IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 3, 2017, accepted June 19, 2017, date of publication July 4, 2017, date of current version July 31, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2723538

Entanglement-Enhanced Quantum-Inspired Tabu Search Algorithm for Function Optimization

SHU-YU KUO AND YAO-HSIN CHOU
National Chi Nan University, Puli 54561, Taiwan
Corresponding author: Yao-Hsin Chou (yhchou@ncnu.edu.tw)

ABSTRACT Many metaheuristic algorithms have been proposed to solve combinatorial and numerical optimization problems. Most optimization problems have high dependence, meaning that variables are strongly dependent on one another. If a method were to attempt to optimize each variable independently, its performance would suffer significantly. When traditional optimization techniques are applied to high-dependence problems, they experience difficulty in finding the global optimum. To address this problem, this paper proposes a novel metaheuristic algorithm, the entanglement-enhanced quantum-inspired tabu search algorithm (Entanglement-QTS), which is based on the quantum-inspired tabu search (QTS) algorithm and the feature of quantum entanglement. Entanglement-QTS differs from other quantum-inspired evolutionary algorithms in that its *Q*-bits have entangled states, which can express a high degree of correlation, rendering the variables more intertwined. Entangled Q-bits represent a state-of-the-art idea that can significantly improve the treatment of multimodal and high-dependence problems. Entanglement-QTS can discover optimal solutions, balance diversification and intensification, escape numerous local optimal solutions by using the quantum not gate, reinforce the intensification effect by local search and entanglement local search, and manage strong-dependence problems and accelerate the optimization process by using entangled states. This paper uses nine benchmark functions to test the search ability of the entanglement-QTS algorithm. The results demonstrate that Entanglement-QTS outperforms QTS and other metaheuristic algorithms in both its effectiveness at finding the global optimum and its computational efficiency.

INDEX TERMS Quantum-inspired tabu search (QTS), quantum entanglement, metaheuristic algorithms, function optimization.

I. INTRODUCTION

Metaheuristic algorithms, which are general and global optimization approaches that have been applied to many combinatorial optimization problems, play a major role in computational intelligence. These optimization algorithms originated from the observations of natural phenomena and were developed into search algorithms. For example, genetic algorithm (GA) [1] adopts the principle of biological evolution, particle swarm optimization (PSO) [2] imitates the behavior of birds foraging, and quantum-inspired evolutionary algorithm (QEA) [3] is based on the concept and principle of quantum computing.

Quantum-inspired evolutionary algorithms, [4] (QIEAs, based on the concept of quantum mechanics [6]–[8], not quantum algorithms) constitute an emerging branch of evolutionary computation [5]. QIEAs solve the problem of premature convergence that is common in traditional evolutionary algorithms, and uses fewer populations to obtain the optimal solution. Compared with conventional evolutionary algorithms, QIEAs can more easily balance exploration and exploitation. First proposed by Han and Kim [3], QEA have two major characteristics: quantum-inspired bits (Q-bits) and quantum-inspired gates (Q-gates). A Q-bit is a probabilistic representation. Q-bit individuals express a linear superposition of all possible states in a search space. Q-gates allow individuals to move toward a better solution and find the global optimum. However, some solvable problems persist, such as the Q-gates not being general and the situation of the algorithm becoming trapped in local optima not being improved. Hence, QIEAs have combined the features of QEA with the advantages of other algorithms to address these concerns: examples include quantum-inspired particle swarm optimization (QPSO), [9], [10], the quantum-inspired immune clonal algorithm (QICA), [11], the quantum-inspired differential evolution (QDE) algorithm [12], and the quantum-inspired tabu search (QTS) algorithm [13], [14]. The last of these (i.e., QTS) is a relatively new algorithm that applies a novel, effective, and efficient global search strategy.

The QTS algorithm includes both Q-bits and Q-gates. It retains the advantages of QIEAs, but the methods used for the Q-gates differ from those used in traditional methods; QTS uses the best and worst solutions to simultaneously move individuals toward the best solution and away from the worst solution. This simultaneous guidance is why QTS quicker and more efficient than other heuristic algorithms. QTS has been applied to many complex combinatorial optimization problems, such as the 0/1 knapsack problem [14], stock market trading [15]–[17], reversible logic circuit synthesis [18], and function optimization [19], to demonstrate its incomparable ability to seek and find the global optimum. However, QTS is limited when solving certain function optimization problems that all exhibit high dependency, which is similar to quantum entanglement. To improve the global search ability, this paper adopts the concept of quantum entanglement and the QTS algorithm to solve global optimization problems more effectively and efficiently.

Many function optimization problems have high dependency, meaning that variables are strongly dependent on one another. If a method were to attempt to optimize each variable independently, its performance would suffer significantly. The results of the present study indicate that this dependency is present in function optimization problem as well as in many other types of optimization problem. Although certain optimization problems appear to be independently overcome, they are actually all tied in constraints. For example, in the 0/1 knapsack problem, the decisions regarding which items are placed in the knapsack appear to be independent. However, when an item is chosen, that choice affects the remaining space in the knapsack, and therefore certain other items cannot be placed in the knapsack; that constitutes dependency. Similar to dependency, each particle in a entangled state cannot be described independently, meaning that the state of any entangled bit affects the states of the other entangled bits.

This paper uses entangled states to enhance the capability of QTS. This novel approach is referred to as the entanglement-enhanced quantum-inspired tabu search algorithm (Entanglement-QTS). Entanglement-QTS differs from other QIEAs in that its Q-bits have entangled states, that can express a high degree of correlation, rendering the relationships of the variable more intertwined. The novel idea of that Q-bit have entangled states offers three important contributions. Firstly, it allows variables to be described and considered collectively, and facilitates solving the problem of strong dependency. Secondly, it expedites the diffusion of Q-bits which have good performance to accelerate the optimization speed. Thirdly, it also affords some strength of diversification. Because it considers multiple variables and changes the values of them, it frees the algorithm to test more than one place. Furthermore, the quantum NOT gate is used to prevent the algorithm from becoming mired in local optima, and local search and entanglement local search can discover better solutions more quickly, thus enhancing the convergence properties and increasing the strength of intensification.

This paper uses various function optimization problems to test the searching ability of Entanglement-QTS algorithm, and these problems, including high-dependency problems that traditional methods are unable to suitably address. Because very few traditional methods consider interaction among variables, the idea of entanglement is a novel and effective means of managing strong-dependency problems. The results presented here indicate that the Entanglement-QTS algorithm outperforms the other methods. The Entanglement-QTS algorithm retains the advantages of QTS (namely a balance between exploration and exploitation), offers rapid searching that is both efficient and effective, applies entangled Q-bits to address the dependency problem, uses a combination of normal and entanglement local searches to discover more quickly, and uses the quantum NOT gate to avoid becoming trapped in local optima.

The rest of this paper is organized as follows: Section II offers an overview of evolutionary computation. Sections III and IV introduce quantum computing and our basic idea of quantum entanglement, respectively. Section V details our algorithm. Performance evaluation is discussed in Section VI. Finally, Section VII presents the conclusion.

II. RELATED WORK

Many metaheuristic algorithms have been proposed, most of which use function optimization problems to test their performance. Metaheuristic algorithms do not guarantee the discovery of the global optimum every time, but are efficient and effective methods for solving complicated engineering problems; they often provide solutions within a reasonable time frame.

Different types of metaheuristic algorithm have been proposed with promising performance, such as GA [1], [22], [23], differential evolution (DE) [25]–[30], PSO [31]–[39] artificial immune system (AIS) [45], [46], artificial bee colony (ABC) algorithm [40]–[44], harmony search (HS) [47]–[49], and estimation of distribution algorithms (EDAs). GA uses selection, crossover and mutation to enable genes to improve through multiple generations of evolution. However, the search ability of GA is not powerful enough, and it exhibits difficulty in solving complex problems. In addition, as the number of generations increases, each group of genes becomes similar in appearance; which causes premature convergence. DE uses a simple cycle of stages that include mutation, recombination and selection similar to the methods used by GA. DE [25] uses the difference vector between two individuals and scales the vector to generate the next generation of individuals. To enhance its performance, several variants of DE have been proposed [26]–[30]. However, DE is limited in that it easily becomes trapped by local optima, and the speed of convergence varies. First proposed by Kennedy and Eberhart, PSO simulates the social behavior of bird flocking or fishing schooling. PSO uses three types of force to avoid premature convergence. It has robust search ability, is simple to implement, and is computationally fast. Many different methods have been designed to improve the performance of PSO and analyze its search ability [31]–[39]. However, PSO has a few drawbacks, particularly its lack of a mechanism to prevent it from becoming trapped at local optima. The ABC [40]–[44] algorithm, based on the intelligent behavior of honey bee swarms, is popular and has been discussed in many recent studies. It uses three groups of bees (employed bees, onlookers, and scouts) to improve the process of locating a food source. The ABC algorithm avoids becoming trapped at local optima and can locate optimal solutions efficiently. However, bottlenecks are experienced when ABC is applied to certain multimodal functions; in addition, ABC lacks a powerful exploratory capacity. The AIS [45], [46] was inspired by the immune system, in which a swarm of cells and molecules protects the body against diseases. AIS has four common techniques include negative selection, clonal selection theory, artificial immune network and clonal selection algorithm. The solution quality of AIS is high because AIS has the memory to retain good solutions and then find a better one. However, AIS is complicated, and computationally expensive. HS [47]–[49] is a derivative-free real parameter optimization algorithm that was inspired by a search for a perfect state of harmony. It exhibits powerful exploratory ability, but its exploiting ability is less favorable. EDAs use different probabilistic models by estimated distributions of individuals of previous generations to generate new promising individuals. EDAs [4], [50]–[52] shows great competitive performance on optimization problems with relatively low computational cost. However, EDAs suffer certain difficulties when presented with multimodal problems.

Recently, QIEAs have been proposed to prove their search ability have good balance between exploration and exploitation. QIEAs can be regard as a type of EDA [4], [50], [51]. First proposed in [3], QEA [3], [53], [54] can explore a search space with fewer individuals and exploit the global solution efficiently. The key elements of QEA are Q-bits and Q-gates, which address the balance between exploration and exploitation. Because the design of Q-gates remains unstandardized, substantial development opportunities exist. Many studies (QIEAs) have combined QEA with other heuristic methods such as PSO, immune clonal algorithm, gravitational search, cuckoo search algorithm, electromagnetism-like mechanism [60] and tabu search, to develop QPSO [9], [10], QICA, quantum-inspired gravitational search algorithm (QGSA), quantum-inspired cuckoo search algorithm (QICSA) [56] or cuckoo search based on quantum mechanism [57], quantum-inspired electromagnetism-like mechanism (QEM) [58], [59] and QTS [14], all of which perform strongly. These approaches (QIEAs) combine the features of QEA with the advantages of other algorithms. However, QIEAs still have certain drawbacks. Many iterations are required to fine-tune the angle at which to drive the individual toward a better solution, and extricating the algorithm from local optima can be difficult. As discussed previously, each metaheuristic algorithm has its own strengths and weakness. Compared with other metaheuristic algorithms, QIEAs have many attractive features, including stronger search abilities, lower computational costs, and easier implementation.

The QTS is a novel, simple and powerful metaheuristic method. Unlike other QIEAs, QTS uses both the best and worst solutions to drive the individual not only toward a better solution but also far away from a worse solution. QTS is effective and efficient, and often can rapidly discover the global optimum. Many applications of the QTS algorithm [14]–[21] have been proposed to demonstrate its promising and incomparable search ability. Hence, QTS is used as a base algorithm in this study.

Despite the robust effectiveness of QTS, it still possesses limits. When QTS or other metaheuristic methods are applied to high-dependency problems, they experience difficulty in finding the global optimum. This paper proposes the Entanglement-QTS method, which uses quantum entanglement to enhance the ability of QTS to solve problems that have high dependency among each variable. A review of the literature has indicated that no substantial research has considered the dependency of each variable as this study does. The entangled Q-bit is a state-of-the-art idea that could significantly improve the treatment of multimodal and high-dependency problems. Moreover, a quantum NOT gate is used to help the algorithm jump out of the local optima, and local search and entanglement local search can intensify the search ability.

III. QUANTUM COMPUTING

Quantum computing is an emerging research field. Quantum computers were first proposed in the 1980s and have since received abundant research attention. Quantum computers provide powerful abilities to solve optimization problems.

Quantum computing has two primary principles: superposition and entanglement [7]. In traditional computers, each bit can be in only one state at a time, either 0 or 1. A qubit, which is the smallest unit of information in quantum computing, can be in a state of superposition of the 0 and 1 states. A qubit state $|\psi\rangle$ is a combination of two basis vectors as shown in Eq. 1, where α and β are complex numbers.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

$|0\rangle$ and $|1\rangle$ can be denoted as two column vectors (Eq. 2).

$$|0\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{2}$$

Because of the linear superposition of individual possible states, $|\alpha|^2$ and $|\beta|^2$ represent the probabilities that the qubit is located in the 0 and 1 state, respectively. The sum of $|\alpha|^2$ and $|\beta|^2$ follows the rule of probability, and can be represented as shown in Eq. 3.

$$|\alpha|^2 + |\beta|^2 = 1 \tag{3}$$

A qubit can be represented as a pair of numbers (Eq. 4).

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{4}$$

If there are n qubits in a quantum system, all qubits can be represented in the following form (Eq. 5).

$$\left[\begin{array}{c|c|c|c|c} \alpha_1 & \alpha_2 & \alpha_3 & \ldots & \alpha_n \\ \beta_1 & \beta_2 & \beta_3 & \ldots & \beta_n \end{array}\right] \tag{5}$$

According to Eq. 3, for each $|\alpha_i|^2 + |\beta_i|^2 = 1, i = 1, 2 \ldots n$.

All entangled states are special cases of superposed states. In an entangled state means, the measuring of one qubit also determines the states of the qubits with which it is entangled at the same time, regardless of the distance between the qubits. Common examples of entangled states can be denoted as Eq. 6.

$$\begin{aligned} |\phi_{00}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ |\phi_{01}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\ |\phi_{10}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\ |\phi_{11}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \end{aligned} \tag{6}$$

For example, $|\phi_{00}\rangle$ in Eq. 6 signifies that there is a 50% chance exists that this qubit can be measured in the $|00\rangle$ or $|11\rangle$ state. When the first qubit is measured and the result is $|0\rangle$, and the state of second qubit is determined as $|0\rangle$, the result of this observation is $|00\rangle$. If the second qubit is in state $|1\rangle$ and the first state of that qubit is determined to be $|1\rangle$, then the result of this observation is $|11\rangle$. These two entangled qubits possess an unseen bond. The principle of quantum entanglement is that the quantum state of each particle cannot be described independently of the others, even when the particles are far apart.

IV. BASIC IDEA: ENTANGLEMENT STATE

This section details with examples the concept of entangled states and how to bring it about; it also gives some examples in detail.

QTS has been applied to many NP-complete problems. When solving these problems, a remarkable phenomenon has been observed. Although the problems' variables may seem independent, they are often dependent on one another because the problem contains constraints by which all variables are bound. In this section, the 0/1 knapsack, deployment, and stock selection problems are used to illustrate this phenomenon. In the 0/1 knapsack problem, whether to choose an item appears to be an independent decision. However, because the carrying weight of the knapsack is limited, the decision to choose or ignore an item affects the ability to choose or ignore other items. When the choice to take a particular item is made, the current and available weight of the knapsack affects future decisions regarding item packing. In the deployment problem of wireless sensor networks, deploying a sensor in the topology affects the locations of sensors that are yet to be deployed. If the sensors are too far apart, the coverage rate may not be achieved, and if they are too close together, the deployment may be inefficient. Therefore, when the location of one sensor is decided, the potential locations of other sensors should maintain a certain distance from one another. In the stock selection problem, a limited amount of money is available to purchase stocks; the decision to buy a particular stock affects the subsequent decisions. The aforementioned examples indicate that most variables in combinatorial problems exhibit dependency because when deciding on the value of one variable, affect decisions about the others.

TABLE 1. Settings of item 6 to 9 of 0/1 knapsack problem.

Item	6	7	8	9
Weight	6	7	8	9
Profit	11	12	13	14

The literature details how approaches to solve the dependency problem are determined. QTS performed very well when applied to the 0/1 knapsack problem [14]. In an experiment performed with 100 items, multiple optimal solutions existed. For example, if the remaining capacity of the knapsack was 15, two solutions could achieve the optimal solution. One was to pick the two items weighing 6 and 9, and the other solution was to choose items weighing 7 and 8. Both approaches achieved the objective, which was to meet the weight restriction while packing as much weight as possible. The profits of these two methods were also the same (i.e., 25). Table 1 illustrates the weight and profit of the items. This case illustrates dependency: if an item weighing 6 is chosen first, then an item weighing 9 must be chosen next. If items weighing 7 or 8 are chosen instead, the optimal solution cannot be achieved. However, if the item weighing 7 is chosen first, then the item weighing 8 must be chosen next. This phenomenon is similar to the concept of entanglement in quantum mechanics. Based on Eq. 6, this situation can be formulated in the following Eq. 7.

$$|\varphi\rangle = \frac{1}{\sqrt{2}}(|1001\rangle + |0110\rangle) \tag{7}$$

Here, $|\varphi\rangle$ indicates that if the first qubit is measured and the result is $|1\rangle$, then the next three qubits are $|001\rangle$. However, if the first qubit is discovered to be $|0\rangle$, then the next three qubits are $|110\rangle$. If the item weighing 6 is chosen, then the states of the remaining items are determined; the items weighing 7 and 8 are not chosen and the item weighing 9 is taken. If the item weighing 6 is not chosen, then the states of the remaining items are determined; the items weighing 7 and 8 are chosen whereas the item weighing 9 is not. As a result, states $|1001\rangle$ or $|0110\rangle$ are optimal solutions. Their neighbor solutions, $|1010\rangle$, $|0101\rangle \ldots$, are local optima.

Most NP-complete problems exhibit strong dependency because their variables are bound with an invisible line, namely their constraints. The capacity of the knapsack is limited in the 0/1 knapsack problem; when one item is placed in the knapsack, the choice of possible other items is affected because the knapsack has weight restrictions. In the deployment problem, the constraint involves using as few sensors as possible while still satisfying the coverage rate. The stock selection problem limits the investment amount; when one stock is chosen, the next investment decision is affected. Regarding function optimization, many functions exist, such as in Griewank and Rosenbrock; therefore, discovering an optimal solution when independently optimizing each variable becomes difficult. The Griewank function: $f(x) = 1 + \frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} cos(\frac{x_i}{\sqrt{i}})$, is designed to deal with the algorithm which optimize each variable independently. The Rosenbrock function: $f(x) = \sum_{i=1}^{d-1}(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$, it can obviously find each value of variable will influence other value of variables. Traditional approaches that optimize each variable independently fall easily into local optima and find it difficult to solve such problems well. Therefore, the idea of entanglement is a novel and effective means of managing strong dependency. Entanglement can consider multiple variables or multiple dimensions simultaneously. In contrast with traditional approaches, entanglement moves and changes variables collectively to better locations. In this paper, function optimization problems are used to evaluate the ability of the Entanglement-QTS algorithm to manage strong dependency, escape from local optima, and obtain optimal solutions.

V. THE ENTANGLEMENT-QTS ALGORITHM

This section details the proposed Entanglement-QTS algorithm, which is a powerful search algorithm that can locate the global optimum effectively and efficiently. It uses entangled states to solve high-dependency problems and accelerate the optimization. Also, it uses local search and entanglement local search to increase the strength of intensification as well as the quantum NOT gate to escape from local optima. Algorithm 1 presents the main procedure of the Entanglement-QTS algorithm for solving the function optimization problem. Parameter t is the generation number, and the detailed description of each step is given as follows.

In the encoding step, binary strings are used to represent real numbers to allow this encoding method to solve both numerical and combinatorial optimization problems. Computer is binary system, it cannot use binary bits to represent all real numbers. Hence, if a problem requires an extremely small value, then it can add more bits to represent this number, just as IEEE 754 does (e.g., float and double). Suppose six bits are used as the length of the binary string and there are two dimensions. In addition, the domain of the function is between -3 and 3. Table 2 provides an example of two binary strings to represent the real numbers 2.5 and -1.25.

1) Initialize Quantum Matrix $Q(t)$ (line 2 of the Entanglement-QTS algorithm): The quantum matrix is composed of $n * d$ Q-bits, and each Q-bit has the probability of existing in different states, as defined in Eq. 1. The

Algorithm 1 Entanglement-Enhanced Quantum-Inspired Tabu Search Algorithm

1: $t \leftarrow 0$
2: Initialize quantum population $Q(t)$
3: **while not** termination-condition **do**
4: $\quad t \leftarrow t + 1$
5: $\quad$ Make neighborhood set N by measuring of $Q(t-1)$
6: $\quad$ Repair $s \in N$ and evaluate $f(s)$
7: $\quad$ Detect whether algorithm stuck in a local optimum
8: $\quad$ **if** stuck **then**
9: $\qquad$ Use quantum NOT gate
10: $\quad$ **end if**
11: $\quad$ Entanglement
12: $\quad$ Local search
13: $\quad$ Entanglement local search
14: $\quad$ Select the best solution s^b and worst solution s^w among N
15: $\quad$ Update $Q(t)$ by s^b and s^w
16: **end while**

TABLE 2. Encoding real numbers 2.5 and −1.25 as binary strings.

sign	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	Value
0	1	0	1	0	0	2.5
1	0	1	0	1	0	-1.25

TABLE 3. Example of $|\beta|^2$ in $Q(0)$ for string length n=6 and dimension d=3, respectively.

sign	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5

matrix $Q(0)$ can be expressed as: $\begin{bmatrix} q_1^1 & q_1^2 & \cdots & q_1^n \\ q_2^1 & q_2^2 & \cdots & q_2^n \\ \vdots & \vdots & \ddots & \vdots \\ q_d^1 & q_d^2 & \cdots & q_d^n \end{bmatrix}$, where n signifies the length of the binary string that represents the value of the variable and d is the number of dimensions. Each q_l^k is initialized as $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$, which indicates that this Q-bit will have the same probability of collapsing into either a "0" or "1" state. Suppose $n = 6$ and $d = 3$; then the $|\beta|^2$ means the probability of the Q-bit is located in the 1 state and $|\beta|^2$ of $Q(0)$ are given in Table 3.

2) The remaining steps, involving lines 4-15 of the Entanglement-QTS algorithm, are executed cyclically until the termination condition is satisfied. The terminal condition in heuristic algorithms can be achieved in many ways. The Entanglement-QTS algorithm has two terminal conditions (line 3 of the Entanglement-QTS algorithm): when the optimal solution is achieved, or when run-time generation (g) achieves a set value.

TABLE 4. Example of random numbers which prepare to produce solution *s*.

sign	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
0.65	0.43	0.13	0.31	0.95	0.52
0.02	0.78	0.25	0.71	0.29	0.63
0.82	0.83	0.14	0.49	0.29	0.99

3) In this algorithm, a new generation of neighborhood solutions (line 5 of the Entanglement-QTS algorithm) is obtained by measuring the quantum matrix. This step is inspired by quantum computing: when measuring a quantum bit, the probability of $|\alpha|^2$ to be seen 0 and $|\beta|^2$ to be seen 1 but only one of its quantum states can be decided. In this step, measuring $Q(t-1)$ repeatedly m times will obtain a neighborhood set $N = \{s_1, s_2, \ldots, s_m\}$, which m is the number of population. Solution s can be produced by following Measure(s), Algorithm 2, and formed as a binary matrix, where $s = \begin{bmatrix} x_1^1 & x_1^2 & \ldots & x_1^n \\ x_2^1 & x_2^2 & \ldots & x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ x_d^1 & x_d^2 & \ldots & x_d^n \end{bmatrix}$.

Algorithm 2 Procedure Measure(s)

for $j = 1$ to d **do**
  for $k = 1$ to n **do**
    $R \in$ random number U[0,1]
    if $R < |\beta|^2$ **then**
      $x_j^k \leftarrow 1$
    else
      $x_j^k \leftarrow 0$
    end if
  end for
end for

In the beginning, this algorithm needs to prepare $n * d$ random numbers r_j^k with values between 0 and 1. Subsequently, a random number, r_j^k, is compared with the probability $|\beta_j^k|^2$ of each quantum matrix. $|\beta_j^k|^2$ refers to the probability to measure $|q_j^k|$ into $|1\rangle$ as Eq. 1 demonstrates. Therefore, if r_j^k less than $|\beta_j^k|^2$, set $x_j^k = 1$; otherwise set $x_j^k = 0$. Each x_j^k is either 1 or 0 to indicate which value of number is selected. For example, to produce s, $n*d$ $(6*3)$ random numbers must be prepared, as shown in Table 4, and compared with $|\beta_j^k|^2$ of each q_j^k, which all initialize with 0.5. If a random number is less than $|\beta_j^k|^2$, then x_j^k is set as 1; otherwise x_j^k is set as 0. The results of binary strings are given in Table 5.

4) In accordance with line 6 of the Entanglement-QTS algorithm, repair $s \in N$ to conform to a feasible domain and evaluate its fitness by using the objective function $f(s)$. The Repair(x) procedure, Algorithm 3, keeps each variable within the domain range. When the variable of x_d is greater than the positive domain or less than the negative domain,

TABLE 5. Example of a solution *s* obtained by measuring *Q*(0).

sign	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	value
0	1	1	1	0	0	3.5
1	0	1	0	1	0	-1.25
0	0	1	1	1	0	1.75

TABLE 6. Example of a solution after repairing.

sign	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	value
0	1	1	**0**	0	0	**3**
1	0	1	0	1	0	-1.25
0	0	1	1	1	0	1.75

it should be repaired within the domain. When x_d is out of domain, x_d^j in x_d should be discarded randomly until x_d is within the domain range. The first number cannot be chosen, as it presents the sign of the number. Each variable should be guaranteed within the domain. The fitness value is calculated using each benchmark function. For example, if the domain of the function is between -3 and 3, the solution (Table 5) produced in the previous step must be repaired because the binary string in dimension 1 represents the value 3.5, which is greater than 3. Therefore, a bit should be chosen randomly in dimension 1. If the x_1^4 is chosen, then x_1^4 is set to 0 and its value is recalculated. After repair, the value in dimension 1 is 3, which is within the domain (Table 6). This step is repeated until all dimensions are within their respective domains. Before evaluating the fitness function, the Entanglement-QTS procedure must run Procedure Repair(x_d) to confirm that each solution is feasible.

Algorithm 3 Procedure Repair(x_d)

$x_d = x_d^1 + x_d^2 + \cdots + x_d^n$
set the *out* − *domain* false
if $x_d > pdomain$ or $x_d < ndomain$ **then**
  set the *out* − *domain* true
end if
while *out* − *domain* **do**
  select random $j, j \neq 1$ and $x_d^j = 1$
  $x_d \leftarrow x_d - x_d^j$
  $x_d^j \leftarrow 0$
  $x_d = x_d^1 + x_d^2 + \cdots + x_d^n$
  if $x_d < pdomain$ and $x_d > ndomain$ **then**
    set the *out* − *domain* false
  end if
end while

5) Detect whether the algorithm is stuck at a local optimum (line 7 of Entanglement-QTS algorithm). Algorithms become stuck at local optima when the best solution is not updated within a certain number (η) of generations. If the algorithm falls into a local optimum, each dimension

TABLE 7. Example of $|\beta|^2$ in dimension 1 of $Q(t)$. Suppose quantum NOT gate is applies to q_1^3 and q_1^5.

sign	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
0.2	0.4	0.9	0.6	0.3	0.8
		X		X	

TABLE 8. Example of $|\beta|^2$ after using quantum NOT gate in q_1^3 and q_1^5 in dimension 1 of $Q(t+1)$.

sign	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
0.2	0.4	**0.1**	0.6	**0.7**	0.8

TABLE 9. Probability of using quantum NOT gate (δ) in each Q-bit.

	sign	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
q_d	q_d^1	q_d^2	q_d^3	q_d^4	q_d^5	q_d^6
δ	0.05	0.10	0.15	0.20	0.25	0.30

has probability δ_1 of applying quantum NOT gate X to escape from this state. A quantum NOT gate is defined as $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. Eq. 8 details the process of applying a quantum NOT gate to a Q-bit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$.

$$X|\psi\rangle = X(\alpha|0\rangle + \beta|1\rangle) = \alpha|1\rangle + \beta|0\rangle = \begin{bmatrix} \beta \\ \alpha \end{bmatrix} \quad (8)$$

When a quantum NOT gate is applied to a Q-bit, the probabilities of choosing 0 and 1 are exchanged. Table 7 provides an example by showing the values of $|\beta|^2$ in dimension 1 of $Q(t)$ before using a quantum NOT gate, whereas Table 8 displays $|\beta|^2$ values after using quantum NOT gate in q_1^3 and q_1^5. Therefore, the $|\beta|^2$ values of q_1^3 and q_1^5 in dimension 1 of $Q(t)$ are 0.9 and 0.3, respectively. In the next generation $Q(t+1)$, q_1^3 and q_1^5 become 0.1 and 0.7, respectively. Hence, using a quantum NOT gate changes the quantum matrix $Q(t)$, thus allowing population to escape from the current state to increase the probability of the algorithm jumping from a local optimum.

Note that the encoding method used in this study alters the bits on the left-hand side, and the value changes a lot. Thus, the bits on the left-hand side are called the most significant bits (MSBs), whereas those on the right-hand side are designated as the least significant bits (LSBs). The probability of using the quantum NOT gate decreases linearly. An example is given in Table 9. If there are five Q-bits in a dimension, the probability of using the quantum NOT gate 0.30 in the LSBs 0.05 (δ_2) in the MSBs, which means that the LSBs are more likely to change state than the MSBs are. Because the MSBs represent high real values, the strength of intensification is reduced when MSBs changing states frequently, resulting in increased difficulty in discovering the global optimum.

6) Entanglement is the primary contribution of this paper (line 11 of the Entanglement-QTS algorithm). The purpose of the Entanglement(x) (Algorithm 4) procedure is to solve strong-dependency problems. When variables in a function are addressed independently, obtaining the global optimum becomes more difficult. Thus, entangled state $|\phi_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ described in section III is used to address a problem dependently. The entanglement step not only entangles Q-bits in different dimensions, but also entangles Q-bits in different orders. Each method results in a different effect. The dependency of dimensions means that it needs to consider multiple values of dimension because changing the value in one dimension may affect other dimension values. The dependency of orders means that the Q-bits in the same order but in different dimensions may need to refer to each other's values. In this study, both dimensions and orders of Q-bits are entangled, hence we can consider the relationship between variables comprehensively. Firstly, when this algorithm runs the entanglement step, it selects an entanglement method. If it chooses to entangle the dimensions, then d_1 and d_2, which are selected randomly, are entangled; when a Q-bit is measured in d_1 and the result x_{d1}^j is 1, then x_{d2}^j is set as 1. If the value of x_{d2}^j is determined to be 0, then x_{d1}^j is 0. The principle is that when a Q-bit is measured in dimension d_2, the other Q-bit in dimension d_1 (with which the first Q-bit is entangled) possesses the same value. If the algorithm chooses to entangle the orders, then the n_1th and n_2th order of Q-bits are entangled. Therefore, when the Q-bits in n_2th order are determined, so are the Q-bits in n_1th order.

This method is elaborated in Entanglement(x), Algorithm 4. Each dimension / order has probability (ρ) of entangling with other dimensions / orders. Firstly, any two dimensions d_i and d_j are chosen and entangled. When the Q-bits in d_j are measured to produce x_j, then x_i of the other dimension d_i is the same as x_j. However, when the algorithm entangles orders, it randomly choose the two orders and entangle them; regardless of which dimensions these Q-bits are in, they are described together. Examples of entangling dimensions and orders are given in Table. 10, 11 and 12, 13. After entanglement, s' is obtained. In evaluating entangled state s', it is removed if it did not outperform s. If the entanglement solution (s') did surpasses outperform s, then s is replaced with s'. The entanglement condition is such that when this solution performs the entanglement, a better performance is yielded and it repeats this method until it cannot find a better solution for several iterations.

Adding an entanglement step improves the QTS algorithm substantially in relation to solution quality and computational cost. The entangled state $|\phi_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ makes three important contributions. Firstly, it describes and considers variables together and can solve the problem of strong dependency very well. Secondly, it can promote the diffusion of Q-bits which have good performance to accelerate the optimization speed. Thirdly, it affords some escape capacity.

TABLE 10. Step 1 of entangling dimensions. Dimensions 1 and 2 are chosen to be entangled.

sign	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	value	
0	1	1	0	0	0	3	←
1	0	1	0	1	0	-1.25	←
0	0	1	1	1	0	1.75	

TABLE 11. Step 2 of entangling dimensions. If a bit state is obtained in dimension 2, the same state will be obtained in dimension 1.

sign	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	value
1	0	1	0	1	0	-1.25
1	0	1	0	1	0	-1.25
0	0	1	1	1	0	1.75

TABLE 12. Step 1 of entangling orders. The 4th and 5th orders are chosen to be entangled.

				↓	↓	
sign	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	value
0	1	1	0	0	0	3
1	0	1	0	1	0	-1.25
0	0	1	1	1	0	1.75

Because it moves multiple Q-bits by a considerable amount at once, it frees the algorithm to test beyond the local area. Not only can the entanglement procedure solve problems with dependency, it can also possess both capabilities of exploration and exploitation.

Algorithm 4 Producer Entanglement(s)

$s = [x_1, x_1, \ldots, x_d]$
$s' \leftarrow s$
while entanglement-condition **do**
  if dimension entanglement **then**
    select random i and j dimensions from s' where $j \neq i$
    $x_i \leftarrow x_j$
  else
    select random i and j orders of Q-bits from s' where $j \neq i$
    $x^i \leftarrow x^j$
  end if
end while
/* In this algorithm, it minimizes the function value.*/
if $f(s') < f(s)$ **then**
  $s \leftarrow s'$
end if

7) The purpose of the local search procedure (line 12 of Entanglement-QTS algorithm) is to search the regions adjacent to the solution to find a better location that gives a better fitness value; the principle is similar to the hill climbing algorithm. The Hamming distance can be used to calculate distance in the binary string. For example, the algorithm selects one bit and changes its state, meaning that the local area is defined as one Hamming distance. If the new state possesses more fitness than that of the old state, the old state is discarded and the new one is adopted. The aforementioned procedure is repeated until the local areas of all bits have been checked; this method is elaborated in Local Search(s), Algorithm 5. To reduce the computational cost in this paper, note that only the best solution uses the procedure of local search.

TABLE 13. Step 2 of entangling orders. The states of these two orders of bits will be the same.

sign	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	value
0	1	1	0	0	0	3
1	0	1	0	0	0	-1
0	0	1	1	0	0	1.5

The local search of the proposed algorithm is a valuable step because better solutions can be discovered more quickly, thus enhancing the convergence properties, increasing the strength of intensification, and also complementing the entanglement step.

Algorithm 5 Procedure Local Search(s)

/* Using Hamming distance to define neighbor */
$s = x_1, x_1, \ldots, x_d$
$s' \leftarrow s$
for $j = 1$ to d **do**
  for $k = 1$ to n **do**
    if $x_j^k = 1$ **then**
      $x_j^k \leftarrow 0$
    else
      $x_j^k \leftarrow 1$
    end if
    if $f(s') < f(s)$ **then**
      $s \leftarrow s'$
    end if
  end for
end for

8) This entanglement local search procedure (line 13 of Entanglement-QTS algorithm) uses entanglement states such as $\frac{1}{\sqrt{2}}(|011\rangle+|100\rangle)$, $\frac{1}{\sqrt{2}}(|0111\rangle+|1000\rangle)$, and $\frac{1}{\sqrt{2}}(|01111\rangle+|10000\rangle)$, which are different from the entangled states of Algorithm 4 already mentioned. This procedure reverses multiple bits simultaneously. Although NOT gate can reverse bits, it just considers and reverses one bit at one time. Not only can an entanglement local search consider multiple bits and change them collectively, it can also be a local search. In the encoding step of the proposed method, in which each binary string represents a real value, something interesting occurs. For example, when there are six bits and the domain is -3 to $+3$, string [001111] represents the value 1.875, and

string [010000] represents the value 2, even though the Hamming distance between these two strings is 5 and their real values are similar. Therefore, this paper proposes the Entanglement Local Search Algorithm as given in Algorithm 6. This uses the distance between the two real values, meaning that the distance represented by the right-most bit ([000001]), which is 0.125 in this example, is defined as the local area.

Entanglement local search differs from the traditional local search in that not only can the former facilitate a local search, it also affords a way to jump from a local optimum. In relation to real values, it is a way to enhance the intensification search. In relation to binary strings, it is a powerful way to jump from local optima, thereby strengthening the ability to diversity. Entanglement local search combines the strength of exploration and exploitation very well. To reduce the computational cost in this paper, only the best solution uses the entanglement local search. The entanglement step provides a significant improvement in solution quality. The local search and entanglement local search step can then reinforce the intensification effect. Therefore, using Local Search and Entanglement Local Search can effectively reduce the computational cost because they both accelerate the search for the global optimum.

Algorithm 6 Procedure Entanglement Local Search(s)

/* Using real value to define neighbor */
$s = [x_1, x_1, \ldots, x_d]$
$s' \leftarrow s$
for $j = 1$ to d **do**
  /* Using Two's complement */
  $R \in$ random number U[0,1]
  if $R < 0.5$ **then**
    $x_j \leftarrow x_j + x_j^n$
  else
    $x_j \leftarrow x_j - x_j^n$
  end if
  if $f(s') < f(s)$ **then**
    $s \leftarrow s'$
  end if
end for

9) Of all the neighborhood solutions, choose the solution $s \in N$ that stores the best fitness value in s^b, and the worst fitness value in s^w (line 14 of the Entanglement-QTS algorithm). In the next step, s^b and s^w are used to update the quantum matrix.

10) The final step is to update the quantum matrix (line 15 of Entanglement-QTS algorithm), which gives rise to the next generation of solutions moving toward the best solution while avoiding the worst solution. The updating step thus increases the probability of choosing the best solution and decreases the probability of choosing the worst solution. When a Q-bit $q_k \in Q$ changes its original state, its new state corresponds to Table 14. If s^b and s^w are the same, this situation is tabu; $q_k \in Q$ is not rotated. If s^b and s^w differ, q_k applies

TABLE 14. Move-gate lookup table.

s^b	s^w	$q_k \in Q$	$\triangle\theta$
0	0	Tabu	0
0	1	false	$+\theta$ in first or third quadrant
			$-\theta$ in second or fourth quadrant
1	0	false	$-\theta$ in first or third quadrant
			$+\theta$ in second or fourth quadrant
1	1	Tabu	0

* $k = 1, 2, \ldots n$ is the number of qubits.

TABLE 15. Best solution s^b.

sign	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	value
0	**0**	**0**	0	0	0	0
1	0	1	0	**1**	**1**	-1.375
1	0	0	1	0	0	-0.5

TABLE 16. Worst solution s^w.

sign	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	value
0	**1**	**1**	0	0	0	3
1	0	1	0	**0**	**0**	-1
1	0	0	1	0	0	-0.5

TABLE 17. Updated $|\beta|^2$ of quantum matrix.

sign	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
0.5	**0.4**	**0.4**	0.5	0.5	0.5
0.5	0.5	0.5	0.5	**0.6**	**0.6**
0.5	0.5	0.5	0.5	0.5	0.5

the move-gate (Q-gate) $U(\triangle\theta) = \begin{bmatrix} cos(\triangle\theta) & -sin(\triangle\theta) \\ sin(\triangle\theta) & cos(\triangle\theta) \end{bmatrix}$ to rotate to a new state, which may acquire improved neighborhood solutions by measuring q in the next iteration. The $\triangle\theta$ are designed in compliance with the application problem.

On the implement side, the quantum matrix is updated by adding or subtracting the value (μ) of $|\beta|^2$ to increase or decrease the probability of choosing the best or the worst solution. Tables 15 and 16 illustrate the best and worst solutions stored in s^b and s^w, respectively. For example, if the value of x_2^5 and x_2^6 in the best solution are 1 and in the worst solution are 0, then in the next generation an increased probability of obtaining 1 rather than 0 at x_2^5 and x_2^6 exists. The set value is therefore added to q_2^5 and q_2^6. Following Algorithm 2, if the value of $|\beta|^2$ is higher, the chance of obtaining 1 increases. However, if a greater probability of obtaining 0 than 1 is desired, such as x_1^2 and x_1^3, in the next generation, the value of q_1^2 and q_1^3 decrease. Table 17 displays the quantum matrix after it has been updated.

VI. EXPERIMENTAL RESULTS

In this section, the performance of the Entanglement-QTS algorithm is evaluated. For these experimental results, this paper uses the number of calculating fitness functions as the comparison criterion. The population and generation numbers are given to inform the reader about the algorithmic parameters, but they cannot represent all function evaluations because many algorithms may have a local search mechanism. Hence, only the number of function evaluations can reveal the total computational cost. In addition, it compares the level of function value, optimization algorithms are designed to find the minimum value of these functions. Hence, in this field, it compares the effectiveness and efficiency of algorithms that which one can use fewer function evaluations and find the minimum value of functions. In other words, it compares which algorithm can incur the lowest computational cost in finding the best solution in a huge solution space. The parameter set is presented in section VI-A. To evaluate the search abilities of the Entanglement-QTS algorithm, nine functions with different effects—include unimodal functions, multimodal functions, and functions with dependency of two variables—were evaluated to test the proposed optimization algorithms. A detailed description and the results of the Entanglement-QTS algorithm are presented in section VI-B. The Entanglement-QTS algorithm is compared in section VI-C with other powerful metaheuristic methods [9], [11], [19], [35], [40], [45], [47], [53] to demonstrate its effectiveness and efficiency, and section VI-D details a performance analysis. The proposed algorithm was implemented in C++.

A. CONDITION OF ENTANGLEMENT-QTS

Table 18 displays the parameter setting of the Entanglement-QTS algorithm. The Q-bits for all test functions were set to 32 bits (per variable). The population size was set to 30, and the terminal conditions were set to the number of generations reaching 500 or the optimal solutions being located. During the entanglement step, each dimension or each bit location had a probability of 0.3 to entangle with other dimensions or locations. After entanglement, if the performance of the new solution was better than that of the old one, the entanglement step was repeated until no improvement was observed over five consecutive iterations. During the local search step, only the best solution discovered other neighbor solutions, and if the neighbor solution was better, then the original one was replaced. This procedure continued until each bit was checked. During the updating step, the value of $|\beta|^2$ was set to 0.3. Here, the algorithm became trapped at local optima, meaning that the best solution had not been updated within 5 consecutive generations. The algorithm applied the quantum NOT gate to escape from local optima. The probability of a solution for each dimension was 0.2, and the MSBs had a probability of 0.05 to use the quantum NOT gate (δ); the probability increased linearly by 0.05 each bit until the LSB was reached. Each result used an average of 100 independent runs. Sections VI-B and VI-C present the experimental results.

TABLE 18. Parameter settings of Entanglement-QTS.

Q-bits	32
Population size (m)	30
Number of generation (g)	500
Range of updated probability (μ)	0.3
Local size	1
Stuck condition (η)	5
The probability of entanglement (ρ)	0.3
δ_1 in dimension	0.2
δ_2 in MSB	0.05

B. BENCHMARK AND PERFORMANCE OF ENTANGLEMENT-QTS

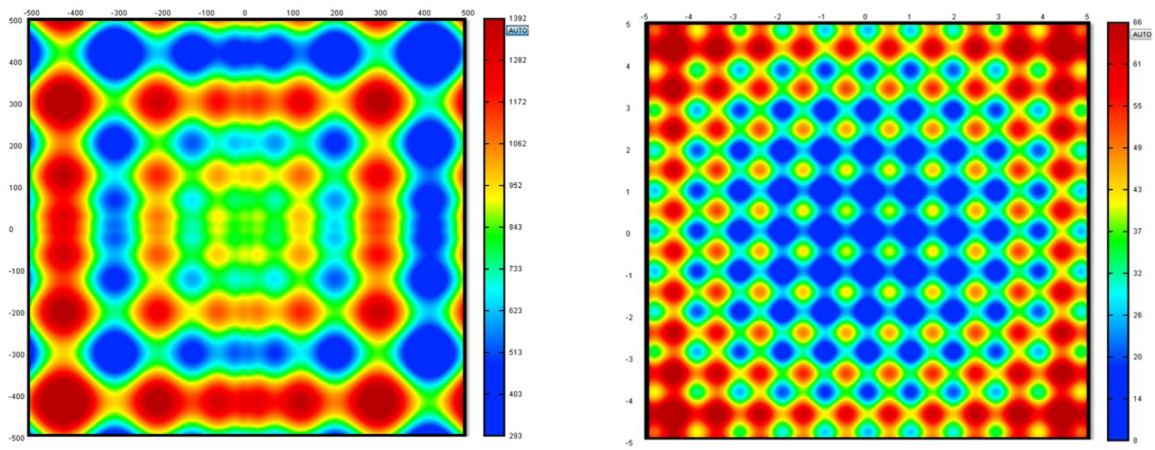
Each function has different properties, such as unimodal, multimodal, global optimum near the bounds, global optimum not on the bounds, and interdependence among the variables, to evaluate the optimization algorithms. Hence, nine functions were prepared (Table 19) to test whether the proposed algorithm can not only solve problems with dependent variables but also other problems with different properties. The dimensions of each function were set to 30, 40, and 50. The f_1, f_7, and f_8 functions were unimodal, meaning that no local optima existed; these functions tested the intensification ability of each algorithm. The f_2, f_3, f_4, f_5, and f_6 functions were multimodal, meaning that they contained more than two local optima. The global optimum in f_6 was near the bounds of the domain. These functions tested the algorithm's ability to diversify and escape from local optima. Moreover, the variable in f_3, f_5 and f_9 exhibited strong dependency, meaning that discovering the global optimum would be difficult for an algorithm that optimized each variable independently. High-dimensional, strongly dependent, and multimodal problems are also more difficult to solve well.

Table 20 represents the performance of Entanglement-QTS algorithm in three dimensions regarding the nine test functions. Table 20 provides the following data to present multiple properties of the proposed algorithm : 1) the best fitness in 100 runs [column (b)]; 2) the average fitness value of the best solution and the standard deviation of the best fitness [column (m)]; 3) the mean number of function evaluations and the standard deviation of evaluations [column (e)]; Columns (b) and (m) indicate that a high-quality solution can be discovered for multiple types of function; the Entanglement-QTS algorithm performed strongly. Column (m) demonstrates that the Entanglement-QTS algorithm is stable and can locate the global optimum most of the time. As seen in column (e), the proposed algorithm used relatively few evaluations to determine the global optimum, indicating that the entanglement and local search steps accelerated the optimal speed efficiently and effectively.

The proposed algorithm has powerful and stable search ability. In the f_6 problem, the global optimum is near

TABLE 19. Classic benchmark functions.

f	Function Name	Function	Global Optimum	Limit
f_1	ABS	$f(x)=\sum_{i=1}^{d}\|x_i\|$	0 at x=(0,0,...0)	[-100,100]
f_2	Ackley	$f(x)=-20e^{-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}}-e^{\frac{1}{d}\sum_{i=1}^{d}cos(2\pi x_i)}+20+e$	0 at x=(0,0,...0)	[-32,32]
f_3	Griewank	$f(x)=1+\frac{1}{4000}\sum_{i=1}^{d}x_i^2-\prod_{i=1}^{d}cos(\frac{x_i}{\sqrt{i}})$	0 at x=(0,0,...0)	[-600,600]
f_4	Rastrigin	$f(x)=10d+\sum_{i=1}^{d}(x_i^2-10cos(2\pi x_i))$	0 at x=(0,0,...,0)	[-5.12,5.12]
f_5	Rosenbrock	$f(x)=\sum_{i=1}^{d-1}(100(x_{i+1}-x_i^2)^2+(x_i-1)^2)$	0 at x=(1,1,...,1)	[-30,30]
f_6	Schwefel	$f(x)=d\times 418.9828873+\sum_{i=1}^{d}-x_i sin(\sqrt{\|x_i\|})$	0 at x=(420.9687,420.9687,...,420.9687)	[-500,500]
f_7	Sphere	$f(x)=\sum_{i=1}^{d}x_i^2$	0 at x=(0,0,...0)	[-100,100]
f_8	Jason	$f(x)=\sum_{i=1}^{d}(x_i-i)^2$	0 at x=(1,2,..,d)	[-100,100]
f_9	Quadric	$f(x)=\sum_{i=1}^{d}(\sum_{j=1}^{i}x_j)^2$	0 at x=(0,0,...0)	[-100,100]



(a) (b)

FIGURE 1. Color bars indicate fitness value: best fitness (dark blue) to worst fitness (dark red). (a) The contour plot of function *f* 6. (b) The contour plot of function *f* 4.

the bounds of the domain, as shown in the contour plot in Fig. 1(a). The Entanglement-QTS algorithm demonstrates its diversification; it does not become trapped in a second minimum. In f_1, f_7, and f_8, the Entanglement-QTS shows its intensification; it can always locate the lowest point in these continuous functions. The Entanglement-QTS algorithm can escape the local optima easily, as seen in f_2 and f_4 shown as Fig. 1(b), that contain numerous local optima. The proposed algorithm can also solve problems with strong dependency in each variable, as in f_3, f_5 and f_9. The Entanglement-QTS can balance diversification and intensification, escaping from local optima and managing the strong-dependency problem.

C. COMPARISON

This section firstly presents information on the performance of the proposed Entanglement-QTS algorithm. The proposed algorithm is also compared with QIEAs (QEA [53], QICA [11], QPSO [9], and QTS [19]), and other powerful swarm algorithms and evolutionary algorithms [ABC [40], Vaccine-AIS [45], Harmony search (HS) [47] and Bare bones PSO (BBPSO) [35])], all of which are major optimization methods in evolutionary computation. The results of compared algorithms are referred from the papers just cited, and the values of parameters such as population and generation are given in the captions of the result tables. Brief descriptions of these eight algorithms are given here.

TABLE 20. Performance of Entanglement-QTS.

F	Function Name	D	Best solution value (b.)	Mean function value (std.) (m.)	Evaluation (std.) (e.)
		30	0.00E+00	0.00E+00 (0.00E+00)	8.3407E+02 (5.63E+02)
f_1	ABS	40	0.00E+00	0.00E+00 (0.00E+00)	1.1253E+03 (7.20E+02)
		50	0.00E+00	0.00E+00 (0.00E+00)	1.2980E+03 (8.54E+03)
		30	0.00E+00	0.00E+00 (0.00E+00)	3.4214E+02 (3.59E+02)
f_2	Ackley	40	0.00E+00	0.00E+00 (0.00E+00)	3.9998E+02 (4.92E+02)
		50	0.00E+00	0.00E+00 (0.00E+00)	8.6975E+02 (9.04E+02)
		30	0.00E+00	0.00E+00 (0.00E+00)	1.0058E+03 (9.15E+02)
f_3	Griewank	40	0.00E+00	0.00E+00 (0.00E+00)	1.7947E+03 (1.38E+02)
		50	0.00E+00	0.00E+00 (0.00E+00)	2.1698E+03 (1.93E+03)
		30	0.00E+00	0.00E+00 (0.00E+00)	1.3364E+03 (1.24E+03)
f_4	Rastrigin	40	0.00E+00	0.00E+00 (0.00E+00)	1.8363E+03 (1.87E+03)
		50	0.00E+00	0.00E+00 (0.00E+00)	3.4752E+03 (2.93E+03)
		30	0.00E+00	0.00E+00 (0.00E+00)	6.9292E+03 (5.12E+03)
f_5	*Rosenbrock	40	0.00E+00	0.00E+00 (0.00E+00)	8.0221E+03 (6.55E+03)
		50	0.00E+00	0.00E+00 (0.00E+00)	9.9109E+03 (9.91E+03)
		30	8.27E-07	8.27E-07 (2.18E-11)	1.0935E+04 (5.13E+03)
f_6	Schwefel	40	1.10E-06	1.10E-06 (1.22E-11)	3.0166E+05 (5.22E+03)
		50	1.38E-06	1.38E-06 (1.38E-11)	3.6934E+05 (4.30E+03)
		30	0.00E+00	0.00E+00 (0.00E+00)	1.0560E+03 (4.87E+02)
f_7	Sphere	40	0.00E+00	0.00E+00 (0.00E+00)	1.3281E+03 (6.50E+02)
		50	0.00E+00	0.00E+00 (0.00E+00)	1.7521E+03 (6.99E+02)
		30	0.00E+00	0.00E+00 (0.00E+00)	2.5752E+04 (4.85E+03)
f_8	Jason	40	0.00E+00	0.00E+00 (0.00E+00)	3.8548E+04 (6.57E+03)
		50	0.00E+00	0.00E+00 (0.00E+00)	5.3054E+04 (7.21E+03)
		30	0.00E+00	0.00E+00 (0.00E+00)	9.8812E+04 (1.97E+04)
f_9	Quadric	40	0.00E+00	0.00E+00 (0.00E+00)	1.6909E+05 (2.54E+04)
		50	0.00E+00	0.00E+00 (0.00E+00)	2.4926E+05 (4.34E+04)

* Rosenbrock is a strong dependent problem, so in this paper, it repeats the entanglement step until there is no improvement over 10 consecutive iterations.

TABLE 21. Comparisons between QEA and Entanglement-QTS. The results of QEA are referred from [53]. The population of QEA [53] is set to 100 and the termination is set to be 1500 generations for f_2 and f_7, 2000 generations for f_3, 5000 generations for f_4, 20000 generations for f_5 and 9000 generations for f_6. The dimensions of these functions are all set to 30.

F	D	Mean function value (Std.)			Evaluation		
		QEA w/H_ϵ	QEA w/R	Entanglement-QTS	QEA w/H_ϵ	QEA w/R	Entanglement-QTS
f_2	30	2.50E-03 (8.10E-04)	4.80E-04 (0.00E+00)	**0.00E+00 (0.00E+00)**	1.50E+05	1.50E+05	**3.4214E+02**
f_3	30	3.60E-02 (3.20E-02)	5.80E-02 (7.50E-02)	**0.00E+00 (0.00E+00)**	2.00E+05	2.00E+05	**1.0058E+03**
f_4	30	3.90E-02 (1.90E-01)	1.87E+01 (7.40E+00)	**0.00E+00 (0.00E+00)**	5.00E+05	5.00E+05	**1.3364E+03**
f_5	30	1.173E+01 (1.836E+01)	7.18E+00 (6.77E+00)	**0.00E+00 (0.00E+00)**	2.00E+06	2.00E+06	**6.9292E+03**
f_6	30	3.80E-04 (3.00E-09)	2.1604E+02 (1.638E+02)	**8.27E-07 (2.18E-11)**	9.00E+05	9.00E+05	**1.0935E+04**
f_7	30	1.80E-04 (1.30E-04)	4.30E-06 (0.00E+00)	**0.00E+00 (0.00E+00)**	1.50E+05	1.50E+05	**1.0560E+03**

1) QEA [53] is a searching algorithm based on the evolutionary algorithm and the principle of quantum computation. It can treat the balance between exploration and exploitation more easily than can a conventional genetic algorithm. Many QEA variants now exist. In [53], QEA uses two different operators which are rotation gate (QEA w/R) and H_ϵ gate (QEA w/H_ϵ) to converge. QEA in [53] is compared with classical evolutionary programming (CEP) and fast EP (FEP) [24], and the results show that the QEA is better than either CEP and FEP. The comparison between QEA and Entanglement-QTS is made in Table 21.

2) QICA [11] is an optimization algorithm based on the concept of quantum computing and a novel immune clonal algorithm. Quantum bits are used to represent the antibodies, and a quantum rotation gate is applied to accelerate convergence. A quantum not gate and quantum recombination are adopted to improve search efficiency. In [11], QICA was compared with the standard immune clonal

TABLE 22. Comparisons between SICA, QICA and Entanglement-QTS. The results of SICA and QICA are referred from [11]. The initial antibody population of QICA is set to 10, the clonal size is 30, and the mutation probability is set to 0.5. The number of Q-bits is set as the number of dimensions. The dimensions of these functions are all set to 30.

F	D	Mean function value (Std.)			Evaluation		
		SICA	QICA	Entanglement-QTS	SICA	QICA	Entanglement-QTS
f_2	30	1.78E-02 (2.12E-03)	4.44E-16 (0.00E+00)	**0.00E+00 (0.00E+00)**	2.12889E+05	1.3691E+04	**3.4214E+02**
f_3	30	1.81E-02 (2.61E-02)	**0.00E+00 (0.00E+00)**	**0.00E+00 (0.00E+00)**	2.23451E+05	1.3745E+04	**1.0058E+03**
f_4	30	4.51E-02 (1.71E-02)	**0.00E+00 (0.00E+00)**	**0.00E+00 (0.00E+00)**	2.23373E+05	1.3992E+04	**1.3364E+03**
f_5	30	1.01E+00 (5.11E-01)	**0.00E+00 (0.00E+00)**	**0.00E+00 (0.00E+00)**	3.02556E+05	1.2421E+04	**6.9292E+03**
f_7	30	1.97E-01 (3.41E-02)	**0.00E+00 (0.00E+00)**	**0.00E+00 (0.00E+00)**	1.08632E+05	1.2491E+04	**1.0560E+03**

TABLE 23. Comparisons between HPSO [32], QPSO [9] and Entanglement-QTS. The results of HPSO and QPSO were referred from [9]. The population sizes of HPSO and QPSO are both set to 80, and termination is set to approach the global optimum or a maximum number of iterations of 5,000. The dimensions of these functions are all set to 40.

F	Limit	D	Best solution (Mean function value)			Evaluation		
			HPSO	QPSO	Entanglement-QTS	HPSO	QPSO	Entanglement-QTS
f_3	[-600,600]	40	5.14E-09 (2.55E-03)	**0.00E+00** (3.22E-07)	0.00E+00 (**0.00E+00**)	4.00E+05	4.00E+05	**1.7947E+03**
f_4	[-5.12,5.12]	40	1.53E+01 (3.00E+01)	1.08E-04 (5.21E-01)	**0.00E+00 (0.00E+00)**	4.00E+05	4.00E+05	**1.8363E+03**
f_5	[-2.048,2.048]	40	2.35E-02 (2.90E+01)	2.72E-09 (2.19E-02)	**0.00E+00 (1.16E-16)**	4.00E+05	4.00E+05	**1.2519E+04**
f_7	[-100,100]	40	2.82E-10 (1.32E-06)	**0.00E+00** (1.09E-23)	**0.00E+00 (0.00E+00)**	4.00E+05	4.00E+05	**1.3281E+03**
f_8	[-100,100]	40	1.50E-10 (6.74E-03)	**0.00E+00** (2.08E-23)	**0.00E+00 (0.00E+00)**	4.00E+05	4.00E+05	**3.8548E+04**

TABLE 24. Comparisons between QTS, PSO and Entanglement-QTS. The results of PSO and QTS were referred from [31] and [19]. The population of PSO is 30 and its number of iterations is more than 1000. The population, iteration, and angle of QTS are referred from [19]. The dimensions of these functions are all set to 30.

F	Limit	D	Mean function value (Std.)			Evaluation		
			Zero Init PSO	QTS	Entanglement-QTS	Zero Init PSO	QTS	Entanglement-QTS
f_1	[-100,100]	30	3.53E-01 (2.87E+00)	9.32E+00 (1.30E+00)	**0.00E+00 (0.00E+00)**	>3.0E+04	7.50E+05	**8.3407E+02**
f_2	[-32.768,32.768]	30	2.49E+00 (1.35E+00)	2.73E+00 (3.54E-01)	**0.00E+00 (0.00E+00)**	>3.0E+04	2.00E+05	**6.0576E+02**
f_3	[-600,600]	30	3.72E-02 (5.26E-02)	4.02E-02 (1.31E-02)	**0.00E+00 (0.00E+00)**	>3.0E+04	2.50E+06	**1.0058E+03**
f_4	[-5.12,5.12]	30	6.66E+01 (1.71E+01)	4.10E+01 (3.02E+01)	**0.00E+00 (0.00E+00)**	>3.0E+04	1.50E+06	**1.3364E+03**
f_5	[-2.048,2.048]	30	2.65E+01 (1.53E+01)	1.13E+02 (1.06E+02)	**2.21E-16 (1.26E-15)**	>3.0E+04	1.00E+05	**1.5590E+04**
f_9	[-100,100]	30	9.04E+01 (8.70E+01)	2.48E+00 (6.31E+00)	**0.00E+00 (0.00E+00)**	>3.0E+04	1.00E+06	9.8812E+04

algorithm (SICA), FEP, DE, TS, and ant colony optimization (ACO), and showed better performance. The comparison between QICA and Entanglement-QTS is made in Table 22.

3) QPSO [9] has stronger search ability than does traditional PSO, because QPSO applies the principle of quantum computing. The PSO formula is used to update the Q-gate. This algorithm also uses two special implementations, namely self-adaptive probability selection and chaotic sequence mutation, to escape from local optima and solve the premature convergence problem. In [9], QPSO was compared with the GA, the immune algorithm, and four PSO variants such as HPSO [32], and QPSO showed the superior performance. The comparison between QPSO and Entanglement-QTS is made in Table 23.

4) QTS [14] is a powerful searching algorithm based on the Tabu search and the principle of quantum computation. QTS uses the best and worst solutions to simultaneously move individuals toward the best solution and away from the worst solution. QTS is quicker and more efficient than other heuristic algorithms. It can find the global optimum efficiently and effectively, and excels at solving traditional NP-complete problems. In [19], function optimization was used to test the performance of QTS in comparison with PSO and a modified evolutionary algorithm; QTS showed its great search ability. The comparison between QTS and Entanglement-QTS is made in Table 24.

5) ABC [40] is a global search algorithm based on the intelligent behavior of honey bee colony. This algorithm is a classic example of a swarm system. It contains three groups of bees—employed bees, onlookers, and scouts—each of which has its own ability to search the solution space. In [40], ABC was compared with GA, PSO, and particle swarm inspired evolutionary algorithm (PS-EA) that is a hybrid algorithm of EA and PSO [61]; the experimental results showed that ABC outperformed the three methods. The comparison between ABC and Entanglement-QTS is made in Table 25.

TABLE 25. Comparisons between ABC and Entanglement-QTS. The results of ABC are referred from [40]. In [40], the population of ABC1 and ABC2 were set to 125. The termination criterion of ABC1 is set to be 500 generations for 10 dimensions, 750 generations for 20 dimensions, and 1,000 generations for 30 dimensions. The termination criterion of ABC2 is set to be 1,000 generations for 10 dimensions, 1,500 generations for 20 dimensions, and 2,000 generations for 30 dimensions.

F	Limit	D	Mean function value (Std.)			Evaluation		
			ABC1	ABC2	Entanglement-QTS	ABC1	ABC2	Entanglement-QTS
		10	7.80E-11 (1.16E-09)	4.60E-11 (5.40E-11)	**0.00E+00 (0.00E+00)**	6.25E+04	1.25E+05	**2.4995E+02**
f_2	[-32.768,32.768]	20	1.60E-11 (1.90E-11)	**0.00E+00** (1.00E-12)	**0.00E+00 (0.00E+00)**	9.375E+04	1.875E+05	**3.0600E+02**
		30	3.00E-12 (5.00E-12)	**0.00E+00 (0.00E+00)**	**0.00E+00 (0.00E+00)**	1.25E+05	2.50E+05	**6.0576E+02**
		10	8.70E-04 (2.54E-03)	3.29E-04 (1.80E-03)	**0.00E+00 (0.00E+00)**	6.25E+04	1.25E+05	**4.8864E+02**
f_3	[-600,600]	20	2.01E-08 (6.76E-08)	**0.00E+00 (0.00E+00)**	**0.00E+00 (0.00E+00)**	9.375E+04	1.875E+05	**5.8059E+02**
		30	2.87E-09 (8.45E-10)	**0.00E+00 (0.00E+00)**	**0.00E+00 (0.00E+00)**	1.25E+05	2.50E+05	**1.0058E+03**
		10	0.00E+00 (0.00E+00)	**0.00E+00 (0.00E+00)**	**0.00E+00 (0.00E+00)**	6.25E+04	1.25E+05	**6.8055E+02**
f_4	[-15,15]	20	1.45E-08 (5.06E-08)	**0.00E+00 (0.00E+00)**	**0.00E+00 (0.00E+00)**	9.375E+04	1.875E+05	**1.2788E+02**
		30	3.39E-02 (1.82E-01)	**0.00E+00 (0.00E+00)**	**0.00E+00 (0.00E+00)**	1.25E+05	2.50E+05	**2.3208E+03**
		10	3.41E-02 (4.56E-02)	1.25E-02 (1.26E-02)	**0.00E+00 (0.00E+00)**	6.25E+04	1.25E+05	**3.1879E+03**
f_5	[-15,15]	20	1.36E-01 (1.32E-01)	1.45E-02 (1.09E-02)	**0.00E+00 (0.00E+00)**	9.375E+04	1.875E+05	**5.2455E+03**
		30	2.20E-01 (1.53E-01)	2.01E-02 (2.18E-02)	**0.00E+00 (0.00E+00)**	1.25E+05	2.50E+05	**5.9173E+03**
		10	1.27E-09 (4.00E-12)	1.27E-09 (4.00E-12)	2.76E-07 (**8.22E-12**)	6.25E+04	1.25E+05	**2.5309E+03**
f_6	[-500,500]	20	1.98E+01 (4.51E+01)	2.55E-04 (0.00E+00)	**5.51E-07** (1.68E-11)	9.375E+04	1.875E+05	**6.2559E+03**
		30	1.47E+02 (8.23E+01)	3.82E-04 (1.00E-12)	**8.27E-07** (2.18E-11)	1.25E+05	2.50E+05	**1.0935E+04**

TABLE 26. Comparison between OGA/Q [23], Vaccine-AIS [45] and Entanglement-QTS. The results of Vaccine-AIS and OGA/Q are referred from [45]. The vaccine size of Vaccine-AIS is set to 30 and the termination was set to approach a maximum number of iterations of 1,000. The dimensions of these functions are all set to 30.

F	Limit	D	Mean function value			Evaluation		
			OGA/Q	Vaccine-AIS	Entanglement-QTS	OGA/Q	Vaccine-AIS	Entanglement-QTS
f_2	[-32,32]	30	4.44E-06	**0.00E+00**	**0.00E+00**	1.12421E+05	9.3363E+04	**3.4214E+02**
f_3	[-600,600]	30	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.34000E+05	5.7054E+04	**1.0058E+03**
f_5	[-5,10]	30	7.52E-01	**0.00E+00**	**0.00E+00**	1.67863E+05	4.2833E+04	**6.1372E+03**
f_7	[-100,100]	30	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.12559E+05	7.26333E+04	**1.0560E+03**
f_9	[-100,100]	30	**0.00E+00**	6.02E-04	**0.00E+00**	1.12576E+05	6.2342E+04	**9.8812E+04**

TABLE 27. Comparison between EHS [47], IHS [48], GHS [49] and Entanglement-QTS. The results of EHS, IHS and GHS are referred from [47]. The dimensions of these functions are all set to 50. EHS, IHS and GHS are constrained to 400,000 evaluations.

F	D	Mean function value (Std.)				Evaluation	
		EHS	IHS	GHS	Entanglement-QTS	EHS/IHS/GHS	Entanglement-QTS
f_2	50	2.03E-15 (6.76E-16)	1.80E+01 (9.47E-01)	8.76E+03 (7.92E+01)	**0.00E+00 (0.00E+00)**	4.00E+05	**8.6975E+02**
f_3	50	5.89E-16 (6.00E-15)	1.92E+02 (3.16E+01)	4.97E+01 (9.84E+00)	**0.00E+00 (0.00E+00)**	4.00E+05	**2.1698E+03**
f_4	50	6.39E-12 (4.05E-12)	2.35E+02 (2.82E+01)	7.55E+01 (4.98E+00)	**0.00E+00 (0.00E+00)**	4.00E+05	**3.4752E+03**
f_5	50	1.21E-03 (3.82E-04)	6.28E+04 (3.82E+03)	4.86E+04 (2.80E+03)	**0.00E+00 (0.00E+00)**	4.00E+05	**9.9109E+03**
f_7	50	3.06E-12 (4.26E-12)	8.34E-01 (4.83E-02)	8.94E-02 (1.04E-02)	**0.00E+00 (0.00E+00)**	4.00E+05	**1.7521E+03**

6) Vaccine-AIS [45] is an optimization method based on the concept of an artificial immune system (AIS). As an emerging branch of evolutionary computation, AIS imitates the behavior of cells in the living body that are fighting against disease. It also uses the vaccine operator to enhance its exploration of global and local optima. In [45], Vaccine-AIS was compared with two existing algorithms, namely artificial immune network and orthogonal genetic algorithm with quantization [23]; Vaccine-AIS found the global optimum using the fewest evaluations. The comparison between Vaccine-AIS and Entanglement-QTS is made in Table 26.

7) HS [47]–[49] is a derivative-free real parameter optimization algorithm that was inspired by searching for a perfect state of harmony. It is a global optimization technique that has better exploratory power than that of traditional HS. In [47], HS was compared with most recently published HS variants such as GHS [49], IHS [48] and other well-known optimization algorithms. The comparison between HS and Entanglement-QTS is made in Table 27.

8) PSO [33]–[35] is a global search technique that simulates the behavior of a flock of birds or a shoal of fish. PSO plays a major role in swarm intelligence. BBPSO [34], [35]

TABLE 28. Comparison with CLPSO [33], BBPSOwJ [34] and SMA-BBPSO [35]. The particles of CLPSO, BBPSOwJ and SMA-BBPSO are set to 30. The termination criterion of CLPSO, BBPSOwJ and SMA-BBPSO were set to 1500 iterations. The dimensions of these functions are all set to 30.

F	D	Mean function value (Std.)				Evaluation	
		CLPSO	BBPSOwJ	SMA-BBPSO	Entanglement-QTS	SMA-BBPSO /CLPSO	Entanglement-QTS
f_2	30	**0.00E+00 (0.00E+00)**	1.54E-01 (3.55E-01)	2.22E-15 (1.81E-15)	**0.00E+00 (0.00E+00)**	4.50E+04	**3.4214E+02**
f_3	30	3.14E-10 (4.64E-10)	3.05E-02 (2.84E-02)	**0.00E+00 (0.00E+00)**	**0.00E+00 (0.00E+00)**	4.50E+04	**1.0058E+03**
f_4	30	4.85E-10 (3.63E-10)	1.11E+01 (3.45E+00)	**0.00E+00 (0.00E+00)**	**0.00E+00 (0.00E+00)**	4.50E+04	**1.3364E+03**
f_5	30	2.10E+01 (2.98E+00)	6.50E+01 (4.22E+01)	2.87E+01 (1.37E-02)	**0.00E+00 (0.00E+00)**	4.50E+04	**6.9292E+03**
f_7	30	4.46E-14 (1.73E-14)	4.34E-03 (2.37E-02)	2.42E-154 (2.71E-154)	**0.00E+00 (0.00E+00)**	4.50E+04	**1.0560E+03**

uses a probability distribution to replace the old formula that was based on velocity. BBPSO can explore a wider range of solution space than can standard PSO. This paper [35] also applies BBPSO with a scale matrix to improve the premature convergence problem of BBPSO. The comparison between PSO and Entanglement-QTS is made in Table 28.

D. PERFORMANCE ANALYSIS OF FUNCTION OPTIMIZATION PROBLEM

As seen in Tables 21-28, the results for the Entanglement-QTS algorithm demonstrate much better performance than the other optimization algorithms, both in terms of solution quality and computational cost. The proposed method uses less computational resources to find better solutions than do QEA, QICA, QPSO, QTS, ABC, Vaccine-AIS, HS, and PSO. Tables 21-28 give the comparison results.

Table 21 compares the Entanglement-QTS algorithm with QEA. Our method outperformed QEA, the results in f_4 revealing that Entanglement-QTS algorithm can escape from local area more effectively than can QEA. The proposed algorithm's performance in f_5 illustrates that the proposed entanglement enhanced method can solve a strongly dependent problem. Table 22 compares the Entanglement-QTS algorithm with QICA. The results indicate that the proposed method applies fewer evaluations than does QICA to locate the global optimum. This is because the Entanglement-QTS algorithm uses the best and worst solutions to guide the quantum matrix to efficiently search for the global optimum, and an entanglement step with a local search step to accelerate the search. Table 23 compares the Entanglement-QTS algorithm with QPSO. The results show that in relation to standard deviation, the proposed method show more stable. Table 24 compares the Entanglement-QTS with QTS. We improve the search ability of traditional QTS and add mechanics, addressing the problem of many local optima and strong dependency of each variable.

Table 25 compares the Entanglement-QTS algorithm with ABC. The results for f_3 and f_5 demonstrate that the Entanglement-QTS algorithm can manage the dependency problem in a more favorable manner than can ABC. QIEAs use probabilistic representation to indicate the entire solution space; the proposed algorithm has a stronger diversification ability in f_6, meaning that compared with ABC, the Entanglement-QTS algorithm discovers more solution space. Table 26 compares the Entanglement-QTS algorithm with Vaccine-AIS. The results demonstrate that both the Entanglement-QTS algorithm and Vaccine-AIS can find the global optimum, but that the Entanglement-QTS algorithm has a lower computational cost because it uses a Q-gate and an entangled state to enhance its search. Table 27 compares the Entanglement-QTS algorithm with HS. Each function dimension is 50. The results indicate that although HS exhibits a powerful exploratory ability, its exploiting ability performs unfavorably. The Entanglement-QTS algorithm has demonstrated its powerful searching ability in terms of diversification and intensification. Table 28 compares the Entanglement-QTS algorithm with BBPSO. BBPSO possesses a strong searching capability, but falters when addressing strongly dependent problem. The Entanglement-QTS algorithm excelled at solving strong-dependency problem.

In summary, the Entanglement-QTS algorithm outperformed other optimization algorithms. It includes advantages of QIEAs, such as balancing diversification and intensification and utilizes the spirit of swarm intelligence, which applies information regarding the best and worst solution to guide the updating process. The main contribution of this paper is the addition of the entangled state to manage strongly dependent problems. The entangled state can solve high-dependency problems, cross dimensions to improve solution quality greatly, accelerate the search, and use fewer evaluations to find the global optimum. In addition, it uses the quantum NOT gate to escape from local optima effectively and local search (traditional and entanglement) mechanisms to intensify the search. The Entanglement-QTS is a powerful search algorithm.

VII. CONCLUSION

This paper indicates that most NP-complete problems exhibit strong dependency because their variables are bound by an invisible line, namely their constraints. To address this problem, this paper proposed a novel and powerful optimization method called the Entanglement-QTS algorithm, which combines QTS and the features of entanglement in quantum computing. QTS is an efficient and effective evolutionary algorithm based on Tabu search and features of quantum superposition. Entanglement is a phenomenon in quantum computing in which the state of quantum bits cannot be described independently.

This study applied the Entanglement-QTS algorithm to solve benchmark function problems with many local optima and strong dependency, in order to test its searching abilities. The Entanglement-QTS algorithm demonstrated its powerful search capabilities when solving such problems. Firstly, the Entanglement-QTS algorithm applies QTS to discover solutions efficiently and effectively and balance diversification and intensification. Secondly, the Entanglement-QTS algorithm uses entangled Q-bits to manage function variables that depend strongly on each other. The entangled state is a state-of-the-art concept that is the most significant contribution of this paper. The entangled state can solve high-dependency problems, cross the dimensions to improve the quality of solution greatly, accelerate the search, and use fewer evaluation to determine the global optimum. Thirdly, the Entanglement-QTS algorithm uses (traditional and entanglement) local search mechanisms to intensify the search speed and the quantum NOT gate to effectively escape from local optima. The experiment results indicate that the Entanglement-QTS algorithm improve QTS greatly and outperforms other global optimization algorithms such as QIEAs (QEA, QICA, and QPSO), and other powerful swarm algorithms and evolutionary algorithms (OGA, ABC, Vaccine-AIS, HS and BBPSO). This work has proposed a new and promising metaheuristic algorithm called Entanglement-QTS not only to solve combinatorial optimization problems efficiently and effectively, but also do manage the strong-dependency problems that are very complex and difficult to solve by traditional optimization algorithms.

There are several issues for future study such as using different states of entanglement and analysing their effects, and applying Entanglement-QTS to many combinatorial optimization problems and multi-objective problems.

REFERENCES

[1] J. H. Holland, "Genetic algorithms," *Sci. Amer.*, vol. 267, no. 1, pp. 66–72, 1992.

[2] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Nov./Dec. 1995, pp. 1942–1948.

[3] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 6, no. 6, pp. 580–593, Dec. 2002.

[4] G. X. Zhang, "Quantum-inspired evolutionary algorithms: A survey and empirical study," *J. Heuristics*, vol. 17, no. 3, pp. 303–351, 2011.

[5] A. Manju and M. J. Nigam, "Applications of quantum inspired computational intelligence: A survey," *Artif. Intell. Rev.*, vol. 42, no. 1, pp. 79–156, 2014.

[6] A. Glassner, "Quantum computing. 1," *IEEE Comput. Graph. Appl.*, vol. 21, no. 4, pp. 84–92, Jul./Aug. 2001.

[7] A. Glassner, "Quantum computing. 2," *IEEE Comput. Graph. Appl.*, vol. 21, no. 5, pp. 86–95, Sep. 2001.

[8] A. Glassner, "Quantum computing. 3," *IEEE Comput. Graph. Appl.*, vol. 21, no. 6, pp. 72–82, Nov. 2001.

[9] K. Meng, H. G. Wang, Z. Dong, and K. P. Wong, "Quantum-inspired particle swarm optimization for valve-point economic load dispatch," *IEEE Trans. Power Syst.*, vol. 25, no. 1, pp. 215–222, Feb. 2010.

[10] Y.-W. Jeong, J.-B. Park, S.-H. Jang, and K. Y. Lee, "A new quantum-inspired binary PSO: Application to unit commitment problems for power systems," *IEEE Trans. Power Syst.*, vol. 25, no. 3, pp. 1486–1495, Aug. 2010.

[11] L. Jiao, Y. Li, M. Gong, and X. Zhang, "Quantum-inspired immune clonal algorithm for global optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 5, pp. 1234–1253, Oct. 2008.

[12] A. Draa, S. Meshoul, H. Talbi, and M. Batouche, "A quantum-inspired differential evolution algorithm for solving the N-queens problem," *Int. Arab J. Inf. Technol.*, vol. 7, no. 1, pp. 21–27, 2010.

[13] Y.-H. Chou, Y.-J. Yang, and C.-H. Chiu, "Classical and quantum-inspired Tabu search for solving 0/1 knapsack problem," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2011, pp. 1364–1369.

[14] H.-P. Chiang, Y.-H. Chou, C.-H. Chiu, S.-Y. Kuo, and Y.-M. Huang, "A quantum-inspired Tabu search algorithm for solving combinatorial optimization problems," *Soft Comput.*, vol. 18, no. 9, pp. 1771–1781, 2014.

[15] Y.-H. Chou, S.-Y. Kuo, C.-Y. Chen, and H.-C. Chao, "A rule-based dynamic decision-making stock trading system based on quantum-inspired Tabu search algorithm," *IEEE Access*, vol. 2, pp. 883–896, 2014.

[16] S.-Y. Kuo, C. Kuo, and Y.-H. Chou, "Dynamic stock trading system based on quantum-inspired Tabu search algorithm," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 1029–1036.

[17] Y.-H. Chou, S.-Y. Kuo, C. Kuo, and Y.-C. Tsai, "Intelligent stock trading system based on QTS algorithm in Japan's stock market," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2013, pp. 977–982.

[18] W.-H. Wang, C.-H. Chiu, S.-Y. Kuo, S.-F. Huang, and Y.-H. Chou, "Quantum-inspired Tabu search algorithm for reversible logic circuit synthesis," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2012, pp. 709–714.

[19] Y.-J. Yang, S.-Y. Kuo, F.-J. Lin, I.-I. Liu, and Y.-H. Chou, "Improved quantum-inspired Tabu search algorithm for solving function optimization problem," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2013, pp. 823–828.

[20] Y.-H. Chou, S.-Y. Kuo, and C. Kuo, "A dynamic stock trading system based on a Multi-objective quantum-inspired Tabu search algorithm," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2014, pp. 112–119.

[21] M.-H. Jao *et al.*, "A wormhole attacks detection using a QTS algorithm with MA in WSN," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2015, pp. 20–25.

[22] J. G. Digalakis and K. G. Margaritis, "An experimental study of benchmarking functions for genetic algorithms," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, vol. 5. Oct. 2000, pp. 3810–3815.

[23] Y.-W. Leung and Y. Wang, "An orthogonal genetic algorithm with quantization for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 5, no. 1, pp. 41–53, Feb. 2001.

[24] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.

[25] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[26] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[27] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 482–500, Apr. 2012.

[28] W. Gong, Z. Cai, and D. Liang, "Adaptive ranking mutation operator based differential evolution for constrained optimization," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 716–727, Apr. 2015.

[29] M. Yang, C. Li, Z. Cai, and J. Guan, "Differential evolution with auto-enhanced population diversity," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 302–315, Feb. 2015.

[30] Y. L. Li, Z. H. Zhan, Y. J. Gong, W. N. Chen, J. Zhang, and Y. Li, "Differential evolution with an evolution path: A DEEP evolutionary algorithm," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1798–1810, Sep. 2015.

[31] A. Engelbrecht, "Particle swarm optimization: Velocity initialization," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2012, pp. 1–8.

[32] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 240–255, Jun. 2004.

[33] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, Jun. 2006.

[34] T. Blackwell, "A study of collapse in bare bones particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 3, pp. 354–372, Jun. 2012.
[35] M. Campos, R. A. Krohling, and I. Enriquez, "Bare bones particle swarm optimization with scale matrix adaptation," *IEEE Trans. Cybern.*, vol. 44, no. 9, pp. 1567–1578, Sep. 2014.
[36] J. Li, J. Zhang, C. Jiang, and M. Zhou, "Composite particle swarm optimizer with historical memory for function optimization," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2350–2363, Oct. 2015.
[37] Q. Qin, S. Cheng, Q. Zhang, L. Li, and Y. Shi, "Particle swarm optimization with interswarm interactive learning strategy," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2238–2251, Oct. 2016.
[38] M. R. Bonyadi and Z. Michalewicz, "Stability analysis of the particle swarm optimization without stagnation assumption," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 814–819, Oct. 2016.
[39] J. Liu, Y. Mei, and X. Li, "An analysis of the inertia weight parameter for binary particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 666–681, Oct. 2016.
[40] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, 2007.
[41] G. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," *Appl. Math. Comput.*, vol. 217, no. 7, pp. 3166–3173, 2010.
[42] W.-F. Gao, S.-Y. Liu, and L.-L. Huang, "A novel artificial bee colony algorithm based on modified search equation and orthogonal learning," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 1011–1024, Jun. 2013.
[43] W.-F. Gao, L.-L. Huang, S.-Y. Liu, and C. Dai, "Artificial bee colony algorithm based on information learning," *IEEE Trans. Cybern.*, vol. 45, no. 12, pp. 2827–2839, Dec. 2015.
[44] J.-Q. Li, Q.-K. Pan, and P.-Y. Duan, "An improved artificial bee colony algorithm for solving hybrid flexible flowshop with dynamic operation skipping," *IEEE Trans. Cybern.*, vol. 46, no. 6, pp. 1311–1324, Jun. 2016.
[45] K. M. Woldemariam and G. G. Yen, "Vaccine-enhanced artificial immune system for multimodal function optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 1, pp. 218–228, Feb. 2010.
[46] W. Zhang, G. G. Yen, and Z. He, "Constrained optimization via artificial immune system," *IEEE Trans. Cybern.*, vol. 44, no. 2, pp. 185–198, Feb. 2014.
[47] S. Das, A. Mukhopadhyay, A. Roy, A. Abraham, and B. K. Panigrahi, "Exploratory power of the harmony search algorithm: Analysis and improvements for global numerical optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 1, pp. 89–106, Feb. 2011.
[48] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Appl. Math. Comput.*, vol. 188, no. 2, pp. 1567–1579, May 2007.
[49] M. G. H. Omran and M. Mahdavi, "Global-best harmony search," *Appl. Math. Comput.*, vol. 198, no. 2, pp. 643–656, 2008.
[50] G.-X. Zhang, "Time-frequency atom decomposition with quantum-inspired evolutionary algorithms," *Circuits, Syst. Signal Process.*, vol. 29, no. 2, pp. 209–233, 2010.
[51] M. D. Platel, S. Schliebs, and N. Kasabov, "Quantum-inspired evolutionary algorithm: A multimodel EDA," *IEEE Trans. Evol. Comput.*, vol. 13, no. 6, pp. 1218–1232, Dec. 2009.
[52] P. Yang, K. Tang, and X. Lu, "Improving estimation of distribution algorithm on multimodal problems by detecting promising areas," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1438–1449, Aug. 2015.
[53] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithms with a new termination criterion, H_ϵ gate, and two-phase scheme," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 156–169, Apr. 2004.
[54] A. V. Abs da Cruz, M. M. B. R. Vellasco, and M. A. C. Pacheco, "Quantum-inspired evolutionary algorithm for numerical optimization," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2006, pp. 2630–2637.
[55] M. Soleimanpour-Moghadam, H. Nezamabadi-Pour, and M. M. Farsangi, "A quantum inspired gravitational search algorithm for numerical function optimization," *Inf. Sci.*, vol. 267, pp. 83–100, May 2014.
[56] A. Layeb, "A novel quantum inspired cuckoo search for knapsack problems," *Int. J. Bio-Inspired Comput.*, vol. 3, no. 5, pp. 297–305, 2011.
[57] N. J. Cheung, X.-M. Ding, and H.-B. Shen, "A nonhomogeneous cuckoo search algorithm based on quantum mechanism for real parameter optimization," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 391–402, Feb. 2017.
[58] Y.-H. Chou, C.-Y. Chen, C.-H. Chiu, and H.-C. Chao, "Classical and quantum-inspired electromagnetism-like mechanism and its applications," *IET Control Theory Appl.*, vol. 6, no. 10, pp. 1424–1433, Jul. 2012.
[59] Y.-H. Chou, C.-C. Chang, C.-H. Chiu, F.-J. Lin, Y.-J. Yang, and Z.-Y. Peng, "Classical and quantum-inspired electromagnetism-like mechanism for solving 0/1 knapsack problems," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2010, pp. 3211–3218.
[60] Ş. I. Birbil and S.-C. Fang, "An electromagnetism-like mechanism for global optimization," *J. Global Optim.*, vol. 25, no. 3, pp. 263–282, 2003.
[61] D. Srinivasan and T. H. Seow, "Particle swarm inspired evolutionary algorithm (PS-EA) for multiobjective optimization problems," in *Proc. IEEE Congr. Evol. Comput.*, vol. 4. Dec. 2003, pp. 2292–2297.

SHU-YU KUO received the B.S. and M.S. degrees in computer science and information engineering from National Chi-Nan University, Nantou, Taiwan, in 2012 and 2013, respectively, where she is currently pursuing the Ph.D. degree with the Department of Computer Science and Information Engineering. Her current research interests are quantum-inspired evolutionary computation, function optimization, and automated trading systems.

YAO-HSIN CHOU received the Ph.D. degree from the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, in 2009. He is currently an Associate Professor with the Department of Computer Science and Information Engineering, National Chi-Nan University, Nantou, Taiwan. He has authored over 60 papers in journals and conference proceedings. His current research interests include artificial intelligence, evolutionary computation, combinatorial optimization, and quantum information science.