UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

COS700
Research Report

Implementing Variations of the Multi-Guided PSO for Dynamic Multi-Objective Optimisation

Student number: u15260870

Supervisor:
A.P. Engelbrecht

Contents

List of Algorithms

List of Tables

Implementing Variations of the Multi-Guided PSO for Dynamic Multi-Objective Optimisation

Abstract

The Multi-guided Particle Swarm Optimisation (MGPSO) algorithm, a variation of the particle swarm Optimisation algorithm, was proposed to solve multi-objective Optimisation problems (MOPs). The purpose of this research report is to develop variations of the MGPSO algorithm, which can execute in dynamic environments. Before any algorithms are developed, a literature review is created to summarise the research found in the relevant fields of: archive management, the Multi-guided Particle Swarm Optimisation algorithm, the Quantum PSO, Dynamic Environments, Multi-Objective Optimisation and Benchmark Functions for Dynamic Environments. The implemented variations of the Multi-guided Particle Swarm Optimisation algorithm are developed as the Dynamic Multi-guided Particle Swarm Optimisation (D-MGPSO) algorithm and the Dynamic Quantum Multi-guided Particle Swarm Optimisation (D-QMGPSO) algorithm, where disparity from the MGPSO is made through modifications to the archive management strategy. Four archive management strategies are implemented and tested against one another for both the D-MGPSO and the D-QMGPSO algorithms. The results show that no single algorithm/archive management strategy performs the best for all performance measures, however it was found that the D-QMGPSO algorithm implemented with the first archive management strategy produces the highest number of wins for both the alternative accuracy and the stability performance measures.

Keywords:

Particle Swarm Optimisation, Multi-Guided, Dynamic Environment/Problems, Multi-Objective Optimisation, Hypervolume

1 Introduction

The particle swarm Optimisation (PSO) algorithm [JK95] was created as a computationally efficient Optimisation algorithm to solve multi-dimensional Optimisation problems. Initially, the algorithm was developed for Optimisation problems which have boundary constraints and a single objective function, executed in a stationary environment. A stationary environment is one in which the objective function's optima does not change with regard to time.

Optimisation problems exist which have multiple objectives. These problems are called MOPs. Due to the contradictory nature of these objective functions [?], it is not possible to find a single solution that would be optimal for all the objectives simultaneously. Instead, multi-objective Optimisation (MOO) algorithms produce a set of optimal solutions, known as the Pareto-optimal solutions. To acquire the Pareto-optimal solutions, the optimal set of decision variables, referred to as the Pareto Optimal Set (POS), are transposed from the decision space to the objective space. The set of Pareto-optimal solutions, referred to as the Pareto-optimal front (POF) give the trade-offs between two or more conflicting objectives.

There are different categories of PSO algorithms which behave differently from one another as they traverse through the search space [Eng07a]. Recently, a variation of the PSO algorithm called the Multi-Guided PSO (MGPSO) [CS18] was developed to solve MOPs. This variation of the particle swarm Optimisation algorithm applied to MOPs has a simple implementation, yet shows competitive performance against alternative state-of-the-art MOO algorithms.

The MGPSO developed by Scheepers and Engelbrecht [CS18] is a multi-swarm adaptation of the PSO algorithm. Therefore, the algorithm utilises multiple sub-swarms [Eng07b], with each sub-swarm optimising a single objective. [PH18]. The MGPSO variation of the PSO algorithm, takes advantage of an archive guiding component in the velocity update equation which encourages the algorithm to search for solutions along the POF.

There is a field of research which exists to solve Optimisation problems which have dynamic environments. A dynamic environment implies that the objective function's various optima changes with regard to time. [?]. Optimisation algorithms which solve Optimisation problems in dynamic environments must be able to continuously track the changing optima over time. Since its creation, alternative variations of the PSO algorithm have been created to solve MOPs [CACCL04] and Optimisation problems with dynamic environments [Eng07c].

Optimisation problems exist with multiple objectives in dynamic environments, referred to as dynamic multi-objective Optimisation problems (DMOPs).

The research to be conducted will focus on how previous static, multi-objective, multi-swarm variations of the PSO have been adapted, or new multi-swarm variation of the PSO have been designed to work in dynamic environments. The goal of the project is to create an efficient and easy to implement variation of the MGPSO to solve DMOPs.

Proceeding the introduction, the problem statement is defined, followed by the background of related literature that aided in the development of the dynamic MGPSO (DMGPSO) variations. Thereafter, the methodology of the implemented variations of the DMGPSO is given, with the results of the these algorithms ensuing. The report is ended off with the conclusion and discussion for future work.

2 Problem Statement

As of October 2018, there are a limited number of algorithms created to solve DMOPs. The current algorithms created to solve DMOPs have complex designs and are complicated to implement. This results in an underutilised field of research where the majority of real world DMOPs are unsolvable.

The purpose of this research report is to develop variations of the MGPSO that are capable of effectively solving DMOPs in a methodology which is easy to implement. The following research questions will need to be addressed:

- Can variations of the MGPSO be created to solve DMOPs?
- How would the various DMGPSO perform against one another?

3 Background and related literature

The literature that was reviewed during the research process comprised of peer reviewed publications and scientific sources such as journal articles, conference papers, books and chapters in books, technical reports, theses and dissertations.

The research conducted was solely focused on DMOPs and explored only the research areas that gained insight into the benefit of the implementation for the variations of the DMGPSO. Therefore, research related to single-objective or many-objective Optimisation problems, as well as any Optimisation algorithms designed to work in a static environment that do not have dynamic environment variations, are out of the scope of the research project.

3.1 Archive Management

The first approach to using a PSO for a MOP was implemented by Raquel and Naval [RN02] with the multi-objective particle swarm Optimisation algorithm (MOPSO). This implementation recorded a set of the best solutions found by a particle. This set is now known as the archive. The archive stored the POF, which is the set of optimal solutions that give the trade-offs between two or more conflicting objectives.

Li [Li03] created the NSPSO. A variation of the MOPSO, which limited the size of the archive. Succeeding the development of the NSPSO, Raquel and Naval [RN05] created the MOPSO-CD. The MOPSO-CD is the first application of a PSO using an archive which incorporates the mechanism of crowding distance. Subsequent to the creation of the MOPSO, the NSPSO and the MOPSO-CD, many variations of the PSO have been implemented to extend the PSO to handle MOPs, while utilising an archive of particles to maintain the POF [RSC06]. The most cited publications include Li And Mohammed's MOPSO [RSC06], Fieldsned and Sing's MOA [Fie06], Sierra and Coello's OMOPSO [SCC05], Mostaghim and Tiech SMOPSO [Tei03], proceeding though to Scheepers and Engelbrecht's MGPSO [CS18].

Archive management and the use of POF are shown to be key aspects to solving MOPs when using a PSO. Keeping track of the set of optimal non-dominated solutions in the search space produces the most effective results. When searching for the diversity of solutions and using an archive to influence the search behaviour of the algorithm it is almost used in every effective modern implementation of a PSO to solve MOPs.

3.2 The Multi-Guided PSO

Scheepers and Engelbrecht developed a new variation of the PSO for MOPs called the MGPSO [CS18]. The MGPSO variation of the PSO algorithm, exploits an archive guiding component in the velocity update equation which guides the PSO algorithm to search for solutions along regions in the POF which have the largest crowding distance. The MGPSO's velocity update equation is formally defined as follows:

$$v_i(t+1) = wv_i(t)+c_1r_1(y_i(t)-x_i(t))+\lambda_i c_2 r_2(\hat{y}(t)-x_i(t))+(1-\lambda_i)c_3r_3(\hat{a}_i(t)-x_i(t)) \tag{1}$$

Where $v_i(t)$ is the velocity of particle i at iteration t, w is the inertia weight, c_1 and c_2 are the acceleration coefficients and r_1 and r_2 are random vectors with components sampled uniformly from (0; 1). $x_i(t)$ is the position of particle i at iteration t, $y_i(t)$ is the personal best position of particle i at iteration t, $\hat{y}_i(t)$ is the knowledge transfer strategy (KTS) selected guide for particle i at iteration t. The KTS may be a global or local, best particle guide. c_3 is an additional acceleration coefficient, r_3 is an additional random vector with components sampled uniformly from $(0, 1)$. λ_i is the exploitation trade-off coefficient for particle i and $\hat{a}_i(t)$ is a randomly selected solution from the archive for particle i at iteration t.

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. Therefore, a randomly selected solution from the archive guides the particle's velocity and directly affects its search behaviour. The more diverse the archive set is the more diverse the collective swarms search behaviour becomes.

Using the MGPSO algorithm, produces an easy to implement MOPSO that effectively solves MOPs. Extending the MGPSO algorithm to execute in dynamic environments has the potential to produce an easy to implement MOPSO that promotes effective results for DMOPs.

3.3 Quantum PSO

Yang, Wang and Jiao [SY] proposed a quantum PSO for dynamic single-objective Optimisation problems. The quantum PSO is based on the quantum model of an atom. In the quantum model of an atom, orbiting electrons are replaced by a quantum cloud. The quantum cloud has a probability distribution governing the position of the electron at a given time. To replicate the quantum model of an atom in an algorithm, the quantum PSO separates its particles into the two types: neutral and charged. Neutral particles follow the standard PSO update equation. However, the set of charged particles are reinitialised within a multi-dimensional sphere around the global best particle after every iteration. This initialisation is based off of a chosen distribution, which may be Uniform, Non-uniform, Gaussian, Cauchy, Exponential, Beta, Triangular or Weibull.

Initialising a set of charged particles around the current global best position after each iteration causes the algorithms search behavior to constantly be in an exploratory state around the current optimal fitness value. This hereditary trait of the quantum model results in the quantum PSO inherently having the capability to handle dynamic environments in which the optimal fitness value of the search objective can change. As of October 2018, a small, limited number variations of the quantum PSO exist to handle MOPs. An an example of a quantum PSO variation for dynamic MOPs can be found in the publication by Blackwell and Branke [**?**]. With the information gained from the research of archive management, an archive approach to the quantum PSO could be developed to create a variation of the quantum PSO which is capable of solving MOPs.

3.4 Dynamic Environments and Multi-Objective Optimisation

In a dynamic environment the fitness landscape can change over time. The change in fitness landscape implies that the fitness optima of the landscape changes or the constraints acting upon the fitness optima change [Hel12]. Therefore, for a single-objective algorithm to execute in a dynamic environment, it must have the ability to track the single optimum fitness value over time. In artificial dynamic environments, the change in the fitness landscape can be controlled with reference to temporal severity and spatial severity. Temporal severity is used to describe the frequency of change that occurs in the environment. Spatial severity is used to describe the extent of the change that occurs for the position of the optima in the fitness landscape.

Duhain [Duh] classifies dynamic environments into the following four categories:

- **Static environments**. Either the environment does not change, or the change is so small that it does not effect the results of the algorithm.
- **Progressively changing environments**. The temporal severity to change is high but the spatial severity of the change is low.
- **Abruptly changing environments**. The temporal severity to change is low but the spatial severity of the change is high.
- **Chaotic environments**. The temporal severity to change and the spatial severity of the change are both high.

For Optimisation problems which have multiple, dynamic objectives, a MOO algorithm must have the ability to track the POF over time. A classification of the dynamic environment types for dynamic MOP have been developed [MFA04] and are defined by Helbig [Hel12] as follows:

- **Type I environment**. When a change in the environment occurs, the Pareto Optimal Set change but the Pareto Optimal Front does not.
- **Type II environment**. When a change in the environment occurs, both the the Pareto Optimal Set and the Pareto Optimal Front change.
- **Type III environment**. When a change in the environment occurs, the the Pareto Optimal Set does not change, but the Pareto Optimal Front does.
- **Type IV environment**. When a change in the environment occurs, the Pareto Optimal Set and the Pareto Optimal Front do not change, even if an objective function or constraint has changed.

With reference to the categories and classifications of dynamic environments, the transformation of the POF shape with regards to time can be summarized as follows:

- The shape of the POF remains the same but the POF shifts in position over time, with reference to the objective space.
- The shape of the POF changes over time. This may change from concave to convex or from a continues POF to a disconnected POF.
- The density of solutions on the POF change over, either becoming more dense or less dense.

- The shape of the POF remains the same, however previous solutions in the POF become dominated and must be removed.

Understanding the types of dynamic environments and the resulting transformation that occur in the POF with regards to time, allow insight to be gained into the benchmark functions which should be used to evaluate the algorithm.

3.5 Benchmark Functions for Dynamic Environments

To sufficiently evaluate a dynamic algorithm's ability to solve DMOOP, Helbig and Engelbrecht [MH13] have observed that certain characteristics should be found in the set of testing benchmark functions. Helbig [Hel12] further defines the ideal set of characteristics as follows:

1. The set of benchmark functions should test for the following difficulties to converge towards the POF:
   - Multimodality.
   - Deception.
   - Isolated optimum.
2. The set of benchmark functions should test for the following difficulties to obtain a diverse set of solutions:
   - Convexity or non-convexity in the POF.
   - Discontinuous POF, i.e. disconnected sub-regions that are continuous.
   - Non-uniform distribution of solutions in the POF.
3. The set of benchmark functions should have various types or shapes of POS's, where the POS's consist of non-linear curves in addition to linear functions.
4. The set of benchmark functions should have decision variables with dependencies or linkages.
5. The set of benchmark functions should have a non-uniform distribution of solutions in the POF, where the distribution of solutions changes over time.

6. The shape of the POFs should change over time from convex to non-convex or vice versa.
7. The set of benchmark functions should have decision variables with different rates of change over time.
8. The set of benchmark functions should include cases where the POF depends on the values of previous POSs or POFs.
9. The set of benchmark functions should enable changing the number of decision variables over time.
10. The set of benchmark functions should enable changing the number of objective functions over time.

With these ideal characteristics taken into account, an ideal set of dynamic multi-objective benchmark functions can be selected from the set of currently used dynamic multi-objective benchmark functions.

4 Methodology

4.1 Algorithms

4.1.1 The Multi-Guided PSO

The Multi-Guided PSO [CS18] is a multi-swarm, multi-objective, particle swarm Optimisation algorithm. In the MGPSO, each objective is optimized by a distinct sub-swarm. Particles in each sub-swarm are evaluated using the corresponding objective function. This allows for multiple objectives to be solved simultaneously. Similar to most modern multi-objective Optimisation algorithms [RSC06], the MGPSO employs an archive to find the pareto-optimal solutions to the MOP. Inspired by the findings of Scheepers and Engelbrecht [CSa] [CSb], the MGPSOs have a bounded archive which exploits crowding distance for its POF management. To guide the sub-swarms towards optimising solutions along the POF, an archive influence component is added. The archive influence component, in conjunction with the social and cognitive components, constitute the velocity update equation. The new velocity equation can be found in the equation 2. The archive guide, $\hat{a}_i(t)$, 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.

The MGPSO algorithm is formally defined as follows:

Algorithm 1 Multi-guided Particle Swarm Optimisation (MGPSO)

for each objective $m = 1, ..., n_m$ **do**
Create and initialise 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 $S_m.y_i$ represent the personal best position of particle $S_m.x_i$, initialised to $S_m.x_i(0)$;
Let $S_m.\hat{y}_i$ represent the global best position that guides particle $S_m.x_i$, initialised to $S_m.x_i(0)$;
initialise $S_m.v_i(0)$ to 0;
initialise $S_m.\lambda_i \sim U(0; 1)$;
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 $f_m(S_m.x_i) < f_m(S_m.y_i)$ **then**
$S_m.y_i = S_m.x_i(t)$;
end if
for particles $\hat{\imath}$ with particle i in their neighborhood **do**
if $f_m(S_m.y_i) < f_m(S_m.\hat{y}_{\hat{\imath}})$ **then**
$S_m.\hat{y}_{\hat{\imath}} = S_m.y_i$;
end if
end for
Update the archive with the solution $S_m.x_i$;
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, $S_m.\hat{a}_i(t)$, from the archive using tournament selection;
$$S_m.v_i(t+1) = wS_m.v_i(t) + c_1 r_1 (S_m.y_i(t) - S_m.x_i(t)) + S_m.\lambda_i c_2 r_r (S_m.\hat{y}_i(t) - S_m.x_i(t)) + (1 - S_m.\lambda_i) c_3 r_3 (S_m.\hat{a}_i(t) - S_m.x_i(t)));$$
$S_m.x_i(t+1) = S_m.x_i(t) + S_m.v_i(t+1)$;
end for
end for
until stopping condition is true

4.1.2 The Quantum Multi-Guided PSO

The Quantum Multi-Guided PSO (QMGPSO) is an extension to MGPSO defined in section 4.1.1 with the addition of techniques taken from the Quantum PSO, described in section 3.3. The QMGPSO capitalises on the Quantum PSO's implementation of the quantum model, utilising the technique of charged and neutral particles. For each constructed sub-swarm, half of the initialised particles are designate as neutral and half of the initialised particles are designate as charged.

The QMGPSO's position update equation is formally defined as follows::

$$x_i(t+1) = \begin{cases} x_i(t) + v_i(t+1), & \text{if } Q_i = 0 \\ U_{cloud}(\hat{y} + a, \hat{y} + b), & \text{if } Q_i \neq 0 \end{cases} \tag{2}$$

Where $x_i(t)$ is the position of particle i at iteration t, $v_i(t+1)$ is the velocity of the particle at iteration $t+1$, $\hat{y}$ is the global best position that guides particle i at iteration t and U_{cloud} defines a uniform distribution around $\hat{y}$, bounded by a and b. Therefore, for every particle i that is a charged particle, its position is re-initialised around the current global best particle, every iteration.

4.1.3 Dynamic Multi-Guided PSO

The first Variation of the MGPSO algorithm, adapted for dynamic environments, is defined as the Dynamic Multi-Guided Particle Swarm Optimisation (D-MGPSO) Algorithm. The modifications of the MGPSO, to facilitate its ability to solve DMOPs, are implemented in its archive management strategy. Aside from the modifications made to the archive management strategy, the methodology of the D-MGPSO remains the same as the MGPSO, described in algorithm 4.1.1. The modifications to the archive management strategy are specified in sections 4.3.1, 4.3.2, 4.3.3 and 4.3.4.

4.1.4 Dynamic Quantum Multi-Guided PSO

The second variation of the MGPSO algorithm, adapted for dynamic environments, is defined as the Dynamic Quantum Multi-Guided Particle Swarm Optimisation (D-QMGPSO) Algorithm. Similar to the D-MGPSO, the alterations of the D-QMGPSO, to facilitate its ability to solve DMOPs, are implemented in its archive management strategy. The modifications to the archive management strategy are specified in sections 4.3.1, 4.3.2, 4.3.3 and 4.3.4.

4.2 Algorithm Parameters

Both the D-MGPSO and the D-QMGPSO algorithms are executed with the following parameters. Each sub-swarm size is set to 25 particles. The maximum archive capacity is set to 50 particles. The tournament selection size for the competition pool used to select the archive guide is set to 3. The values for the constants w, c_1, c_2 and c_3 are set to $[0.475, 1.80, 1.10, 1.80]$ respectfully.

Both the D-MGPSO and the D-QMGPSO algorithms have the following boundary constraint handling mechanism applied to the decision variables. If a decision variable's value goes above the upper bound, the value for the decision variable is set to the upper bound. If a decision variable's value goes below the lower bound, the value for the decision variable is set to the lower bound.

Each algorithm is tested for a low, medium and large category of decision variables. In the low category, each benchmark function found in section 4.4, is tested with either 2 or 3 decision variables, as required for the minimum number of decision variables specified for the respective benchmark function. In the medium category, each benchmark function found in section 4.4 is tested with 15 decision variables. In the large category, each benchmark function found in section 4.4 is tested with 30 decision variables.

All experiments consisted of 30 independent runs and each run continued for 1000 iterations. For all benchmark functions found in section 4.4, the severity of change (n_t) was set to 1, 10 and 20 and the frequency of change (τ_t) was set to either 10, 25 or 50. This selection of n_t and τ_t values enables the evaluation of the various algorithms in both a fast and slowly changing environment, and an environment that changes either gradually or severely over time.

All code was implemented on an system with 16 Gb of Memory and a i7-8700 CPU running at 3.20GHz.

4.3 Archive Management

For both the D-MGPSO and the D-QMGPSO, an archive is used to hold the set of pareto-optimal solution particles. At every instance that a new particle is requested to be added to the archive, two rules must be passed to allow the request to be granted. Firstly, the particle must be feasible, having all decision values within the respective bounds. Secondly, the particle cannot be dominated by any other particle currently in the archive. If a particle conforms to these rules, a search is made through the archive to remove any particles currently in the archive which are dominated by this new particle. The new particle is then added to the archive. If the number of particles in the archive is greater than the archive's capacity, the particle which results in the smallest crowding distance is removed from the archive.

For both the D-MGPSO and the D-QMGPSO algorithms, an additional modification is made to the archive management of each respective algorithm. This modification facilitates the algorithms ability to solve DMOPs. The modification can be implemented in several distinct strategies, with each strategy resulting in a unique variation of the algorithm. These distinct strategies are executed respectfully, when the algorithm is notified that a change has occurred in the environment. When the algorithm is notified that a change has occurred in the environment, the archive is saved. The description of each archive management strategy can be found in sections 4.3.1, 4.3.2, 4.3.3 and 4.3.4.

4.3.1 Strategy 1: Standard MGPSO

Archive management strategy 1 follows the archive management described in section 4.3. Compared to the standard the MGPSO, there are no additional modifications added to this strategy's archive management, other than saving the archive when the respective algorithm is notified that the environment has changed.

4.3.2 Strategy 2: Total Archive Re-initialisation

When the algorithm is notified that the environment has changed, the following modifications are executed. All particles have their global best and previous best positions reset. All particles have their velocity set to zero. The archive is emptied, having all current particle solutions removed.

4.3.3 Strategy 3: Remove Dominated Solutions

When the respective algorithm is notified that the environment has changed, the following modifications are executed. All particles have their global best and previous best positions reset. All particles have their velocity set to zero. The archive is emptied, having all current particle solutions temporarily removed. Each previous archive solution is requested to be added to the archive, where it is assessed against the rules stated in section 4.3, before its request is granted.

4.3.4 Strategy 4: Linearly decrease Archive Influence

At the beginning of the algorithms execution, each particle has λ, the archive guide's influence on the velocity, set to 1.
At each iteration, the value of λ is linearly decreased by the equation:

$$\lambda = \lambda - \frac{1}{n_t} \tag{3}$$

Where n_t is the frequency of change for the current parameter combination. When the respective algorithm is notified that the environment has changed, the following modifications are executed. All particles have their global best and previous best positions reset. All particles have their velocity set to zero. The archive is emptied, having all current particle solutions temporarily removed. Each previous archive solution is requested to be added to the archive, where it is assessed against the rules stated in section 4.3, before its request is granted. Each particle has λ set to 1.

4.4 Benchmark Functions

Heeding the findings stated in section 3.5, a set of sixteen ideal benchmark function where selected from [Hel12] to evaluate the DMGPSO and D-QMGPSO algorithms. A description of each benchmark function is listed below:

$$DIMP2 = \begin{cases} Minimise: f(x,t) = (f_1(x_{\text{I}}), g(x_{\text{II}},t) \cdot (f_1(x_{\text{I}}), g(x_{\text{II}},t))) \\ f_1(x_{\text{I}}) = x_1, \\ g(x_{\text{II}},t) = 1 + 2(n-1) + \sum_{x_i \in, x_{\text{II}}}[(x_i - G_i(t))^2 - 2\cos(3\pi(x_i - G_i(t)))], \\ h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}, \\ where: \\ G_i(t) = \sin(0.5\pi t + 2\pi(\frac{i}{n+1}))^2,\ t = \frac{1}{n_t}\lfloor\frac{\tau}{\tau_t}\rfloor, \\ x_{\text{I}} \in [0,1];\ x_{\text{II}} \in [-2,2]^{n-1} \end{cases} \tag{4}$$

With reference to 3.4, $DIMP2$ is a Type I DMOOP. The true POF is $f_2 = 1 - \sqrt{f_1}$ and the true POS is $x_i = G(t)$, $\forall x_i \in x_{_{II}}$.

$$FDA1 = \begin{cases} Minimise: f(x,t) = (f_1(x_{_I}), g(x_{_{II}}, t) \cdot (f_1(x_{_I}), g(x_{_{II}}, t))) \\ f_1(x_{_I}) = x_1, \\ g(x_{_{II}}, t) = 1 + \sum_{x_i \in x_{_{II}}} (x_i - G(t))^2, \\ h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} \\ where: \\ G(t) = \sin(0.5\pi t),\ t = \frac{1}{n_t} \lfloor \frac{\tau}{\tau_t} \rfloor, \\ x_{_I} \in [0,1];\ x_{_{II}} = (x_2, ..., x_n) \in [-1,1]^{n-1} \end{cases} \tag{5}$$

With reference to 3.4, FDA1 is a Type I DMOOP. The true POF is $f_2 = 1 - \sqrt{f_1}$ and the true POS is $x_i = G(t)$, $\forall x_i \in x_{_{II}}$.

$$FDA1_{Zhou} = \begin{cases} Minimise: f(x,t) = (f_1(x_{_I}), g(x_{_{II}}, t) \cdot (f_1(x_{_I}), g(x_{_{II}}, t))) \\ f_1(x_{_I}) = x_1, \\ g(x_{_{II}}, t) = 1 + \sum_{x_i \in x_{_{II}}} \left(x_i - G(t) - x_1^{H(t)}\right)^2, \\ h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)} \\ where: \\ G(t) = \sin(0.5\pi t), \\ H(t) = 1.5 + G(t),\ t = \frac{1}{n_t} \lfloor \frac{\tau}{\tau_t} \rfloor, \\ x_{_I} \in [0,1];\ x_{_{II}} = (x_2, ..., x_n) \in [-1,2]^{n-1} \end{cases} \tag{6}$$

With reference to 3.4, $FDA1_{Zhou}$ is a Type II DMOOP. The true POF is $f_2 = 1 - f_1^{H(t)}$ and the true POS is $x_i = G(t) + x_1^{H(t)}$, $\forall x_i \in x_{_{II}}$.

$$FDA2 = \begin{cases} Minimise: f(x,t) = (f_1(x_{\text{I}}), g(x_{\text{II}},t) \cdot (f_1(x_{\text{I}}), g(x_{\text{II}},t))) \\ f_1(x_{\text{I}}) = x_1, \\ g(x_{\text{II}}) = 1 + \sum_{x_i \in x_{\text{II}}} x_i^2, \\ h(x_{\text{III}}, f_1, g, t) = 1 - \left(\frac{f_1}{g}\right)^{H_2(t)} \\ where: \\ H(t) = 0.75 + 0.75\sin(0.5\pi t),\ t = \frac{1}{n_t}\lfloor\frac{\tau}{\tau_t}\rfloor, \\ H_2(t) = (H(t) + \sum_{x_i \in x_{\text{III}}}(x_i - H(t))^2)^{-1}, \\ x_{\text{I}} \in [0,1];\ x_{\text{II}}, x_{\text{III}} \in [-1,1] \end{cases} \tag{7}$$

With reference to 3.4, $FDA2$ is a Type II DMOOP. The true POF is $f_2 = 1 - f_1^{H(t)-1}$ and the true POS is $x_i = 0$, $\forall x_i \in x_{\text{II}}$ and $x_i = H(t)$, $\forall x_i \in x_{\text{III}}$.

$$FDA2_{Camara} = \begin{cases} Minimise: f(x,t) = (f_1(x_{\text{I}}), g(x_{\text{II}},t) \cdot (f_1(x_{\text{I}}), g(x_{\text{II}},t))) \\ f_1(x_{\text{I}}) = x_1, \\ g(x_{\text{II}}) = 1 + \sum_{x_i \in x_{\text{II}}} x_i^2, \\ h(x_{\text{III}}, f_1, g, t) = 1 - \left(\frac{f_1}{g}\right)^{H_2(t)} \\ where: \\ H_2(t) = H(t) + \sum_{x_i \in x_{\text{III}}}(x_i - H(t)/2)^2, \\ H(t) = z^{-\cos(\pi t/4)},\ t = \frac{1}{n_t}\lfloor\frac{\tau}{\tau_t}\rfloor, \\ x_{\text{I}} \in [0,1];\ x_{\text{II}}, x_{\text{III}} \in [-1,1] \end{cases} \tag{8}$$

With reference to 3.4, $FDA2_{Camara}$ is a Type III DMOOP. The true POF is $f_2 = 1 - f_1^{H_2(t)}$ and the true POS is $x_i = 0$, $\forall x_i \in x_{\text{II}}$ and $x_i = -1$, $\forall x_i \in x_{\text{III}}$.

$$FDA3 = \begin{cases} Minimise: f(x,t) = (f_1(x_{\text{I}}), g(x_{\text{II}},t) \cdot (f_1(x_{\text{I}}), g(x_{\text{II}},t))) \\ f_1(x_{\text{I}},t) = \sum_{x_i \in x_{\text{I}}} x_i^{F(t)}, \\ g(x_{\text{II}},t) = 1 + G(t) + \sum_{x_i \in x_{\text{II}}}(x_i - G(t))^2, \\ h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} \\ where: \\ G(t) = |\sin(0.5\pi t)|, \\ F(t) = 10^{2\sin(0.5\pi t)},\ t = \frac{1}{n_t}\lfloor\frac{\tau}{\tau_t}\rfloor, \\ x_{\text{I}} \in [0,1];\ x_{\text{II}} \in [-1,1] \end{cases} \tag{9}$$

With reference to 3.4, $FDA3$ is a Type II DMOOP. The true POF is $f_2 = (1+G(t))(1-\sqrt{\frac{f_1}{1+G(t)}})$ and the true POS is $x_i = G(t)$, $\forall x_i \in x_{\shortparallel}$.

$$FDA3_{Camara} = \begin{cases} Minimise: f(x,t) = (f_1(x_{\shortmid}), g(x_{\shortparallel},t)\cdot(f_1(x_{\shortmid}), g(x_{\shortparallel},t))) \\ f_1(x_{\shortmid},t) = x_1^{F(t)}, \\ g(x_{\shortparallel},t) = 1 + G(t) + \sum_{x_i\in x_{\shortparallel}}(x_i - G(t))^2, \\ h(f_1,g) = 1 - \sqrt{\frac{f_1}{g}} \\ where: \\ G(t) = |\sin(0.5\pi t)|, \\ F(t) = 10^{\sin(0.5\pi t)},\ t = \frac{1}{n_t}\lfloor\frac{\tau}{\tau_t}\rfloor, \\ x_{\shortmid} \in [0,1];\ x_{\shortparallel} \in [-1,1],\ x_{\shortparallel} = (x_2,...,x_n) \end{cases} \tag{10}$$

With reference to 3.4, $FDA3_{Camara}$ is a Type II DMOOP. The true POF is $f_2 = (1+G(t))(1-\sqrt{\frac{f_1}{1+G(t)}})$ and the true POS is $x_i = G(t)$, $\forall x_i \in x_{\shortparallel}$.

$$dMOP2 = \begin{cases} Minimise: f(x,t) = (f_1(x_{\shortmid}), g(x_{\shortparallel},t)\cdot(f_1(x_{\shortmid}), g(x_{\shortparallel},t))) \\ f_1(x_{\shortmid}) = x_1, \\ g(x_{\shortparallel},t) = 1 + 9\sum_{x_i\in x_{\shortparallel}}(x_i - G(t))^2, \\ h(f_1,g,t) = 1 - \left(\frac{f_1}{g}\right)^{H(t)} \\ where: \\ H(t) = 0.75\sin(0.5\pi t) + 1.25, \\ G(t) = \sin(0.5\pi t),\ t = \frac{1}{n_t}\lfloor\frac{\tau}{\tau_t}\rfloor, \\ x_i \in [0,1];\ x_{\shortmid} \in (x_1),\ x_{\shortparallel} = (x_2,...,x_n) \end{cases} \tag{11}$$

With reference to 3.4, $dMOP2$ is a Type II DMOOP. The true POF is $f_2 = 1 - f_1^{H(t)}$ and the true POS is $x_i = G(t)$, $\forall x_i \in x_{\shortparallel}$.

$$
dMOP3 = \begin{cases} Minimise: f(x,t) = (f_1(x_{\text{I}}), g(x_{\text{II}},t) \cdot (f_1(x_{\text{I}}), g(x_{\text{II}},t))) \\ f_1(x_{\text{I}}) = x_r, \\ g(x_{\text{II}},t) = 1 + 9\sum_{x_i \in x_{\text{II}} \backslash x_r} (x_i - G(t))^2, \\ h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} \\ where: \\ G(t) = \sin(0.5\pi t),\ t = \frac{1}{n_t}\lfloor\frac{\tau}{\tau_t}\rfloor, \\ x_i \in [0,1];\ r = U(1,2,...,n) \end{cases} \tag{12}
$$

With reference to 3.4, $dMOP3$ is a Type I DMOOP. The true POF is $f_2 = 1 - \sqrt{f_1}$ and the true POS is $x_i = G(t)$, $\forall x_i \in x_{\text{II}} \backslash x_r$.

$$
dMOP2_{iso} = \begin{cases} Minimise: f(x,t) = (f_1(x_{\text{I}}), g(x_{\text{II}},t) \cdot (f_1(x_{\text{I}}), g(x_{\text{II}},t))) \\ f_1(x_{\text{I}}) = x_1, \\ g(x_{\text{II}},t) = 1 + 9\sum_{x_i \in x_{\text{II}}} (y(x_i) - G(t))^2, \\ h(f_1, g, t) = 1 - \left(\frac{f_1}{g}\right)^{H(t)} \\ where: \\ y(x_i) = G(t) + min(0, \lfloor x_i - 0.001 \rfloor)\frac{G(t)(0.001 - x_i}{0.001} - \\ \qquad min(0, \lfloor 0.05 - x_i \rfloor)\frac{(1-G(t))(x_i - 0.05)}{1 - 0.05}, \\ H(t) = 0.75\sin(0.5\pi t) + 1.25, \\ G(t) = \sin(0.5\pi t),\ t = \frac{1}{n_t}\lfloor\frac{\tau}{\tau_t}\rfloor, \\ x_{\text{I}} \in [0,1];\ x_{\text{I}} \in (x_1),\ x_{\text{II}} = (x_2, ..., x_n) \end{cases} \tag{13}
$$

With reference to 3.4, $dMOP2_{iso}$ is a Type II DMOOP. The true POF is $f_2 = 1 - f_1^{H(t)}$ and the true POS is $x_i = G(t)$, $\forall x_i \in x_{\text{II}}$.

$$dMOP2_{dec} = \begin{cases} Minimise: f(x,t) = (f_1(x_{\text{I}}), g(x_{\text{II}},t) \cdot (f_1(x_{\text{I}}), g(x_{\text{II}},t))) \\ f_1(x_{\text{I}}) = x_1, \\ g(x_{\text{II}},t) = 1 + 9\sum_{x_i \in x_{\text{II}}} (y(x_i) - G(t))^2, \\ h(f_1,g,t) = 1 - \left(\frac{f_1}{g}\right)^{H(t)} \\ where: \\ y(x_i) = \Big(\frac{\lfloor x_i - 0.35 + 0.001 \rfloor \left(1 - 0.05 + \frac{0.35-0.001}{0.001}\right)}{0.35-0.001} + \frac{1}{0.001} + \\ \qquad \frac{\lfloor 0.35 + 0.001 - x_i \rfloor \left(1 - 0.05 + \frac{1-0.35-0.001}{0.001}\right)}{1-0.35-0.001}\Big)(|x_i - 0.35| - 0.001) + 1, \\ H(t) = 0.75\sin(0.5\pi t) + 1.25, \\ G(t) = \sin(0.5\pi t),\ t = \frac{1}{n_t}\lfloor \frac{\tau}{\tau_t} \rfloor, \\ x_{\text{I}} \in [0,1];\ x_{\text{I}} \in (x_1),\ x_{\text{II}} = (x_2, ..., x_n) \end{cases} \tag{14}$$

With reference to 3.4, $dMOP2_{dec}$ is a Type II DMOOP. The true POF is $f_2 = 1 - f_1^{H(t)}$ and the true POS is $x_i = G(t)$, $\forall x_i \in x_{\text{II}}$.

$$HE1 = \begin{cases} Minimise: f(x,t) = (f_1(x_{\text{I}}), g(x_{\text{II}},t) \cdot (f_1(x_{\text{I}}), g(x_{\text{II}},t))) \\ f_1(x_{\text{I}}) = x_1, \\ g(x_{\text{II}}) = 1 + \frac{9}{n-1}\sum_{x_i \in x_{\text{II}}} x_i, \\ h(f_1,g,t) = 1 - \sqrt{\frac{f_1}{g}} - \frac{f_1}{g}\sin(10\pi t f_1) \\ where: \\ t = \frac{1}{n_t}\lfloor \frac{\tau}{\tau_t} \rfloor, \\ x_i \in [0,1];\ x_{\text{I}} = (x_1);\ x_{\text{II}} = (x_2, ..., x_n) \end{cases} \tag{15}$$

With reference to 3.4, $HE1$ is a Type III DMOOP. The true POF is $f_2 = 1 - \sqrt{f_1} - f_1\sin(10\pi t f_1)$ and the true POS is $x_i = 0$, $\forall x_i \in x_{\text{II}}$.

$$HE2 = \begin{cases} Minimise: f(x,t) = (f_1(x_{\text{I}}), g(x_{\text{II}},t) \cdot (f_1(x_{\text{I}}), g(x_{\text{II}},t))) \\ f_1(x_{\text{I}}) = x_i, \\ g(x_{\text{II}}) = 1 + \frac{9}{n-1}\sum_{x_i \in x_{\text{II}}} x_i, \\ h(f_1,g,t) = 1 - \left(\sqrt{\frac{f_1}{g}}\right)^{H(t)} - \left(\frac{f_1}{g}\right)^{H(t)} \sin(10\pi f_1) \\ where: \\ H(t) = 0.75\sin(0.5\pi t) + 1.25;\ t = \frac{1}{n_t}\lfloor\frac{\tau}{\tau_t}\rfloor, \\ x_i \in [0,1];\ x_{\text{I}} = (x_1);\ x_{\text{II}} = (x_2, ..., x_n) \end{cases} \quad (16)$$

With reference to 3.4, $HE2$ is a Type III DMOOP. The true POF is $f_2 = 1 - \sqrt{f_1}^{H(t)} - f_1^{H(t)}\sin(10\pi f_1)$ and the true POS is $x_i = 0$, $\forall x_i \in x_{\text{II}}$.

$$HE3 = \begin{cases} Minimise: f(x,t) = (f_1(x_{\text{I}}), g(x_{\text{II}},t) \cdot (f_1(x_{\text{I}}), g(x_{\text{II}},t))) \\ f_1(x_i) = x_1 + \frac{2}{|J_1|}\sum_{j \in J_1}\left(x_j - x_1^{0.5\left(1.0+\frac{3(j-2)}{n-2}\right)}\right)^2, \\ g(x_i) = 2 - \sqrt{x_1} + \frac{2}{|J_2|}\sum_{j \in J_2}\left(x_j - x_1^{0.5\left(1.0+\frac{3(j-2)}{n-2}\right)}\right)^2, \\ h(f_1,g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)} \\ where: \\ H(t) = 0.75\sin(0.5\pi t) + 1.25;\ t = \frac{1}{n_t}\lfloor\frac{\tau}{\tau_t}\rfloor, \\ J_1 = \{j|\text{ j is odd and } 2 \le j \le n\}, \\ J_2 = \{j|\text{ j is even and } 2 \le j \le n\}, \\ x_i \in [0,1] \end{cases} \quad (17)$$

With reference to 3.4, $HE3$ is a Type III DMOOP. The true POF is $f_2 = (2 - \sqrt{x_1})\left[1 - \left(\frac{x_1}{2-\sqrt{x_1}}\right)^{H(t)}\right]$ and the true POS is $x_j = x_1^{0.5\left(\frac{3(j-2)}{n-2}\right)}$, $\forall j = 2, 3, ..., n$.

$$HE6 = \begin{cases} Minimise: f(x,t) = (f_1(x_{\text{I}}), g(x_{\text{II}},t) \cdot (f_1(x_{\text{I}}), g(x_{\text{II}},t))) \\ f_1(x_i) = x_1 + \frac{2}{|J_1|}\sum_{j\in J_1}\left(x_j - 0.8x_1\cos\left(\frac{6\pi x_1 + \frac{j\pi}{n}}{3}\right)\right)^2, \\ g(x_i) = 2 - \sqrt{x_1} + \frac{2}{|J_2|}\sum_{j\in J_2}\left(x_j - 0.8\cos\left(6\pi x_1 + \frac{j\pi}{n}\right)\right)^2, \\ h(f_1,g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)} \\ where: \\ H(t) = 0.75\sin(0.5\pi t) + 1.25;\ t = \frac{1}{n_t}\lfloor\frac{\tau}{\tau_t}\rfloor, \\ J_1 = \{j|\text{ j is odd and } 2 \le j \le n\}, \\ J_2 = \{j|\text{ j is even and } 2 \le j \le n\}, \\ x_1 \in [0,1];\ x_i \in [-1,1];\ \forall i = 2,3,...,n \end{cases} \tag{18}$$

With reference to 3.4, $HE6$ is a Type III DMOOP. The true POF is $f_2 = (2-\sqrt{x_1})\left[1-\left(\frac{x_1}{2-\sqrt{x_1}}\right)^{H(t)}\right]$ and the true POS is $x_j = 0.8x_1\cos\left(\frac{6\pi x_1 + \frac{j\pi}{n}}{3}\right)$, $\forall j \in J_1$ and $x_j = 0.8x_1\sin(6\pi x_1 + \frac{j\pi}{n})$, $\forall j \in J_2$.

$$HE7 = \begin{cases} Minimise: f(x,t) = (f_1(x_{\text{I}}), g(x_{\text{II}},t) \cdot (f_1(x_{\text{I}}), g(x_{\text{II}},t))) \\ f_1(x_i) = x_1 + \frac{2}{|J_1|}\sum_{j\in J_1}\left(x_j - \left[0.3x_1^2\cos\left(24\pi x_1 + \frac{4j\pi}{n}\right)\right.\right. \\ \qquad\qquad \left.\left. +0.6x_1\right]\cos\left(6\pi x_1 + \frac{j\pi}{n}\right)\right)^2, \\ g(x_i) = 2 - \sqrt{x_1} + \frac{2}{|J_2|}\sum_{j\in J_2}\left(x_j - \left[0.3x_1^2\cos(24\pi x_1 + \frac{4j\pi}{n})\right.\right. \\ \qquad\qquad \left.\left. +0.6x_1\right]\sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right)^2, \\ h(f_1,g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)} \\ where: \\ H(t) = 0.75\sin(0.5\pi t) + 1.25;\ t = \frac{1}{n_t}\lfloor\frac{\tau}{\tau_t}\rfloor, \\ J_1 = \{j|\text{ j is odd and } 2 \le j \le n\}, \\ J_2 = \{j|\text{ j is even and } 2 \le j \le n\}, \\ x_1 \in [0,1];\ x_i \in [-1,1];\ \forall i = 2,3,...,n \end{cases} \tag{19}$$

With reference to 3.4, $HE6$ is a Type III DMOOP. The true POF is $f_2 = (2-\sqrt{x_1})\left[1-\left(\frac{x_1}{2-\sqrt{x_1}}\right)^{H(t)}\right]$ and the true POS is

$x_j = a\cos\left(\frac{6\pi x_1 + \frac{j\pi}{n}}{3}\right)$, $\forall j \in J_1$ and $x_j = a\sin(6\pi x_1 + \frac{j\pi}{n})$, $\forall j \in J_2$, with $a = \left[0.3x_1^2 \cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1\right]$.

4.5 Performance Measures

Adhering to the analyses made by Helbig [Hel12], the performance measures: NS, *stab* and acc_{alt} are be used to evaluate the performance of the D-MGPSO and D-QMGPSO algorithms. These measures are applied to the results of all benchmark functions defined in section 4.4. For these experiments, POF solutions are created for each DMOOP by dividing the range of the first objective value into one thousand equally sized intervals. The respective second objective values are generating using the equation of the true POF, as defined in 4.4. This process was followed for each $n_t - \tau_t$ combination.

4.5.1 Performance Measure 1: NS

The number of non-dominated solutions (NS) in the found POF. Although this measure does not provide any information with regards to the quality of the found solutions, it provides additional information which can be used to compare the performance of the two distinct algorithms.

4.5.2 Performance Measure 2: acc_{alt}

Alternative Accuracy Measure
The hypervolume distance (HVD) is used to measure the quality of the found POF.

$HVD = HV(POF')$ - $HV(POF^*)$,

where POF^* is the POF found by the algorithm and POF' is the true POF generated for the benchmark problem.

$acc_{alt}(t) = |HV(POF'(t))$ - $HV(POF^*(t))|$

where $acc_{alt}(t)$, the alternative accuracy measure, is the absolute HVD with regards to time t. A lower acc_{alt} portrays a more desirable performance.

4.5.3 Performance Measure 3: *stab*

Stability Measure
The affect of the changes in the environment on the accuracy of the algorithm can be quantified by the measure of stability:

$$stab(t) = max\{0, acc_{alt}(t-1) - acc_{alt}(t)\}$$

Where $stab(t)$, the stability measure, is the difference between alternative accuracy with regards to time t. A lower *stab* portrays a more desirable performance.

5 Results

This section discusses the results obtained from all executed experiments. Results for each performance measure are given with a pair of tables, for the respective D-MGPSO and D-QMGPSO algorithms. For each pair of tables, the D-MGPSO and the D-QMGPSO are compared according to the archive management strategy per dimension category. The results of each category are depicted in wins, losses, difference between wins and losses and ranked score. Each value displayed in the table, represents the overall average for the respective performance measure for all archive's found. There are over 30 independent runs of each benchmark function, for all n_t, τ_t combinations.

5.1 Results for Non-Dominated Solutions

Table 1: D-MGPSO Non-dominated solutions for all n_t, τ_t combinations

			Archive Management Strategy			
PM	Dimension	Results	1	2	3	4
NS	Large	Wins	56	3	77	8
NS	Large	Losses	88	141	67	136
NS	Large	Diff	−32	−138	10	−128
NS	Large	Rank	4	7	1	5
NS	Low	Wins	57	5	73	9
NS	Low	Losses	87	139	71	135
NS	Low	Diff	−30	−134	2	−126
NS	Low	Rank	3	7	1	6
NS	Medium	Wins	57	17	69	1
NS	Medium	Losses	87	127	75	143
NS	Medium	Diff	−30	−110	−6	−142
NS	Medium	Rank	3	5	2	7

Table 2: D-QMGPSO Non-dominated solutions for all n_t, τ_t combinations

			Archive Management Strategy			
PM	Dimension	Results	1	2	3	4
NS	Large	Wins	59	3	74	8
NS	Large	Losses	85	141	70	136
NS	Large	Diff	−26	−138	4	−128
NS	Large	Rank	3	7	2	5
NS	Low	Wins	51	5	73	15
NS	Low	Losses	93	139	71	129
NS	Low	Diff	−42	−134	2	−114
NS	Low	Rank	4	7	1	5
NS	Medium	Wins	56	1	82	5
NS	Medium	Losses	88	143	62	139
NS	Medium	Diff	−32	−142	−20	−134
NS	Medium	Rank	4	7	1	6

5.2 Results for Alternative Accuracy

Table 3: D-MGPSO acc_{alt} results for all n_t, τ_t combinations

			Archive Management Strategy			
PM	Dimensions	Results	1	2	3	4
acc_{alt}	Large	Wins	34	11	30	53
acc_{alt}	Large	Losses	94	117	98	75
acc_{alt}	Large	Diff	-60	-106	-68	-22
acc_{alt}	Large	Rank	4	7	5	1
acc_{alt}	Low	Wins	86	16	8	18
acc_{alt}	Low	Losses	42	112	120	110
acc_{alt}	Low	Diff	44	-96	-112	-92
acc_{alt}	Low	Rank	2	5	7	4
acc_{alt}	Medium	Wins	49	11	31	37
acc_{alt}	Medium	Losses	79	117	97	91
acc_{alt}	Medium	Diff	-30	-106	-66	-54
acc_{alt}	Medium	Rank	2	7	5	4

Table 4: D-QMGPSO acc_{alt} for all n_t, τ_t combinations

			Archive Management Strategy			
PM	Dimensions	Results	1	2	3	4
acc_{alt}	Large	Wins	42	48	9	29
acc_{alt}	Large	Losses	86	80	119	99
acc_{alt}	Large	Diff	-44	-32	-110	-70
acc_{alt}	Large	Rank	3	2	8	6
acc_{alt}	Low	Wins	87	30	2	9
acc_{alt}	Low	Losses	41	98	126	119
acc_{alt}	Low	Diff	46	-68	-124	-110
acc_{alt}	Low	Rank	1	3	8	6
acc_{alt}	Medium	Wins	59	39	10	20
acc_{alt}	Medium	Losses	69	89	118	108
acc_{alt}	Medium	Diff	-10	-50	-108	-88
acc_{alt}	Medium	Rank	1	3	8	6

5.3 Results for Stability

Table 5: D-MGPSO *stab* for all n_t, τ_t combinations

			Archive Management Strategy			
PM	Dimensions	Results	1	2	3	4
stab	Large	Wins	27	33	22	46
stab	Large	Losses	101	95	106	82
stab	Large	Diff	−74	−62	−84	−36
stab	Large	Rank	4	3	6	2
stab	Low	Wins	65	29	8	26
stab	Low	Losses	63	99	120	102
stab	Low	Diff	2	−70	−112	−76
stab	Low	Rank	2	4	7	5
stab	Medium	Wins	32	36	25	35
stab	Medium	Losses	96	92	103	93
stab	Medium	Diff	−64	−56	−78	−58
stab	Medium	Rank	4	2	5	3

Table 6: D-QMGPSO *stab* for all n_t, τ_t combinations

			Archive Management Strategy			
PM	Dimensions	Results	1	2	3	4
stab	Large	Wins	18	73	12	25
stab	Large	Losses	110	55	116	103
stab	Large	Diff	−92	18	−104	−78
stab	Large	Rank	7	1	8	5
stab	Low	Wins	69	33	5	21
stab	Low	Losses	59	95	123	107
stab	Low	Diff	10	−62	−118	−86
stab	Low	Rank	1	3	8	6
stab	Medium	Wins	24	63	18	23
stab	Medium	Losses	104	65	110	105
stab	Medium	Diff	−80	−2	−92	−82
stab	Medium	Rank	6	1	8	7

5.4 Analysis

5.4.1 Non-Dominated Solutions

According to Table 5.1, it can be seen that the D-QMGPSO algorithm, at the medium dimension category provides the best ranked result for the number of non-dominated solutions with 82 wins. In Table 5.1, it can be seen that the D-MGPSO algorithm, at the large dimension category, provides the best ranked result for the number of non-dominated solutions with 77 wins. The best ranked result for the number of non-dominated solutions for the low dimension category is shared between the D-QMGPSO and the D-MGPSO algorithms, with 73 wins for both. It is found that the best ranked results for all three dimension categories are a result of either the D-MGPSO or the D-QMGPSO algorithms with the strategy 3 applied.

The result of strategy 3 performing the best can be expected. As stated in section 4.3.1, after the environment changes, strategy 1 keeps all solutions in the archive, allowing old dominated solutions to guide new particle positions. This allows the archive to remain populated with dominated solutions. As stated in Section 4.3.2, after the environment changes, strategy 2 removes all solutions out of the archive. Therefore, solutions that remain non-dominated after the environment changes are removed as well. This behaviour never allows the archive to reach a large capacity of solutions. However, as stated in Section 4.3.3 the behaviour of strategy 3 is to keep non-dominated solutions in the archive after each environment change. This behaviour ensues that the number of non-dominated solutions in archive (NS) remains high throughout the algorithms execution.

5.4.2 Alternative Accuracy

From Tables 5.2, 5.2, it can be seen that the D-QMGPSO algorithm, at the low and medium categories, provides the best ranked result for alternative accuracy, with 87 and 59 wins respectively. The D-MGPSO algorithm, at the low and medium categories, provides the second best ranked results for alternative accuracy, with 86 and 49 wins respectively. It is found that the best ranked results for both the low and medium dimension categories are a result of either the D-MGPSO or the D-QMGPSO algorithms with the first strategy applied.

The alternative accuracy performance measure gives an insight into how well the found POF matches the true POF for the respective benchmark

n_t, τ_t combination. As shown by the stated results, the D-QMGPSO algorithm implemented with strategy 1, produces the best overall performance for the alternative accuracy performance measure. With reference to Section 4.1.4, the behaviour of the D-QMGPSO to re-initialised half of its particles around the current gbest particle, after each iteration could be the reason that this algorithm performance so well with reference alternative accuracy performance measure. Strategy 1 performing the second best result for the D-MGPSO is unexpected. Strategy 1 applied to the D-MGPSO allows for non-dominated solutions to remain in the archive after a change in the environment occurs. This strategy facilitates a POF that is inconsistent with the respective true POF, which should result in a poor alternative accuracy measure.

5.4.3 Stability

From Table 5.3 it can be seen that the D-QMGPSO algorithm, for all dimension categories, provides the best ranked result for the stability performance measure, with 73, 69 and 63 wins respectfully. From Table 5.3, it can be seen that the D-MGPSO algorithm, provided inconsistently ranked results for varying management strategies at all dimension categories.

The stability performance measure measures how much the alternative accuracy performance measure changes with regards to time. Therefore, the stability performance measure gives insight into how well the algorithm tracks the changing POF with each change in the environment. The results show that the D-QMGPSO with the 2nd strategy performs the most consistently with regards to alternative accuracy as the environment changes.

6 Conclusions and future work

In conclusion, the results show that variations of the MGPSO can be created to solve DMOPs. The D-MGPSO with the third archive management strategy produces the best ranked result for the non-dominated solutions performance measure. The D-QMGPSO, with first archive management strategy, produces the best ranked result for both the alternative accuracy and stability performance measure. Overall the D-QMGPSO with the first archive management strategy produces more wins for all specified performance measures than the D-MGPSO implemented with any archive management strategy.

Future work on research project could include the following. The D-QMGPSO could be implemented and tested with different distributions for the re-initialisation of the charged particles. The D-QMGPSO algorithm could be implemented and tested for a variation which is not informed when the environment changes. This would require the algorithm to monitor when the environment has changed and apply its archive management strategy appropriately. Parameter tuning could be applied to the D-MGPSO and D-QMGPSO algorithms for all archive management strategies with respect to the sixteen benchmark functions that they are tested against. Archive management strategy 4 could be implemented with the linear degrading archive influence variable initialised to a random value between 0 and 1 each time the environment changes. This would follow the logic of the original MGPSO, where the archive influence variable is originally initialised to a random value between 0 and 1. The D-QMGPSO, for archive management strategies, could be compare against other State-of-the-Art dynamic multi-Objective Optimisation algorithms.

References

[CACCL04] IEEE-Gregorio Toscano Pulido Carlos A. Coello Coello, Member and Maximino Salazar Lechuga. Handling multiple objectives with particle swarm optimization. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, 8(3):256–279, JUNE 2004.

[CSa] Andries P Engelbrecht Christiaan Scheepers. 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*.

[CSb] Andries P Engelbrecht Christiaan Scheepers. Vector evaluated particle swarm optimization exploration behavior part i: Explorative analysis. In *Proceedings of the IEEE Congress on Evolutionary Computation*.

[CS18] Andries P. Engelbrecht Christiaan Scheepers. Multi-guide particle swarm optimization a multi-swarm multi-objective particle swarm optimizer. 2018.

[Duh] Duhain. Empirical analysis of particle swarm optimization on dynamic environments. Master's thesis, University of Pretoria, Pretoria.

[Eng07a] Andries P. Engelbrecht. Particle Swarm Optimization. In Andries P. Engelbrecht, editor, *Computational Intelligence An Introduction, Second Edition*, chapter 16, pages 343–345. John Wiley Sons Ltd, 2007.

[Eng07b] Andries P. Engelbrecht. Particle Swarm Optimization. In Andries P. Engelbrecht, editor, *Computational Intelligence An Introduction, Second Edition*, chapter 16, pages 326–333. John Wiley Sons Ltd, 2007.

[Eng07c] Andries P. Engelbrecht. Particle Swarm Optimization. In Andries P. Engelbrecht, editor, *Computational Intelligence An Introduction, Second Edition*, chapter 16, pages 346–350. John Wiley Sons Ltd, 2007.

[Fie06] Sameer Fieldsend, Jonathan E. Singh. A multi-objective algorithm based upon particle swarm optimisation, an efficient data structure and turbulence. 2006.

[Hel12] Marde Helbig. *Solving dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation*. PhD thesis, University of Pretoria, Pretoria, 2012.

[JK95] R. Eberhart J. Kennedy. Particle swarm optimization. *Proceedings of the. 1995 IEEE International Conference on Neural Networks*, (4), Nov 1995.

[Li03] Xiaodong Li. A non-dominated sorting particle swarm optimizer for multiobjective optimization. In *Genetic and Evolutionary Computation — GECCO 2003 pp 37-48*. Springer Berlin Heidelberg, 2003.

[MFA04] K. Deb M. Farina and P. Amato. Dynamic multiobjective optimization problems: test cases, approximations, and applications. In *IEEE Transactions on Evolutionary Computation*, pages 425–442,, 2004.

[MH13] Andries P. Engelbrecht Marde Helbig. In *Analysing the Performance of Dynamic Multi-objective Optimisation Algorithms*. "IEEE", 2013.

[PH18] CAO Liang-lin LI Li-xian PENG Hu, LI Rong. Multiple swarms multi-objective particle swarm optimization based on decomposition. 2018.

[RN02] Carlo R. Raquel and Prospero C. Naval. Mopso: a proposal for multiple objective particle swarm optimization. In *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress*. IEEE, 2002.

[RN05] Carlo R. Raquel and Prospero C. Naval. An effective use of crowding distance in multiobjective particle swarm optimization. In *In Proc. of Genetic and and Evolutionary Conference (GECCO 2005),257–64, ACM*. Press, 2005.

[RSC06] Margarita Reyes-Sierra and Carlos A. Coello. Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. In *International Journal of Computational Intelligence Research*, number 3, page 287–308, 2006.

[SCC05] Margarita Reyes Sierra and Carlos A. Coell Coello Coello. Improving pso-based multi-objective optimization using crowding, mutation and -dominance. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization*, pages 505–519, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[SY] Licheng jiao Shuyuan Yang, Min Wang. A quantum particle swarm optimization. In *In: Evolutionary Computation, 2004. CEC2004. Congress on*. IEEE.

[Tei03] S. Mostaghim; J. Teich. Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In *Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE*, number 3, 2003.