JOURNAL OF MODERN MANUFACTURING SYSTEMS AND TECHNOLOGY

Homepage: http://journal.ump.edu.my/jmmst
ISSN (Online): 2636-9575

A hybrid SP-QPSO algorithm with parameter free adaptive penalty method for constrained global optimization problems

Fatemeh D B[1], Loo C K[2], Kanagaraj G[3*], Ponnambalam S G[4]

*Correspondence
gkmech@tce.edu

[1]Fatemeh Divan Bachari
Department of Artificial Intelligence,Faculty of Computer Science & Information Technology,University of Malaya, 50603 Malaysia.
[2] Advanced Robotics Lab
Department of Artificial Intelligence
Faculty of Computer Science and Information Technology,
University of Malaya, Malaysia
[3]Department of Mechatronics Engineering, Thiagarajar College of Engineering, Madurai, India
[4]Faculty of Manufacturing Engineering University Malaysia Pahang 26600 Pekan, Pahang, Malaysia
Email:sgponnambalam@ump.edu.my

Articles Info:
Received **11 June 2018**
Received in revised form **24 Aug 2018**
Accepted **11 Sept 2018**
Available Online **13 Sept 2018**

Keywords:
Constrained optimization
Hybrid algorithm
Swarm intelligence
Penalty method

ABSTRACT

Most real-life optimization problems involve constraints which require a specialized mechanism to deal with them. The presence of constraints imposes additional challenges to the researchers motivated towards the development of new algorithm with efficient constraint handling mechanism. This paper attempts the suitability of newly developed hybrid algorithm, Shuffled Complex Evolution with Quantum Particle Swarm Optimization abbreviated as SP-QPSO, extended specifically designed for solving constrained optimization problems. The incorporation of adaptive penalty method guides the solutions to the feasible regions of the search space by computing the violation of each one. Further, the algorithm's performance is improved by Centroidal Voronoi Tessellations method of point initialization promise to visit the entire search space. The effectiveness and the performance of SP-QPSO are examined by solving a broad set of ten benchmark functions and four engineering case study problems taken from the literature. The experimental results show that the hybrid version of SP-QPSO algorithm is not only overcome the shortcomings of the original algorithms but also outperformed most state-of-the-art algorithms, in terms of searching efficiency and computational time.

INTRODUCTION

Nowadays the advancements in technology make the human life easier which brings new problems to the optimization field. In real-world applications of engineering field, designers are mostly simulating the systems that have complicated structures that lead to more complex systems. Because of various practical requirements, the effective approaches to solve such problems are still in urgent need for high-quality decision making, has made the topic a valuable and challenging research area. Therefore, researchers are mostly dealing with constrained optimization problems (COPs) and to develop an efficient algorithm integrated with a suitable mechanism to guide the individuals moving towards the feasible region on finding the optimal solution [1]. In general, the mathematical form of COPs is expressed as a nonlinear optimization problem as follows [2]

$$Minimize\ \ f(\vec{x}) \tag{1}$$

$$Subject\ to\ g_j(\vec{x}) \leq 0 \quad j = 1, \dots, p$$
$$h_i(\vec{x}) = 0 \quad i = p+1, \dots, m$$

where $f(\vec{x})$ is a real objective or criterion function defined on a feasible set, and the solution $\vec{x} = (x_1, x_2, \dots, x_n)\ \epsilon R^n$ is the n-dimensional real vector. The inequality $g_j(\vec{x}) \leq 0$ represents the j^{th} inequality constraint and $h_i(\vec{x})$ shows the i^{th} equality constraint and there are p inequality and $m - \mathrm{p}$ equality constraints. Both the objective function and constraints of COPs could be either linear or nonlinear. Vector x is a feasible solution of the problem if it satisfies all the constraints and the feasible region is constituted by all of the feasible solutions. An inequality constraint of the mathematical form as: $\mathrm{g_j(x)} \geq 0$ is not restrictive and can be converted to the $-\mathrm{g_j(x)} \leq 0$ (J. Sun et al., 2007). Also to make it more convenient, in this paper the equality constrained is converted to inequality as:$|\mathrm{h_i(x)} - \varepsilon \leq 0$; Where the value of ε for these numerical experiments is set to 10^{-4} or 10^{-3} which depends on the problem [3].

According to the previous studies, Evolutionary Algorithms (EAs) are stochastic population-based optimization algorithm that efficient and more flexible for solving COPs since they only require to knowing the objective function [4-8]. The rest of the information such as differentiability; continuity of objective function or constraints are not required. Moreover, EA generates solutions near to the global optimum, and the chance of being trapped by a local minimum is lower; hence EAs are more powerful and efficient than the traditional numerical methods [9]. It is noteworthy that EAs are unconstrained search methods and lack an explicit mechanism to bias the search in constrained search space. Therefore, a number of attempts to solve optimization problems, while hybrid algorithms have shown outstanding reliability and efficiency to solve these problems. In fact, the hybrid techniques, being powerful, yields promising results in solving specific problems [10].

In this paper, a novel hybrid of two common EAs, QPSO and SP-UCI algorithms called SP-QPSO algorithm using a parameter-free penalty method is proposed for solving constrained optimization problems. However, in order to increase the optimizer's performance, the Centroidal Voronoi Tessellations (CVT) method for point initialization, is applied on the SP-QPSO to assure that points visit the entire search space. The proposed SP-QPSO algorithm is tested on a group of standard constrained test benchmarks which are selected from [11], they are used to evaluate the efficiency of the SP-QPSO's performance as compared with the results of native EAs such as QPSO and SP-UCI algorithms.

The remaining part of the paper is organized as follows. In section 2, a review of a number of common EAs and existing penalty based techniques in solving COPs are presented. In section 3, the description of the proposed hybrid algorithm and the methodology is described in detail. The experimental results based on well-known theoretical benchmarks as well as the algorithm comparison are presented in Section 4. Section 5 concludes the paper and also the potential future works of the proposed algorithm are discussed.

LITERATURE REVIEW

In the last decades, different meta-heuristics have been proposed employing new ideas, and hybrid algorithms that improve the original meta-heuristics. One of the most successfully employed meta-heuristics is Particle swarm optimization (PSO) and its hybrid variants. PSO incorporates swarming behaviours, from which the idea is emerged initially by Kennedy and Eberhart [12] and received increasing attention regarding its potential as a faster global optimization technique. However, it might be caught in the trap of local optimum caused by premature convergence. PSO had difficulties in controlling the balance between exploration (global investigation of the search place) and exploitation (the fine search around a local optimum). Thus, special trade-off between exploration and exploitation is required to balance optimization reliability and convergence speed. The resultant algorithm has the ability to improve the robustness of the algorithm in terms of solution quality and faster convergence. Researchers are constantly working on adding some good features from any method to the PSO algorithm to improve the exploration and exploitation ability.

Many hybrid variants of PSO have been proposed to improve the performance of the original PSO algorithm [13] (Kaur & Kaur, 2015). Liang et al. (2006) implemented a novel learning strategy in standard PSO called Comprehensive Learning Particle Swarm Optimizer (CLPSO). In this algorithm, a particle can learn from any other particles' previous best positions. In addition, each dimension of a particle can learn from a different sample. Its velocity's update equation is different from the one in the original PSO algorithm. However, it does not add any complicated equation to the original algorithm and tries to improve the premature convergence of the original PSO algorithm.

He and Wang [14] presented a hybrid PSO with a feasibility-based rule has the aim of overcoming the deficiencies of penalty function methods. Fan and Zahara [15] have demonstrated Nelder–Mead simplex search method with PSO (NM-PSO) to be a promising and viable tool for solving unconstrained nonlinear optimization problems. The performance of the algorithm improved further by [16] with an embedded constraint handling method. As a real-world application, NM-PSO was employed to solve the optimization problem of silicon carbide grinding and it was proved that NM-PSO is superior in comparison with GA and PSO.

A hybrid heuristic particle swarm ant colony optimization (HPSACO) is proposed by Kaveh and Talatahari [7]. HPSACO utilizes a particle swarm optimization with a passive congregation algorithm as a global search, the idea of ant colony approach (ACO) worked as a local search and the harmony search (HS) utilized to handle the boundary constraints. Liu et al., [17] proposed a novel hybrid algorithm named PSO–DE, which integrates particle swarm optimization with differential evolution to solve constrained optimization problems. DE is used to update the best position of particles in order to force them to jump out of stagnation.

Cagnina et al., [18] implemented a hybrid particle swarm optimization algorithm for solving COPs, which utilizes different methods to update the particle's information, as well as the use of a double population and a special shake mechanism with a simple constraint-handling technique designed to avoid premature convergence. Kanagaraj et al., [19] introduced hybrid Cuckoo search and Genetic algorithm approach utilise the advantages of both cuckoo search technique and genetic algorithm for solving reliability- redundancy allocation problems in series, series–parallel, and complex (bridge) systems. Kaveh et al.,[13] presented a Hybrid Particle Swallow Swarm Optimization algorithm (HPSSO) for highly nonlinear dynamic truss shape and size optimization with multiple natural frequency constraints. A review article proposed by Xin et al., [20] provides a detailed study of hybridized DE-PSO versions and concluded that the hybridization of DE and PSO came out as a giant optimizer for the optimization problems.

Some researchers achieved excellent improvements in PSO algorithm, in which a variant called Quantum Particle swarm Optimization (QPSO) presented by Sun et al.,[21] attracted wide attention in recent years for its promising results [22-23]. In QPSO, the particles in space have quantum behaviour and wave functions instead of position and velocity which improves the original PSO by solving the issues related to premature convergence and having many parameters to adjust. Fu et al., [23] proposed the phase angle-encoded and quantum-behaved particle swarm optimization (Θ-QPSO) to generate a safe and flyable path for the UAV in the presence of different threat environments.

When solving constrained optimization problems by EAs, an important issue is how to balance constraints and objective function. Montes & Coello [24] and Salcedo-Sanz [25] mentioned various constraint handling methods utilized in evolutionary algorithms (EAs) and categorized the majority of them into five types-penalty functions, repair algorithms, special representations and operators, separate objective and constraints, as well as hybrid methods. The most common method utilized by EAs is penalty functions, originally proposed by Richard Courant in the 1940s and were later expanded by Carroll and Fiacco & McCormick [26]. Its key idea is to incorporate a penalized term into the objective function so that a constrained optimization problem can be transformed into an unconstrained one. The methods which are based on the penalty method and the rejection of infeasible solution (special representations and operators) have performed better when they are applied with the PSO algorithm together (Mazhoud et al., 2013). Usually, adaptive penalty function methods are very effective for constrained optimization since they can make use of the information obtained during the evolution to adapt their parameters.

The main problem with penalty functions is that the “ideal” penalty factor to be adopted in a penalty function cannot be known a priori for an arbitrary problem. If the penalty adopted is too high and the optimum lies at the boundary of the feasible region, the EA will be pushed inside the feasible region very quickly, and will not be able to move back towards the boundary with the infeasible region. On the other hand, if the penalty is too low, a lot of the search time will be spent exploring the infeasible region because the penalty will be negligible with respect to the objective function. Motivated by the above consideration, this paper proposes a new hybrid algorithm called SP-QPSO with the incorporation of CVT method of point initialization and parameter- free adaptive penalty of constraint handling for solving COPs.

HYBRID SP-QPSO

In this section, the description of proposed hybrid SP-QPSO, parameter-free adaptive penalty technique for constraint handling, Centroidal Voronoi Tessellations method of point initialization and the procedural steps of SP-QPSO are depicted.

Algorithm description

Recently, a number of hybridization techniques are offered to develop a new hybrid algorithm [27]. PSO is an effective optimization method that belongs to the category of swarm intelligence methods, originally developed by Kennedy and Eberhart in 1995 [12]. PSO is initialized with a population of random designs, named as particles. Each particle in the swarm is propelled to move through the search space from their current positions with increased velocity that are dynamically adjusted according to their current velocity, best self-experienced position and the best global-experienced position. Sometimes, PSO had struggle in controlling the balance between exploration and exploitation and trapped by premature convergence.

In order to improve the performance of PSO, this paper considers the positive features of QPSO and SP-UCI are combined to form a hybrid algorithm called SP-QPSO suitable for high dimensional constrained optimization tasks. The performance of QPSO algorithm in solving simple low dimensional optimization problems is significantly efficient, due to its local search ability. The SP-UCI algorithm exhibit the efficient performance in solving complex high dimensional optimization problems, due to its global search ability and adopt Modified Competitive Complex Evolution (MCCE) strategy instead of the traditional Competitive Complex Evolution (CCE) strategy in solving high dimensional problems [11]. Adding these features will further enhance the algorithm performance towards local and global search ability.

In QPSO the particle converged to its own local attractor, its personal best position converges to the global best position. This process converge the QPSO algorithm to global optima. Therefore, the particle's local attractor will gather toward the global best position. As a result, it makes the current position of the particle converges to the global best position. So, the movement of p_{id} can be guided by the global best position which affects the convergence behaviour of the particle's current position. By using Trajectory analysis [28], the local stochastic attractor of particles is calculated as follows [21]:

$$P = \varphi \, . \, p_{id} + (1 - \varphi) \, . \, p_{gd} \tag{2}$$

By applying Monte Carlo method, we obtained the following equation :

$$x = P \pm \frac{L}{2} \ln\left(\frac{1}{u}\right), \quad u = \mathrm{rand}(0,1) \tag{3}$$

The value of L and the mean of the pbest positions of all the particles (mbest) are calculated as:

$$L = 2\alpha . |mbest - x| \quad , \quad mbest = 1/N \sum_{i=1}^{N} P_i(t) \tag{4}$$

Where the variable $P_i$is the personal best position of particle i and parameter α must be set as less than 1.781 ($\alpha < 1.781$) to guarantee convergence of the particle.
The evolution equation of the QPSO is:

$$X_i(t+1) = p_i \pm \beta \, . \, |mbest - X_i(t)| . \ln\left(\frac{1}{u}\right) \tag{5}$$

Where p_i is the local optimal point and X_i presents particle's current position; The algorithm will lead to premature convergence if the global best position is trapped into p_i, since $X_i$will also be dragged into that point. Therefore, apart from the strong local search ability, QPSO algorithm needs to improve its global search ability especially when the algorithm deals with the complex high dimensional real-world optimization problems. The detailed flow diagram of SP-QPSO is outlined in Fig. 1.

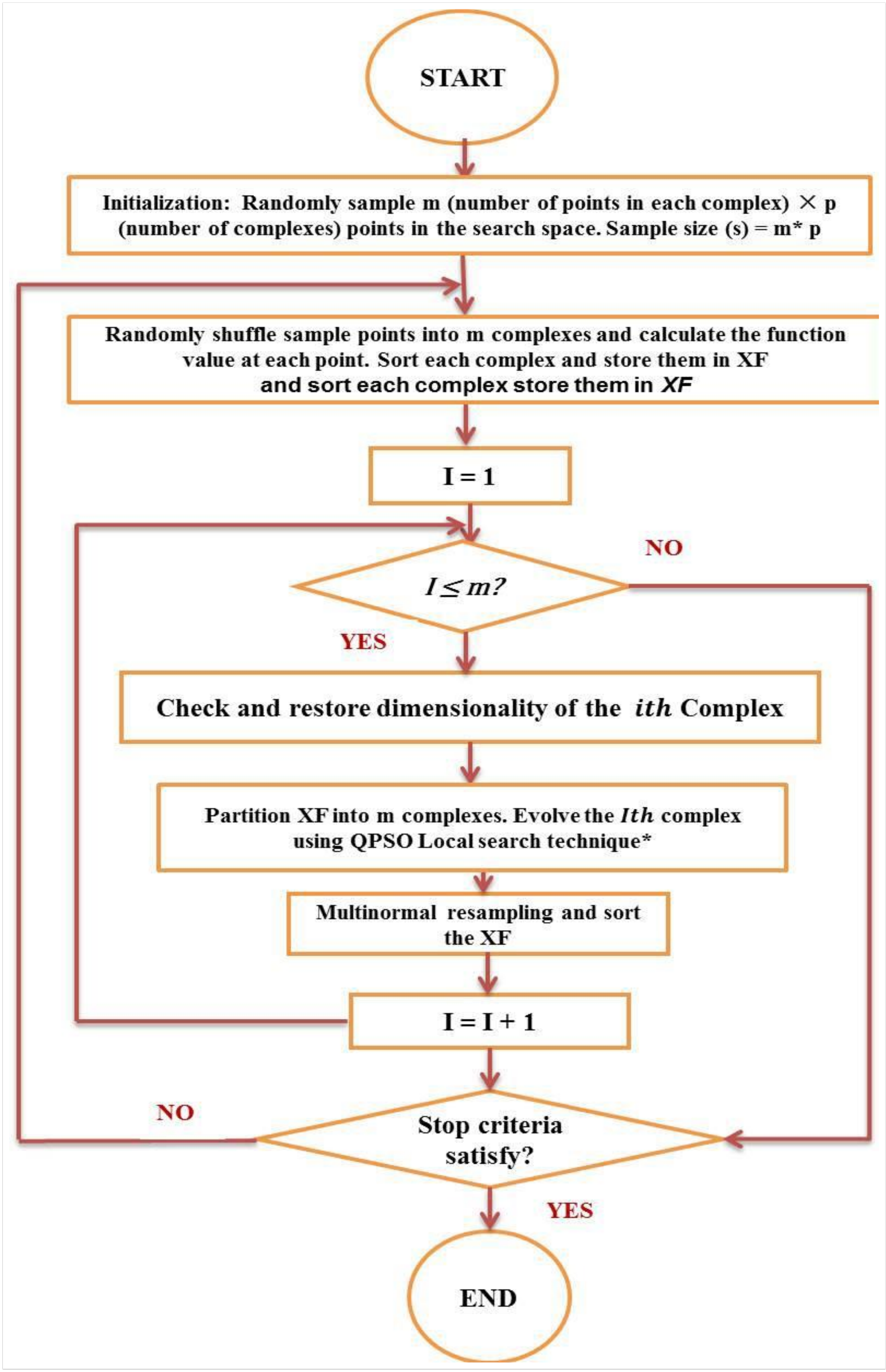
START
Initialization: Randomly sample m (number of points in each complex) × p (number of complexes) points in the search space. Sample size (s) = m* p
Randomly shuffle sample points into m complexes and calculate the function value at each point. Sort each complex and store them in XF
and sort each complex store them in XF
I = 1
I ≤ m?
NO
YES
Check and restore dimensionality of the ith Complex
Partition XF into m complexes. Evolve the Ith complex using QPSO Local search technique*
Multinormal resampling and sort the XF
I = I + 1
Stop criteria satisfy?
NO
YES
END

Figure 1 Flow diagram of SP-QPSO

Parameter free Adaptive Penalty Method for constraint handling

To deal with constraints, penalty function based methods for constraint handling have a long history of development. In its native form, a penalty is applied to the fitness of a solution in the event it violates one or more constraints [3]. Generally, the penalty based techniques used in EAs for constraint optimization have parameters and the quality of their performance is totally depend on tuning those parameters which need many trial experiences to determine the right value for the penalty parameter. To overcome this limitation, adaptive penalty function methods are very effective for constrained optimization since they can make use of the information obtained during the evolution to adapt their parameters.

To select various penalties for each constraint during the process of searching for the global best point, an adaptive scheme is implemented that utilizes population information such as, the mean of the objective function and the level of violation of each constraint [29]. The reason is that higher penalty coefficient values should be allocated for constraints which are more difficult to be satisfied. The number of elements violating a given constraint and the amount of violation is the indication of the above-mentioned constraint difficulties. The j^{th} coefficient is represented by $\mathrm{k_j}$. The proposed fitness function is:

$$\mathrm{F(x)} = \begin{cases} \mathrm{f(x)} & \text{if x is feasible} \\ \mathrm{f(x)} + \sum_{\mathrm{j=1}}^{\mathrm{m}} \mathrm{k_j v_j(x)} & \text{otherwise} \end{cases} \tag{6}$$

Where the penalty parameter ($\mathrm{k_j}$) at each generation is defined as follows (Barbosa & Lemonge, 2003):

$$\mathrm{k_j} = |< f(\mathrm{x}) >| \frac{<\mathrm{v_j(x)}>}{\sum_{\mathrm{l=1}}^{\mathrm{m}}[<\mathrm{v_l(x)}>]^2} \quad \text{or} \tag{7}$$

$$\mathrm{k_j} = \frac{\sum_{\mathrm{i=1}}^{\mathrm{pop}} \mathrm{f(x^i)}}{\sum_{\mathrm{l=1}}^{\mathrm{m}}\left[\sum_{\mathrm{i=1}}^{\mathrm{pop}} \mathrm{v_l(x^i)}\right]^2} \sum_{\mathrm{i=1}}^{\mathrm{pop}} \mathrm{v_j(x^i)} \tag{8}$$

Where the mean of the objective function values in the current population is represented by $< f(\mathrm{x}) >$ and the violation of the lth constraint averaged over the current population is defined by$< \mathrm{v_l(x)} >$. "Pop" is denoting the population size. The adaptive penalty method [30] is used without any type of pre-defined penalty parameter to be incorporated with hybrid SP-QPSO, QPSO and SPUCI algorithms, which efficiently handles constraints and dynamically adjusts the penalty during the optimization.

Centroidal Voronoi Tessellations method of initialization

In the case of population-based algorithms, the distribution of agents over the search space plays a vital role and can affect the performance of the algorithm. A variety of distribution techniques are proposed to initialize the agents and start the optimization task [31]. The most important drawback of random distribution of points is leading to local minima in the beginning of search process. Thus, one of the ways to improve the result of optimization is to manage the initialization of points all over the search space. Centroidal Voronoi Tessellation (CVT) is one of the best techniques repeatedly used in many fields of science and engineering to partition the surface to the sub-regions and scatter points to all the sub-regions. Promising results obtained for both low dimensional problems as well as a high dimensional problem by the incorporation of CVT techniques [11]. Traditional PSO is easy to fall into stagnation when no particle discovers a position that is better than its previous best position for several generations. This could be avoided by CVT on initialization stage, which helps the algorithm to visit all the regions in the search space and avoiding decoy in local minima at the very beginning of optimization process.

THE PROCEDURAL STEPS OF SP-QPSO

The proposed hybrid approach is performed in two stages, the outer stage and inner stage (as shown in Fig. 1). In the outer stage (step 1-8), agents are known as points which are generated by CVT method of initialization at the beginning. Points are distributed randomly through the search space and the first results are saved and sorted from the smallest value to the largest in an array. After sorting the values in an array, it is divided into the number of complexes; in each subdivision "*m*" points are placed and called simplex. In the inner stage (step 5), the optimization procedure is applied to each simplex. The outer stage is performed by SP-UCI strategy is to check and monitor the dimensionality of the population in each iteration and avoids population degeneration to accomplish the optimization task towards global search. Following this, the inner stage is the bound with QPSO strategy on each complex to look for the best result over the feasible search space. In this QPSO strategy, all the particles follow the best points found so far in all the past iterations (pbest) individually and also the best point found by all the particles which is known as global best (gbest).

Applying SP-UCI strategy accomplished by CVT helps the algorithm to check and monitor the dimensionality and avoid population degeneration and also ensure to visit all the regions in the search space at the very beginning of optimization process. Correspondingly to ensure the performance of global search, QPSO is applied to find out the optimum result by employing each particle.

The procedural steps of the proposed hybrid SP-QPSO algorithm are as follows:

Step 1: *Initialize the number of complex, number of points in each complex, sample size and dimension of the problem.*
Step 2: *Initialize sample points using Halton step method of Centroidal Voronoi Tessellations (CVT) over the feasible space using a generator. The search space can be partitioned into sections evenly and sort the points based on their calculated function value.*
Step 3: *Apply dimensionality check for rough fitness landscapes in high dimensional space.*
Step 4: *Store the population points in an array and divide the array into complexes.*
Step 5: *Apply QPSO strategy on each complex, so that each complex looks for the optimum function value.*

- *a. Initializing the population size and maximum number of iteration.*
- *b. Building a sub swarm containing all the population points and calculating their function values (local best).*
- *c. Finding the best value by comparing all the local bests and assigning it to the global best.*
- *d. Calculating the mean best position of the population (mbest) and the movements of particle toward the global best for the next iteration.*
- *e. Repeating the process until the maximum number of iteration satisfied.*

Step 6: *Checking the probability of trapping in local minima by applying the multi-normal resampling.*
Step 7: *Shuffling and replacing the complexes into an array and sorting it.*
Step 8: *Repeating the process until the stopping criteria satisfied.*

PERFORMANCE AND EXPERIMENTAL RESULTS

Experimental setup

The proposed hybrid SP-QPSO algorithm, QPSO and SP-UCI was coded in MATLAB 8.4, and then the results were obtained by running the algorithms on Windows 8 with Intel Core personal computer having 2.5 GHz processor and 4 GB RAM. For each test function, 25 independent runs are conducted and each run is stopped when reaches a maximum number of iterations.

Parameter setting

For SP-QPSO, the parameters include population size, number of iterations, number of complexes and numbers of points in each complex. The remaining of parameters are assigned by default. Through an extensive experimental fine tuning, the parameters recommended to best suit the proposed SP-UCI, QPSO and SP-QPSO algorithms are listed in Table1.

Table 1 Setting of parameters

Parameter	SP-UCI	QPSO	SP-QPSO
Population size	-	20	-
Maximum iteration number	10,000	10,000	10,000
Maximum number of complexes	2	-	2
Maximum number of points	-	100	100

Benchmark functions

To evaluate the optimization ability and performance of the proposed hybrid algorithm, 10 constrained benchmark test functions were considered. The details of test functions are given in Annexure. The test functions considered for the study is chosen with varying dimension, type (linear, non-linear, quadratic) and type of constraints (linear equality and inequality, non-linear equality and inequality constraints). The main characteristics of the test functions are summarized in Table 2, which lists the number of decision variables (n), the type of objective function, the number of linear inequality constraints (LInC), the number of nonlinear inequality constraints (NLInC), the number of linear equality constraints (LEC), the number of nonlinear equality constraints (NLEC), and the known optimal value (fglobal).

Table 2 Characteristics of test functions

Test functions	n	Obj-*f*	LInC	NLInC	LEC	NLEC	α(%)
G03	10	polynomial	0	0	0	1	0.0000
G04	5	Quadratic	0	6	0	0	26.925
G05	4	Cubic	2	0	0	3	0.0000
G06	2	Cubic	0	2	0	0	0.0054
G08	2	Non-linear	0	2	0	0	0.8611
G09	7	Polynomial	0	4	0	0	0.5314
G11	2	Quadratic	0	0	0	1	0.0000
G13	5	Non-linear	0	0	0	3	0.0000
G15	3	Quadratic	0	0	1	1	0.0000
G24	2	Linear	0	2	0	0	74.268

Table 3 reports the statistical results of the test functions in terms of the best, the mean, the worst, and the standard deviation of the objective value of the solutions obtained by SP-UCI, QPSO and SP-QPSO.

Table 3 Statistical of results for benchmark functions with QPSO, SP-UCI and SP- QPSO

Function	Optimal solution	Criterion	QPSO	SPUCI	SP-QPSO
G03	-1.00	Best	-1.000450036	-1.00027426	-1.051010033
		Median	-1.000449954	-1.000060147	-1.051009986
		Worst	-1.000449879	-0.999495167	-0.956813069
		Mean	-1.000449964	-1.000014815	-1.033082021
		Std	4.60237E-08	0.000175711	0.031673834
G04	-30665.539	Best	-30665.53867	-30665.98468	-30665.82104
		Median	-30665.53867	-30665.80689	-30665.82104
		Worst	-30665.53867	-30665.53465	-30665.82104
		Mean	-30665.53867	-30665.76242	-30665.82104
		Std	3.73249E-12	0.143768368	0
G05	5126.497	Best	5120.512676	5102.824196	5125.343394
		Median	5126.396077	5126.24158	5126.455471
		Worst	5136.490147	5142.80568	5129.593164
		Mean	5126.467709	5126.335643	5126.314437
		Std	2.877328754	7.336755318	0.945735948
G06	-6961.814	Best	-6961.817179	-6961.83341	-6961.81771
		Median	-6961.813876	-6961.815485	-6961.814135
		Worst	-6961.813876	-6961.765813	-6961.810773
		Mean	-6961.814041	-6961.814216	-6961.814733
		Std	0.000738673	0.012180327	0.001952029
G08	−0.095825	Best	-0.095825041	-0.095819301	-0.095824812
		Median	-0.095825041	-0.095773684	-0.095823522
		Worst	-0.095825041	-0.09557091	-0.095822238
		Mean	-0.095825041	-0.095741566	-0.095823579
		Std	6.25859E-12	8.32235E-05	-7.6865E-07
G09	680.630	Best	680.0593129	680.1245066	680.0192237
		Median	680.7027416	680.5598827	680.3625003
		Worst	681.0886926	681.1419486	680.8679251
		Mean	680.6337115	680.6168825	680.4143854
		Std	0.34474419	0.349821989	0.238895904
G11	0.75	Best	0.7499	0.740000009	0.74
		Median	0.7499	0.740002159	0.74
		Worst	0.749900007	0.740008935	0.74
		Mean	0.749900001	0.740002169	0.74
		Std	1.98731E-09	2.11005E-06	8.5008E-10
G13	0.053942	Best	0.003444395	0.003444441	0.003444395
		Median	0.003577757	0.010053257	0.003444395
		Worst	0.003581249	0.032665924	0.003577725
		Mean	0.003518152	0.013280591	0.003457728
		Std	6.84526E-05	0.010545019	4.10374E-05
G15	961.715	Best	961.700361	961.7019105	961.7001591
		Median	961.812475	961.7127172	961.7154342
		Worst	964.981852	961.7292587	961.7251997
		Mean	962.452291	961.7123561	961.7120884
		Std	1.12143941	0.007820518	961.7120884
G24	−5.508013	Best	-5.508013172	-5.50801373	-5.508013993
		Median	-5.508013023	-5.508013257	-5.508013747
		Worst	-5.508013002	-5.50801253	-5.50801285
		Mean	-5.50801305	-5.508013258	-5.508013606
		Std	5.4841E-08	3.49385E-07	-3.7197E-07

Table 4 summarize the results based on a well- known benchmarks and comparisons with previously reported algorithms CVI-PSO[32], DECV[33], AATM[34] and TCELL[35]. These results include the *best* objective value found, the *mean*, the *standard deviation*, and the *worst*. The statistics are based on feasible solutions only. The above experiment have shown that the CVI-PSO can achieve substantially good results for all problems. As described in Table 3, the SP-QPSO is able to find the best global optimum in G03, G04, G05, G06, G07, G08, G09, G11, G13, G15 and G24 and it found solutions very close to the global optimum. It is also interesting to note that for the problems g01, g03, g11 and g12 the best solutions of our results correspond to "theoretical" optimum with a standard deviation very close to 0 for G04, G08 and G11 (respectively, 0.46e−8 and 0.37e−12) and equal to 0 for G15 and G24.

Figure 2 shows an evolving process of the fitness value when solving the problem **G03, G04, G05, G11, G13 and G24** using the QPSO, SP-UCI and SP-QPSO. For the convergence rate, Figure 2 clearly indicates that SP-QPSO converges faster than other three algorithms for all the test functions. SP-QPSO improves the performance of QPSO and SP-UCI in terms of the quality of the final functions, the convergence rate, and the successful rate in the majority of benchmark functions.

Table 4 Comparison against state-of-the-art algorithms in benchmark functions

Function	Optimal solution	Criterion	CVI- PSO [32]	DECV [33]	AATM[34]	TCELL[35]	SP-QPSO
G03	-1.00	Best	-1	-0.461	-1	-1	-1.051010033
		Worst	-0.99999999999	-0.002	-1	-1	-0.956813069
		Mean	-0.99999999999	-0.134	-1	-1	**-0.03308202**
		Std	0.37e-15	0.117	3.5e -04	0.0	0.031673834
		NFEs	25,000	240,000	120,000	350,000	10,000
G04	-30665.539	Best	-30665.821710220	-30665.539	-30665.53	-30665.538	-30665.82104
		Worst	-30665.803241136	-30665.539	-30665.53	-30665.538	-30665.82104
		Mean	-30665.820996110	-30665.539	-30665.53	-30665.538	**-30665.8210**
		Std	0.003391	1.56e-06	1.0e-11	1.0e-4	0
		NFEs	25,000	240,000	120,000	350,000	10,000
G05	5126.497	Best	5127.27766734565	5126.497	5126.498	5126.6255	5125.343394
		Worst	5127.27766734565	5126.497	5128.824	6112.1181	5129.593164
		Mean	5127.27766734565	5126.497	5126.714	5378.2678	**5126.314437**
		Std	0.0	0.0	0.43	298.0173	0.945735948
		NFEs	25,000	240,000	120,000	350,000	10,000
G06	-6961.814	Best	-6961.8138755801	-6961.814	-6961.814	-6961.8138	-6961.81771
		Worst	-6961.8138755801	-6961.814	-6961.814	-6961.8136	-6961.810773
		Mean	-6961.8138755801	-6961.814	-6961.814	-6961.8138	**-6961.81473**
		Std	0.0	0.0	7.1e-12	3.9e-5	0.001952029
		NFEs	25,000	240,000	120,000	350,000	10,000
G08	−0.095825	Best	-0.1054595050884	-0.095825	-0.095825	-0.095825	-0.095824812
		Worst	-0.1054595050884	-0.095825	-0.095825	-0.095825	-0.095822238
		Mean	**-0.105459505088**	-0.095825	-0.095825	-0.095825	-0.095823579
		Std	0.0	4.23e-17	5.8e-18	0.0	-7.6865E-07
		NFEs	25,000	240,000	120,000	350,000	10,000
G09	680.630	Best	680.635400792284	680.630	680.630	680.63	680.0192237
		Worst	680.863957829483	680.630	680.646	680.70	680.8679251
		Mean	680.755705150086	680.630	680.634	680.65	**680.4143854**
		Std	0.079232	3.45e- 07	0.0045	0.0167	0.238895904
		NFEs	25,000	240,000	120,000	350,000	10,000
G11	0.75	Best	0.75	0.75	0.75	0.7499	0.74
		Worst	0.75	0.75	0.75	0.7499	0.74
		Mean	0.75	0.75	0.75	0.7499	**0.74**
		Std	0.0	1.12e-16	3.8e-06	0.0	8.5008E-10
		NFEs	25,000	240,000	120,000	350,000	10,000
G13	0.053942	Best	0.05555582974209	0.059798	NA	0.054638	0.003444395
		Worst	0.09372815700166	0.999094	NA	0.994983	0.003577725
		Mean	0.06559074439931	0.382401	NA	0.458857	**0.003457728**
		Std	0:010177	0.268	NA	0.344995	4.10374E-05
		NFEs	25,000	240,000	120,000	350,000	10,000
G15	961.715022	Best	961.715707147147	961.715022	961.715	961.71502	961.7001591
		Worst	961.718781406392	961.715022	961.716	970.59467	961.7251997
		Mean	961.718595474823	961.715022	961.715	963.37482	**961.7120884**
		Std	6.87e-04	2.31e-13	3.0e-04	2.27562	961.7120884
		NFEs	25,000	240,000	120,000	350,000	10,000
G24	−5.508013	Best	-5.5080132715953	-5.508013	-5.508013	-5.508013	-5.508013993
		Worst	-5.5080132715952	-5.508013	-5.508013	-5.508013	-5.50801285
		Mean	-5.5080132715953	-5.508013	-5.508013	-5.508013	**-5.50801360**
		Std	9.46e-15	2.71e-15	1.8e-15	0.0	-3.7197e-07
		NFEs	25,000	240,000	120,000	350,000	10,000

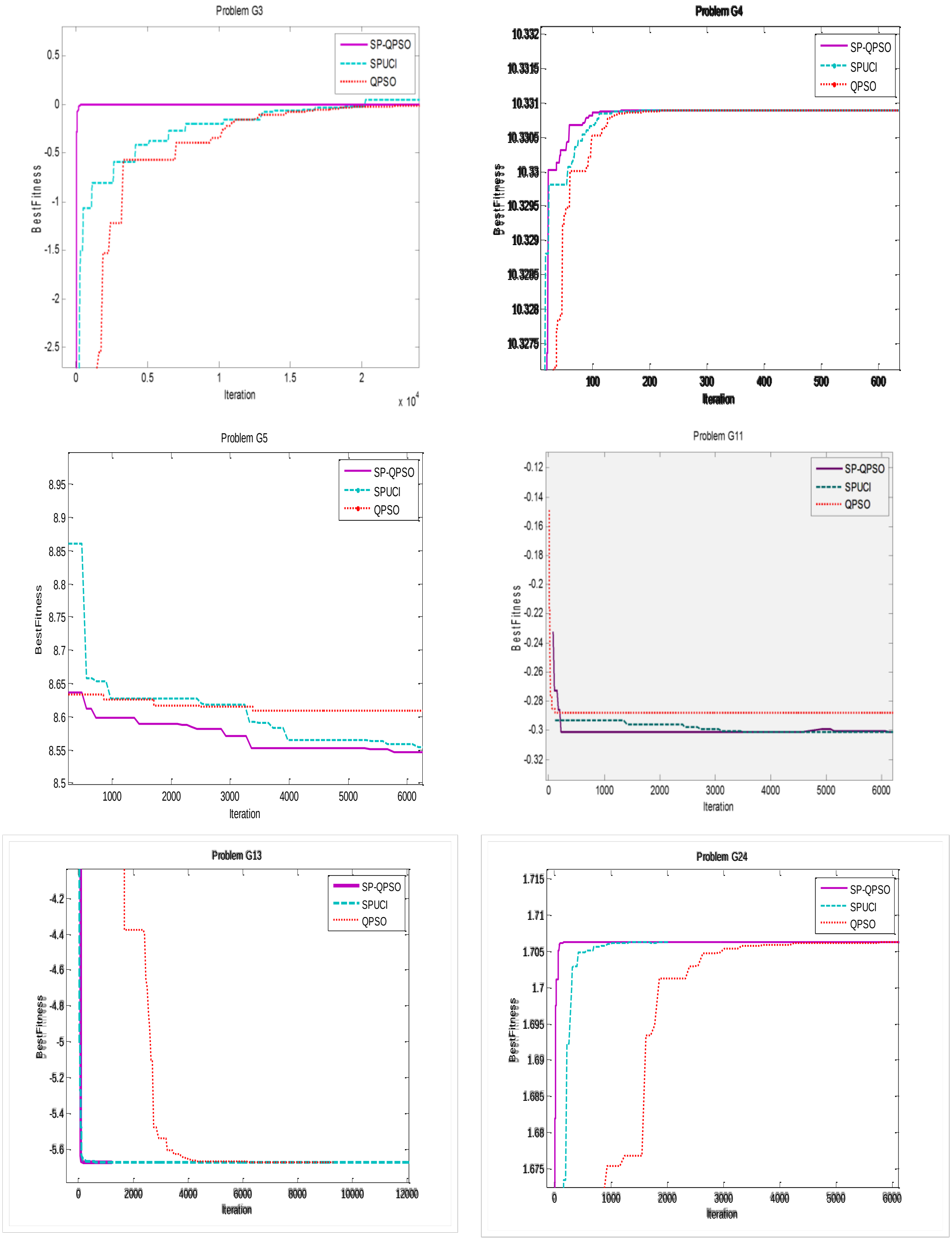

Figure 2 Convergence graphs for test functions G3, G4, G5, G11, G13 and G24

CONCLUSION

This paper presents a hybrid SP-QPSO algorithm for solving constrained optimization problems. The hybrid SP-QPSO makes use of Centroidal Voronoi Tessellations (CVT) method of point initialization, and parameter free adaptive penalty function for constraints handling. This integration ensures that the improved local and global search ability of the algorithm by the quantum movements of particles, to guide the individuals moving towards the feasible region without violating the constraints. Since, the population is divided into number of complexes, particles can learn not only from the best globally-experienced particle, but also from the best particle of each complex.

Simulation results based on ten well-known test functions demonstrate the effectiveness, efficiency and robustness of the SP-QPSO over QPSO and SPUCI. Experimental results clearly illustrate the attractiveness of this method by handling several types of constraints. The convergence curves of QPSO and SP-UCI algorithms are shown in Fig.2. From Fig 2, it is observed that the algorithm requires only 50 generations reaching the global optimum solution. The SP-QPSO convergence diagrams show how this algorithm benefits all of its defined number of iterations as the stopping criteria for deeply search the design space in all test problems. And we can also see that the SP-QPSO has a faster convergence rate and better searching result, which reveals that the proposed SP-QPSO has an improvement on QPSO and SP-UCI algorithms. From the comparative study, SP-QPSO has shown its potential to handle various COPs, and its performance is much better than eight other state-of-the-art Constrained Optimization EAs in terms of the selected performance metrics.

REFERENCES

[1] Sayed, G. I., Darwish, A., & Hassanien, A. E. (2018). *A new chaotic multi-verse optimization algorithm for solving engineering optimization problems*. Journal of Experimental & Theoretical Artificial Intelligence, 30(2), 293-317.

[2] Liu, H., Cai, Z., & Wang, Y. (2010). *Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization*. Applied Soft Computing, 10(2), 629-640.

[3] Coello, C. A. C. (2002). *Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art*. Computer methods in applied mechanics and engineering, 191(11-12), 1245-1287.

[4] Michalewicz, Z., & Schoenauer, M. (1996). *Evolutionary algorithms for constrained parameter optimization problems*. Evolutionary computation, 4(1), 1-32.

[5] Michalewicz, Z., Deb, K., Schmidt, M., & Stidsen, T. (2000). *Test-case generator for nonlinear continuous parameter optimization techniques*. IEEE Transactions on Evolutionary Computation, 4(3), 197-215.

[6] He, S., Prempain, E., & Wu, Q. H. (2004). *An improved particle swarm optimizer for mechanical design optimization problems*. Engineering Optimization, 36(5), 585-605.

[7] Kaveh, A., & Talatahari, S. (2009). *Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures*. Computers & Structures, 87(5-6), 267-283.

[8] Dong, Y., Tang, J., Xu, B., & Wang, D. (2005). *An application of swarm optimization to nonlinear programming*. Computers & Mathematics with Applications, 49(11-12), 1655-1668.

[9] Coello, C. A. C., Lamont, G. B., & Van Veldhuizen, D. A. (2007*). Evolutionary algorithms for solving multi-objective problems* (Vol. 5). New York: Springer.

[10] Das, G., Panda, S., & Padhy, S. K. (2018). *Quantum Particle Swarm Optimization Tuned Artificial Neural Network Equalizer*. In Soft Computing: Theories and Applications (pp. 579-585). Springer, Singapore.

[11] Taherzadeh, G., & Loo, C. K. (2014, October). *Comparison of Applying Centroidal Voronoi Tessellations and Levenberg-Marquardt on Hybrid SP-QPSO Algorithm for High Dimensional Problems*. In International Conference in Swarm Intelligence (pp. 332-341). Springer, Cham.

[12] Kennedy, J., & Eberhart, R. C. (1997, October). *A discrete binary version of the particle swarm algorithm. In Systems, Man, and Cybernetics*, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on (Vol. 5, pp. 4104-4108). IEEE.

[13] Kaveh, A., Bakhshpoori, T., & Afshari, E. (2015). *Hybrid PSO and SSO algorithm for truss layout and size optimization considering dynamic constraints*. Structural Engineering and Mechanics, 54(3), 453-474.

[14] He, Q., & Wang, L. (2007). *A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. Applied mathematics and computation*, 186(2), 1407-1422.

[15] Fan, S. K. S., & Zahara, E. (2007). *A hybrid simplex search and particle swarm optimization for unconstrained optimization*. European Journal of Operational Research, 181(2), 527-548.

[16] Zahara, E., & Hu, C. H. (2008*). Solving constrained optimization problems with hybrid particle swarm optimization*. Engineering Optimization, 40(11), 1031-1049.

[17] Liu, H., Cai, Z., & Wang, Y. (2010). *Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization*. Applied Soft Computing, 10(2), 629-640.

[18] Cagnina, L. C., Esquivel, S. C., & Coello Coello, C. A. (2011). *Solving constrained optimization problems with a hybrid particle swarm optimization algorithm*. Engineering Optimization, 43(8), 843–866.

[19] Kanagaraj, G., Ponnambalam, S. G., & Jawahar, N. (2013). *A hybrid cuckoo search and genetic algorithm for reliability–redundancy allocation problems*. Computers & Industrial Engineering, 66(4), 1115-1124.

[20] Xin, B., Chen, J., Zhang, J., Fang, H., & Peng, Z. H. (2012). *Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: a review and taxonomy*. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(5), 744-767.

[21] Sun, J., Feng, B., & Xu, W. (2004, June). *Particle swarm optimization with particles having quantum behavior*. In Evolutionary Computation, 2004. CEC2004. Congress on (Vol. 1, pp. 325-331). IEEE.

[22] Wu, S. (2018). *A Quantum Particle Swarm Optimization Algorithm Based on Self-Updating Mechanism*. International Journal of Swarm Intelligence Research (IJSIR), 9(1), 1-19.

[23] Fu, Y., Ding, M., & Zhou, C. (2012). *Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for UAV*. IEEE transactions on systems, man and cybernetics, part A: systems and humans, 42(2), 511-526.

[24] Mezura-Montes, E., & Coello, C. A. C. (2011). *Constraint-handling in nature-inspired numerical optimization: past, present and future.* Swarm and Evolutionary Computation, 1(4), 173-194.

[25] Salcedo-Sanz, S. (2009). *A survey of repair methods used as constraint handling techniques in evolutionary algorithms.* Computer science review, 3(3), 175-192.

[26] Fiacco, A. V., & McCormick, G. P. (1964). *The sequential unconstrained minimization technique for nonlinear programing, a primal-dual method.* Management Science, 10(2), 360-366.

[27] Talbi, E. G. (2002). A taxonomy of hybrid metaheuristics. Journal of heuristics, 8(5), 541-564.

[28] Clerc, M., & Kennedy, J. (2002). *The particle swarm-explosion, stability, and convergence in a multidimensional complex space*. IEEE transactions on Evolutionary Computation, 6(1), 58-73.

[29] Lemonge, A. C., & Barbosa, H. J. (2004). *An adaptive penalty scheme for genetic algorithms in structural optimization.* International Journal for Numerical Methods in Engineering, 59(5), 703-736.

[30] Tessema, B., & Yen, G. G. (2006, July). *A self adaptive penalty function based algorithm for constrained optimization*. In Evolutionary Computation, 2006. CEC 2006. IEEE Congress on (pp. 246-253). IEEE.

[31] Du, Q., Emelianenko, M., & Ju, L. (2006). *Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations*. SIAM journal on numerical analysis, 44(1), 102-119.

[32] Mazhoud, I., Hadj-Hamou, K., Bigeon, J., & Joyeux, P. (2013). *Particle swarm optimization for solving engineering problems: a new constraint-handling mechanism*. Engineering Applications of Artificial Intelligence, 26(4), 1263-1273.

[33] Mohamed, A. W., & Sabry, H. Z. (2012). *Constrained optimization based on modified differential evolution algorithm.* Information Sciences, 194, 171-208.

[34] Wang, Y., Cai, Z., & Zhou, Y. (2009*). Accelerating adaptive trade-off model using shrinking space technique for constrained evolutionary optimization*. International journal for numerical methods in engineering, 77(11), 1501-1534.

[35] Aragón, V. S., Esquivel, S. C., & Coello, C. A. C. (2010). *A modified version of a T-Cell Algorithm for constrained optimization problems*. International Journal for Numerical Methods in Engineering, 84(3), 351-378.