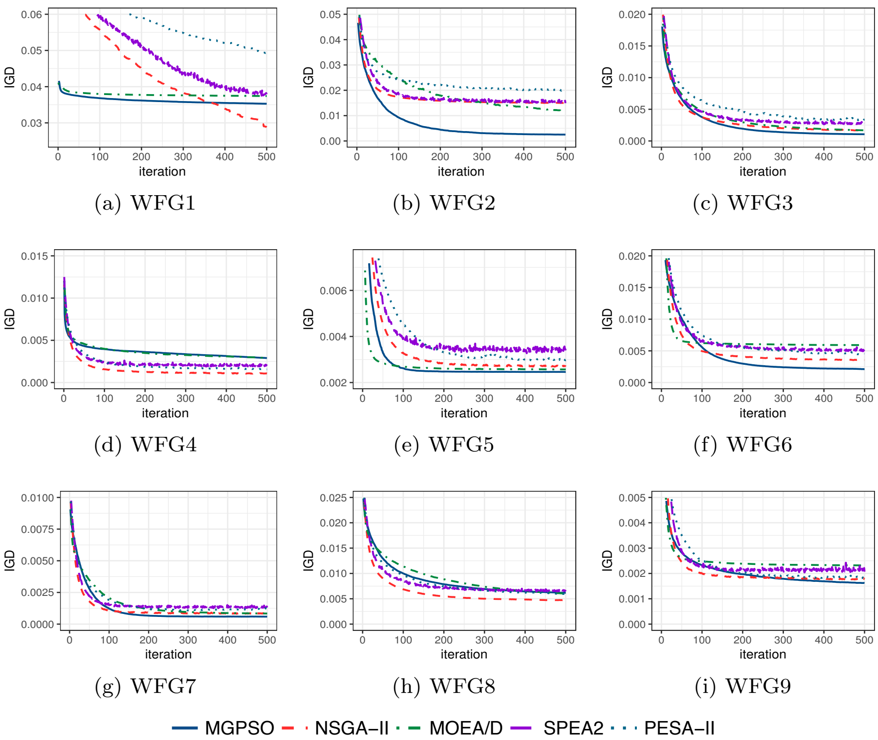
Table 10 depicts the porcupine measure results for MGPSO compared against *state of the art* MOEAs. Results where one algorithm outperformed another by more than 5% are shown in bold. The results indicate that MGPSO managed to outperform the other algorithms by more than 5% in 66 of the 92 comparisons. For 30 of the comparisons, MGPSO performed statistically significantly better than



(a) ZDT1 (b) ZDT2 (c) ZDT3

(d) ZDT4 (e) ZDT6

Fig. 9 MOO inverted generational distance (IGD) for ZDT problems

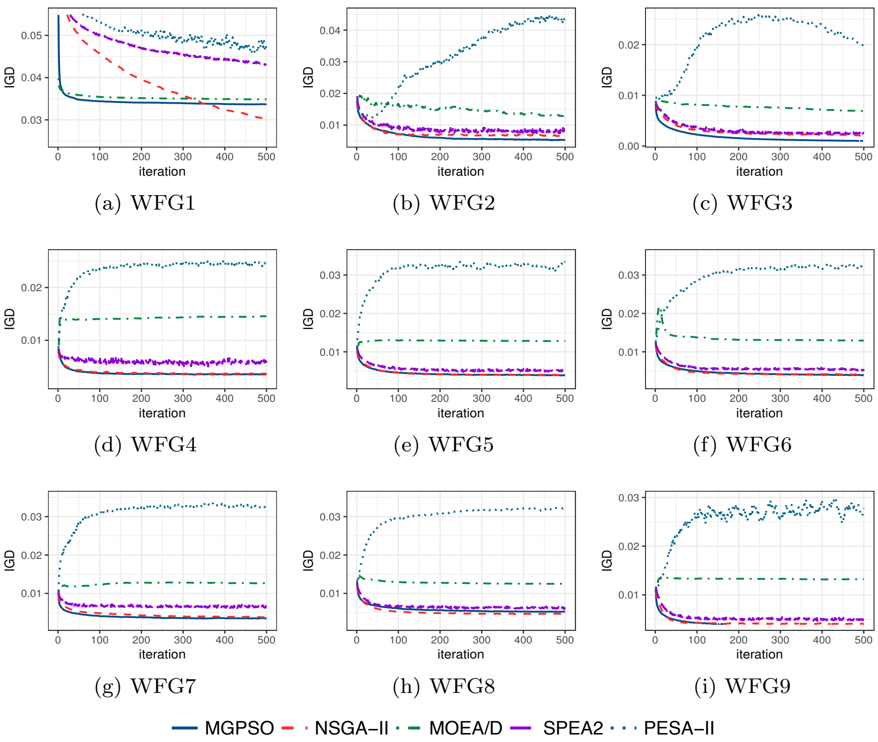


(a) WFG1 (b) WFG2 (c) WFG3
(d) WFG4 (e) WFG5 (f) WFG6
(g) WFG7 (h) WFG8 (i) WFG9

Fig. 10 MOO inverted generational distance (IGD) for 2-objective WFG problems

Table 7 Inverted Generational Distance Ranking - State of the Art MOEAs

Algorithm	**Result**	**ZDT Function 1**	**2**	**3**	**4**	**6**	**Overall**
MGPSO	Wins	4	4	4	0	3	15
	Losses	0	0	0	4	0	4
	Difference	4	4	4	-4	3	11
	Rank	**1**	**1**	**1**	5	**1**	**1**
NSGA II	Wins	2	2	3	3	2	12
	Losses	2	2	1	0	2	7
	Difference	0	0	2	3	0	5
	Rank	3	3	2	**1**	3	3
MOEA/D	Wins	3	3	0	3	3	12
	Losses	1	1	3	0	0	5
	Difference	2	2	-3	3	3	7
	Rank	2	2	5	**1**	**1**	2
SPEA2	Wins	0	0	1	2	0	3
	Losses	3	4	2	2	4	15
	Difference	-3	-4	-1	0	-4	-12
	Rank	4	5	3	3	5	4
PESA-II	Wins	0	1	0	1	1	3
	Losses	3	3	2	3	3	14
	Difference	-3	-2	-2	-2	-2	-11
	Rank	4	4	4	4	4	4



(a) WFG1 (b) WFG2 (c) WFG3

(d) WFG4 (e) WFG5 (f) WFG6

(g) WFG7 (h) WFG8 (i) WFG9

Fig. 11 MOO inverted generational distance (IGD) for 3-objective WFG problems

Table 8 Inverted Generational Distance Ranking - State of the Art MOEAs

Algorithm	**Result**	**2-objective WFG Function**									**Overall**
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	
MGPSO	Wins	3	4	4	0	4	4	4	1	3	27
	Losses	1	0	0	3	0	0	0	1	0	5
	Difference	2	4	4	-3	4	4	4	0	3	22
	Rank	2	**1**	**1**	4	**1**	**1**	**1**	2	**1**	**1**
NSGA II	Wins	4	2	2	4	2	3	2	4	2	25
	Losses	0	2	1	0	2	1	1	0	0	7
	Difference	4	0	1	4	0	2	1	4	2	18
	Rank	**1**	3	2	**1**	3	2	2	**1**	2	2
MOEA/D	Wins	1	2	2	0	3	0	2	1	0	11
	Losses	2	1	1	3	1	4	1	1	1	15
	Difference	-1	1	1	-3	2	-4	1	0	-1	-4
	Rank	3	2	2	4	2	5	2	2	3	3
SPEA2	Wins	1	1	1	2	0	1	0	0	0	6
	Losses	2	2	3	2	4	3	4	4	3	27
	Difference	-1	-1	-2	0	-4	-2	-4	-4	-3	-21
	Rank	3	4	4	3	5	4	5	5	5	5
PESA-II	Wins	0	0	0	3	1	2	1	1	1	9
	Losses	4	4	4	1	3	2	3	1	2	24
	Difference	-4	-4	-4	2	-2	0	-2	0	-1	-15
	Rank	5	5	5	2	4	3	4	2	3	4

Table 9 Inverted Generational Distance Ranking - State of the Art MOEAs

Algorithm	Result	3-objective WFG Function 1	2	3	4	5	6	7	8	9	Overall
MGPSO	Wins	3	4	4	4	4	4	4	3	4	34
	Losses	1	0	0	0	0	0	0	1	0	2
	Difference	2	4	4	4	4	4	4	2	4	32
	Rank	2	**1**	**1**	**1**	**1**	**1**	**1**	2	**1**	**1**
NSGA II	Wins	4	3	3	3	3	3	3	4	3	29
	Losses	0	1	1	1	1	1	1	0	1	7
	Difference	4	2	2	2	2	2	2	4	2	22
	Rank	**1**	2	2	2	2	2	2	**1**	2	2
MOEA/D	Wins	2	1	1	1	1	1	1	1	1	10
	Losses	2	3	3	3	3	3	3	3	3	26
	Difference	0	-2	-2	-2	-2	-2	-2	-2	-2	-16
	Rank	3	4	4	4	4	4	4	4	4	4
SPEA2	Wins	1	2	2	2	2	2	2	2	2	17
	Losses	3	2	2	2	2	2	2	2	2	19
	Difference	-2	0	0	0	0	0	0	0	0	-2
	Rank	4	3	3	3	3	3	3	3	3	3
PESA-II	Wins	0	0	0	0	0	0	0	0	0	0
	Losses	4	4	4	4	4	4	4	4	4	36
	Difference	-4	-4	-4	-4	-4	-4	-4	-4	-4	-36
	Rank	5	5	5	5	5	5	5	5	5	5

the algorithm being compared against by more than 90%. MGPSO only performed worse by more than 5% for 21, or less than 23%, of the comparisons. For five of the comparisons, neither algorithm outperformed the other by more than 5%.

With the exception of the comparison against NSGA II for the 3-objective WFG problems, the results overwhelmingly show that MGPSO outperformed the other MOO algorithms. For the 3-objective WFG problems, MGPSO outperformed NSGA II for only WFG3 and WFG7. However, the IGD statistical analysis and IGD graphs indicated that MGPSO outperformed NSGA II, at least, in terms of the IGD measure. On the other hand, the porcupine measure indicated that there are areas of the attainment surface where NSGA II outperformed MGPSO.

7 Conclusion

This article introduced the mutli-guided particle swarm optimization (MGPSO) algorithm, a multi-objective, multi-swarm particle swarm optimization (PSO) algorithm. MGPSO was compared against current multi-objective optimization (MOO) PSOs and *state of the art* MOEAs. Inverted generational distance (IGD) and the porcupine measure were used to evaluate MGPSO's performance.

The results indicate that MGPSO performed on-par and in some cases outperformed the algorithms against which it was compared. MGPSO was shown to perform well on both $2-$ and 3-objective problems using the ZDT and WFG test sets. Overall, MGPSO succeeded in solving, with the exception of ZDT4, all the MOO test problems evaluated in this study and was shown to be highly competitive.

Table 10 Porcupine Measure - State of the Art MOEAs

Problem	**Objectives**	**MGPSO vs** **NSGAII**	**MOEA/D**	**SPEA2**	**PESA2**
ZDT1	2	[**49.8**, 1.6]	[**76.4**, 3.6]	[**97.9**, 0.0]	[**86.1**, 0.0]
ZDT2	2	[**34.5**, 1.8]	[**67.0**, 7.4]	[**98.8**, 0.0]	[**66.8**, 0.0]
ZDT3	2	[0.6, **77.6**]	[**100.0**, 0.0]	[**44.3**, 1.2]	[29.0, 27.3]
ZDT4	2	[15.8, **75.3**]	[0.0, **82.9**]	[**45.3**, 29.7]	[32.9, **63.2**]
ZDT6	2	[**80.7**, 0.4]	[37.9, **44.1**]	[**99.7**, 0.1]	[**99.9**, 0.0]
WFG1	2	[17.3, **78.8**]	[**92.8**, 2.0]	[25.0, **69.7**]	[46.4, 50.9]
WFG2	2	[**99.2**, 0.0]	[**99.9**, 0.0]	[**99.9**, 0.0]	[**99.9**, 0.0]
WFG3	2	[**99.9**, 0.0]	[**90.7**, 0.0]	[**99.9**, 0.0]	[**99.9**, 0.0]
WFG4	2	[0.0, **99.9**]	[20.0, **28.5**]	[1.3, **97.4**]	[4.5, **92.7**]
WFG5	2	[**43.8**, 4.7]	[**42.1**, 36.2]	[**97.6**, 0.0]	[**89.5**, 0.0]
WFG6	2	[**99.9**, 0.0]	[**99.9**, 0.0]	[**99.9**, 0.0]	[**99.9**, 0.0]
WFG7	2	[**95.1**, 0.0]	[**81.5**, 2.9]	[**99.9**, 0.0]	[**99.9**, 0.0]
WFG8	2	[0.0, **99.7**]	[28.0, **45.6**]	[**35.6**, 7.3]	[17.3, **70.6**]
WFG9	2	[0.0, **7.0**]	[**99.7**, 0.0]	[**29.4**, 0.1]	[**9.0**, 0.0]
WFG1	3	[5.5, **89.8**]	[**86.9**, 9.2]	[**64.4**, 25.2]	[**68.7**, 25.5]
WFG2	3	[29.2, **50.8**]	[**60.0**, 20.7]	[28.5, **55.0**]	[**89.2**, 1.4]
WFG3	3	[**57.6**, 22.5]	[**84.2**, 15.0]	[**71.3**, 13.7]	[**56.7**, 31.1]
WFG4	3	[2.7, **78.3**]	[**64.8**, 21.7]	[**50.4**, 3.9]	[**90.7**, 5.7]
WFG5	3	[6.9, **55.7**]	[**50.9**, 38.8]	[**42.3**, 15.1]	[**90.7**, 3.6]
WFG6	3	[10.0, 13.7]	[**69.3**, 9.2]	[**93.9**, 0.0]	[**94.1**, 3.1]
WFG7	3	[**26.8**, 5.7]	[**71.3**, 12.4]	[**100.0**, 0.0]	[**90.4**, 6.5]
WFG8	3	[0.09, **82.5**]	[**69.6**, 22.0]	[37.8, 38.8]	[**94.4**, 2.5]
WFG9	3	[18.0, 16.0]	[**54.0**, 34.9]	[**40.5**, 7.0]	[**85.6**, 7.9]

A theoretical analysis of the convergence behavior and characteristics of MGPSO is currently underway. Variations of the MGPSO's λ variable and the influence this has on MGPSO's performance is currently being investigated. Variations of the archive term's selection are also being investigated.

A study of how MGPSO behaves in the within dynamic environments and the presence of constraints is currently underway.

Acknowledgements This work is based on the research supported by the National Research Foundation (NRF) of South Africa (Grant Number 46712). The opinions, findings and conclusions or recommendations expressed in this article is that of the author(s) alone, and not that of the NRF. The NRF accepts no liability whatsoever in this regard.

References

Coello Coello CA, Sierra MR (2004) A Study of the Parallelization of a Coevolutionary Multi-Objective Evolutionary Algorithm. In: Proceedings of the Third Mexican International Conference on Artificial Intelligence, pp 688–697, DOI 10.1007/978-3-540-24694-7

Corne D, Jerram N, Knowles JD, Oates MJ (2001) PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001), pp 283–290, DOI citeulike-article-id:8133801

Deb K (2001) Multi-Objective Optimization using Evolutionary Algorithms. Wiley, DOI 10.1109/TEVC.2002.804322

Deb K, Agrawal S, Pratap A, Meyarivan T (2000) A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. Parallel Prob-

lem Solving from Nature PPSN VI pp 849–858, DOI 10.1007/3-540-45356-3, URL http://repository.ias.ac.in/83498/
Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2):182–197, DOI 10.1109/4235.996017
Fonseca CM, Fleming PJ (1995) On the Performance Assessment and Comparison of Stochastic Multiobjective Optimisers. In: Parallel problem solving from nature - PPSN IV, vol 1141, pp 584–593
Franken CJ (2009) Visual Exploration of Algorithm Parameter Space. Proceedings of IEEE Congress on Evolutionary Computation pp 389–398, DOI 10.1109/CEC.2009.4982973
Gibbons JD, Chakraborti S (2010) Nonparametric statistical inference, 5th edn. Chapman and Hall/CRC Press
Grobler J (2009) Particle swarm optimization and differential evolution for multi objective multiple machine scheduling. Master's thesis, University of Pretoria
Grobler J, Engelbrecht AP (2009) Hybridizing PSO and DE for improved vector evaluated multi-objective optimization. Proceedings of IEEE Congress on Evolutionary Computation pp 1255–1262, DOI 10.1109/CEC.2009.4983089
Harrison KR, Ombuki-Berman BM, Engelbrecht AP (2013) Knowledge transfer strategies for vector evaluated particle swarm optimization. In: Proceedings of the 7th International Conference on Evolutionary Multi-Criterion Optimization, pp 171–184, DOI 10.1007/978-3-642-37140-0
Huband S, Barone L, While L, Hingston P (2005) A Scalable Multi-objective Test Problem Toolkit. In: Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization, pp 280–295, DOI 10.1007/978-3-540-31880-4
Huband S, Hingston P, Barone L, While L (2006) A review of multiobjective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation 10(5):477–506, DOI 10.1109/TEVC.2005.861417
Kennedy J, Eberhart RC (1995) Particle Swarm Optimization. In: Proceedings of the IEEE International Conference on Neural Networks, vol 4, pp 1942–1948, DOI 10.1109/ICNN.1995.488968
Knowles JD, Corne DW (2000) Approximating the nondominated front using the Pareto Archived Evolution Strategy. Evolutionary Computation 8(2):149–172, DOI 10.1162/106365600568167
Matthysen W, Engelbrecht AP, Malan KM (2013) Analysis of Stagnation Behavior of Vector Evaluated Particle Swarm Optimization. In: Proceedings of the IEEE Symposium on Swarm Intelligence, pp 155–163, DOI 10.1109/SIS.2013.6615173
Nebro A, Durillo J, García-Nieto J, Coello Coello CA, Luna F, Alba E (2009) SMPSO: A New PSO-based Metaheuristic for Multi-objective Optimization. In: Proceedings of the IEEE Symposium on Multi-Criteria Decision-Making, 2, pp 66–73, DOI 10.1109/MCDM.2009.4938830
Nebro AJ, Durillo JJ, Vergne M (2015) Redesigning the jMetal Multi-Objective Optimization Framework. Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation pp 1093–1100, DOI 10.1145/2739482.2768462, arXiv:1508.06655v1
Pampara G, Engelbrecht A, Cloete T (2008a) CIlib: A collaborative framework for Computational Intelligence algorithms-Part I. In: Proceedings of the IEEE International Joint Conference on Neural Networks, pp 1750–1757, DOI 10.1109/IJCNN.2008.4634035
Pampara G, Engelbrecht A, Cloete T (2008b) CIlib: A collaborative framework for Computational Intelligence algorithms-Part II. In: Proceedings of the IEEE International Joint Conference on Neural Networks, pp 1764–1773
Parsopoulos KE, Vrahatis MN (2002a) Particle swarm optimization method in multiobjective problems. In: Proceedings of the ACM Symposium on Applied Computing, pp 603–607, DOI 10.1145/508791.508907
Parsopoulos KE, Vrahatis MN (2002b) Recent approaches to global optimization problems through Particle Swarm Optimization. Natural Computing 1(2):235–306
Parsopoulos KE, Tasoulis DK, Pavlidis NG, Plagianakos VP, Vrahatis MN (2004) Vector Evaluated Differential Evolution for Multiobjective Optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation, vol 1, pp 204–211, DOI 10.1109/CEC.2004.1330858

Schaffer JD (1985) Multiple objective optimization with vector evaluated genetic algorithms. In: Proceedings of the First International Conference on Genetic Algorithms, JANUARY 1985, pp 93–100, URL http://dl.acm.org/citation.cfm?id=657079

Scheepers C (2018) Multi-guided Particle Swarm Optimization: A Multi-objective Particle Swarm Optimizer. Doctor's dissertation, University of Pretoria

Scheepers C, Engelbrecht AP (2016a) Vector Evaluated Particle Swarm Optimization Archive Management: Pareto Optimal Front Diversity Sensitivity Analysis. In: Proceedings of the IEEE Symposium Series on Computational Intelligence, DOI 10.1109/SSCI.2016.7850264

Scheepers C, Engelbrecht AP (2016b) Vector Evaluated Particle Swarm Optimization Exploration Behavior Part I: Explorative Analysis. In: Proceedings of the IEEE Congress on Evolutionary Computation

Scheepers C, Engelbrecht AP (2016c) Vector Evaluated Particle Swarm Optimization Exploration Behavior Part II: Quantitative Analysis. In: Proceedings of the IEEE Congress on Evolutionary Computation

Scheepers C, Engelbrecht AP (2017a) Comparing Performance of Multi-objective Algorithms using the Porcupine Measure. Technical Report pp 1–12

Scheepers C, Engelbrecht AP (2017b) Vector Evaluated Particle Swarm Optimization: The Archive's Influence on Performance. In: Proceedings of the IEEE Congress on Evolutionary Computation

Sierra MR, Coello Coello CA (2004) A New Multi-Objective Particle Swarm Optimizer with Improved Selection and Diversity Mechanisms. Tech. rep., Evolutionary Computation Group at CINVESTAV-IPN, México

Sierra MR, Coello Coello CA (2005) Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and -Dominance. In: Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization, vol 3410, Berlin, Heidelberg, pp 505–519, DOI 10.1007/978-3-540-31880-4

Zhang Q, Li H (2007) MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Transactions on Evolutionary Computation 11(6):712–731, DOI 10.1109/TEVC.2007.892759

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

Zitzler E, Laumanns M, Thiele L (2001) SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Tech. rep., Swiss Federal Institute of Technology (ETH) Zurich