

On Extending Quantum Behaved Particle Swarm Optimization to MultiObjective Context

Heyam AlBaity
Computer Science Department
University of Birmingham, UK &
King Saud University, Riyadh, KSA
hha090@cs.bham.ac.uk

Souham Meshoul
CS Dept. , MISC Laboratory
University Mentouri, Constantine
smeshoul@umc.edu.dz

Ata Kaban
Computer Science Department
University of Birmingham, UK
A.Kaban@cs.bham.ac.uk

Abstract— Quantum behaved particle swarm optimization (QPSO) is a recently proposed metaheuristic, which describes bird flocking trajectories by a quantum behavior. It uses only one tunable parameter and suggests a new and interesting philosophy for moving in the search space. It has been successfully applied to several problems. In this paper, we investigate the possibility of extending QPSO to handle multiple objectives. More specifically, we address the way global best solutions are recorded within an archive and used to compute the local attractor point of each particle. For this purpose, a two level selection strategy that uses sigma values and crowding distance information has been defined in order to select the suitable guide for each particle. The rationale is to help convergence of each particle using sigma values while favoring less crowded regions in the objective space to attain a uniformly spread out Pareto front. The proposed approach has been assessed on test problems for function optimization from convergence and diversity points of view. Very competitive results have been achieved compared to some state of the art algorithms.

Keywords- multi objective optimization, quantum behaved particle swarm optimization, local attractor, function optimization.

I. INTRODUCTION

Recently quantum computing principles have been introduced into evolutionary algorithms (EAs) and Particle swarm Optimization (PSO). While some of them focus on quantum representation of individuals [1], others suggest the use of a quantum behavior defining a new philosophy for exploring the search space [2]. This later was the basic idea behind Quantum Behaved Particle Swarm Optimization (QPSO) algorithm, a quantum variant of PSO algorithm. It has been introduced by Sun et al. [2] and has been shown as a promising algorithm for many optimization problems. Like PSO, QPSO is characterized by its simplicity and easy implementation. Besides, it has better search ability and fewer parameters to set when compared against PSO [3]. Actually, only one tunable parameter is required. With the new search philosophy it suggests, QPSO could improve the convergence capability of the global optimization [4]. QPSO has been successfully applied to many real world applications [5,6]. The promising results reported in [2,4] comparing QPSO to PSO in the single objective domain motivate us to develop and investigate an extension of QPSO to handle multi-objective optimization problems. Many real world optimization problems often have multiple objectives

that are generally conflicting and need to be taken into account during the optimization process. In general, a multi-objective optimization problem (MOP) does not have a single solution that could optimize all objectives simultaneously. In MOP, the major challenge is to find the set of solutions that achieve the best compromise with regard to the whole set of objectives. Although a great deal of effort has been devoted to solve multi-objective optimization problems, the problem is still open and the related issues still attract interest of researchers. Most of the proposed approaches make use of metaheuristics. Their basic idea is to introduce the Pareto dominance concept into nature inspired algorithms such as genetic algorithms (GAs) and particle Swarm Optimization (PSO) [7,8]. Evolutionary multi-objective optimization techniques are well suited for tackling the MOP because as they are population based, they are able to generate a set of solutions from which the non-dominated solutions can be found in a single simulation run. Moreover, in [9], it has been stated that the behavior of an evolutionary algorithm is not affected by the Pareto surface shape. Quantum principles like states superposition and interference have been introduced to multiobjective genetic algorithms [10, 11].

However, GA based algorithms for MOP require the setting of many genetic parameters [12]. Therefore, following the success of the use of PSO to solve single-objective optimization problems and due to its simplicity and easy implementation, it is natural to ask whether it would be possible to extend its use to tackle MOPs especially that it requires less tunable parameters than evolutionary algorithms do. Multi-objective Particle swarm optimization algorithms (MOPSO) have shown to be highly competitive with Multi-objective genetic algorithms [13]. This fact further motivates the use of QPSO in multiobjective context. With its current form, QPSO cannot be used in a straightforward way for multiobjective optimization. Many issues should be addressed among which the most important is how to deal with the global best particles and how to select the one to compute the local attractor of a given particle. Extending QPSO to multiobjective optimization is not yet well investigated in the literature. To our knowledge, only one published paper treated the topic [14] to deal with a specific application related to laminated composite structures. The authors of the paper incorporated a vector evaluated technique within

QPSO taking inspiration from Vector Evaluated PSO (VEPSO) [15].

In an attempt to achieve good convergence of particles and uniformly spread out fronts, we propose selecting a particle leader based on two criteria namely sigma values [16] and crowding distance information [17]. The first one fosters convergence of particles while the second one fosters uniform distribution of nondominated solutions along the Pareto Front. A global best archive is used to keep nondominated solutions encountered during the search process and a two level selection strategy is developed and introduced into a QPSO algorithm to solve multiobjective optimization.

The remainder of the paper is organized as follows: In section 2, we provide the background that underlies this work. It encompasses the general MOP formulation along with some useful definitions as well as a descriptions of QPSO. In section 3, a review of the work done to solve multi-objective optimization problems is presented with a focus on PSO based techniques. Section 4 is devoted to the framework we propose to extend QPSO to multi-objective context. Section 5, reports on conducted experiments and obtained results. Finally, conclusions and perspectives are provided in Section 6.

II. BACKGROUND

A. Multiobjective optimization problem formulation

Three basic elements define any MOP namely: a set of decision variables, a set of objectives and a set of constraints. Basically, a MOP consists in exploring the space of decision variables in order to find the vector of variables that optimizes the set of objectives while satisfying the set of constraints [8]. More formally, a MOP is defined as follows [8]:

$$\begin{aligned} & \text{Optimize} && f_m(\vec{x}), && m = 1, 2, \dots, M \\ & \text{subject to} && \\ & && g_j(\vec{x}) \geq 0, && j = 1, 2, \dots, J \\ & && h_k(\vec{x}) = 0, && k = 1, 2, \dots, K \\ & \text{and} && x_i^{(L)} \leq x_i \leq x_i^{(U)}, && i = 1, 2, \dots, n \end{aligned} \quad (1)$$

Where,

- \vec{x} is a vector of n decision variables. $\vec{x} = (x_1, x_2, \dots, x_n)^T$.
- M is the number of the objective functions and $\vec{F}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \dots, f_M(\vec{x}))^T$ is the vector of objective functions.
- $g_j(\vec{x})$ and $h_k(\vec{x})$ refer to inequality and equality constraint functions.
- $x_i^{(L)}$ and $x_i^{(U)}$ are the lower bound and the upper bound of decision variable x_i .

Without loss of generality, optimization may refer to a minimization as well as a maximization problem. In the following we suppose the minimization case.

In order to compare two different solutions, we make use of the notion of Pareto dominance [18].

Pareto Dominance: A vector $u = (u_1, \dots, u_k)$ is said to dominate vector $v = (v_1, \dots, v_k)$ denoted by $(u \preceq v)$ if and only if (in the minimization case) :

$$\begin{aligned} & \forall i \in \{1, 2, \dots, k\}, \quad u_i \leq v_i \text{ and} \\ & \exists i \in \{1, 2, \dots, k\} : u_i < v_i \end{aligned} \quad (2)$$

Therefore, the best solutions in the sense of Pareto dominance constitute the Pareto optimal set.

Pareto optimal set: A solution is said to be Pareto optimal if and only if it is not dominated by any other solution in the search space. The set of all Pareto optimal solutions is called Pareto optimal set (P^*) and is defined as [7,18]:

$$P^* = \{ \vec{x} \in \Omega \mid \neg \exists \vec{x}' \in \Omega \vec{f}(\vec{x}') \preceq \vec{f}(\vec{x}) \} \quad (3)$$

Pareto front: Each solution vector in the Pareto optimal set corresponds to a vector of the related objectives values. Therefore, a Pareto front represents the set of Pareto optimal solutions in the objective space. Each M -dimensional point of the Pareto front is related to a solution vector [8,18].

B. Quantum Particle swarm optimisation

PSO has been added as a powerful addition to the pool of nature inspired heuristics. Developed by Kennedy and Eberhart [19], PSO is a population based metaheuristic that has been motivated by social behavior of bird flocking or fish schooling. The PSO model is derived from Newtonian physics. In a D -dimensional problem space, each particle representing a potential solution to the problem is defined by a position vector and a velocity vector. In classical PSO model, the velocity vector is updated using influence of self best performance and global best performance. A good review of PSO metaheuristic can be found in [20]. Motivated by the idea that the social behavior of a swarm of birds is much more complicated to be depicted by classical mechanics, Sun et al. [2] suggest Quantum Behaved Particle Swarm optimization using quantum mechanics. In QPSO, the particles are considered to lie in a potential field. The position of each particle is depicted by using a wave function $\Psi(x, t)$ instead of position and velocity. As such, the position and velocity cannot be specified simultaneously. Instead, the probability of a particle appearing in position x can only be obtained from the probability density function $|\Psi(x, t)|^2$ which depends on the potential field [3,4]. In QPSO, the position of a particle is updated according to the following equation:

$$x_{ij}^{t+1} = p_{ij}^t \pm \beta \cdot |mbest_j^t - x_{ij}^t| \cdot \ln(1/u_{ij}^t) \quad (4)$$

Where, u_{ij}^t is a random number within the range (0,1), β is the Contraction Expansion coefficient (CE). It is the only tunable parameter of QPSO and has a significant impact on controlling the convergence speed of the algorithm [3]. p_{ij}^t is the local attractor of particle i and is evaluated by :

$$p_{ij}^t = \phi_{ij}^t \cdot self\ best_{ij}^t + (1 - \phi_{ij}^t) \cdot global\ best_j^t \quad (5)$$

ϕ_{ij}^t is a random number within the range (0,1). Finally, $mbest$ is called the Mainstream Thought point or the Mean best position. It is the mean of self best positions of all particles and is evaluated by :

$$mbest_j^t = \frac{1}{N} \sum_{i=1}^N selfbest_i = \left(\frac{1}{N} \sum_{i=1}^N SP_{i1}, \frac{1}{N} \sum_{i=1}^N SP_{i2}, \dots, \frac{1}{N} \sum_{i=1}^N SP_{iD} \right) \quad (6)$$

Where, N is the population size, SP_i is the self best position of particle P_i and D is the dimension of the search space.

Starting from an initial swarm that is a set of particles positions, QPSO iteratively evolves each particle position according to equation (4) above. At each iteration, the self best position for each particle as well as the global best are updated when necessary. Since its inception, many improved versions have been proposed to enhance the search capabilities of the algorithm from both convergence and diversity points of view. A good review of QPSO with related improvements can be found in [3].

III. RELATED WORK

A great deal of effort had been devoted to solve multi-objective optimization problems. A good review of evolutionary algorithms is presented in [8]. In this section we focus only on PSO based methods. A survey of the related state of the art can be found in [13]. Different Pareto-based PSO approaches have been reported in the literature. This includes MultiObjective PSO (MOPSO) developed by Coello et al. [21] in which an external repository is used to keep the non-dominated solutions found during the search process. In MOPSO the objective space is divided into hypercubes using an adaptive grid in order to produce well distributed fronts. However the number of grid subdivisions is problem specific and should be well set to achieve the goal. A fitness value is assigned to each hypercube depending on the number of non dominated solutions lying into it. A roulette wheel selection scheme is used to select a hypercube and then a global best (leader) is selected randomly from the chosen hypercube to update particles positions. A mutation operator whose range of action varies over time has been used to prevent PSO premature convergence. Fieldsend and Singh [22] proposed an approach that overcomes the inefficiency of the limited archive size of previous proposals by using an

unconstrained archive called dominated tree. The global best of each particle is selected based on the closest member of the dominated tree archive. The algorithm also uses a mutation operator to promote diversity. Another approach proposed by Mostaghim and Teich [16] called Sigma-MOPSO proposes a new density measure scheme called sigma method to select the global best for each particle. Each particle in the swarm is assigned a sigma value. Then to select a gbest for a particle, the external archive member with closet sigma value of the given particle is selected as its gbest guide. The algorithm uses a mutation operator on decision variable space. Arias et al. [23] suggested an approach ST-MOPSO that introduces a new mechanism called stripes to maintain diversity for MOPs. The global best guide for each particle is its closest stripe centre. Such strip centers are the uniformly distributed points along the line obtained by objective function values. By another side NSGAI mechanism [17] has been introduced into PSO by Li [24] leading to NSPSO. As NSGAI works with 2N population, The algorithm combines the pbest positions of the swarm with all the new positions to form the 2N solutions. The global best is selected randomly according to the two niching methods (niche count and crowding distance). A mutation operator is adopted. Finally, taking inspiration from vector Evaluated Genetic Algorithm, Parsopoulos and Vrahatis proposed Vector Evaluated PSO (VEPSO) [15] to handle bi-objective optimization. In VEPSO, two swarms were used, one for each objective function. In addition, the weighted aggregation approach was also adopted. None of these methods has shown to be the best in all situations. From QPSO side, there is little insight into using it to multiobjective context. VEQPSO principles have been used by [14] to tackle an engineering application.

IV. THE PROPOSED FRAMEWORK FOR MULTIOBJECTIVE QUANTUM BEHAVED PARTICLE SWARM OPTIMIZATION

Extending QPSO to multiobjective optimization should be done in a way to find a Pareto front as close as possible to the optimal one while ensuring a smooth and uniform distribution of the nondominated solutions within it. In QPSO with single objective, the computation of each particle position requires three main components as described in section II.B namely:

- The local attractor point of the particle which is a function of the self best performance of the particle and global best performance of the entire swarm.
- The mainstream thought or mean best position which is obtained by averaging out all self best positions.
- The contraction expansion (CE) parameter β .

Therefore, in a multiobjective case, the following legitimate questions should be addressed:

- How should the local attractor be updated for a given particle?

- How should the mean-best be derived?
- What is the impact of the CE parameter on the convergence of the algorithm?

Hence, we describe the proposed framework by bringing answers to the above questions.

A. Computing local attractor points in multiobjective context

The convergence of an individual particle relies heavily on the local attractor point [2]. It helps achieving a certain balance between the self best performance of the particle and the global best performance as it is a function of both of them. With several objectives, global best positions and self best positions should be computed in a different manner than in single objective case using Pareto dominance relation.

- For global best positions, an archive is used to keep non dominated solutions found during the search process. We call such an archive Global Best Archive (GBA). Like with PSO, one issue should be clarified concerning which solution should be selected from the GBA to compute the local attractor point of a particle. The selection method we propose is inspired from the sigma method proposed in [16] and makes use of crowding distance information [17]. The rational is to help convergence of each particle using sigma values while favoring less crowded regions in the objective space to attain a uniformly spread out Pareto front using crowding distance values. The selection mechanism we propose is simple and operates in two levels as follows. Given a particle for which we need to select a global leader in order to compute its local attractor point, the procedure first determines the members in GBA that are close to the particle in terms of sigma values. Then, the less crowded solutions among these neighbors is chosen as the global best solution for the particle. This principle is illustrated in figure 1. More formally, the proposed selection method can be described as follows:

Selection_Method (GBA, P_i)

```

For Each non_dominated solution in GBA
    Compute_Sigma_value (GBAk);

Compute_Sigma_value ( $P_i$ );

Record_k_nearest_neighbors;

Sort_k_nearest_neighbors (crowding_distance_values);

Choose_less_crowded_solution;
END.

```

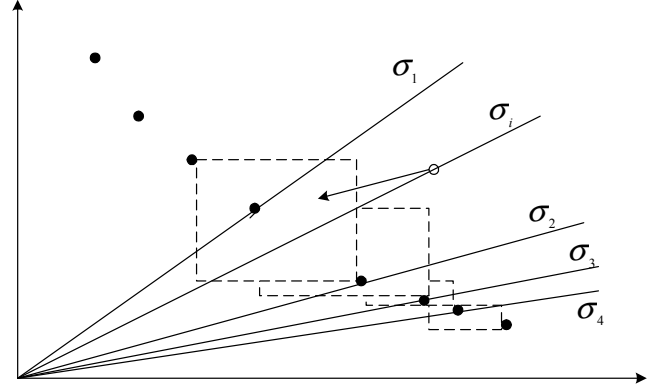


Figure 1. Selecting global leader from GBA: filled circles are GBA members, empty circle represents particle P_i

As described in [16], a sigma value of a particle P_i characterizes the line joining the corresponding point in the objective space to the center point (0,0,0). The closeness of two sigma values is in fact an indication that the two corresponding particles lie on two lines that are close to each other. This fact is used to guide the particle by the suitable leader. That is why the k nearest neighbors are selected according to the ascending order of the distance between the particle sigma value and each of the GBA member sigma value. The sigma value for any particle P_i whose corresponding point in the objective space is the vector $\vec{F}_i = (f_{i1}, f_{i2}, \dots, f_{im})^T$ is given by:

$$\vec{\sigma}_i = \begin{pmatrix} f_{i1}^2 - f_{i2}^2 \\ \dots \\ f_{i1}^2 - f_{im}^2 \\ f_{i2}^2 - f_{i3}^2 \\ \dots \\ f_{i2}^2 - f_{im}^2 \\ \dots \\ f_{i(m-1)}^2 - f_{im}^2 \end{pmatrix} / \sum_{j=1}^m f_{ij} \quad (7)$$

Crowding distance computation is done in a similar way as in NSGAII [17]. Solutions in the GBA are first sorted in the objective space then the overall crowding distance is calculated as the sum of individual distance values corresponding to each objective.

- For self best position, the strategy used is to keep only one solution as a self best point for each particle. Once a new position is computed for a particle, three cases may appear:
 - The new position dominates the self best position in which case this latter should be updated.
 - The new position is dominated by the self best position in which case this latter remains the same.
 - None of the two dominates the other, in this case one of the randomly selected is kept as the self best position for the particle.

- Finally, once self best and global best positions are determined for a particular particle, the local attractor can be computed as given in equation (5).

B. Calculating the mean best position

As only one solution is kept as a self best position for each particle, the mean best position is computed as in single objective optimization which is shown in equation (6).

C. Setting of the CE parameter β

We use a time varying CE parameter β . It has been shown [3] that in the case of mono objective optimization, good results have been obtained when decreasing β linearly from an initial value β_0 to a final value β_1 during the search process. At each iteration t , the new value of β is calculated as a function of β_0 , β_1 and the maximum number of iterations T :

$$\beta^{t+1} = \beta^t + (\beta_1 - \beta_0)/T \quad (8)$$

D. Outline of Multiobjective QPSO

Multiobjective QPSO (MOQPSO) can be described as follows:

Input: MOP specification

- Initialize population size (N)
- Initialize problem dimension (D)
- Initialize positions of particles randomly
- Initialize self best positions.
- For each particle P_i Compute the objectives vector \vec{F}_i
- Initialize GBA with non-dominated solutions in the initial population ;

repeat

Decrease linearly β according to eq. 8 ;

Compute mean best position using eq. 6 ;

For each particle P_i

Select-leader from GBA;

Compute local attractor using eq.5 ;

$u = \text{rand}(0,1)$;

if ($\text{rand}(0,1) \geq 0.5$) **Then**

$$x_{ij}^{t+1} = p_{ij}^t + \beta \cdot |mbest_j^t - x_{ij}^t| \cdot \ln(1/u_{ij}^t)$$

Else

$$x_{ij}^{t+1} = p_{ij}^t - \beta \cdot |mbest_j^t - x_{ij}^t| \cdot \ln(1/u_{ij}^t)$$

Endif

Update self best position;

End for

Update GBA

Until a maximum number of iterations

Output GBA

V. EXPERIMENTAL RESULTS

To assess the performance of the proposed algorithm, we used test problems for function optimization that are described in [17] and reported in table 1. These functions are related to studies of Schaffer, Fongesca et al. and Zitzler [17]. As shown in this table, a specification of the true Pareto optimal front is given. The initial and final value of the CE parameter β have been set to 1.2 and 0.5 respectively. The number k of nearest neighbors in the selection strategy has been set to 4. Several experiments have been conducted to assess the ability of the proposed approach to converge to the Pareto optimal front as well as its ability to spread out the non-dominated solutions uniformly along the Pareto front. For this purpose, we used the two metrics related to convergence and diversity described in [17] and we adopted the same testing procedures described there for sake of comparison. To derive the Pareto optimal front, a set of 500 uniformly distributed solutions has been derived for each test problem using the specification given in column 5 of table 1. First, the convergence metric Θ which reflects the closeness of the obtained Pareto front to the Pareto optimal front has been measured as the average of the sum of minimum distances of each obtained solution in GBA to the Pareto optimal front. The smaller the Θ , the better the convergence. In figure 2, we show the obtained front against the optimal one for each of the five test problems reported in table 1. Qualitatively, as can be seen very good convergence has been achieved in all cases. Quantitatively, through several runs of the algorithm for each case, the mean and variance of the metric Θ have been recorded. Table 2 shows the obtained results along with results of some state of the art algorithms for the same test problems as given in [17]. Very competitive results have been obtained. The variance of the metric is almost equal to zero in all cases which can be interpreted as an indication of the robustness of the algorithm. MOQPSO achieves better results than the other algorithms on Fongesca function and Zitzler 3 function. For the remaining functions it is in between while being close to the best ones. Second, the diversity metric Δ has been measured by calculating the nonuniformity in the distribution of solutions along the front according to the equation given in [17]. Like the convergence metric, the smaller the Δ metric value, the most widely and uniformly the set of non-dominated solutions. The mean and variance of the metric Δ have been recorded after several runs of the algorithm on each of the used test problems. Table 3 shows the results. It is clearly apparent from figure 2 that an excellent spread among the set of obtained solutions has been achieved. This fact is corroborated by the results reported in table 3 as all values are close to zero. We can also see that MOQPSO outperforms the other algorithms in all cases.

Table 1: Used test problems

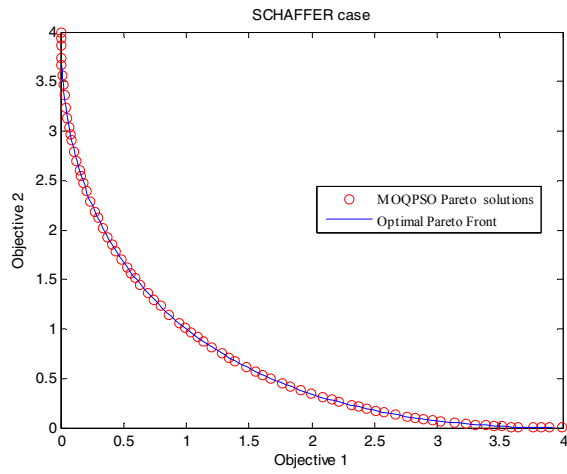
Test Problem	D	Variable Bounds	Objective Functions	Optimal Solutions	Type
Schaffer (SCH[17])	1	$[-10^3, 10^3]$	$f_1(x) = x^2$ $f_2(x) = (x - 2)^2$	$x \in [0, 2]$	Convex
Fonsesca et al. (FON [17])	3	$[-4, 4]$	$f_1(x) = 1 - \exp(-\sum_{i=1}^3 (x_i - 1/\sqrt{3})^2)$ $f_2(x) = 1 - \exp(-\sum_{i=1}^3 (x_i + 1/\sqrt{3})^2)$	$x_1 = x_2 = x_3$ $x \in [-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}]$	Non convex
Zitzler 1 (ZDT1[17])	30	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{x_1/g(x)}]$ $g(x) = 1 + 9(\sum_{i=2}^D x_i)/(D - 1)$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, \dots, D$	Convex
Zitzler 2 (ZDT2[17])	30	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - (x_1/g(x))^2]$ $g(x) = 1 + 9(\sum_{i=2}^D x_i)/(D - 1)$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, \dots, D$	Non convex
Zitzler 3 (ZDT6[17])	10	$[0, 1]$	$f_1(x) = 1 - \exp(-4x_1) \sin^6(6\pi x_1)$ $f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$ $g(x) = 1 + 9[\sum_{i=2}^D x_i/(D - 1)]^{0.25}$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, \dots, D$	Non convex , Non uniformly Spaced

Table 2: Results of convergence assessment (Mean (first row) and variance (second row) of the used metric Θ)

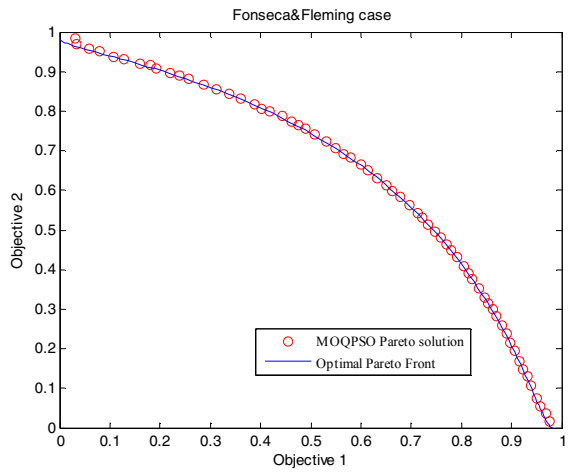
Algorithms	Schaffer	Fonsesca	Zitzler 1	Zitzler2	Zitzler3
MOQPSO	0.0032	0.0018	0.0241	0.0109	0.0789
	0	0	0	0	0
NSGAII real coded [17]	0.003391	0.001931	0.033482	0.072391	0.296564
	0	0	0.004750	0.031689	0.013135
NSGA II Binary coded[17]	0.002833	0.002571	0.000894	0.000824	7.806798
	0.000001	0	0	0	0.001667
SPEA [17]	0.003403	0.125692	0.001799	0.001339	0.221138
	0	0.000038	0.000001	0	0.000449
PAES [17]	0.001313	0.151263	0.082085	0.126276	0.085469
	0.000003	0.000905	0.008679	0.036877	0.006664

Table 3: Results of Diversity assessment (Mean (first row) and variance (second row) of the used metric Δ)

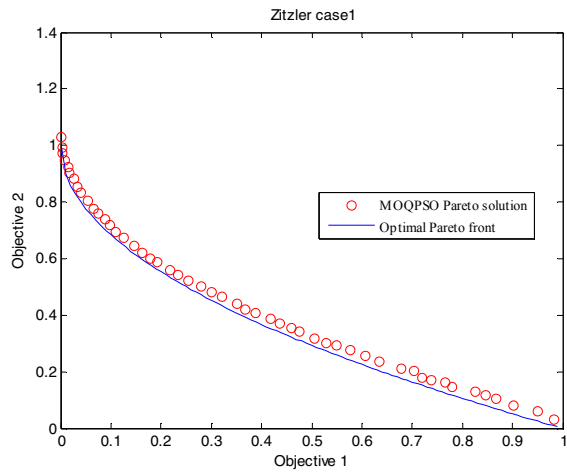
Algorithms	Schaffer	Fonsesca	Zitzler 1	Zitzler2	Zitzler3
MOQPSO	0.0574	0.0456	0.0559	0.0404	0.0616
	0.0077	0.0083	0.0117	0.0061	0.0150
NSGAII real coded [17]	0.477899	0.378065	0.390307	0.430776	0.668025
	0.003471	0.000639	0.001876	0.004721	0.009923
NSGA II Binary coded[17]	0.449265	0.395131	0.463292	0.435112	0.644477
	0.002062	0.001314	0.041622	0.024607	0.035042
SPEA [17]	1.021110	0.792352	0.784525	0.755148	0.849389
	0.004372	0.005546	0.004440	0.004521	0.002713
PAES [17]	1.063288	1.162528	1.229794	1.165942	1.153052
	0.002868	0.008945	0.004839	0.007682	0.003916



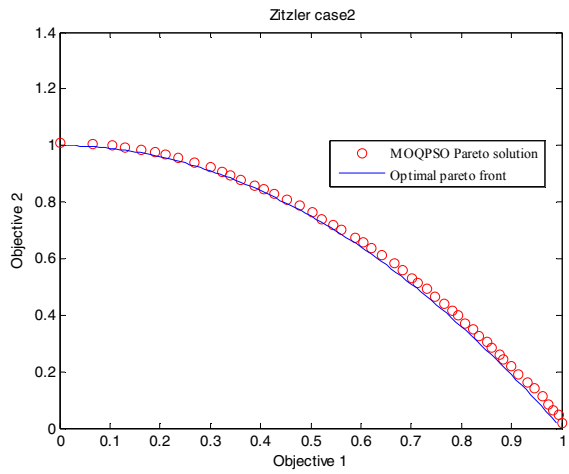
(a)



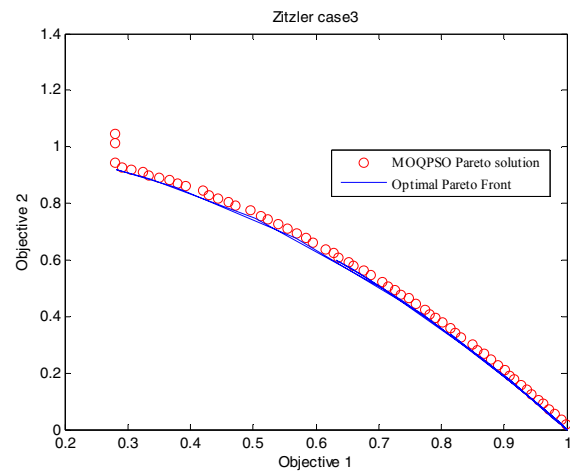
(b)



(c)



(d)



(e)

Figure 2. Obtained Pareto front against Pareto optimal Front:

(a) Schaffer function (b) Fonseca and Fleming function (c) Zitzler function 1(ZDT1[17]) (d) Zitzler function 2(ZDT2[17]) (e) Zitzler function 3(ZDT6[17])

VI. CONCLUSIONS AND FUTURE WORK

In this study, we investigated features related to the application of QPSO to handle multiple objectives. The main feature is the proposal of a strategy to compute the local attractor point for each solution that consists in keeping one self best solution for each solution at each iteration and a selection of the global best position based on sigma values and crowding distance. The CE parameter has been decreased linearly. Very promising and competitive results have been obtained on some test problems compared to the state of the art algorithms when assessing the convergence and diversity abilities of the algorithm. As a future work, we intend to further investigate the capabilities of QPSO for multiobjective optimization by tackling the issues related to constraint handling and global best archive update.

REFERENCES

- [1] K. Han and J. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization". *IEEE Transactions on Evolutionary Computation*, pp. 580–593, 2002.
- [2] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior". *In proceedings of the IEEE Congress on Evolutionary Computation*, 2004.
- [3] W. Fang, J. Sun, Y. Ding, X. Wu, and W. Xu, "A review of quantum-behaved particle swarm optimization". IETE Technical Review, 2010.
- [4] J. Sun, W. Xu, and B. Feng, "A global search strategy of quantum behaved particle swarm optimization". *In proceedings of the IEEE Conference on Cybernetics and Intelligent Systems*, pp. 111–116, 2004.
- [5] S. Meshoul and M. Batouche, "A novel quantum behaved particle swarm optimization with chaotic search for image alignment". *In proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pp. 557–562, 2010.
- [6] S. Meshoul and T. Al-Owaisheq, "A quantum behaved particle swarm optimization for consensus pattern identification". *In proceedings of ISICA, Communications on computers and information sciences, CCIS 51*, Springer, pp. 369–378, 2009.
- [7] A. Abraham and L. Jain, "Evolutionary multiobjective optimization", In Abraham Ajith, Jain Lakhmi, and Goldberg Robert, editors. *Evolutionary Multiobjective Optimization, Advanced Information and Knowledge Processing*, Springer, pp. 1–6, 2005.
- [8] K. Deb, "Multi-objective optimization using evolutionary algorithms". Chichester : Wiley, 2001.
- [9] A. Baltar and D. Fontane, "A generalized multi-objective particle swarm optimization solver for spreadsheet models: application to water quality", hydrology days, 2006.
- [10] S. Meshoul, K. Mahdi, and M. Batouche, "A quantum inspired evolutionary framework for multiobjective optimization". *In proceedings of the 12th Portuguese Conference on Artificial Intelligence (EPIA 2005)*, LNAI 3808, pp. 190–201, 2005.
- [11] Y. Kim, J. H. Kim, and K. H. Han, "Quantum-inspired multiobjective evolutionary algorithm for multiobjective 0/1 Knapsack problems". *In proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pp. 9151–9156, 2006.
- [12] H. Kuo, J. Chang, and C. Liu, "Particle swarm optimization for global optimization problems". *Marine Science and Technology*, pp. 170–181, 2006.
- [13] M. Sierra and C. Coello, "Multi-objective particle swarm optimizers: A survey of the state-of-the-art". *International Journal of Computational Intelligence Research*, pp. 287–308, 2006.
- [14] S. Omkar, R. Khandelwal, T. Ananth, G. N. Naik, and S. Gopalakrishnan, "Quantum behaved particle swarm optimization (qpso) for multi-objective design optimization of composite structures". *Expert Systems with Applications*, pp. 11312–11322, 2009.
- [15] K. Parsopoulos and M. Vrahatis, "Particle swarm optimization method in multiobjective problems". *In proceedings of ACM Symp. Applied Computing 2002 (SAC 2002)*, pp. 603–607, 2002.
- [16] S. Mostaghim and J. Teich, "Strategies for finding good local guides in multi-objective particle swarm optimization (mopso)". *In proceedings of the IEEE Swarm Intelligence Symposium*, pp. 26–33, 2003.
- [17] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGAII". *IEEE Transactions on Evolutionary Computation*, pp. 182 –197, 2002.
- [18] C. Coello, G. Pulido, and M. Lechuga, "Handling multiple objectives with particle swarm optimization". *IEEE Transactions on Evolutionary Computation*, pp. 256 –279, 2004.
- [19] D. Rini, S. Shamsuddin, and S. Yuhaniz, "Particle swarm optimization: Technique, system and challenges". *International Journal of Computer Applications*, pp. 19–27, 2011.
- [20] R. Umrani and V. Selvi, "Particle swarm optimization - evolution, overview and applications". *International Journal of Engineering Science and Technology*, pp. 2802–2806, 2010.
- [21] C. Coello and M. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization". *In Congress on Evolutionary Computation (CEC 2002)*, pp. 1051–1056, 2002.
- [22] J. Fieldsend and S. Singh, "A multiobjective algorithm based upon particle swarm optimization, an efficient data structure and turbulence". *In proceedings of the 2002 U.K. Workshop on Computational Intelligence*, pp. 37–44, 2002.
- [23] M. Arias, G. Pulido, and C. Coello, "A proposal to use stripes to maintain diversity in a multi-objective particle swarm optimizer". *In proceedings of the IEEE Swarm Intelligence Symposium*, pp. 22–29, 2005.
- [24] X. Li, "A non-dominated sorting particle swarm optimizer for multiobjective optimization". *In proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, Springer, pp. 37–48, 2003.