Multi-modal multi-guide particle swarm optimisation for multi-objective optimisation

JW du Toit
Computer Science Department
Stellenbosch University
Stellenbosch, South Africa
22808892@sun.ac.za
22808892

***Abstract*—The multi-guide particle swarm optimisation (MGPSO) algorithm was developed to solve multi- and many-objective optimisation problems (MOPs) due to the inability of the standard inertia weight particle swarm optimisation (PSO) algorithm to solve such problems. The MGPSO algorithm uses a multi-swarm approach, where each swarm is used to optimise a separate objective function independent of the other objective functions. Current MGPSO implementations use a standard inertia weight PSO for each swarm. This report presents a new MGPSO-based algorithm named multi-modal multi-guide particle swarm optimisation (MMMGPSO). The MMMGPSO algorithm substitutes the standard inertia weight PSO swarms used in MGPSO with multi-modal particle swarm optimisation (MMPSO) swarms. The performance of the MGPSO and MMMGPSO algorithms are compared by evaluating the algorithms over five benchmark MOPs. The two algorithms are compared to determine whether or not the substitution of the standard inertia weight PSO swarms with MMPSO swarms in the MGPSO algorithm lead to improved performance for the MOPs. The empirical analysis indicates that the MMMGPSO algorithm outperforms the MGPSO algorithm for the benchmark MOPs due to the exploration capabilities of the MMPSO swarms used in the MMMGPSO algorithm.**

I. Introduction

Optimisation problems that consist of multiple objectives are called multi- and many-objective optimisation problems (MOPs). These problems do not provide a single solution to the optimisation problem, but rather provide a set of solutions with optimal trade-offs for the objectives of the problem. Many multi-objective optimisation algorithms have been developed to solve MOPs and the particle swarm optimisation (PSO) [6]-based algorithm called multi-guide particle swarm optimisation (MGPSO) [9] is an example of such an algorithm.

The MGPSO algorithm was developed to solve MOPs due to the incapability of the PSO algorithm to solve such problems. The MGPSO uses a separate swarm to optimise each objective function independent of the other objective functions. Current implementations of MGPSO use a standard inertia weight PSO [10] for each swarm. The objective of this report is to investigate the impact of using a multi-modal particle swarm optimisation (MMPSO) algorithm instead of the standard inertia weight PSO for each of the swarms. The reasoning behind this substitution is the exploration ability of MMPSO algorithms which may lead to improved performance for MOPs. This report presents a new MGPSO-based algorithm named multi-modal multi-guide particle swarm optimisation (MMMGPSO). The MMMGPSO algorithm substitutes the standard inertia weight PSO swarms used in the MGPSO with MMPSO swarms.

In order to derive an outcome for the objective of the report, the standard MGPSO algorithm with inertia weight PSO swarms is compared to the MMMGPSO algorithm. The two algorithms are evaluated over five benchmark MOPs from the Zitzler-Deb-Thiele (ZDT) and walking fish group (WFG) test sets for different performance metrics. The performance measures for the two algorithms are compared to establish whether or not the substitution of the inertia weight PSO with a MMPSO in each swarm improves performance.

The main observation obtained from this study is the improved performance of the MGPSO algorithm through the substitution of the inertia weight PSO swarms with MMPSO swarms. The proposed algorithm, MMMGPSO, outperformed the MGPSO algorithm across the five benchmark MOPs used in this report.

The rest of the report proceeds as follows: Section II provides a high-level discussion of MOPs, MGPSO and MMPSO. Section III provides an explanation of the proposed MMMGPSO algorithm and the approach taken to derive an outcome for the objective of the report. Section IV introduces the procedure followed to obtain empirical results. Section V discusses the results obtained from the experiments and Section VI draws a conclusion for the report.

II. Background

A. Multi-objective optimisation

MOPs are optimisation problems that consist of multiple objectives such that they do not provide a single solution to the optimisation problem, but rather provide a set of solutions with optimal trade-offs for the objectives of the problem. This set of solutions is referred to as the Pareto-optimal set (POS) and the corresponding set of objective vectors obtained from the POS is called the Pareto-optimal front (POF). Multi-objective optimisation algorithms attempt to find POFs for MOPs so that the solutions obtained are as close to the true POF as possible and are diverse along the POF.

Without loss of generality and under the assumption of minimisation, a multi-objective optimisation problem with n_k objectives is of the form:

$$\min(f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_{n_k}(\mathbf{x})) \tag{1}$$

with $\mathbf{x} \in \mathcal{F}, f_k : \mathbb{R}^{n_x} \to \mathbb{R}$ for all $k \in \{1, ..., n_k\}$, and $\mathcal{F} \subset \mathbb{R}^{n_x}$ is the feasible space as determined by the constraints where n_x is the dimension of the search space.

The following definitions are used throughout the report. **Definition 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 $f_k(\mathbf{x_1}) \leq f_k(\mathbf{x_2}) \ \forall \ k \in \{1, ..., n_k\}$ and $\exists \ k \in \{1, ..., n_k\}$ such that $f_k(\mathbf{x_1}) < f_k(\mathbf{x_2})$.

Definition 2 (*Pareto-optimal*) A decision vector $\mathbf{x}^* \in \mathcal{F}$ is Pareto-optimal if there does not exist a decision vector $\mathbf{x} \neq \mathbf{x}^* \in \mathcal{F}$ that dominates it.

Definition 3 (*Pareto-optimal set*) The set of all Pareto-optimal decision vectors form the Pareto-optimal set (POS) $\mathcal{P}^*$, where $\mathcal{P}^* = \{\mathbf{x}^* \in \mathcal{F} \,|\, \nexists \ \mathbf{x} \in \mathcal{F} : \mathbf{x} \prec \mathbf{x}^*\}$.

Definition 4 (*Pareto-optimal front*) Given the objective vector $\mathbf{f}(\mathbf{x})$ and the Pareto-optimal set $\mathcal{P}^*$, the Pareto-optimal front $\mathcal{PF}^* \subseteq \mathbb{R}^{n_k}$ is defined as $\mathcal{PF}^* = \{\mathbf{f} = (f_1(\mathbf{x}^*), f_2(\mathbf{x}^*), ..., f_k(\mathbf{x}^*)) \mid \mathbf{x}^* \in \mathcal{P}^*\}$.

B. Multi-guide particle swarm optimisation

MGPSO [9] is a PSO-based algorithm that was developed to solve MOPs due to standard PSO algorithms not being able to solve such problems. The MGPSO uses a multi-swarm approach, where a different swarm is assigned to each objective function in the MOP. The personal best and neighbourhood best positions within a swarm are evaluated and updated according to the objective function associated with that swarm.

The MGPSO uses an archive guide to enable an exchange of information between the swarms. The archive is bounded using a crowding distance [3]-based archive implementation. Once a new solution has been found and it is not dominated by any solution already in the archive, this new solution is added to the archive if the archive is not full. If the archive is full, a crowding distance measure is used to determine the most crowded solution in the archive, this solution will be removed from the archive and the new non-dominated solution is added. Solutions that become dominated by the newly added solution are removed. This ensures that no solution in the archive is dominated by another solution in the archive. The set of solutions contained in the archive after the execution of the algorithm forms the POS for the MOP which is used to generate the POF.

In order to obtain the archive guide $\hat{\mathbf{a}}_i(t)$, a competition pool is formed through a random selection of a certain number of solutions where the least crowded solution in the competition pool is chosen to be $\hat{\mathbf{a}}_i(t)$. The crowding distance measure used in the archive guide selection process ensures that MGPSO particles are guided towards more sparsely populated areas of the objective space which leads to a diverse spread of solutions in the POF.

The MGPSO velocity update equation is

$$\begin{aligned}\mathbf{v}_i(t+1) =& w\mathbf{v}_i(t) + c_1\mathbf{r}_1(\mathbf{y}_i(t) - \mathbf{x}_i(t)) + \lambda_i c_2\mathbf{r}_2(\hat{\mathbf{y}}_i(t) \\ & - \mathbf{x}_i(t)) + (1-\lambda_i)c_3\mathbf{r}_3(\hat{\mathbf{a}}_i(t) - \mathbf{x}_i(t))\end{aligned} \tag{2}$$

where $\mathbf{v}_i(t+1)$ is the velocity of particle i at iteration $t+1$, w is the inertia weight, c_1, c_2 and c_3 are the acceleration coefficients, $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$ are random vectors where components are 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 and $\hat{\mathbf{y}}_i(t)$ is the neighbourhood best position within a swarm. $\hat{\mathbf{a}}_i(t)$ is the archive guide selected from the archive for particle i at iteration t and λ_i is the exploitation trade-off coefficient for particle i. λ_i determines the influence that the archive guide has on the velocity of a particle, where larger values increase the influence of the neighbourhood guide and decrease the influence of the archive guide.

The MGPSO algorithm has certain stability conditions which have a strong link to the performance of the algorithm [9]. If these stability conditions are satisfied, the particles in a swarm are guaranteed to reach an equilibrium state, *i.e.*, the particles will stop moving. This implies that parameter tuning can be focused within the stability region to ensure good performance of the algorithm. The stability conditions are given by

$$\begin{aligned}&0 < c_1 + \lambda c_2 + (1-\lambda)c_3 < \\ &\frac{4(1-w^2)}{1 - w + \frac{(c_1{}^2 + \lambda^2 c_2{}^2 + (1-\lambda)^2 c_3{}^2)(1+w)}{3(c_1 + \lambda c_2 + (1-\lambda)c_3)^2}}, |w| < 1\end{aligned} \tag{3}$$

The pseudo code for the MGPSO algorithm is provided in Algorithm 1.

C. Multi-modal particle swarm optimisation

Multi-modal problems are problems which require more than one solution. The standard inertia weight PSO is not capable of efficiently providing multiple solutions to multi-modal problems. Algorithms such as NichePSO [1], LSEPSO [8] and speciation particle swarm optimisation (SPSO) [7] have been proposed to solve such problems. This report will use SPSO and thus the rest of this section describes the SPSO algorithm.

The SPSO partitions the particles of a swarm into groups based on the distance between the particles measured by the Euclidean distance between them. The Euclidean distance between two particle positions is given by

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \tag{4}$$

Algorithm 1 Multi-guide Particle Swarm Optimisation (MGPSO)

```
for each objective m = 1, ..., n_m do
  Let f_m be the objective function;
  Create and initialise a swarm, S_m, to contain S.n_{s_m} particles;
  for each particle i = 1, ..., S.n_{s_m} do
    Initialise position S_m.x_i(0) uniformly within a predefined hypercube of dimension n_x;
    Initialise the personal best position as S_m.y_i(0) = S_m.x_i(0);
    Determine the neighbourhood best position, S_m.ŷ_i(0);
    Initialise the velocity as S_m.v_i(0) = 0;
    Initialise S_m.λ_i ~ U(0,1);
  end for
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.x_i(t)) < f_m(S_m.y_i(t)) then
        S_m.y_i(t) = S_m.x_i(t);
      end if
      for particles î with particle i in their neighbourhood do
        if f_m(S_m.y_i(t)) < f_m(S_m.ŷ_î(t)) then
          S_m.ŷ_î(t) = S_m.y_i(t);
        end if
      end for
      Update the archive with the solution S_m.x_i(t);
    end for
  end for
  for each objective m = 1, ..., n_m do
    for each particle i = 1, ..., S_m.n_s do
      Select a solution S_m.â_i(t) from the archive using tournament selection;
      S_m.v_i(t+1) = w S_m v_i(t) + c_1 r_1 (S_m.y_i(t) − S_m.x_i(t)) + S_m.λ_i c_2 r_2 (S_m.ŷ_i(t) − S_m x_i(t)) + (1 − S_m.λ_i) c_3 r_3 (S_m.â_i(t) − S_m x_i(t))
      S_m.x_i(t+1) = S_m.x_i(t) + S_m.v_i(t+1)
    end for
  end for
  t = t + 1
until stopping condition is true
```

where $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{in})$ and $\mathbf{x}_j = (x_{j1}, x_{j2}, ..., x_{jn})$ are vectors of real numbers that represent two particles.

SPSO introduces a new control parameter, r_s which denotes the radius measured in Euclidean distance from the species seed to the boundary of the species. The species seed is the particle with the best fitness in the species. Thus, all particles that are within the distance r_s of the species seed, form a species. The pseudo code for determining species seeds is provided in Algorithm 2.

The SPSO algorithm can be summarised in the following steps:

1. Generate an initial population with randomly generated particles;
2. Evaluate all particle individuals in the population;
3. Sort all particles in descending order of their fitness;
4. Determine the species seeds for the current population according to Algorithm 2;
5. Assign each species seed identified as the neighbourhood best to all individuals identified in the same species;
6. Adjust velocity and position of each particle;
7. If termination criterion is not met, goto step 2;

Algorithm 2 Algorithm for determining species seeds

```
Let L_sorted = all particles sorted in decreasing order of fitness;
Let S_seed = {};
while L_sorted contains unprocessed particles do
  Get the first unprocessed particle x_L in L_sorted;
  Let found = FALSE;
  for each particle x_S in S_seed do
    if d(x_S, x_L) ≤ r_s then
      found = TRUE;
      break;
    end if
  end for
  if (not found) then
    Let S_seed = S_seed ∪ {x_L};
  end if
  Mark x_L as processed
end while
```

III. METHODOLOGY

The objective of this report is to determine whether the MGPSO algorithm can be improved through the substitution of the standard inertia weight PSO with a MMPSO in the different swarms. The reasoning behind this investigation is the idea that the MMPSO algorithm should provide a higher level of exploration compared to the standard inertia weight PSO algorithm which leads to improved performance for the MOP. A new MGPSO-based multi-objective optimisation algorithm, named MMMGPSO, is proposed which uses a SPSO in each of the swarms as opposed to the inertia weight PSO swarms used in the MGPSO algorithm. The pseudo code for the MMMGPSO algorithm is provided in Algorithm 3.

The initialisation of the particles, maintenance of the archive, selection of the archive guide and the velocity and position updates of the MMMGPSO algorithm is identical to that of the MGPSO algorithm described in Section II. The POS generated by the MMMGPSO algorithm is the set of solutions contained in the archive after the algorithm has been executed. The main difference between the algorithms is the manner in which the neighbourhood best positions are chosen. Unlike the MGPSO algorithm that assigns the global best position of a swarm to the neighbourhood best position of each particle in the associated swarm, the MMMGPSO algorithm performs the procedure of the SPSO algorithm to obtain the neighbourhood best positions used in the velocity update calculation.

For each iteration of the algorithm the following procedure is followed to update the neighbourhood best positions for a given swarm:
1. All particles are evaluated and personal best positions are updated.
2. The particles are sorted in decreasing order of fitness.
3. The speciation process is applied where particles are grouped into species if they are within a certain distance of the species seed.
4. The species seed becomes the neighbourhood best for all of the particles in that species.

This procedure followed for the allocation of neighbourhood bests allows particles to potentially explore larger areas of the search space compared to the standard inertia weight PSO swarms used in the MGPSO algorithm. This is due to the fact that the different species in a swarm explore and exploit different areas of the search space as opposed to the MGPSO swarms which are all drawn to the global best of the associated swarm. However, the exploration ability and performance of the SPSO swarms is highly dependent on the control parameter r_s.

The value of r_s influences the performance of the MMMG-PSO algorithm as it determines the radius size for each species in the swarm. The value of r_s is chosen to be a fraction of the Euclidean distance between the vectors representing the lower and upper bounds of the search space. The value of r_s value is determined by

$$r_s = z\sqrt{\sum_{k=1}^{n}(x_k^u - x_k^l)^2} \tag{5}$$

where $x_1^l, x_2^l, ...x_n^l$ and $x_1^u, x_2^u, ...x_n^u$ are the lower and upper bounds of the search space respectively and z is the parameter used to control the size of r_s such that $0 < z < 1$. The performance of the MMMGPSO algorithm is investigated under different values of z to establish at which values of r_s the algorithm performs best.

In order to determine whether the MMMGPSO algorithm can improve the standard MGPSO algorithm, the two algorithms are compared over five benchmark MOPs from the ZDT and WFG test sets. The problems in these test sets provide different MOP characteristics which prevent deriving conclusions from problem dependent results and provide a general performance profile for each of the algorithms. These algorithms are measured with respect to two standard performance metrics for MOPs, namely inverted generational distance (IGD) and hypervolume. The IGD [2] value measures the Euclidean distance that a set of solutions are to the true POF of the given problem. The IGD thus provides a measure of the performance of an algorithm with respect to the true POF for a given MOP, where lower values imply better performance due to the POF obtained being closer to the true POF. The hypervolume [4] value measures the area dominated by the POF with respect to a reference point. The hypervolume thus provides a measure of the performance of an algorithm, where larger values imply better performance due the POF having lower values which leads to a greater dominated area. These measures are used along with a visualisation of the POFs generated by the two algorithms to determine which algorithm performs best.

IV. Empirical Procedure

This section describes the procedure that was followed to obtain the empirical results for this report.

Both the MGPSO and the MMMGPSO algorithms were evaluated over five benchmark MOPs. The problems included the ZDT1, ZDT2, ZDT3, WFG1 and WFG3 functions from the ZDT [11] and WFG [5] test sets respectively. The properties of these benchmark MOPs are given in Table I, where N is the number of objective functions.

For each benchmark MOP the two algorithms were compared with respect to two performance measures, namely IGD and hypervolume. The IGD [2] value measures the Euclidean distance that a set of solutions are to the true POF of the given problem. The hypervolume [4] measures the area dominated by the provided set of solutions with respect to a reference point. Reference points of [5, 5] and [10, 10, 10] were used for the calculation of the hypervolume for the ZDT and WFG problems respectively. These performance measures were taken over time to generate a performance profile for the algorithm with respect to the associated performance metric.

In order to obtain the performance profiles for a specified algorithm, the following procedure was performed:
1. The algorithm is executed for 2000 iterations where at each iteration the results of the two performance measures are collected.
2. Step 1 is executed 20 times and the average and standard deviation of the values over the 20 independent runs is obtained at each iteration.
3. The average at each iteration is plotted on a graph to indicate how the algorithm performed over time according to the associated performance measure.
4. The standard deviation is added to and subtracted from the average at each iteration and these points are plotted to indicate the standard deviation of the associated measures over time.
5. The POF generated by each algorithm after 2000 iterations is plotted against the true POF of the given MOP in order to visualise the performance of the algorithm.
6. The steps above are executed for different values of z to determine which r_s values lead to improved performance for the MMMGPSO algorithm. The values of z used include $1, 0.75, 0.5, 0.4, 0.3, 0.2, 0.1, 0.075$ and 0.05.

For each of the five benchmark MOPs the procedure mentioned above was executed with both algorithms.

The strategy used to obtain the control parameter values was done according to the following procedure: For each particle,

Algorithm 3 Multi-modal Multi-guide Particle Swarm Optimisation (MMMGPSO)

```
for each objective $m = 1, ..., n_m$ do
  Let $f_m$ be the objective function;
  Create and initialise a swarm, $S_m$, to contain $S.n_{s_m}$ particles;
  for each particle $i = 1, ..., S.n_{s_m}$ do
    Initialise position $S_m.\mathbf{x}_i(0)$ uniformly within a predefined hypercube of dimension $n_x$;
    Initialise the personal best position as $S_m.\mathbf{y}_i(0) = S_m.\mathbf{x}_i(0)$;
    Determine the neighbourhood best position, $S_m.\hat{\mathbf{y}}_i(0)$;
    Initialise the velocity as $S_m.\mathbf{v}_i(0) = \mathbf{0}$;
    Initialise $S_m.\lambda_i \sim U(0, 1)$;
  end for
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 $f_m(S_m.\mathbf{x}_i(t)) < f_m(S_m.\mathbf{y}_i(t))$ then
        $S_m.\mathbf{y}_i(t) = S_m.\mathbf{x}_i(t)$;
      end if
    end for
  end for
  for each objective $m = 1, ..., n_m$ do
    Let $L_{sorted}$ = all particles in $S_m$ sorted in decreasing order of fitness;
    Let $S_{seed} = \{\}$;
    for each particle $i = 1, ..., S_m.n_s$ do
      Let $found$ = FALSE;
      for each particle $\mathbf{x}_S$ in $S_{seed}$ do
        if $d(\mathbf{x}_S, L_{sorted}.\mathbf{x}_i(t)) \leq r_s$ then
          $L_{sorted}.\hat{\mathbf{y}}_i(t) = \mathbf{x}_S$
          $found$ = TRUE;
          break;
        end if
      end for
      if ($not\ found$) then
        Let $S_{seed}$ = $S_{seed} \cup \{\mathbf{x}_L\}$;
      end if
    end while
    Update the archive with the solution $S_m.\mathbf{x}_i(t)$;
  end for
  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
```

TABLE I
PROPERTIES OF THE ZDT AND WFG PROBLEMS

Problem	N	Separability	Modality	Geometry
ZDT1	2	Separable	Unimodal	Convex
ZDT2	2	Separable	Unimodal	Concave
ZDT3	2	Separable	Unimodal	Disconnected
WFG1	3	Separable	Unimodal	Convex, mixed
WFG3	3	Non-separable	Unimodal	Linear, degenerate

TABLE II
DEFAULT PARAMETERS

Problem	$\lvert S_1 \rvert$	$\lvert S_2 \rvert$	$\lvert S_3 \rvert$	w	c_1	c_2	c_3
ZDT1	33	17		0.475	1.80	1.10	1.80
ZDT2	8	42		0.075	1.60	1.35	1.90
ZDT3	33	17		0.050	1.85	1.90	1.90
WFG1	37	4	9	0.125	1.2	1.3	1.75
WFG3	29	10	11	0.525	1.65	1.75	0.75

new control parameter values are assigned at each iteration. The values of c_1, c_2, c_3 are sampled uniformly from $(0, 2)$ and the value of w sampled uniformly from $(0, 1)$. These generated values are used along with the λ value associated with the particle to test if the stability conditions are satisfied. The stability conditions used for the experiments of both the MGPSO and MMMGPSO are given by (3). If the stability conditions are satisfied, these control parameter values are used. If the conditions are not satisfied, new values are sampled and tested again. If after 10 failed attempts the parameter values still do not satisfy the stability conditions, a default selection of values are assigned.

The default control parameter values and the partition of the number of particles for each swarm used for both algorithms are given in Table II. These control parameter values and partition of the number of particles are optimal for the MGPSO algorithm [9]. A competition pool size of three was used for all of the experiments.

V. RESEARCH RESULTS

This section presents the empirical results obtained from Section IV and provides a discussion on the performance of the standard MGPSO algorithm compared to the MMMGPSO algorithm. Due to the new control parameter r_s introduced for the MMMGPSO algorithm, different values of r_s were compared to see which values lead to improved performance. A selection of POFs obtained from the MGPSO and MMMGPSO algorithms were visually compared to the true POF of the associated problem. The performance of the two algorithms with respect to IGD values over time were compared. Finally, the two algorithms were compared over time with respect to the hypervolume performance measure.

A. Radius tuning

The MMMGPSO algorithm introduces a new control parameter, r_s, which controls the radius of a species such that particles join a species if they are within the specified radius from the species seed. The parameter r_s thus has an influence

TABLE III
MMMGPSO PERFORMANCE AT ITERATION 2000 FOR ZDT1 PROBLEM FOR DIFFERENT z VALUES

z	IGD	Hypervolume
1	0.69	20.44
0.75	0.67	20.57
0.5	0.69	20.47
0.4	0.65	20.71
0.3	0.61	21.01
0.2	0.60	21.87
0.1	1.64	14.86
0.075	1.6	15.05
0.05	1.58	15.16

on the performance of the MMMGPSO algorithm and this section investigates the performance for different values of z which directly influences the radius size.

Fig. 1 shows the average IGD values obtained from the MMMGPSO algorithm with a z value 0.5 for the ZDT1 MOP over 2000 iterations. From Fig. 1 it was observed that the IGD values decreased as the solutions in the POF improved up to iteration 2000. Fig. 1 is representative of the trend observed for IGD values obtained for different values of z and all benchmark MOPs used in this report.

The average hypervolume values obtained over 2000 iterations by the MMMGPSO algorithm with a z value 0.5 for the ZDT1 MOP problem are given in Fig. 2. From Fig. 2 it was observed that the hypervolume increased steadily over time up to iteration 2000 as the POF improved. Fig. 2 is representative of the trend observed for the hypervolume performance for different values of z and all benchmark MOPs used in this report.

Table III shows the performance of the MMMGPSO for different values of z for the ZDT1 MOP. From Table III it was observed that as the value of z decreased from a value of 1 to 0.2 the performance of the MMMGPSO algorithm improved. This observation can be seen by the decreasing IGD values and the increasing hypervolume values as z decreases to 0.2. For z values smaller than 0.2 the performance of MMMGPSO significantly decreased. Through the investigation of different z values for the other benchmark MOPs, it was observed that although the performance for different z values is problem dependent, a z value of 0.2 consistently obtained good results and thus the remainder of the experiments were performed with a z value of 0.2. It was observed that a z value of 1, which implies that the SPSO performs in a similar fashion to a standard global best PSO, was outperformed by smaller values of z up to a certain point. This observation indicates the performance improvements that the MMMGPSO with a good r_s value can obtain. A z value of 1 implies that a SPSO swarm functions as a global best PSO because the radius will include the entire search space, but the expensive sorting calculations will lead to worse performance than a standard global best PSO swarm.

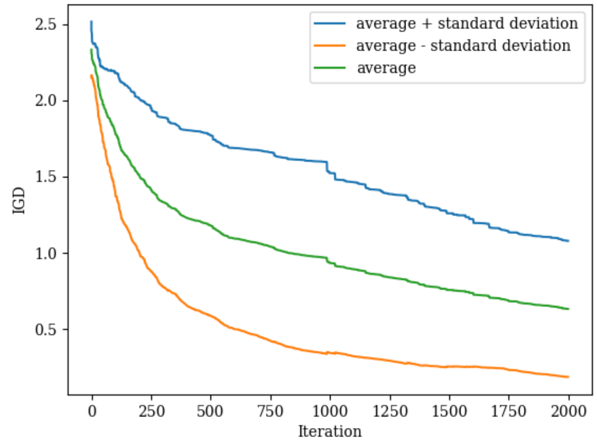

Fig. 1. MMMGPSO with $z = 0.5$ IGD over time for ZDT1

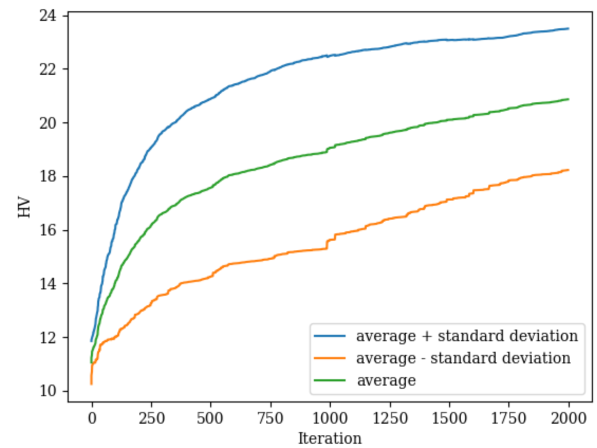

Fig. 2. MMMGPSO with $z = 0.5$ hypervolume over time for ZDT1

B. Pareto-optimal front visualisation

The POFs for the two algorithms were compared visually to the true POF for the benchmark MOPs in order to gauge the performance of the two algorithms. A representative POF for the MMMGPSO compared to the true POF for the ZDT1 benchmark MOP is given by Fig. 3. Similarly, Fig. 4 and Fig. 5 provide examples of the POFs generated by the MMMGPSO algorithm for the ZDT3 and WFG3 problems respectively. It was observed that the POFs obtained from the MGPSO algorithm for the ZDT1, ZDT3 and WFG3 problems followed the same shape and general range of solutions as shown in Fig. 3, Fig. 4 and Fig. 5 respectively. Fig. 5 indicated that the MMMGPSO algorithm was able to generate a POF that was close to the true POF after only 2000 iterations for the WFG3 problem.

Even though the POFs generated by the MGPSO and MM-MGPSO after 2000 iterations were worse than the true POF

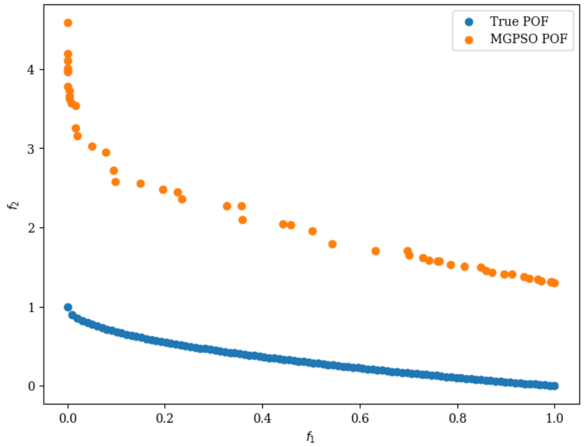

Fig. 3. MMMGPSO vs true POF for ZDT1

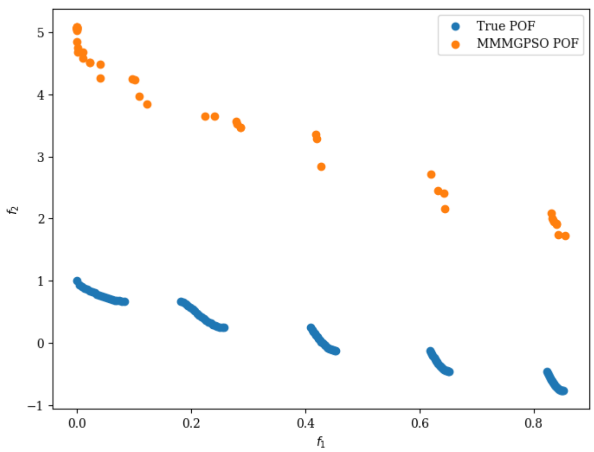

Fig. 4. MMMGPSO vs true POF for ZDT3

for the associated benchmark POF, they portrayed a similar shape to the true POFs. Furthermore, the observation that the MMMGPSO generates similar POF shapes to the MGPSO and the true POF indicates that the algorithm performed as expected and that the non-dominance relation for the archive is maintained.

C. Inverted generational distance comparison

The IGD values obtained over time by the two algorithms for the five benchmark MOPs are compared in this section. The IGD value provides a performance measure for the algorithm with respect to the true POF of the given MOP, where smaller IGD values imply better performance. Fig. 6 shows the IGD values obtained by the MGPSO for the ZDT2 MOP over 2000 iterations. From Fig. 6 it was observed that the IGD values steadily decreased as the solutions in the POF improved up to iteration 2000. Fig. 6 is representative of the trend shown for IGD values obtained from the MGPSO and MMMGPSO algorithms over time for each of the benchmark MOPs used in this report.

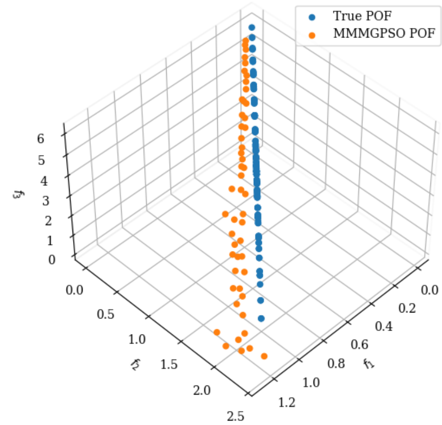

Fig. 5. MMMGPSO vs true POF for WFG3

TABLE IV
IGD AT ITERATION 2000 FOR ZDT2 PROBLEM

Algorithm	Average	Standard Deviation
MGPSO	0.66	0.18
MMMGPSO	0.41	0.14

Table IV provides the average IGD and the standard deviation obtained by the MGPSO and MMMGPSO algorithms for the ZDT2 problem at iteration 2000. From Table IV it was observed that the MMMGPSO algorithm outperformed the MGPSO algorithm after 2000 iterations with respect to IGD values due to the smaller value of 0.41 obtained by the MMMGPSO algorithm compared to the value of 0.66 obtained by the MGPSO algorithm. Furthermore, Table IV indicated that the MMMGPSO algorithm obtained a slightly lower standard deviation compared to the MGPSO with respect to the IGD values which implies that the MMMGPSO provides more stable results.

This observation that the MMMGPSO algorithm improved the quality of the IGD values compared to the MGPSO algorithm was seen throughout the different benchmark MOPs. The performance improvement with respect to the IGD value for the MMMGPSO algorithm can be attributed to the exploration capabilities of the SPSO swarms used in this algorithm. However, it should be noted that the performance difference between the two algorithms were not significant for some problems and that the performance of the MMMGPSO algorithm is dependent on the value of r_s.

D. Hypervolume comparison

The hypervolume values obtained over time by the two algorithms for the five benchmark MOPs are compared in this section.The hypervolume measures the area dominated by the provided set of solutions with respect to a reference point. The hypervolume thus provides a measure for the quality of the POF, where better POFs lead to larger hypervolume values and worse POFs lead to smaller hypervolume values. The hypervolume over time obtained by the MMMGPSO

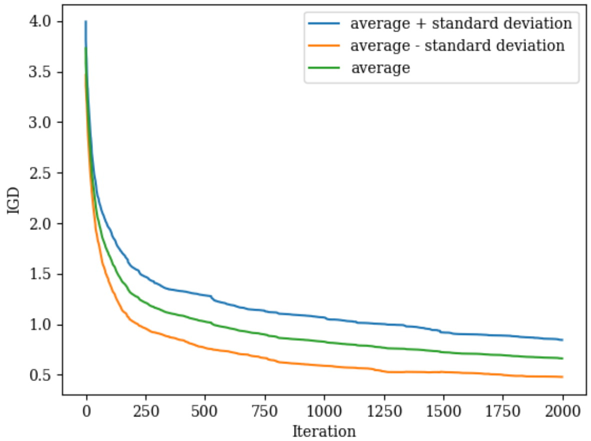

Fig. 6. MGPSO IGD over time for ZDT2

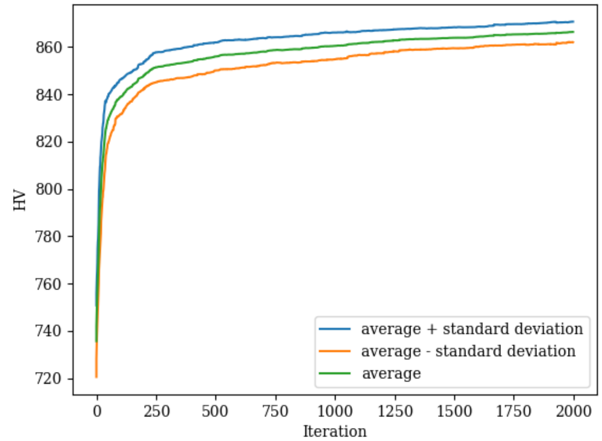

Fig. 7. MMMGPSO hypervolume over time for WFG3

TABLE V
HYPERVOLUME AT ITERATION 2000 FOR WFG3 PROBLEM

Algorithm	Average	Standard Deviation
MGPSO	862.06	6.03
MMMGPSO	866.35	4.43

algorithm for the WFG3 MOP is given in Fig. 7. Fig. 7 is representative of the trend of the hypervolume performance for the MMMGPSO and MGPSO algorithms for the five benchmark MOPs. From Fig. 7 it was observed that the hypervolume steadily increases over time up to iteration 2000 as the POF improves.

Table V provides the average hypervolume and the standard deviation obtained by the MGPSO and MMMGPSO algorithms at iteration 2000 for the WFG3 problem. From Table V it was observed that the MMMGPSO algorithm outperformed the MGPSO after 2000 iterations with respect to hypervolume due to the larger value of 866.35 obtained by the MMMGPSO compared to the value of 862.06 obtained by the MGPSO. Furthermore, table V indicated that the MMMGPSO algorithm obtained a slightly lower standard deviation compared to the MGPSO for the hypervolume values which indicated better stability for the MMMGPSO.

It was observed throughout the different benchmark MOPs that the MMMGPSO algorithm improved the quality of the hypervolume values compared to the MGPSO algorithm. This performance improvement with respect to the hypervolume value for the MMMGPSO algorithm can be attributed to the exploration capabilities of the SPSO swarms used in this algorithm compared to the standard inertia weight PSO swarms used in the MGPSO algorithm. However, it should be noted that the difference in performance between the two algorithms were not significant for some problems and that the performance of the MMMGPSO algorithm is dependent on the value of the parameter r_s.

VI. CONCLUSION

The objective of this report is to determine whether the multi-guide particle swarm optimisation (MGPSO) algorithm can be improved through the substitution of the standard inertia weight particle swarm optimisation (PSO) with a multi-modal particle swarm optimisation (MMPSO) in the different swarms. In order to derive an outcome for this objective, a new MGPSO-based algorithm, named multi-modal multi-guide particle swarm optimisation (MMMGPSO) is proposed. MMMGPSO uses a speciation particle swarm optimisation (SPSO) algorithm in each swarm as opposed to the inertia weight PSO used in MGPSO. The performance of these algorithms are compared across five benchmark multi-objective optimisation problems (MOPs) to establish whether the use of SPSO swarms improves the performance of the MGPSO algorithm.

From the results discussed in Section V it is derived that the proposed MMMGPSO algorithm outperforms the MGPSO with respect to two standard MOP performance metrics across five benchmark MOPs. The MMMGPSO provides improved explorative ability compared to the MGPSO which leads to the generation of pareto-optimal fronts (POFs) that are closer to the true POF.

This report can be extended through additional experiments using more benchmark MOPs. Furthermore, the MMMGPSO can be modified to use a different MMPSO algorithm other than the SPSO algorithm in each of the swarms.

REFERENCES

[1] T. Crane, B. Ombuki-Berman and A. Engelbrecht, "NichePSO and the Merging Subswarm Problem," 2020 7th International Conference on Soft Computing Machine Intelligence (ISCMI), 2020, pp. 17-22, 2020.

[2] C. Coello Coello and M. Sierra, "A study of the parallelization of a coevolutionary multi- objective evolutionary algorithm", In Proceedings of the 3rd Mexican international conference on artificial intelligence, pp. 688–697, 2004.

[3] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," in IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197, 2002.

[4] A. Guerreiro, C. Fonseca and L. Paquete, ”The Hypervolume Indicator”, ACM Computing Surveys, vol. 54, no. 6, pp. 1-42, 2020.

[5] S. Huband, P. Hingston, L. Barone and L. While, ”A review of multiobjective test problems and a scalable test problem toolkit”, IEEE Transactions on Evolutionary Computation, vol. 10, no. 5, pp. 477-506, 2006.

[6] J. Kennedy and R. Eberhart, ”Particle swarm optimization”, Proceedings of ICNN'95 - International Conference on Neural Networks, 1995, vol. 4, pp. 1942-1948, 1995.

[7] X. Li, ”Adaptively Choosing Neighbourhood Bests Using Species in a Particle Swarm Optimizer for Multimodal Function Optimization”, GECCO 2004, pp. 105–116, 2004.

[8] T. Rahkar-Farshi, S. Behjat-Jamal and M. Derakhshi, ”An Improved Multimodal PSO Method Based on Electrostatic Interaction using N-Nearest-Neighbor Local Search”, International Journal of Artificial Intelligence amp; Applications, vol. 5, no. 5, pp. 75-84, 2014.

[9] C. Scheepers, A. Engelbrecht and C. Cleghorn, ”Multi-guide particle swarm optimization for multi-objective optimization: empirical and stability analysis”, Swarm Intelligence, vol. 13, no. 3-4, pp. 245-276, 2019.

[10] Y. Shi and R. Eberhart, ”A modified particle swarm optimizer,” 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), 1998, pp. 69-73. 1998.

[11] E. Zitzler, K. Deb and L. Thiele, ”Comparison of Multiobjective Evolutionary Algorithms: Empirical Results”, Evolutionary Computation, vol. 8, no. 2, pp. 173-195, 2000.