Multi-dimensional Particle Swarm Optimization for Dynamic Environments

Serkan Kiranyaz, Jenni Pulkkinen and Moncef Gabbouj
Tampere University of Technology, Tampere, Finland
{serkan.kiranyaz, jenni.pulkkinen, moncef.gabbouj}@tut.fi

Abstract

The particle swarm optimization (PSO) was introduced as a population based stochastic search and optimization process for static environments; however, many real problems are dynamic, meaning that the environment and the characteristics of the global optimum can change over time. Thanks to its stochastic and population based nature, PSO can avoid being trapped in local optima and find the global optimum. However, this is never guaranteed and as the complexity of the problem rises, it becomes more probable that the PSO algorithm gets trapped into a local optimum due to premature convergence. In this paper, we propose novel techniques, which successfully address several major problems in the field of Particle Swarm Optimization (PSO) and promise efficient and robust solutions for multi-dimensional and dynamic problems. The first one, so-called Multi-Dimensional (MD) PSO, re-forms the native structure of swarm particles in such a way that they can make inter-dimensional passes with a dedicated dimensional PSO process. Therefore, in a multidimensional search space where the optimum dimension is unknown, swarm particles can seek for both positional and dimensional optima. This eventually removes the necessity of setting a fixed dimension a priori, which is a common drawback for the family of swarm optimizers. To address the premature convergence problem, we then propose Fractional Global Best Formation (FGBF) technique, which basically collects all the best dimensional components and fractionally creates an artificial global-best particle (aGB) that has the potential to be a better "guide" than the PSO's native gbest particle. To establish follow-up of (current) local optima, we then introduce a novel multi-swarm algorithm, which enables each swarm to converge to a different optimum and use FGBF technique distinctively. We then propose a multi-dimensional extension of the Moving Peaks Benchmark (MPB), which is a publicly available for testing optimization algorithms in a multi-modal dynamic environment. In this extended benchmark an extensive set of experiments show that MD PSO using FGBF technique with multi-swarms exhibits an impressive performance and tracks the global maximum peak with the minimum error.

1. Introduction

Many real-world problems are dynamic and thus require systematic re-optimizations due to system and/or environmental changes. Even though it is possible to handle such dynamic problems as a series of individual processes via restarting the optimization algorithm after each change, this may lead to a significant loss of useful information, especially when the change is not too drastic. Since most of such problems have multi-modal nature, which further complicates the dynamic optimization problems, the need for powerful and efficient optimization techniques is imminent. Conceptually speaking, PSO [6] is originated from the computer simulation of individuals in a bird flock or fish school [9], which basically show a natural behavior when they search for some target (e.g. food). Their goal is, therefore, to converge to the global optimum of a possibly nonlinear function or system. Similarly, in a PSO process, a swarm of particles (or agents), each of which represents a potential solution to an optimization problem, navigate through the search space. The particles are initially distributed randomly over the search space with a random velocity and the goal is to converge to the global optimum of a function or a system. Each particle keeps track of its position in the search space and its best solution so far achieved. This is the personal best value (the so-called *pbest* in [6]) and the PSO process also keeps track of the global best solution so far achieved by the swarm by remembering the index of the best particle (the so called *gbest* in [6]). During their journey with discrete time iterations, the velocity of each agent in the next iteration is affected by the best position of the swarm (the best position of the particle *gbest* as the social component), the best personal position of the particle (*pbest* as the cognitive component), and its current velocity (the memory term). Both social and cognitive components contribute randomly to the velocity of the agent in the next iteration.

There are some efforts for simulating dynamic environments in a standard and configurable way. Some early works such as [1] and [5] use experimental setup introduced by Angeline in [1]. However, this setup enables testing only in an uni-modal environment. Branke in [4] has provided a publicly available Moving Peaks Benchmark (MPB) to enable dynamic optimization algorithms to be tested in a standard way in a multi-modal environment. MPB allows creation of different dynamic fitness functions consisting of a number of peaks with varying location, height and width. The primary measure for performance evaluation is offline error, which is the average difference between the optimum and the best evaluation since the last environment change. Obviously, this value is always a positive number and it is zero only for perfect tracking. Several PSO methods are developed and tested using MPB such as [2], [3], [7], and [8]. Particularly Blackwell and Branke in [2] proposed a successful multi-swarm approach. The idea behind this is that different swarms can converge to different peaks and track them when the environment changes. The swarms interact only by mutual repulsion that keeps two swarms from converging to the same peak.

In this paper, we shall first introduce a novel algorithm that significantly improves the global convergence performance of PSO by forming an artificial Global Best particle (*aGB*) fractionally. This algorithm, the so-called Fractional GB Formation (FGBF), collects the best dimensional components from each swarm particle and fractionally creates the *aGB* particle, which will replace *gbest* as guide for the swarm, if it turns out to be better than the swarm's native *gbest*. We then propose a novel multi-swarm algorithm, which combines multi-swarms with the FGBF technique so that each swarm can apply FGBF distinctively. For the multi-dimensional dynamic environments where the optimum dimension also changes over time, we shall introduce a Multi-Dimensional (MD) PSO technique, which re-forms the native structure of swarm particles in such a way that they can make inter-dimensional passes with a dedicated dimensional PSO process. Therefore, in a multi-dimensional search space where the optimum dimension is unknown, swarm particles can seek for both positional and dimensional optima. This eventually pushes the frontier of the optimization problems in dynamic environments towards a global search in a multi-dimensional space, where the problems in each dimension are possibly multi-modal and dependent on each other in a certain manner. Since the conventional MPB is created for a unique (fixed) dimension, we shall propose an extension of the benchmark within a dimension range in which there exists a unique optimum dimension where the global (highest) peak is located. In order to provide a certain degree of dependency among individual dimensions, peaks in different dimensions share the common coordinates of the peak locations. This is basically accomplished by subtracting a penalty term, whose magnitude depends on a dimensional error, from the landscape height in non-optimal dimensions. As a result, over the extended MPB, MD PSO can seek for both optimum dimension and the global peak in it. FGBF, multi-swarms and MD PSO are generic and independent, i.e. one can be performed without others and furthermore, no additional parameter is needed to perform the proposed techniques whereas MD PSO further voids the need of fixing the dimension of the solution space in advance.

The rest of the paper is organized as follows. Section 2 surveys related work on MPB and multi-swarm PSO. The proposed techniques, MD PSO using multi-swarms with FGBF and their applications over the extended (multi-dimensional) MPB are presented in detail in Section 3. Section 4 provides the experiments conducted and discusses the results. Finally, Section 5 concludes the paper.

2. Related Work

2.1 Moving Peaks Benchmark

Conceptually speaking, MPB developed by Branke in [4], is a simulation of a configurable dynamic environment changing over time. The environment consists of a certain number peaks with varying location, height and width. The dimensionality of the fitness function is fixed in advance and thus is an input parameter of the benchmark. Type and number of peaks along with their initial heights and widths, environment dimension and size, change severity, level of change randomness and change frequency can be defined. To facilitate standard comparative evaluations among different algorithms, three standard settings of such MPB parameters, so called "*Scenarios*", have been defined. *Scenario 2* is the most widely used. Where the scenario allows a range of values, the following are commonly used: number of peaks = 10, change severity *vlength* = 1.0, correlation *lambda* =0.0 and peak change frequency = 5000. In *Scenario 2* no basis landscape is used and peak type is a simple cone. Due to the page limit more formal description and further details are skipped and can be obtained from [4].

2.2 Multi-swarm PSO

The main problem of using the basic PSO algorithm in a dynamic environment is that eventually the swarm will converge to a single peak – whether global or local. When another peak becomes the global maximum as a result of an environmental change, it is likely that the particles keep circulating close to the peak to which the swarm has converged and thus they cannot find the new global maximum. Blackwell and Branke have addressed this problem in [2] and [3] by introducing multi-swarms. Multi-swarms are actually separate PSO processes. Each particle is now a member of one of the swarms only and it is unaware of other swarms. The main idea is that each swarm can converge to a separate peak. Swarms interact only by mutual repulsion that keeps them from converging to the same peak. For a single swarm it is essential to maintain enough diversity so that the swarm can track small location changes of the peak to which it is converging. For this purpose Blackwell and Branke introduced charged and quantum swarms, which are analogues to an atom having a nucleus and charged particles randomly orbiting it. The particles in the nucleus take care of the fine tuning of the result while the charged particles are responsible of detecting the position changes. However, it is clear that, instead of charged or quantum swarms, any method can be used to ensure sufficient diversity among particles of a single swarm so that the peak can be tracked despite of small location changes. As one might expect, the best results are achieved when the number of swarms is set equal to the number of peaks.

The repulsion between swarms is realized by simply re-initializing worse of two swarms if they move within a certain range from each other. Using physical repulsion could lead to equilibrium, where swarm repulsion prevents both swarms from getting close to a peak. A proper limit closer to which the swarms are not allowed to move, r_{rep} is attained by using the average radius of the peak basin, r_{bas}. If p peaks are evenly distributed in X^N, $r_{rep}=r_{bas}=X/p^{1/N}$.

3. The Proposed Techniques for Multi-dimensional Dynamic Environments

3.1 MD PSO Algorithm

Instead of operating at a fixed dimension N, the MD PSO algorithm is designed to seek both positional and dimensional optima within a dimension range, ($D_{\min} \leq N \leq D_{\max}$). In order to accomplish this, each particle has two sets of components, each of which has been subjected to one of two independent and consecutive processes. The first one is a regular positional PSO, i.e. the traditional velocity updates and due positional shifts in *N*-dimensional search (solution) space. The second one is a dimensional PSO, which allows the particle to navigate through dimensions. Accordingly, each particle keeps track of its last position, velocity and personal best position (*pbest*) in a particular dimension so that when it re-visits the same dimension at a later time, it can perform its regular "positional" fly using this information. The dimensional PSO process of each particle may then move the particle to another dimension where it will remember its positional status and keep "flying" within the positional PSO process in this dimension, and so on. The swarm, on the other hand, keeps track of the *gbest* particles in all dimensions, each of which respectively indicates the best (global) position so far achieved and can thus be used in the regular velocity update equation for that dimension. Similarly the dimensional PSO process of each particle uses its personal best dimension in which the personal best fitness score has so far been achieved. Finally, the swarm keeps track of the global best dimension, *dbest*, among all the personal best dimensions. The *gbest* particle in *dbest* dimension represents the optimum solution and dimension, respectively.

In a MD PSO process at time (iteration) t, each particle a in the swarm, $\xi=\{x_1,..,x_a,..,x_S\}$, is represented by the following characteristics:

$xx_{a,j}^{xd_a(t)}(t)$: j^{th} component (dimension) of the position in dimension $xd_a(t)$

$vx_{a,j}^{xd_a(t)}(t)$: j^{th} component (dimension) of the velocity in dimension $xd_a(t)$

$xy_{a,j}^{xd_a(t)}(t)$: j^{th} component (dimension) of the personal best (*pbest*) position in dimension $xd_a(t)$

gbest(d) : Global best particle index in dimension d

$x\hat{y}_j^d(t)$: j^{th} component (dimension) of the global best position of swarm, in dimension d

$xd_a(t)$**:** Current Dimension

$vd_a(t)$**:** Current Dimensional velocity

$x\tilde{d}_a(t)$**:** Personal best dimension

Let f denote the dimensional fitness function that is to be optimized within a certain dimension range, $\{D_{\min}, D_{\max}\}$. Without loss of generality assume that the objective is to find the maximum (position) of f at the optimum dimension within a multi-dimensional search space. Assume that the particle a visits (back) the same dimension after T iterations (i.e. $xd_a(t)=xd_a(t+T)$), then the personal best position can be updated in iteration $t+T$ as follows,

$$xy_{a,j}^{xd_a(t+T)}(t+T)=\begin{cases} xy_{a,j}^{xd_a(t)}(t) & if\ f(xx_a^{xd_a(t+T)}(t+T))<f(xy_a^{xd_a(t)}(t)) \\ xx_{a,j}^{xd_a(t+T)}(t+T) & else \end{cases} \tag{1}$$

Furthermore, the personal best dimension of particle a can be updated in iteration $t+1$ as follows,

$$x\tilde{d}_a(t+1)=\begin{cases} x\tilde{d}_a(t) & if\ f(xx_a^{xd_a(t+1)}(t+1))<f(xy_a^{x\tilde{d}_a(t)}(t)) \\ xd_a(t+1) & else \end{cases} \tag{2}$$

Let *gbest(d)* be the index of the global best particle in dimension d and let *S(d)* be the total number of particles in dimension d, then it follows that, $x\hat{y}^{dbest}(t)=xy_{gbest(dbest)}^{dbest}(t)$, where $gbest(dbest)=\arg\max\left(f(xy_1^{dbest}(t)),...,f(xy_S^{dbest}(t))\right)$. For a particular iteration t, and for a particle $a\in\{1,S\}$, first the positional components are updated in its current dimension, $xd_a(t)$ and then the dimensional update is performed to determine its next (t+1$^{\text{st}}$) dimension, $xd_a(t+1)$. The positional update is performed for each dimension component, $j\in\{1,xd_a(t)\}$, as follows:

$$\begin{aligned} & vx_{a,j}^{xd_a(t)}(t+1)= \\ & w(t)\,vx_{a,j}^{xd_a(t)}(t)+c_1r_{1,j}(t)\left(xy_{a,j}^{xd_a(t)}(t)-xx_{a,j}^{xd_a(t)}(t)\right)+ \\ & c_2r_{2,j}(t)\left(x\hat{y}_j^{xd_a(t)}(t)-xx_{a,j}^{xd_a(t)}(t)\right) \\ & xx_{a,j}^{xd_a(t)}(t+1)=xx_{a,j}^{xd_a(t)}(t)+C_{vx}\left[vx_{a,j}^{xd_a(t)}(t+1),\{V_{\min},V_{\max}\}\right] \\ & xx_{a,j}^{xd_a(t)}(t+1)\leftarrow C_{xx}\left[xx_{a,j}^{xd_a(t)}(t+1),\ \{X_{\min},X_{\max}\}\right] \end{aligned} \tag{3}$$

where $C_{xx}[.,.]\equiv C_{vx}[.,.]$ are the clamping operators applied over each positional component, $xx_{a,j}^d$ and $vx_{a,j}^d$. $C_{xx}[.,.]$ may or may not be applied depending on the optimization problem but $C_{vx}[.,.]$ is basically needed to avoid exploding.

Due to space limitations the details of the clamping operators and pseudo-code for MD PSO are skipped in this paper.

3.2 FGBF Algorithm over MD PSO

Fractional GB formation (FGBF) is designed to avoid the premature convergence by providing a significant diversity obtained from a proper *fusion* of the swarm's best components (the individual dimension(s) of the current position of each particle in the swarm). At each iteration in a *bPSO* process, an artificial GB particle (*aGB*) is (fractionally) formed by selecting the best particle (dimensional) components from the entire swarm. Take for instance the function minimization problem as illustrated in Figure 1 where 2D space is used for illustration purposes. In the figure, three particles in a swarm are ranked as the 1$^{\text{st}}$ (or the *gbest*), the 3$^{\text{rd}}$ and the 8$^{\text{th}}$ with respect to their proximity to the target position (or the global solution) of some function. Although *gbest* particle (i.e. 1$^{\text{st}}$ rank particle) is the closest in the overall sense, the particles ranked 3$^{\text{rd}}$ and 8$^{\text{th}}$ provide the best x and y dimensions (closest to the target's respective dimensions) in the entire swarm and hence the *aGB* particle via FGBF yields a better (closer) particle than the swarm's *gbest*. Therefore, especially during the initial steps, the FGBF can be and most of the time, is a better alternative than the native *gbest* particle since it has the advantage of assessing each dimension of every particle in the swarm individually, and forming the *aGB* particle fractionally by using the best components among them.

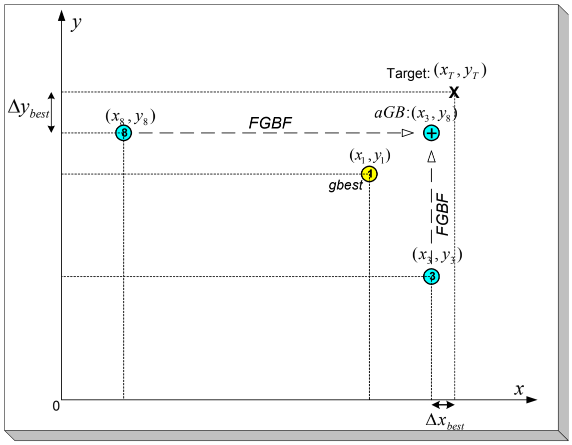

Figure 1: A sample FGBF in 2D space.

Due to space limitations the details of the FGBF operation over MD PSO are skipped in this paper.

3.3 Application over Multi-Dimensional MPB

For testing the proposed multi-dimensional optimization algorithms, we extended Branke's MPB into a multi-dimensional version, in which there exists many search space dimensions within a dimensional range $\{D_{\min}, D_{\max}\}$, and the optimal dimension changes over time in addition to the dynamic nature of the conventional MPB. Peak locations in different dimensions share the common coordinates in the fitness function and thus such an extension further allows exploitation of the information gathered in other search space dimensions. The initialization and changes of peak locations must now be done in the highest possible search space dimension, $D_{\max}$. Locations in the other dimensions can be obtained simply by leaving out the redundant coordinates (non-existing dimensions). The optimal dimension is chosen randomly every time the environment is changed. Therefore, the fitness function with m peaks in multi-dimensional environment can be expressed as,

$$F(\vec{x}^d, t) = \max\left(B\left(\vec{x}^d\right), \max_{p=1\ldots m} P\left(\vec{x}^d, d, h_p(t), w_p(t), \vec{c}_p^{\,d}(t)\right)\right) \quad (4)$$

where $d \in \{D_{\min}, D_{\max}\}$ is the dimension of position $\vec{x}^d$, where $B\left(\vec{x}^d\right)$ is time invariant basis landscape, whose utilization is optional, and P is the function defining the height of the p^{th} peak at location $\vec{x}^d$. Each of the m peaks have its own dynamic parameters height, $h_p(t)$, width, $w_p(t)$ and location (first d coordinates) of the peak center, $\vec{c}_p^{\,d}(t)$. A *cone* peak is expressed with the following expression:

$$P\left(\vec{x}^d, h_p(t), w_p(t), \vec{c}_{pd}(t)\right) = h_p(t) - s_p(t) * \left\|\vec{x}^d - \vec{c}_p^{\,d}(t)\right\| / d - (D_{opt} - d)^2 \quad \textit{where}$$

$$s_p(t) = \frac{h_p(t)}{w_p(t)} \quad \textit{and} \quad \left\|\vec{x}^d - \vec{c}_p^{\,d}(t)\right\| = \sqrt{\sum_{i=1}^{d}\left(x_i^d - c_{pi}^d\right)^2} \quad \forall x_i^d \in \vec{x}^d, \forall c_{pi}^d \in \vec{c}_p^{\,d}(t) \quad (5)$$

Where $s_p(t)$ is the slope and $\|.\|$ is the *Euclidean* distance between two d-dimensional vectors, $\vec{x}^d$ and $\vec{c}_p^{\,d}(t)$ and D_{opt} is the current optimal dimension. If compared with the conventional MPB, for all non-optimal dimensions a penalty term $(D_{opt} - d)^2$ is subtracted from the whole environment. In addition to that the peak slopes are scaled by the term $1/d$. The purpose of this scaling is to prevent the benchmark from favoring the lower dimensions. Otherwise a solution, whose every coordinate differs from the optimum by 1.0 would be a lot better solution in a lower dimension as the *Euclidian* distance is used.

Similar to the uni-dimensional (PSO) case, each positional component $xx_a^d(t)$ of MD PSO particle represents a potential solution in dimension *d*. Only difference is that now the dimensionality of the optimal solution is not known beforehand, but it can vary within the defined range. Even a single particle can provide potential solutions in different dimensions as it makes inter-dimensional passes as a result of MD-PSO process. Our dynamic multi-dimensional optimization algorithm combines multi-swarms and FGBF with MD-PSO. As in the different dimensions the common coordinates of the peak locations are the same, it does not seem purposeful for two swarms to converge to a same peak in different dimensions. Therefore, the mutual repulsion between swarms is extended to affect swarms that are in different dimensions. Obviously, only the common coordinates are considered when the swarm distance is computed.

Recall that in *Scenario 2* of MPB the peaks used are all in *cone* shape, as given in (5), $h_p(t)$ and $s_p(t)$ are both set by MPB, finding the highest peak is equivalent to minimizing the $\left\|\vec{x} - \vec{c}_p(t)\right\|$ term, yielding $f(a, j) = -\left(x_j - c_{pj}\right)^2$. We do not expect that dimensional fitness scores can be evaluated with respect to the optimum peak since this requires the *a priori* knowledge of the global optimum, instead we use either the *current* peak where the particle resides on or the peak to which the swarm is converging (swarm peak). We shall thus consider and evaluate both *modes* separately.

4. Experimental Results

On the extended MPB we conducted experiments with the settings of *Scenario 2* and the change frequency used is set to 15000. The search space dimension range used is $d \in \{5, 15\}$.

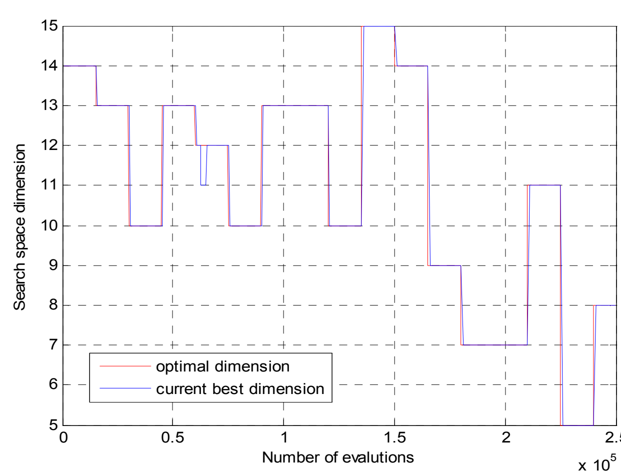

Figure 2: Optimum dimension tracking in a MD PSO run.

Figure 2 shows how the global optimal dimension changes over time and how MD PSO is tracking these changes. Current best dimension represents the dimension, where the best solution is achieved among all swarms' *dbest* dimensions. 10 multi-swarms are used with 7 particles in each. It can be seen that the algorithm always finds the optimal dimension, even though the difference in peaks heights between the optimal dimension and its neighbor dimensions is quite insignificant (=1) compared to the peak heights (30-70). Figure 3 shows how the current error behaves during the first 250000 evaluations, when the same settings are used. We note that the initial converging phase, when the algorithm has not been yet behaving at its best is longer. Similarly it takes longer time to regain the optimal behavior if follow-up of some peaks is lost for some reason (it is, for example, possible that higher peaks hide other lower peaks under them).

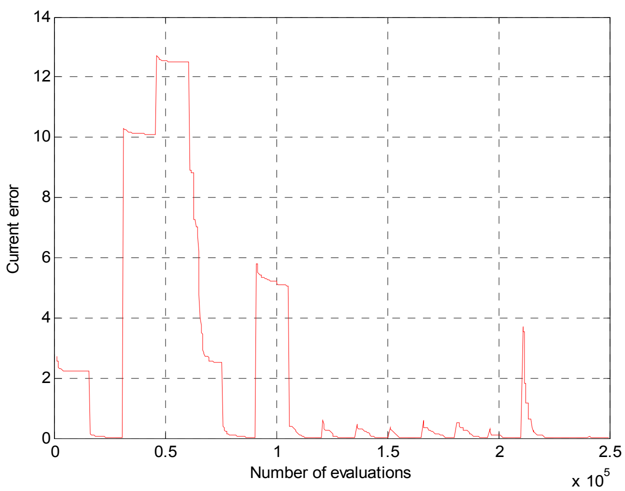

Figure 3: Current error in a MD PSO run.

The numerical results in terms of offline error are given in Table 1. Each result given is the average of 50 runs, where each run consists of 500000 function evaluations. As in the uni-modal case, best results are achieved when the number of swarms is equal to the number of peaks, which is 10. Interestingly when the swarm peak *mode* is used the optimal number of particles becomes 7 while with current peak mode it is still 4. Note that these results cannot be directly compared with the results on conventional MPB since the objective function of multi-dimensional MPB is somewhat different.

Table 1: Offline error on extended MPB

No. of swarms	No. of particles	Swarm peak	Current peak
10	4	2.01±0.98	**3.29±1.44**
10	5	1.77±0.83	3.41±1.69
10	6	1.79±0.98	3.64±1.60
10	7	**1.69±0.75**	3.71±1.74
10	8	1.84±0.97	4.21±1.83
10	10	1.96±0.94	4.20±2.03
8	7	1.79±0.91	3.72±1.86
9	7	1.83±0.84	4.30±2.15
11	7	1.75±0.91	3.52±1.40
12	7	2.03±0.97	4.01±1.97

5. Conclusions

In this paper, we proposed two novel PSO techniques, namely, FGBF with the multi-swarms and MD PSO for an efficient and robust optimization over the dynamic systems. Both techniques can also be used over the static optimization problems particularly as a cure to common drawbacks of the family of PSO methods such as *a priori* fixation of the search space dimension and pre-mature convergence to local optima. MD PSO efficiently addresses the former drawback by defining a new particle structure and embedding the ability of dimensional navigation into the core of the process. It basically allows particles to make inter-dimensional 'passes' with a dedicated PSO process whilst performing regular positional updates in every dimension they visit. Although the ability of determining the optimum dimension where the global solution exists is gained with MD PSO, its convergence performance is still limited to the same level as *bPSO*, which suffers from the lack of diversity among particles. When performed with FGBF and multi-swarms, MD PSO exhibits a global and faster convergence ability. The experiments conducted over the extended (multi-dimensional) MPB approve that the proposed MD PSO technique with multi-swarms and FGBF always finds and tracks the optimum dimension where the global peak resides. The mutual applications of the proposed techniques generally find and track the global optimum peak, yet it can occasionally converge to a near-optimum peak, particularly if the height difference happens to be insignificant.

Overall, the proposed techniques fundamentally upgrade the particle structure and the swarm guidance, both of which accomplish substantial improvements in terms of speed and accuracy. Both techniques are modular and independent from each other, i.e. one can be performed without the other whilst other PSO methods/variants can also be performed conveniently with (either of) them.

References

[1] P.J. Angeline, "Tracking extrema in dynamic environments", *In Proc. Of the 6th Conf. on Evolutionary Programming,* pp.335-345, Springer Verlag, 1997.

[2] T.M. Blackwell and J. Branke, "Multi-Swarm Optimization in Dynamic Environments", *Applications of Evolutionary Computation*, vol. 3005, pp. 489-500, Springer, 2004.

[3] T.M. Blackwell and J. Branke, "Multiswarms, Exclusion, and Anti-Convergence in Dynamic Environments", *IEEE Transactions on Evolutionary Computation,*, vol. 10/4, pp. 51-58, 2004.

[4] J. Branke, "Moving Peaks Benchmark", http://www.aifb.uni-karlsruhe.de/~jbr/MovPeaks/ , viewed 26/06/08 .

[5] R. Eberhart and Y. Shi, "Tracking and Optimizing Dynamic Systems with Particle Swarms", in *Proc. of Computational Evolution Conference (CEC 2001)*, NJ, US, pp. 94-100, 2001.

[6] J. Kennedy, R Eberhart., "Particle swarm optimization", in *Proc. of IEEE Int. Conf. On Neural Networks*, vol. 4, pp. 1942–1948, Perth, Australia, 1995.

[7] X. Li, J. Branke and T. Blackwell, "Particle Swarm with Speciation and Adaptation in a Dynamic Environment", *Proc. of Genetic and Evolutionary Computation Conf.,* pp. 51-58, US, 2006.

[8] R. Mendes and A. Mohais, "DynDE: a Differential Evolution for Dynamic Optimization Problems", *IEEE Congress on Evolutionary Computation,* pp. 2808-2815, 2005.

[9] E.O. Wilson, *Sociobiology: The new synthesis*, Cambridge, MA: Belknap Press, 1975.