Multi-Guide Particle Swarm Optimization: Evaluating The Success of Using a Multi-Modal Algorithm Compared to Inertia Weight

*A Comparison of Multi-Guide Particle Swarm Optimization techniques

JC De Witt
Stellenbosch University
(Department of Computer Science)
Stellenbosch, South Africa
21663904@sun.ac.za

***Abstract*—This paper provides a comparison between two multi-objective particle swarm optimization algorithms for solving optimization problems in which a single optimal solution does not exist. Multi-guide particle swarm optimization, and a speciation based extension are both fit for solving this class of optimization problem. To determine which is more effective The algorithms are compared using a series of quality assessment mechanisms. The results suggest that there is not a statistically significant difference in performance between the two algorithms. However,when comparing the single best Pareto-optimal front obtained throughout the simulation procedure, a species based multi modal optimization algorithm is most effective.**

***Keywords*—particle swarm optimization, multi-guide particle swarm optimization, multi-objective optimization problems, Pareto-optimal front**

I. Introduction

It is not uncommon for real-world optimization problems to require a trade off between multiple conflicting objectives. Such problems are referred to as multi-objective optimization problems (MOPs). Due to the conflicting nature of MOPs, a set of optimal trade-offs with respect to the provided objectives is obtained, rather than a single solution. This set of solutions is referred to as the Pareto-optimal set (POS). The objective vectors corresponding to the set of Pareto-optimal solutions form the Pareto-optimal front (POF). Multi-objective optimization algorithms have been developed to find solutions in the search space such that

1) the solutions approximate the true POF
2) the diversity of solutions along the obtained POF are as diverse as possible.

Multi-guide Particle swarm optimization (MGPSO) as proposed by Scheepers *et al* [1] is a multi-objective variant of traditional particle swarm optimization [2] designed for MOPs. MGPSO is a multi-swarm adaptation where each objective function is optimized using a separate sub-swarm. To facilitate optimal trade-off balance between conflicting objectives, an archive guide is added to the velocity update equation. The archive guide is a randomly chosen non-dominated candidate solution that acts as an attractor to particles in each sub-swarm. Traditionally each sub-swarm performs velocity updates according to the standard inertia-weight technique proposed by Shi and Eberhart [3]. Various PSO techniques have been developed since that specialize their performance on certain search space characteristics. Li [4] proposes a speciation particle swarm optimization (SPSO) technique that specializes in multi-modal search environments. This paper intends to provide an empirical evaluation on the performance of MGPSO when the underlying PSO technique is changed to a multi-modal optimization algorithm. An empirical procedure is established to compare the two MGPSO techniques, in order to determine if there is incentive to optimize the underlying PSO algorithm based on the search space characteristics. This is accomplished by comparing the performance of both algorithms across various independent simulations. Multiple quality measurement techniques are employed to evaluate the performance of MGPSO compared to multi-modal MGPSO. It was found that under certain conditions the multi-modal variation of MGPSO obtained POFs of superior quality. A non-parametric statistical test used to compare the two algorithms found that there is not a statistically significant improvement in using a multi-modal variation of MGPSO.

The remainder of this paper is organized as follows: Section II supplies an overview of MGPSO and the influence each control parameters imposes on the swarms behaviour. Section III outlines the evaluation metrics used to analyze the quality of PSO performance. Section IV discloses the benchmark functions and experimental procedure followed to obtain results. Subsequently, the results obtained are presented in Section V. Lastly, section VI concludes the paper.

II. Background

A minimization orientated multi-objective optimization problem (MOP) with n_m objectives is of the form:

$$min(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_{n_m}((\mathbf{x})) \quad (1)$$

with $\mathbf{x} \in \mathcal{F}, f_m: \mathbb{R}^{n_m} \rightarrow \mathbb{R}$ for all $m \in$ 1, ..., n_m, where the feasible space determined by the constraints is $\mathcal{F} \subset \mathbb{R}^{n_m}$ with search space dimension n_x.

The following definitions [1] are referenced frequently throughout this paper:

- **Definition 1** (*Domination*) A decision vector $\mathbf{x_1} \in \mathcal{F}$ dominates a decision vector $\mathbf{x_2}$ (denoted $\mathbf{x}_1 \prec \mathbf{x}_2$) if and only if $f_m(\mathbf{x_1}) \leq f_m(\mathbf{x_2})$ for all $m \in 1, \ldots, n_m$ and there exists $m \in 1, \ldots, n_m$ such that $f_m(\mathbf{x}_1) < f_m(\mathbf{x}_2)$
- **Definition 2** (*Pareto optimal*) A decision vector $\mathbf{x}_1 \in \mathcal{F}$ is said to be Pareto optimal if no other decision vector $\mathbf{x}_2 \in \mathcal{F}$ exists such that $\mathbf{x}_1 \prec \mathbf{x}_2$
- **Definition 3** (*Pareto-optimal set*) The set $P \subseteq \mathbb{R}^{n_x}$ where $P = \mathbf{x}_1 \in \mathcal{F} \mid \nexists\ \mathbf{x}_1 \in \mathcal{F}$ such that $\mathbf{x}_2 \prec \mathbf{x}_1$, is referred to as the Pareto-optimal set.
- **Definition 4** (*Pareto-optimal front*) The set $Q \subseteq \mathbb{R}^{n_x}$ where $Q = \mathbf{f} = (f_1(\mathbf{x}^*),\ f_1(\mathbf{x}^*),\ \ldots,\ f_{n_m}(\mathbf{x}^*)),\ \forall \mathbf{x}^* \in P$, is referred to as the Pareto-optimal front (POF).

A. MGPSO Velocity Update Equation

In contrast to single-objective inertia weight PSO [3], MGPSO provides an expanded velocity update equation where an archive guide is added to facilitate communication between locally exploiting sub-swarms. The MGPSO velocity update equation using inertia-weighting as the underlying technique is formally defined as [1]

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

The notation of the velocity update equation is clarified below.

- $\mathbf{x}_i(t)$ and $\mathbf{x}_i(t+1)$ specify the position of the i^{th} particle during iteration t and $t+1$ respectively. The position of a particle represents a potential solution to the optimization problem.
- $\mathbf{v}_i(t)$ and $\mathbf{v}_i(t+1)$ denote the velocity of the i^{th} particle during iteration t and $t+1$ respectively. It is important to note that the velocity vector of a particle specifies both the direction this particle is moving, as well as the magnitude of the movement. The velocity of all particles are initialized to the zero vector at iteration one.
- $\hat{\mathbf{y}}_i(t)$ denotes the position which yields the best objective function evaluation that any particle in the neighborhood has visited. This point can be interpreted as the "best personal best" of particles in this sub-swarm.
- $\mathbf{r_1}$, $\mathbf{r_2}$ and $\mathbf{r_3}$ are n dimensional random vectors, where each element in the vector is chosen from a *uniform(0,1)* distribution. The non-deterministic characteristics of these vectors are critical in facilitating exploration of the search space.
- c_1, c_2 and c_3 are control coefficients which dictate the influence of the cognitive, social, and archive components respectively. These parameters are sampled such that derived [1] stability conditions are satisfied.
- λ_i controls the influence that the archive guide has on the particle's velocity. Small values for λ_i decrease the influence of the archive guide while simultaneously decreasing the influence of the neighborhood guide.
- $\hat{\mathbf{a}}_i(t)$ is the most crowded candidate solution participating in the tournament selection procedure. The competition pool is established by randomly selecting a predefined number of candidate solutions from the archive. The tournament size has direct influence on the selection pressure [5]. Empirical procedure has indicated that a tournament size of 2 or 3 yields good results [6].

B. The Archive

Scheepers and Engelbrecht [8] [9] found that a bounded-archive using crowding distance [10] to replace crowded candidate solutions yields desirable performance characteristics. The archive is a collection of non-dominated objective function evaluations obtained by particles in the sub-swarms. If a new solution is added to the archive, contained solutions that are dominated by the newfound candidate solution are removed from the archive. If the archive is full, crowding distance is used to determine the most crowded non-dominated solution, which is subsequently removed [1]. The crowding distance calculation procedure is

Algorithm 1 Crowding Distance Calculation

```
l = |I|
for each i, set I[i]_distance = 0 do
    for each objective m do
        I = sort(I, m)
        I[1]_distance = I[l]_distance = ∞
        for i = 2 to (l − 1)) do
            I[i]_distance += (I[i+1].m − I[i−1].m)
        end for
    end for
end for
```

as suggest by Deb *et al* [10]. The crowding distance of a particular solution represents the average distance between its two neighboring solutions. Endpoints are given an infinite crowding distance so that they are always selected for removal if a less crowded non-dominated candidate solution is obtained. Once the simulation is finished, the entries in the archive form the obtained Pareto-optimal front.

C. MGPSO Procedure

Multi-guide particle swarm optimization [1] is a multi-swarm extension of traditional PSO. By using crowding distance as part of the archive guide process, MGPSO focuses more on populated ares of the objective space. The MGPSO pseudo code is shown in Algorithm 2

D. Stability Condition

Inherent to PSO is the problem of particle roaming. Particle roaming occurs when the velocity magnitude of particles within the sub-swarm grows to such an extent, that the

Algorithm 2 Multi-guide Particle Swarm Optimization

```
for each objective, m = (1,...,n_m) do
    Let f_m be the objective function;
    for each particle, i = (1,...,S.n_{s_m}) do
        Initialize position S_m.x_i(0) in hypercube of dimension n_x;
        Initialize personal best position as S_m.y_i(0) = Sm.x_i(0);
        Determine the neighborhood best position, S_m.ŷ_i(0)
        Initialize the velocity as S_m.v_i(0) = 0
        Initialize S_m.λ_i distributed U(0, 1);
    end for
end for
repeat
    for each objective, m = (1,...,n_m) do
        for each particle, i = (1,...,S.n_{s_m}) do
            if f_m(S_m.x_i) < f_m(S_m.y_i) then
                S_m.yi = S_m.x_i(t);
            end if
            for particles î with particle i in their neighborhood do
                if f_m(S_m.y_i) < f_m(S_m.ŷ_î) then
                    S_m.ŷ_{î=S_m.y_i(t);}
                end if
            end for
            Update the archive with the solution Sm.x_i;
        end for
    end for
    for each objective, m = (1,...,n_m) do
        for each particle, i = (1,...,S.n_{s_m}) do
            Select a solution, S_m.â_i, from the archive using tournament selection;
            Calculate S_m.v_i(t + 1) using the MGPSO velocity update equation
            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
```

particles permanently leave the search domain. Scheepers *et al* [1] derived

$$0 < c_1 + \lambda c_2 + (1-\lambda)c_3 < \frac{(4-w^2)}{1 - w + \frac{(c_1^2+\lambda^2 c_2^2+(1-\lambda)^2 c_3^2)(1+w)}{3(c_1+\lambda c_2+(1-\lambda)c_3)^2}} \tag{3}$$

for MGPSO based upon the work of Cleghorn *et al* [11] which guarantee particle convergence when adhered to. Cleghorn and Engelbrecht [7] demonstrate that if stability conditions are not followed, PSO performance is often worse than random search. As a result, this paper adheres to the derived stability conditions throughout simulation.

E. Multi-modal MGPSO

The principle behind PSO as proposed by Kennedy and Eberhart [2] is for particles to identify a single globally attractive solution guided by a particle's cognitive and social influence. For search spaces with multiple optimal solutions, particles in the swarm population may be misled to local optima. A possible solution is to allow the population to simultaneously search for multiple optima. Species-based PSO (SPSO) proposed by Li [4] dynamically assigns particle neighborhoods based upon fitness, allowing different portions of the swarm to converge to different optima. Central to SPSO is the task of segmenting particles in the population into a cluster of species. Goldberg and Richardson [12] propose a niching method where a population is classified into groups based upon similarity measure using Euclidean distance. If a particle's Euclidean distance is below a certain threshold from the species center (species seed), that particle is said to be a member of the species. The predefined distance threshold is known as a the species radius. The algorithm for determining species seeds is

Algorithm 3 Algorithm to Determine Species Seeds

Input L_{sorted} - particles sorted in decreasing order fitness
Output S - particles identified as species seeds

```
S = φ
while Not reaching the end of L_sorted do
    found ← FALSE;
    for all P ∈ S do
        if d(s, p) ≤ r_s then
            found ← TRUE;
            break;
        end if
    end for
    if (notfound) then
        let S ← S ∪ {S}
    end if
end while
```

The species seed is initialized to ϕ, an empty set. The set of particles is then iterated in sorted order of fitness. In the context of a objective function minimization, fit particles have low objective function evaluations. If the particle currently being evaluated does not fall with in the radius r_s of any particle in the set S, the particle will become a new seed and be added to S. The species seed is the best fit particle of any given species. Particles within the same species can be made to follow the species seed through PSO neighborhood best assignment. This allows particles within the same species to be drawn to the most-fit particle, in turn increasing their fitness.

III. Comparison Methodology

This section describes the implementation decisions made to compare MGPSO and Multi-modal MGPSO. This paper implements both variations in Java according the pseudo code presented in algorithm 2 and algorithm 3.

To determine if it advantageous to use a multi-modal MGPSO algorithm for multi-objective optimization problems a performance analysis is made against inertial weight MGPSO as proposed by Scheepers *et al*. Due to the non-deterministic nature of PSO, the evaluation procedure compares many independently simulated trials.

A. Evaluation tournament

To compare the performance of Multi-modal MGPSO to traditional MGPSO, a tournament based comparison procedure is conducted. The performance of both optimizers are compared across many trials, to obtain an overall performance rank.

B. Mann–Whitney U test

The IGD data of the final iteration of both optimization algorithms are evaluated using a Mann-Whitney U non-parametric statistical test. A left tailed statistical test is conducted to test the hypothesis:

- $\mathbf{H}_0$: The inverted generation distance between the POFs obtained by MGPSO and Multi-modal MGPSO are similar across ZDT benchmark functions
- $\mathbf{H}_a$: Multi-modal MGPSO obtains POFs with lower inverted generational distance to the true POF across ZDT benchmark functions

If the test indicates a statistically significant difference between the performance of the optimizers at the 95% confidence level, the null hypothesis $\mathbf{H}_0$ will be rejected in favor of the alternate.

C. Control parameter configuration

The performance of the model is highly sensitive to control parameter configurations. To mitigate poor performance as a result of unfavorable configuration assignment, control parameters are sampled randomly on the intervals:

- $(w) \sim Uniform(0, 1)$
- $(c_1, c_2, c_3) \sim Uniform(0, 2)$

Scheepers *et al* [1] derived optimized control parameter configurations for MGPSO for various search environments. The parameter domains mentioned above encapsulate the set of tuned parameter configurations. As a result, these bounds are used to randomly sample control parameter configurations. Once a control configuration value is selected for $w, c_1, c_2, and c_3$ a corresponding particle specific value for λ is sampled such that the stability condition given in equation 3 is satisfied.

With regards to the niching radius of SPSO, it was empirically determined that a radius size of 0.05 per dimension yields favorable performance [4].

Throughout the simulation procedure, each sub-swarm is allocated 25 particles. Particle count does influence the performance of PSO [1], in future research this parameter will be tuned corresponding to the search space being traversed.

IV. Simulation Procedure

This section highlights the empirical process used to compare the performance of MGPSO against multi-modal MG-PSO. To diversify the testing procedure, a variety of multi-objective benchmark functions [14] have been selected to gauge performance.

A. Selected Benchmark Functions

Five of the Zitzler–Deb–Thiele (ZDT) benchmark functions were be used to evaluate the performance of the PSO implementations.

- **ZDT1** *has a convex Pareto-optimal front*

$$
\begin{aligned}
&f_1(x) = x_1 \\
&g(x_2, ..., x_m) = 1 + 9 \cdot \sum_{i=1}^{m} \frac{x_i}{(m-1)} \\
&h(f_1, g) = 1 - \sqrt{f_1/g} \\
&f_2(\mathbf{x}) = g(x_2, ..., x_m) \cdot h(f_1(x_1), g(x_2, ..., x_m))
\end{aligned}
$$

where m = 20, and $x_i \in [0, 1]$. *The Pareto-optimal front is formed when g(x) = 1.*

- **ZDT2** *the non-convex counterpart to* **ZDT1**

$$
\begin{aligned}
&f_1(x) = x_1 \\
&g(x_2, ..., x_m) = 1 + 9 \cdot \sum_{i=1}^{m} \frac{x_i}{(m-1)} \\
&h(f_1, g) = 1 - (f_1/g)^2 \\
&f_2(\mathbf{x}) = g(x_2, ..., x_m) \cdot h(f_1(x_1), g(x_2, ..., x_m))
\end{aligned}
$$

where m = 20, and $x_i \in [0, 1]$. *The Pareto-optimal front is formed when g(x) = 1.*

- **ZDT3** *represents the discreteness feature; its Pareto-optimal front consists of several noncontiguous convex parts*

$$
\begin{aligned}
&f_1(x) = x_1 \\
&g(x_2, ..., x_m) = 1 + 9 \cdot \sum_{i=1}^{m} \frac{x_i}{(m-1)} \\
&h(f_1, g) = 1 - \sqrt{f_1/g} - (f_1/g)\sin(10\pi f_1) \\
&f_2(\mathbf{x}) = g(x_2, ..., x_m) \cdot h(f_1(x_1), g(x_2, ..., x_m))
\end{aligned}
$$

where m = 20, and $x_i \in [0, 1]$. *The Pareto-optimal front is formed when g(x) = 1. The introduction of the sine function causes discontinuity in the Pareto-optimal front. However, there is no discontinuity in the parameter space.*

- **ZDT4** *contains* 21^9 *local Pareto-optimal fronts and tests for the PSO's ability to deal with search space multi modality*

$$
\begin{aligned}
&f_1(x) = x_1 \\
&g(x_2, ..., x_m) = 1 + 10(m-1) + \sum_{i=1}^{m} (x_i^2 - 10(\cos(4\pi x_i)) \\
&h(f_1, g) = 1 - \sqrt{f_1/g} \\
&f_2(\mathbf{x}) = g(x_2, ..., x_m) \cdot h(f_1(x_1), g(x_2, ..., x_m))
\end{aligned}
$$

where m = 10, $x_1 \in [0, 1]$, *and* $x_2, ..., x_m \in [-5, 5]$. *The global Pareto-optimal front is formed with g(x) = 1.*

- **ZDT6** *the Pareto-optimal solutions are nonuniformly distributed along the global Pareto front. The density of* *the solutions is lowest near the Pareto-optimal front and highest away from the front*

$$f_1(x) = 1 - exp(-4x_1)\sin^6(6\pi x_1)$$
$$g(x_2, ..., x_m) = 1 + 9 \cdot (\sum_{i=1}^{m} \frac{x_i}{(m-1)})^{0.25}$$
$$h(f_1, g) = 1 - (f_1/g)^2$$
$$f_2(\mathbf{x}) = g(x_2, ..., x_m) \cdot h(f_1(x_1), g(x_2, ..., x_m))$$

where m = 10, and $x_i \in [0, 1]$. *The Pareto-optimal front is formed when g(x) = 1, and is non-convex*

B. Inverted Generational Distance

To compare the quality of the Pareto-optimal fronts identified by MGPSO and multi-modal MGPSO, this paper uses Inverted Generational Distance (IGD) as proposed by Coello and Reyes-Sierra [13]. Low IGD values suggest the optimization algorithm successfully exploited the true Pareto-optimal front. IGD is calculated by

$$IGD = \frac{\sqrt{\sum_{k=1}^{|POF*|} b_k^2}}{|POF*|} \tag{4}$$

Where POF* is a set of Pareto-optimal solutions representing the true Pareto-optimal front, and b_k is the shortest Euclidean distance in the objective space between obtained solution k and the true POF.

C. Performance Analysis Procedure

To determine if using a multi-modal MGPSO variation performs competitively against MGPSO, a tournament-based evaluation procedure is conducted. For each benchmark function, 50 independent trials of 5000 iterations each are performed per algorithm. For each independent trial a comparison will be made based upon the quality of the obtained POFs of both algorithms. The algorithm that produced a POF with the smallest IGD evaluation is the winner for that trial. The results across all 50 trials will be used to indicate which optimization procedure yields desirable results most frequently.

Both optimization algorithms are highly sensitive to parameter configurations, often deviating significantly between trials, despite tuning efforts. As a result, the variance between independent iterations is also collected to aid in accurate comparison. The minimum IGD value obtained for each benchmark function is also noted.

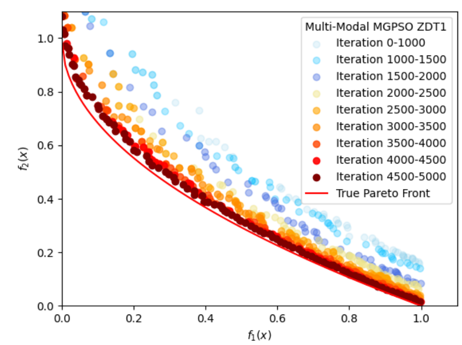
To visually interpret the performance of the optimizers, a plotting technique is used that highlights the improvement of POF quality over time.

V. Simulation Results

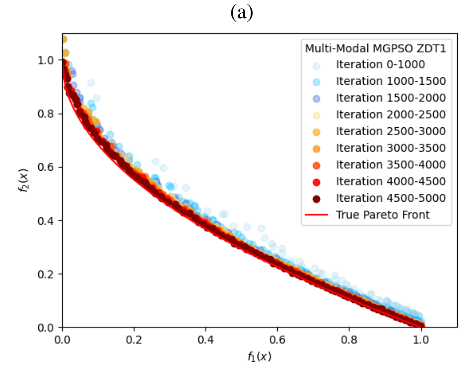
This section presents empirical results and observations regarding the performance comparisons between MGPSO and Multi-guide MGPSO.

A. Addressing the variability between trials

As discussed, the performance of multi-objective particle swarm optimization is highly sensitive to control parameter configurations. Although control parameter configurations are sampled as to satisfy the stability condition of equation 3, desirable performance is not guaranteed.



(a)



(b)

Fig. 1: Both Pareto-optimal fronts were obtained in two independent simulations. The non-deterministic nature of PSO optimizer's account for the obtained variance.

This behaviour is not unique to the ZDT1 benchmark function, as displayed in Figure 1. This result suggests in order to supplement the credibility of the evaluation procedure, results should be reduced across multiple independent simulations. Future research will be invested in the effort of determining effective techniques to curtail the variability of MGPSO for more accurate analysis. The state of the POF is recorded throughout the simulation process. The archive is plotted using warmer colours as the iterations progress. Using this technique facilitates inference as to the progression of the optimizer over time. All examined MOPs exhibit significant variance between trials.

B. Implementation success

The figures indicate that both Multi-modal MGPSO and traditional MGPSO managed to successfully explore and exploit the Pareto-optimal fronts of the Zitzler–Deb–Thiele test set. For certain functions however, the quality of the optimization procedure deviate significantly between trials. For multi-objective optimization functions ZDT4 and ZDT6, there was significant variance between trials regardless of the optimization algorithm used. Multi-modal MGPSO performed marginally better in terms of solution quality obtained from ZDT4 and ZDT6, however the variance between trials exceeded that of MGPSO.

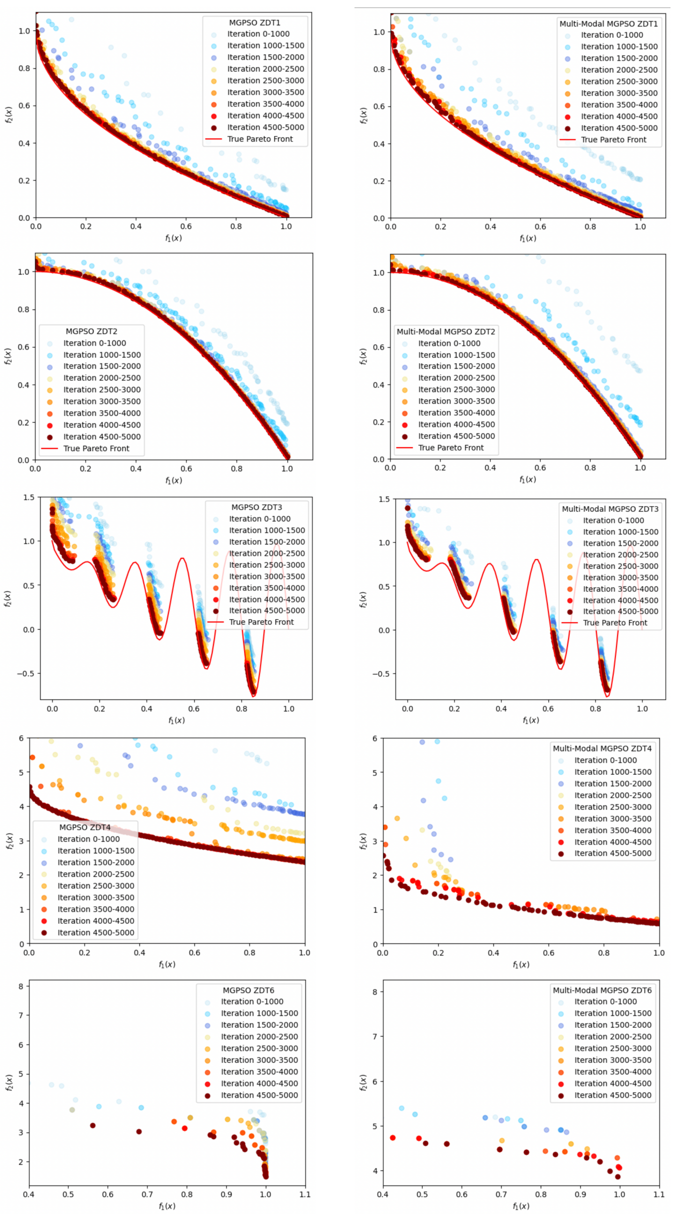


Fig. 2: MGPSO and Multi-modal MGPSO calculated POFs for 2-objective ZDT problems. The left subplots are obtained by MGPSO, and the right by Multi-modal MGPSO. Dot colouring is related to iteration count.

C. Optimizer stability/reliability

To assess the stability of MGPSO against multi-modal MGPSO, IGD information was plotted along with bounding standard deviation curves obtained from the independent simulations.

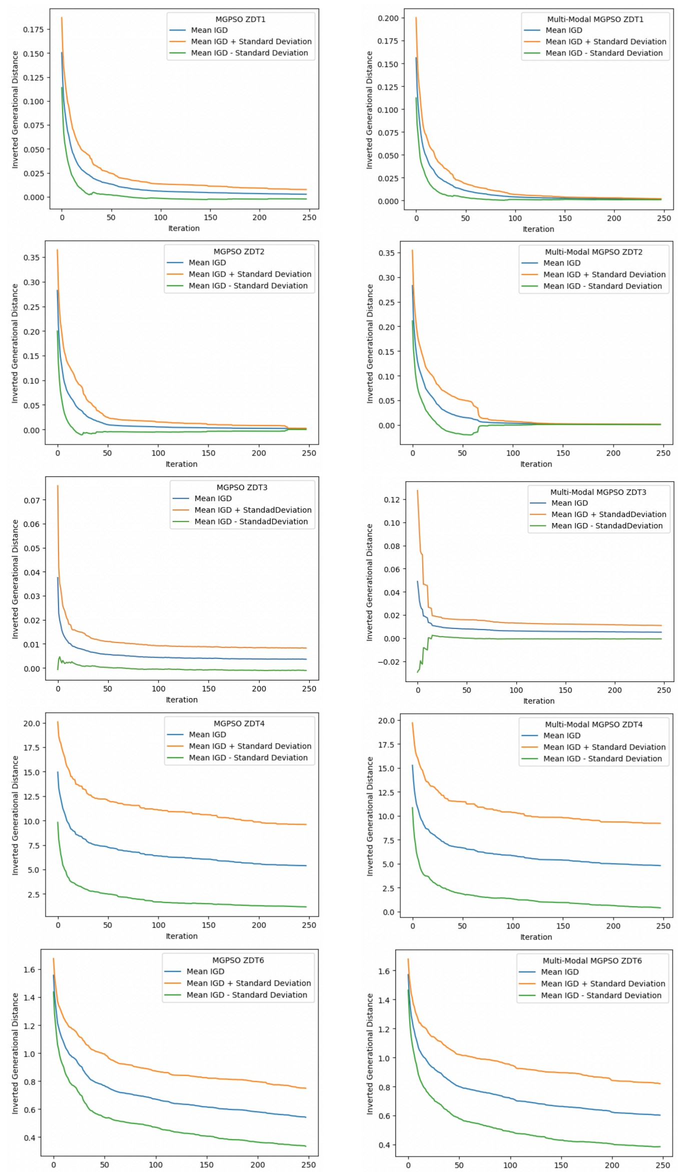


Fig. 3: MGPSO and Multi-modal MGPSO IGD bounding curves for 2-objective ZDT problems. The left subplots are obtained by MGPSO, and the right by Multi-modal MGPSO.

D. Multi-objective optimization tournament results

Due to the similarity of the obtained data, a graphic analysis is not sufficient to infer which optimizer yields preferred performance most frequently. To assist in this analysis a tournament of 50 independent trials each was conducted. The algorithm which produced the POF with the lowest final IGD evaluation was noted, as well as the mean IGD variance. The results of the optimizer pair-wise comparison were found to be:

TABLE I: Inverted Generational Distance Ranking

	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
MGPSO Wins	22	17	27	21	22
Multi-modal Wins	28	33	23	29	28
Total trials	50	50	50	50	50

TABLE II: Mean Inverted Generational Distance Variance

	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
MGPSO Var.	1.35e-4	5.06e-4	2.59e-3	21.54	4.91e-2
Multi-modal Var	8.46e-5	4.56e-4	5.31e-4	20.76	4.31e-2
Lowest Mean Var	Multi	MGPSO	Multi	Multi	MGPSO

The results of this tournament evaluation indicate that for multi-objective optimization problems ZDT1, ZDT2, ZDT4, and ZDT6 the Multi-modal MGPSO algorithm exploited the Pareto-optimal fronts with lowest IGD. Multi-modal MGPSO demonstrated an astute advantage over traditional MGPSO when optimizing the ZDT2 function. Furthermore, multi-modal MGPSO produced lower variance between independent trials than MGPSO for three of the five MOPs evaulated. This suggests that in most cases the niching abilities of multi-modal MGPSO allows for POFs to be obtained at increased reliability compared to traditional MGPSO. To determine if this improvement can be accredited to algorithmic improvement, or stochastic deviation, a Mann-Whitney U test can be conducted on IGD distances of the obtained POF.

E. Results of the Mann-Whitney U Test

After performing a lower-tail statistical test, the p values obtained were:

TABLE III: Lower Tail Mann-Whitney p Values

	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
p value	0.131	0.148	0.092	0.441	0.271
Statistically significant	No	No	No	No	No

At the 95% confidence level it cannot be concluded that using a Multi-modal implementation of MGPSO yields improved performance over traditional MGPSO. It is however noted that had this test been conducted at a 90% confidence level, there is a statistically significant difference between the algorithm's performance on the ZDT3 multi-objective optimization problem. At the 85% confidence level it can be concluded that Multi-modal MGPSO is statistically superior to MGPSO on three of the five ZDT benchmark functions.

In light of these results, the null hypothesis cannot be rejected in favor of the alternate using a 95% confidence level.

VI. Conclusion

The intention of this paper was to investigate the performance quality of multi-guide particle swarm optimization (MGPSO) compared to an extension designed for multi-modal multi-objective search spaces. Evaluation metrics were established as a means of assessing the quality, stability, and reliability of both algorithms. A Mann-Whitney U statistical test suggests that at the 95% confidence level there is not a statistically significant deviation in performance that would indicate that extending traditional MGPSO for multi-modal objective spaces is worthwhile. In conducting this research, the stochasticity of multi-objective particle swarm optimization was surprising. If effort is made to reduce inter-trial variation, the comparison may be different.

A. Future research

As mentioned in the paper, the performance of both algorithms are highly sensitive to the control parameter configurations. Although the configurations used throughout this paper satisfy the stability conditions derived by Scheepers *et al* [1], performace is still highly variable. As a result- future research will be performed on determining a class of optimum control parameter configurations for each benchmark function. The parameter domain that should be tuned specifically is λ. Dynamically updating the configurations for λ during the optimization process may yield less variance in the quality of the POF obtained.

References

[1] C. Scheepers, A. Engelbrecht, C. Cleghorn, 'Multi-guide particle swarm optimization for multi-objective optimization: empirical and stability analysis', Swarm Intelligence, 13, 12 2019.

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

[3] 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

[4] X. Li, 'Adaptively Choosing Neighbourhood Bests Using Species in a Particle Swarm Optimizer for Multimodal Function Optimization', 08 2004, 3102.

[5] A. P. Engelbrecht, Computational Intelligence: An Introduction, 3rd ed. Hoboken, NJ: Wiley-Blackwell, 2020.

[6] Scheepers, C. (2018). Multi-guided particle swarm optimization: A multi-objective particle swarm optimizer. Doctor's dissertation, University of Pretoria.

[7] Cleghorn, C., Engelbrecht, A. (2016a). Particle swarm optimizer: The impact of unstable particles on performance. In Proceedings of the IEEE symposium series on swarm intelligence (pp. 1–7). Piscataway, NJ: IEEE Press.

[8] C. Scheepers, A. Engelbrecht, 'Vector evaluated particle swarm optimization archive management: Pareto optimal front diversity sensitivity analysis', 12 2016

[9] C. Scheepers, A. Engelbrecht, 'Vector evaluated particle swarm optimization: The archive's influence on performance'

[10] 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, April 2002
[11] Cleghorn, C. W., Scheepers, C., Engelbrecht, A. P. (2018). Stability analysis of a multi-objective particle swarm optimizer. In Proceedings of the 11th international conference on swarm intelligence.
[12] Goldberg, D.E. and Richardson, J.: Genetic algorithms with sharing for multi- modal function optimzation. In: Grefenstette, J.J. (eds.): Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms. Lawrence Earlbaum, Hillsidale, New Jersey (1987) 41–49
[13] Coello Coello, C.A. and Reyes Sierra, M., A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. Monroy, R., ArroyoFigueroa, G., Sucar, L.E. and Sossa, H. (editors), Proceedings of the Mexican International Conference on Artificial Intelligence, vol. 2972, pp. 688–697. Springer, 2004.
[14] Zitzler, E., Deb, K., and Thiele, L., "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," Evolutionary Computation, Vol. 8, No. 2, pp. 173-195, 2000.