

Discrete Quantum-Behaved Particle Swarm Optimization Based on Estimation of Distribution for Combinatorial Optimization

Jiahai Wang, Yunong Zhang, Yalan Zhou and Jian Yin

Abstract—Particle swarm optimization (PSO) is a population-based swarm intelligence algorithm. A quantum-behaved particle swarm optimization (QPSO) is also proposed by combining the classical PSO philosophy and quantum mechanics. These algorithms have been very successful in solving the global continuous optimization, but their applications to combinatorial optimization have been rather limited. Estimation of distribution algorithm (EDA) samples new solutions from a probability model which characterizes the distribution of promising solutions. This paper proposes a novel discrete QPSO based on EDA for the combinatorial optimization problem. The proposed algorithm combines global statistical information extracted by EDA with local information obtained by discrete QPSO to create promising solutions. To demonstrate the performance of the proposed algorithm, experiments are carried out on the unconstrained binary quadratic programming problem which numerous hard combinatorial optimization problems can be formulated as. The results show that the discrete QPSO based on EDA have superior performance to other algorithms.

I. INTRODUCTION

THE particle swarm optimization (PSO) is inspired by observing the bird flocking or fish school [1]. A large number of birds/fishes flock synchronously, change direction suddenly, and scatter and regroup together. Each individual, called a particle, benefits from the experience of its own and that of the other members of the swarm during the search for food. Comparing with genetic algorithm, the advantages of PSO lie on its simple concept, easy implementation and quick convergence. However, the evolution equation of standard PSO (SPSO) cannot guarantee the algorithm to find out the global optimum with probability 1, that is, SPSO is not a global optimization algorithm, as van den Bergh has demonstrated [2]. Sun et al. [3] [4] proposed a global convergence-guaranteed search technique, quantum-behaved particle swarm optimization algorithm (QPSO). The QPSO

algorithm, kept to the philosophy of SPSO, is based on Delta potential well and depicted only with the position vector without velocity vector, which is a simpler algorithm. And the results show that QPSO performs better than SPSO on several benchmark test functions and is a promising algorithm due to its global convergence guaranteed characteristic. Therefore, QPSO is applied to some problems in practice, for example, clustering problem [5], and multi-period financial planning problem [6], electromagnetic design [7][8][9].

The advantages of PSO, such as simple structure, immediately accessible for practical applications, ease of implementation, quick convergence, and robustness, are shown in the literature [5-10]. The PSO algorithms have been very successful in solving the global continuous optimization, but their applications to combinatorial optimization have been rather limited and are not as effective as in global continuous optimization [11]. To solve combinatorial optimization problems, Kennedy and Eberhart also developed a discrete version of PSO (DPSO) [12]. Yin [13] proposed a genetic particle swarm optimization (GPSO) with genetic reproduction mechanisms, namely crossover and mutation to facilitate the applicability of PSO to combinatorial optimization. Recently, the QPSO has also been extended to solve the discrete optimization problems [14] [15].

On the other hand, the search in PSO is mainly based on the local information. More specifically, the search in PSO is based on particle oneself best position information and global best position information found so far. All information guided the search of PSO is a kind of local information and therefore PSO has no mechanism to extract and use global information about the search space. Estimation of distribution algorithm (EDA) [16-19] is a new class of EAs. The search in EDA is mainly based on the global information. EDA directly extracts the global statistical information about the search space from the search so far and builds a probability model of promising solutions. New solutions are sampled from the probability model. An efficient evolutionary algorithm should make use of both the local information of solutions found so far and the global information about the search space. The local information of solutions found so far is helpful for exploitation, while the global information can guide the search for exploring promising areas.

In this paper, a novel discrete QPSO based on EDA is proposed for combinatorial optimization. In the proposed algorithm, firstly, all personal best solutions from the current swarm are selected; then, a probability model is constructed to estimate the distribution of good regions over the search space based on the selected personal best solutions. Then

This work was supported in part by the Guangdong Provincial Natural Science Foundation of China (07300630, 06104916), the Specialized Research Fund for the Doctoral Program of Higher Education (20070558052, 20050558017), the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry (2007-1108), the National Science Foundation of China (60643004, 60573097, 60773198, 60703111), and the Science and Technology Office of Sun Yat-Sen University.

Jiahai Wang is with the Department of Computer Science, Sun Yat-sen University, Guangzhou, 510275, China. (corresponding author to: wjiahai@hotmail.com).

Yunong Zhang is with the Department of Electronics & Communication Engineering, Sun Yat-sen University, Guangzhou, 510275, China. (e-mail: zhyunong@mail.sysu.edu.cn).

Yalan Zhou is with the Department of Computer Science, Sun Yat-sen University, Guangzhou, 510275, China. (e-mail: zhouylan@163.com).

Jian Yin is with the Department of Computer Science, Sun Yat-sen University, Guangzhou, 510275, China. (e-mail: issjyin@mail.sysu.edu.cn).

mainstream point of discrete QPSO is generated by sampling the constructed probability distribution model, which increases the diversity of the swarm and can lead the swarm to escape from local optima. Moreover, in the proposed algorithm, one part of a solution generated by the original discrete QPSO, the other part of a solution is sampled in the search space from the constructed probability distribution model. Therefore, a solution generated by the offspring generation scheme of the proposed algorithm is based on the local information and the global statistical information. Thus, the proposed algorithm makes use of both the local information of solutions found so far and the global information about the search space. The local information of solutions found so far is helpful for exploitation, while the global information can guide the search for exploring promising areas. In order to keep the diversities in the population, a bit flip mutation operator is also incorporated into the proposed algorithm. To demonstrate the performance of the proposed algorithm, experiments are carried out on the unconstrained binary quadratic programming problem which numerous hard combinatorial optimization problems can be formulated as. The results show that the EDA can significantly improve the performance of discrete QPSO and discrete QPSO based on EDA have superior performance to other evolutionary algorithms.

The rest of the paper is organized as follows. Section 2 briefly reviews the SPSO, QPSO and DQPSO. In Section 3, the EDA is first briefly introduced, and then a novel discrete QPSO based on EDA is proposed. The experimental simulation results are reported in Section 4. Section 5 concludes this paper.

II. PSO AND QUANTUM-BEHAVED PSO

A. Standard PSO (SPSO)

PSO is initialized with a group of random particles (solutions) and then searches for the optima by updating each generation. In each iteration, each particle is updated by the following two best values. The first one is the local best solution a particle has obtained so far. This value is called personal best solution. Another best value is that the whole swarm has obtained so far. This value is called global best solution. The philosophy behind the original PSO is to learn from individual's own experience (personal best solution) and the best individual experience (global best solution) in the whole swarm.

Denote by N particle number in the swarm. Let $X_i(t) = (x_{i1}(t), \dots, x_{id}(t), \dots, x_{iD}(t))$, be particle i with D bits at iteration t , where being treated as a potential solution. Denote the velocity as $V_i(t) = (v_{i1}(t), \dots, v_{id}(t), \dots, v_{iD}(t))$, $v_{id}(t) \in R$. Let $PBest_i(t) = (pbest_{i1}(t), \dots, pbest_{id}(t), \dots, pbest_{iD}(t))$ be the best solution that particle i has obtained until iteration t , and $GBest(t) = (gbest_1(t), \dots, gbest_d(t), \dots, gbest_D(t))$ be the best solution obtained from $PBest_i(t)$ in the whole swarm at iteration t . Each particle adjusts its velocity according to

previous velocity of the particle, the cognition part and the social part. The algorithm is described as follows [1]:

$$v_{id}(t+1) = v_{id}(t) + c_1 \cdot r_1 \cdot (pbest_{id}(t) - x_{id}(t)) + c_2 \cdot r_2 \cdot (gbest_d(t) - x_{id}(t)), \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1), \quad (2)$$

where c_1 is the cognition learning factor and c_2 is the social learning factor; r_1 and r_2 are the random numbers uniformly distributed in $[0,1]$;

In [20], Clerc and Kennedy analyze the trajectory and prove that each particle i in the PSO system converges to its local attractor point g_i , whose coordinates are $g_{id} = \varphi \cdot pbest_{id} + (1 - \varphi) \cdot gbest_d$ so that the best previous position of all particles will converge to an exclusive global position with $t \rightarrow \infty$, where φ is a random number distributed uniformly on $[0,1]$. In fact, collectiveness of particles in traditional PSO confines the search scope of the particle, and consequently it is not allowed that the particle appears at a position far from the particle swarm, which may be a solution with better fitness than the current global best solution.

B. Quantum-behaved PSO (QPSO)

SPSO is not a global convergence-guaranteed optimization algorithm, as van den Bergh has demonstrated [2]. Therefore, Sun et al. [3] [4] proposed a global convergence-guaranteed search technique, quantum-behaved particle swarm optimization algorithm (QPSO), whose performance is superior to the standard PSO (SPSO).

In the quantum model of a PSO, the state of a particle is depicted by wavefunction $\psi(x, t)$, instead of position and velocity. The dynamic behavior of the particle is widely different from that of the particle in traditional PSO systems in that the exact values of position and velocity cannot be determined simultaneously. We can only learn the probability of the particle's appearing in position x from probability density function $|\psi(x, t)|^2$, the form of which depends on the potential field the particle lies in. The particles move according to the following iterative equation [3] [4]:

$$x_{id}(t+1) = \begin{cases} g_{id} - \beta \cdot |mbest_d - x_{id}(t)| \cdot \ln(1/u) & \text{if } \text{rand}() \geq 0.5 \\ g_{id} + \beta \cdot |mbest_d - x_{id}(t)| \cdot \ln(1/u) & \text{otherwise} \end{cases} \quad (3)$$

where

$$g_{id} = \varphi \cdot pbest_{id} + (1 - \varphi) \cdot gbest_d, \quad (4)$$

and

$$mbest_d = \frac{1}{N} \sum_{i=1}^N pbest_{id}, \quad (5)$$

$mbest$ (mean best position or mainstream thought point) is defined as the mean value of all particles' the best position, φ and u are random number distributed uniformly on $[0,1]$ respectively. $L = \beta \cdot |mbest_d - x_{id}(t)| \cdot \ln(1/u)$ can be viewed as the strength of creativity or imagination because it characterizes the knowledge seeking scope of the particle,

and therefore the larger the value L , the more likely the particle find out new knowledge. The parameter, β , called contraction-expansion coefficient, is the only parameter in QPSO algorithm. From the results of stochastic simulations, QPSO has relatively better performance by varying the value of β from 1.0 at the beginning of the search to 0.5 at the end of the search to balance the exploration and exploitation [21].

The QPSO algorithm, kept to the philosophy of PSO, is based on Delta potential well and depicted only with the position vector without velocity vector, which is a simpler algorithm.

C. Discrete QPSO (DQPSO)

QPSO can be generalized to discrete binary search space by redefining local attractor g_i , the $mbest$ (mean best position) and the strength of creativity $L = \beta \cdot |mbest_d - x_{id}(t)| \cdot \ln(1/u)$ in binary search space [14] [15].

Firstly, we describe how to compute the local attractor for each particle in binary space. Eq. (4) is used for getting the coordinate of the local attractor g_i for particle i in the continuous space. Therefore, g_i is generated through arithmetic crossover between $pbest_i$ and $gbest$, and the coordinate of g_i lies between $pbest_i$ and $gbest$. In binary space, The point g_i can be generated through binary or discrete crossover operation like that used in genetic algorithm (GA) with discrete coding. That is, g_i is randomly selected from two offspring that are generated by exerting crossover on the two parents, $pbest_i$ and $gbest$. In this paper, uniform crossover is used to compute the local attractor.

Secondly, we describe how to compute the $mbest$ (mean best position) in binary space. At first, DQPSO compute the center of personal best positions m_d :

$$m_d = \frac{\sum_{i=1}^N pbest_{id}}{N}, \quad (6)$$

then the mean best position is determined by m_d by the following way:

if $m_d > 0.5$, set $mbest_d = 1$,
if $m_d < 0.5$, set $mbest_d = 0$,
if $m_d = 0.5$ set $mbest_d$ to be 1 or 0, with probability 0.5 for either state.

That is, if more particles take on 1 at the d th bit of their own $pbests$, the d th bit of $mbest$ will be 1; otherwise the bit will be 0. However, if half of the particles take on 1 at the d th bit of their $pbests$, the d th bit of $mbest$ will be set randomly to be 1 or 0, with probability 0.5 for either state.

Thirdly, we describe how to define the strength of creativity or imagination L in binary space. In the binary space, the basic operation is the bit flip, and an individual moves to nearer or farther corners of the hypercube (searching space) by flipping bits in its position vector. Therefore the distance (difference) between two solutions is measured by Hamming distance. Since an individual moves to nearer or farther corners of the hypercube (searching space) by flipping bits in its position vector, the distance vector between the $mbest$ (mean best position) and population individual X_i (denoted

as $|mbest_d - x_{id}(t)|$ of the d th dimension) can be described by a vector whose bit is set to 1 if the corresponding bits of $mbest$ and X_i are different and 0 otherwise. Thus, the operation $|mbest_d - x_{id}(t)|$ in Eq.(3) can be substituted with $mbest_d \oplus x_{id}$, where “ \oplus ” is the bit-wise exclusive-or (XOR) operator.

For each bit, $\beta \ln(1/u)$ can be seen as the probability that a change is applied to the corresponding bit of g_i to produce the mutant local attractor vector, that is, $\beta \ln(1/u)$ is like the mutation probability in GA. If a bit in the distance vector is 1 and change is activated by the probability $\beta \ln(1/u)$ (that is, $rand() < \beta \ln(1/u)$), the corresponding bit in the g_i vector will be reversed. This reverse operation is also an equivalence of the XOR operator.

Discrete QPSO can be described as follows:

$$x_{id}(t+1) = g_{id} \oplus (R_d(F) \otimes (mbest_d \oplus x_{id}(t))) \quad (7)$$

where

$$R_d(F) = \begin{cases} 1 & \text{if } rand() < F \\ 0 & \text{otherwise} \end{cases}, \quad (8)$$

$$F = \text{Min}(\beta \ln(1/u), 1), \quad (9)$$

and “ \otimes ” is the bit-wise AND operator.

III. PROPOSED NOVEL DISCRETE QPSO BASED ON EDA

In this section, EDA is incorporated into discrete QPSO and thus a novel DQPSO based on EDA is proposed for combinatorial optimization problem. Then we first briefly review the EDA.

A. Estimation of Distribution Algorithm (EDA)

Evolutionary algorithms, which first use information obtained during the optimization process to build probabilistic models of the distribution of good regions in the search space, and then use these models to generate new solutions, are called estimation of distribution algorithms (EDAs) [16-19]. These algorithms, which have a theoretical foundation in probability theory, are also based on populations that evolve as the search progresses. EDAs use probabilistic modeling of promising solutions to estimate a distribution over the search space, which is then used to produce the next generation by sampling the search space according to the estimated distribution. After every iteration, the distribution is re-estimated.

An algorithmic framework of most EDAs can be described as:

```
Pop = InitializePopulation( )
while Stopping criteria are not satisfied do
    Popsel = Select(Pop)/*Selection*/
    Prob = Estimate(Popsel)/*Estimation*/
    Pop = Sample(Prob)/*Sampling*/
endwhile
```

An EDA starts with a solution population Pop and a solution distribution model $Prob$. The main loop consists of three principal stages. The first stage is to select the best individuals (according to some fitness criteria) from

the population. These individuals are used in the second stage in which the solution distribution model $Prob$ is updated or recreated. The third stage consists of sampling the updated solution distribution model to generate new solutions offspring. The reviews of EDA can be found in Refs. [17] [18]. There has been a growing interest for EDAs in the last year. More comprehensive presentation of the EDA field can be found in Ref. [19].

B. Novel DQPSO Based on EDA (DQPSO-EDA)

In this section, EDA is incorporated into discrete QPSO and thus a novel DQPSO based on EDA is proposed for combinatorial optimization problem.

Inspired by EDA, which try to guide its search towards a promising area by sampling new solutions from a probability model, we incorporate the EDA mechanism into the DQPSO algorithm in order to create solutions which are more promising, and consequently, to explore the search space more effectively. The position updates for the DQPSO based on EDA can be described as follows:

$$x_{id}(t+1) = \begin{cases} g_{id} \oplus (R_d(F) \otimes (mbest_d \oplus x_{id}(t))) & \text{if } \text{rand}() < CR \\ EDA(p_d) & \text{otherwise} \end{cases}, \quad (10)$$

where $P = (p_1, p_2, \dots, p_d, \dots, p_D)$ is a probability vector to characterize the distribution of promising solutions in the search space, and CR is crossover rate to balance contributions of the local information and the global information.

Several different probability models have been introduced in EDAs for modeling the distribution of promising solutions. The univariate marginal distribution (UMD) model is the simplest one and has been successfully population-based incremental learning (PBIL) [22], compact GA [23] and evolutionary algorithm with guided mutation [24]. Therefore, in this paper the UMD model is adopted to estimate the distribution of good regions over the search space based on the personal best solutions (according to some fitness criteria) from the population.

In the proposed algorithm, firstly, all personal best solutions from the current swarm are selected; then, the UMD model is adopted to estimate the distribution of good regions over the search space based on the selected personal best solutions. The UMD uses a probability vector $P = (p_1, p_2, \dots, p_d, \dots, p_D)$ to characterize the distribution of promising solutions in the search space, where p_d is the probability that the value of the d -th position of a promising solution is 1.

In the proposed algorithm described by Eq.(10), $mbest$ is also generated by sampling the solution distribution model in the following way:

$$mbest_d = \begin{cases} 1 & \text{if } \text{rand}() < p_d \\ 0 & \text{otherwise} \end{cases}. \quad (11)$$

New offspring solutions are partly generated by sampling the updated solution distribution model in the following way:

$$EDA(p_d) = \begin{cases} 1 & \text{if } \text{rand}() < p_d \\ 0 & \text{otherwise} \end{cases}. \quad (12)$$

In the proposed algorithm described by Eq.(10), a bit of the particle is copied from the mutant local attractor, or sampled from the probability vector P randomly, which is controlled or balanced by the parameter CR .

The probability vector P is initialized by the following rule:

$$p_d = m_d = \frac{\sum_{i=1}^N pbest_{id}}{N}. \quad (13)$$

The probability vector in the proposed algorithm can be learned and updated at each iteration for modelling the distribution of promising solutions. Since some elements of the offspring are sampled from the probability vector P , it can be expected that should fall in or close to a promising area. The sampling mechanism can also provide diversity for the search. At each iteration t in the proposed algorithm, the best personal best solutions of all the particles are selected and used for updating the probability vector P . Then, the probability vector P can be updated in the same way as in the PBIL algorithm [22]:

$$p_d = (1 - \lambda)p_d + \lambda \frac{\sum_{i=1}^N pbest_{id}}{N}. \quad (14)$$

where $\lambda \in [0, 1]$ is the learning rate. As in PBIL [22], the probability vector P is used to generate the next set of sample points; the learning rate also affects which portions of the problem space will be explored. The setting of the learning rate has a direct impact on the trade-off between global exploration of the problem space and local exploitation of the exploration already conducted. For example, if the learning rate is 0, there is no exploitation of the information gained through search. As the learning rate is increased, the amount of exploitation increases, and the ability to search large portions of the problem space diminishes.

In order to keep the diversities in the swarm, a bit flip mutation operator is also incorporated into the proposed algorithms. After each bit is decided in accordance with Eq.(10), the mutation operator independently flips the bit of a particle with a mutation probability.

From Eq.(10), Eq.(11) and Eq.(12), we can find that two key modifications or characteristics are introduced in the proposed discrete QPSO.

The first characteristic of the proposed algorithm is that the mean best point $mbest$ is sampled from probability model according to Eq.(11). In original discrete QPSO, if more particles take on 1 at the j th bit of their own pbests, the j th bit of $mbest$ will be 1; otherwise the bit will be 0. However, if half of the particles take on 1 at the d th bit of their pbests, the d th bit of $mbest$ will be set randomly to be 1 or 0, with probability 0.5 for either state. But in the proposed algorithm, the mainstream thought point or mean best position ($mbest$) is sampled from probability vector described by Eq. (11) which describes the distribution of good regions over the search space.

The probability vector in the proposed algorithm can be learned and updated at each iteration according Eq.(14). Eq.

(14) can also be expressed as

$$p_d = p_d + \lambda(m_d - p_d), \quad (15)$$

m_d is the center of personal best solutions. The values in the probability vector are gradually shifted or moved towards the center of the new personal best solutions. The probability update rule Eq.(15), which is based upon the update rule of competitive learning, is more similar to the mainstream thought evolution in the reality world. This probability update rule determines a shift of the current mainstream toward new mainstream. The size of this shift is determined by the learning parameter λ . In the extreme cases, there is either very little learning or update in the mainstream (when λ is very close to zero), or the current mainstream is replaced by the new mainstream (when $\lambda = 1$). Furthermore, the random sampling also provides diversity for the mean best position, which also increases the diversity of the swarm and can lead the swarm to escape from local optima.

The second characteristic of the proposed algorithm is that the solution generation scheme is based on local information and the global statistical information. In the proposed algorithm, one part of a solution comes from the original discrete QPSO, the other part of a solution is sampled in the search space from the constructed probability distribution model. Therefore, a solution generated by the offspring generation scheme Eq.(10) is based on the local information and the global statistical information. CR is used to balance contributions of the local information and the global information. Therefore, the proposed algorithm makes use of both the local information of solutions found so far and the global information about the search space. The local information of solutions found so far is helpful for exploitation, while the global information can guide the search for exploring promising areas.

IV. SIMULATION RESULTS

In this section, the proposed algorithms are tested on the unconstrained binary quadratic programming problem (UBQP) which numerous hard combinatorial optimization problems can be formulated as [25-28].

The UBQP can be formulated as [25-28]

$$\max f(X) = \sum_{i=1}^n \sum_{j=1}^n q_{ij} x_i x_j, \quad (16)$$

subject to

$$x_i \in \{0, 1\}, i = 1, 2, \dots, n,$$

where q_{ij} are entries of a given $n \times n$ symmetric rational matrix $[q_{ij}]$.

It is well-known that any generalized objective involving both linear and quadratic terms such as:

$$\max f(X) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j + \sum_{i=1}^n b_i x_i, \quad (17)$$

can be converted into the objective shown in Eq.(16) above involving the symmetric matrix $[q_{ij}]$ by noting that $x_i =$

TABLE I
MAIN CHARACTERISTICS OF 15 BENCHMARK TEST
PROBLEMS FOR UBQP.

Class	No.	n	Density	Linear coef. q_{ii}	Quadr. coef. q_{ij}	Best known value [27]
F_1	1	500	0.1	[-75,75]	[-50,50]	61194
	2	500	0.25	[-75,75]	[-50,50]	100161
	3	500	0.5	[-75,75]	[-50,50]	138035
	4	500	0.75	[-75,75]	[-50,50]	172771
	5	500	1	[-75,75]	[-50,50]	190507
F_2	1	500	0.1	[0,100]	[-50,0]	4029
	2	500	0.25	[0,100]	[-50,0]	2010
	3	500	0.5	[0,100]	[-50,0]	1094
	4	500	0.75	[0,100]	[-50,0]	685
	5	500	1	[0,100]	[-50,0]	418
G_2	1	1000	0.1	[0,100]	[-50,0]	5216
	2	1000	0.3	[0,100]	[-50,0]	2122
	3	1000	0.5	[0,100]	[-50,0]	1272
	4	1000	0.7	[0,100]	[-50,0]	866
	5	1000	1	[0,100]	[-50,0]	452

$(x_i)^2$ (as $x_i \in \{0, 1\}$) and setting $q_{ii} = a_{ii} + b_i$ for $\forall i$, and for $q_{ij} = a_{ij} + a_{ji}/2$ for $\forall i \neq j$.

UBQP is known to be NP-hard [29]. It is a unified model for a variety of combinatorial optimization problems [25-28]. The model (16) or (17), sometimes with some constraints added (which, however, can be reformulated as the model (16) [25]), is of considerable practical significance. Numerous hard problems in many diverse areas have been formulated as UBQP problems, including VLSI design, manufacturing, computer aided design, reliability theory and statistics, and so on. Moreover, many graph problems can be converted to UBQP including the maximum clique problem, maximum cut problem, minimum covering problem, and graph coloring problem [25].

The 15 most difficult benchmark test problems used in other studies of UBQP [26] [27] were also tested in this paper. The basic parameters and the best known solutions of these problems are shown in Table 1. Density of the problems means the ratio of the number of nonzero coefficients of quadratic terms in Eq.(16) or Eq.(17). These benchmark test problems, F_1 , F_2 and G_2 can be downloaded from the Hearin Center for Enterprise Science website (<http://hces.bus.olemiss.edu/tools.html>). There are also several group test instances contained in the ORLIB [30]. These instances are not considered in our experiments, since they are solved easily by heuristics and thus provide no challenge for the proposed algorithm [26] [27].

The original DQPSO (described in Section 2.3), the proposed DQPSO with only the first modification or characteristic (DQPSO with EDA mbest described by Eq.(7) and Eq. (11)), the proposed DQPSO with all the modifications or characteristics (DQPSO-EDA described by Eq.(10), Eq.(11) and Eq.(12)), and existing powerful evolutionary algorithm GPSO [13] were implemented and evaluated for comparison. The GPSO was proposed and applied to solve the polygonal approximation problem, which is one of the instances in combinatorial optimization [13]. The experimental results showed that the GPSO with genetic reproduction features

TABLE II
COMPARISON OF THE RESULTS OBTAINED BY THE ORIGINAL DQPSO, DQPSO WITH EDA MBEST, DQPSO-EDA, AND GPSO.

Class	No.	DQPSO [14]		DQPSO with EDA mbest		GPSO [13]		DQPSO-EDA	
		Best	Av. Std.	Best	Av. Std.	Best	Av. Std.	Best	Av. Std.
F_1	1	26776	22467.6 2144.96	53656	51450.8 1578.85	60823	59937.4 521.39	60856	60419.3 253.76
	2	43151	36878.6 3900.72	91209	85412.8 3319.73	99607	98302.1 623.27	99974	99088.4 529.1
	3	60149	52072.2 4107.46	122333	116483 3693.45	136533	135026 945.46	137432	136305 688.78
	4	72589	65773.9 3929.29	156504	147397 4178.51	171739	169571 1453.7	172595	171207 623.73
	5	83690	73703.9 4188.46	171736	159267 5731.45	188128	185883 1674.77	189811	188158 1023.3
F_2	1	-43803	-54870.7 6509.03	1230	-666.1 1288.19	3626	3235.6 202.63	3601	3359.95 143.74
	2	-114016	-148674 16988.1	-572	-7085.2 3864.31	1685	1551.1 87.65	1805	1606 113.54
	3	-259147	-311848 31880.6	-3821	-13545.4 6967.66	902	813.95 47.63	1033	827.55 87.65
	4	-391543	-480143 50606.5	-8400	-25677 11836.1	580	515.45 32.71	585	515.1 39
	5	-522114	-636735 55930.6	-23651	-45010.3 13662.8	355	303.1 28.73	384	317.2 35.83
G_2	1	-288852	-311088 14453.4	-4270	-8413.6 3017.13	3849	3265.75 173.82	4466	4031.4 190.52
	2	-833751	-970984 67538.3	-7357	-15212.2 4987.31	1538	1263.9 109.37	1805	1519.45 121.77
	3	-1.4962E+6	-1.64425E+6 86359.6	-36185	-68573 25459.4	851	751.5 64.43	1035	839.05 109.72
	4	-1.98974E+6	-2.35327E+6 150317	-52830	-135018 55978.1	611	498.5 58.58	685	571.25 75.08
	5	-2.88123E+6	-3.30784E+6 172982	-86664	-137326 29578.3	385	280.1 69.77	383	306.15 58.32

can significantly improve the search efficacy of DPSO for combinatorial optimization [13], and further, is comparable to or better than GA and ant colony optimization (ACO) [13]. At the same time, the implement of GPSO is simpler than GA and ACO [13]. Therefore, we select the GPSO for comparison. All the algorithms were implemented in C on a PC (Pentium4 2.80 GHz). The comparison between DQPSO with EDA mbest, DQPSO-EDA and original DQPSO is to show how the modifications based on EDA can improve the performance of the DQPSO. The comparison with the GPSO is to show the superior search ability of the proposed hybrid algorithms to other evolutionary algorithm.

In the proposed algorithms, there are several parameters to be selected. Following the suggestion in [21], the value of contraction-expansion or creativity coefficient parameter, β , is dynamically tuned from 1.0 to 0.5 according to the number of generations such that more exploration search is pursued during the early generations and the exploitation search is emphasized afterward. The crossover rate CR is used to

balance contributions of the local information and the global information and therefore we set $CR = 0.4$ experimentally. The learning rate λ is set to a random number selected from a uniform distribution in $[0,1]$. Thus, different dimensions of the probability vector adopt different random learning rates, which is a random way to balance the exploration and exploitation ability. The role of the bit flip mutation in the proposed algorithms is just like that in GA where mutation probability is set to a small value. Therefore, mutation probability is set to 0.001 in the proposed algorithm.

Certainly, the parameters are chosen experimentally, and tuning of these parameters may be necessary when solving different optimization problems. But from our experience, the tuning is only on a small scale because we have discussed the meanings or impacts of the parameters. Therefore, several preliminary experiments were undertaken to select the parameters mentioned above.

In the GPSO, the standard parameters are adopted from Ref. [13].

In all algorithms, the population size and maximum iteration number are set to 40 and 1500, respectively.

We implemented each algorithm 20 times for each benchmark problem, keeping the best, average and standard deviation values provided by each algorithm. Table 2 shows the comparison of the results obtained by the original DQPSO, DQPSO with EDA mbest, DQPSO-EDA, and existing powerful evolutionary algorithm GPSO. From Table 2, we can find that the proposed hybrid algorithms based on EDA outperform the other algorithms. More specifically, several results can be observed.

First, the DQPSO with EDA mbest performs better than the original DQPSO. Then mainstream point or mean best point of the DQPSO with EDA mbest is generated by sampling the constructed probability distribution model, which increases the diversity of the swarm and can lead the swarm to escape from local optima.

Second, the DQPSO-EDA performs better than the DQPSO and the DQPSO with EDA. A solution generated by the offspring generation scheme of the DQPSO-EDA is based on the local information and the global statistical information, which improve the performance of the proposed algorithm.

Third, the DQPSO-EDA performs better than the GPSO, which indicated DQPSO based on EDA is better than other evolutionary algorithm.

It is now well-known that the performance of pure EAs is not satisfactory to the complex combinatorial optimization problems because pure EAs are not well suited for fine-tuning solutions [31]. EAs with local search techniques, called memetic algorithms, can greatly improve the efficiency of search [28]. Therefore we do not claim that the proposed DQPSO based on EDA can provide competitive results compared with other specialized UBQP algorithms or memetic algorithms [28]. Instead, our main purpose of this paper is to show how the EDA can improve the performance of the DQPSO. We also show the good relative performance of the proposed algorithm by comparing them with other evolutionary algorithms. In order to improve the absolute performance of the proposed algorithm for UBQP, some powerful local search algorithms with linear time complexity like Ref. [28] can be directly incorporated into the proposed hybrid algorithms. Therefore, hybridizing the proposed algorithm with suitable local search and applications of the proposed algorithm to other combinatorial optimization problems are future research topics.

V. CONCLUSIONS

This paper proposes a novel discrete QPSO based on EDA for the combinatorial optimization problem. The proposed algorithm combines global statistical information extracted by EDA with local information obtained by discrete QPSO to create promising solutions. The simulation results on the unconstrained binary quadratic programming problem show that the EDA can significantly improve the performance of DQPSO and DQPSO based on EDA have superior performance to other evolutionary algorithms. Further, hybridizing

the proposed algorithms with suitable local search and applications to other combinatorial optimization problems is our future work.

REFERENCES

- [1] J. Kennedy, R.C. Eberhart, "Particle swarm optimization," Proceedings of IEEE International Conference on Neural Networks. NJ: Piscataway, pp.1942–1948, 1995.
- [2] F. Van den Bergh, An analysis of particle swarm optimizers. PhD Thesis. University of Pretoria, Nov. 2001.
- [3] J. Sun et al., "Particle Swarm optimization with particles having quantum behavior," Proc.of Congress on Evolutionary Computation, pp.325–331, 2004.
- [4] J. Sun et al., "A global search strategy of quantum-behaved particle swarm optimization," Proc. IEEE Conference on Cybernetics and Intelligent Systems, pp.111–116, 2004.
- [5] J. Sun, W. Xu, B. Ye, "Quantum-behaved particle swarm optimization clustering algorithm," Proceedings Advanced Data Mining and Applications, Lecture Notes in Artificial Intelligence, vol.4093, pp.340–347, 2006.
- [6] J. Sun, W. Xu, W. Fang, "Solving multi-period financial planning problem via quantum-behaved particle swarm algorithm," Lecture Notes In Artificial Intelligence, vol.4114, pp.1158–1169, 2006.
- [7] L.S. Coelho, "Novel Gaussian quantum-behaved particle swarm optimizer applied to electromagnetic design," IET Science, Measurement & Technology, vol.1, no.5, pp.290–294, Sept. 2007.
- [8] L.S. Coelho, "A quantum particle swarm optimizer with chaotic mutation operator," Chaos, Solitons & Fractals, 2008 (In press).
- [9] S.M. Mikkilä, and A.A. Kishk, "Quantum Particle Swarm Optimization for Electromagnetics," IEEE Transactions on Antennas and Propagation, vol.54, no.10, pp. 2764–2775, Oct. 2006.
- [10] K.E. Parsopoulos and M.N. Vrahatis, "Recent approaches to global optimization problems through Particle Swarm Optimization," Natural Computing, vol.1, no.2–3, pp.235–306, 2002.
- [11] X. Hu, Y. Shi and R.C. Eberhart, "Recent advance in particle swarm," Proceedings of the 2004 Congress on Evolutionary Computation. Piscataway, NJ: IEEE Press, pp.90–97, 2004.
- [12] J. Kennedy, R.C. Eberhart, "A discrete binary version of the particle swarm algorithm," Proceedings of the World Multiconference on Systems, Cybernetics and Informatics. NJ: Piscataway, pp. 4104–4109, 1997.
- [13] P.Y. Yin, "Genetic particle swarm optimization for polygonal approximation of digital curves," Pattern Recognition and Image Analysis, vol.16, no.2, pp.223–233, 2006.
- [14] J. Sun, W. Xu, W. Fang, Z. Chai, "Quantum-Behaved Particle Swarm Optimization with Binary Encoding," Proceedings of 8th International Conference on Adaptive and Natural Computing Algorithms, Part I. Lecture Notes in Computer Science, vol. 4431, pp.376–385, Springer, 2007.
- [15] D. Zhou, J. Sun, W. Xu, "Polygonal approximation of curves using binary quantum-behaved particle swarm optimization," Journal of Computer Applications, vol.27, no. 8, pp.2030–2032, 2007 (In Chinese).
- [16] H.Muhlenbein, G. Paaß, "From recombination of genes to the estimation of distributions," Proceedings of the 4th Conference on Parallel Problem Solving from Nature-PPSN IV, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds. Lecture Notes in Computer Science, vol. 1411. Springer, Berlin, pp.178–187, 1996.
- [17] M. Zlocchin, M. Birattari, N. Meuleau, M. Dorigo, "Model-based search for combinatorial optimization: A critical survey," Annals of Operations Research, vol.131, no.1–4, pp.373–395, Oct. 2004.
- [18] S. Kern, S.D. Muller, N. Hansen, D.Buche, J. Ocenasek, P. Koumoutsakos, "Learning probability distributions in continuous evolutionary algorithms-a comparative review," Natural Computing, vol.3, no.1, pp.77–112, 2004.
- [19] P. Larranaga, J. Lozano, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Genetic Algorithms and Evolutionary Computation, vol. 2. Springer, 2001.
- [20] M. Clerc and J. Kennedy, "The Particle Swarm: Explosion, Stability and Convergence in a Multi-Dimensional Complex Space", IEEE Transaction on Evolutionary Computation, vol.6, pp.58–73, 2002.
- [21] J. Sun, W. Xu, J. Liu, "Parameter Selection of Quantum-Behaved Particle Swarm Optimization," Proceedings Advances in Natural Computation, Lecture Notes in Computer Science, vol.3612, pp.543–552, 2005.

- [22] S. Baluja, "Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning," School of Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-94-163, 1994.
- [23] G.R. Harik, F.G. Lobo, and D.E. Goldberg, "The compact genetic algorithm," *IEEE Trans. Evol. Comput.*, vol.3, no.4, pp. 287–297, 1999.
- [24] Q. Zhang, J. Sun and E. Tsang, "An evolutionary algorithm with guided mutation for the maximum clique problem," *IEEE Trans. on Evolutionary Computation*, vol.9, no.2, pp.192–200, 2005.
- [25] G. Kochenberger, F. Glover, B. Alidaee and C. Rego, "A unified modeling and solution framework for combinatorial optimization problems," *OR Spectrum*, vol.26, pp.1–14, 2004.
- [26] F. Glover, B. Alidaee, C. Rego, and G. Kochenberger, "One-pass heuristics for large-scale unconstrained binary quadratic problems," *European Journal of Operational Research*, vol.137, pp.272–287, 2002.
- [27] E. Boros, P.L. Hammer, G. Tavares, "Local search heuristics for Quadratic Unconstrained Binary Optimization (QUBO)," *Journal of Heuristics*, vol.13, pp.99–132, 2007.
- [28] P. Merz, and K. Katayama, "Memetic algorithms for the unconstrained binary quadratic programming problem," *BioSystems*, vol.78, pp.99–118, 2004.
- [29] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.
- [30] J.E. Beasley, OR-Library, <http://people.brunel.ac.uk/mas-tjjb/jeb/orlib/bqpinfo.html>.
- [31] N. Krasnogor and J. Smith, "A tutorial for competent memetic algorithms: model, taxonomy, and design issues," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 5, pp.474–488, Oct. 2005.