Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 676275, 13 pages
http://dx.doi.org/10.1155/2013/676275

Research Article

Artificial Bee Colony Algorithm Merged with Pheromone Communication Mechanism for the 0-1 Multidimensional Knapsack Problem

Junzhong Ji, Hongkai Wei, Chunnian Liu, and Baocai Yin

College of Computer Science and Technology, Beijing University of Technology, Beijing Municipal Key Laboratory of Multimedia and Intelligent Software Technology, Beijing 100124, China

Correspondence should be addressed to Junzhong Ji; jjz01@bjut.edu.cn

Received 24 March 2013; Accepted 11 June 2013

Academic Editor: Vishal Bhatnagar

Given a set of n objects, the objective of the 0-1 multidimensional knapsack problem (MKP_01) is to find a subset of the object set that maximizes the total profit of the objects in the subset while satisfying m knapsack constraints. In this paper, we have proposed a new artificial bee colony (ABC) algorithm for the MKP_01. The new ABC algorithm introduces a novel communication mechanism among bees, which bases on the updating and diffusion of inductive pheromone produced by bees. In a number of experiments and comparisons, our approach obtains better quality solutions in shorter time than the ABC algorithm without the mechanism. We have also compared the solution performance of our approach against some stochastic approaches recently reported in the literature. Computational results demonstrate the superiority of the new ABC approach over all the other approaches.

1. Introduction

Given a set $J = \{o_1, o_2, \ldots, o_n\}$ of n objects and a knapsack with a set $C = \{c_1, c_2, \ldots, c_m\}$ of m dimensions, the 0-1 multidimensional knapsack problem (MKP_01) seeks a subset of J in such a way that the total profit of objects included in the subset is maximized, while m resource constraints remain satisfied. More formally, each object $o_j \in J$ has profit p_j and weight r_{ij} in dimension i ($1 \le i \le m$), and each dimension of the knapsack has a capacity c_i. By introducing binary decision variable x_j to indicate whether object o_j is included into the knapsack ($x_j = 1$) or not ($x_j = 0$), the MKP_01 can be formulated as

$$\begin{aligned} &\text{Maximize} \quad \sum_{j=1}^{n} p_j x_j \\ &\text{Subject to} \quad \sum_{j=1}^{n} r_{ij} x_j \le c_i, \quad i = 1, \ldots, m, \\ &\qquad x_j \in \{0, 1\}, \quad j = 1, \ldots, n. \end{aligned} \tag{1}$$

The MKP_01 is a well-known NP-Hard problem. There are many practical applications which can be formulated as a MKP_01, for example, the capital budgeting problem, the cargo loading, the processor allocation in distributed systems, and the project selection. Therefore, more and more people recently focus on the research for solving the MKP_01. In general, the solving algorithms can be divided into two kinds: exact and heuristic methods [1]. The exact methods used to employ some typical search techniques, such as Enumeration algorithm [2], Branch and Bound method [3], and Approximate Dynamic programming [4]. These methods can be only applied to some small-scaled MKP_01 because the computation complexity is rather high. Subsequently, many heuristic search methods, including Genetic Algorithm (GA) [5, 6], Evolutionary Algorithm (EA) [7], Particle Swarm Optimization (PSO) [8], Ant Colony Optimization (ACO) [9–12], and Artificial Bee Colony (ABC) [13, 14], were proposed by simulating some natural phenomena. As these population-based methods are versatile and robust, thus they have been proved to be very effective methods. Thereinto, ABC algorithm is a recently proposed method, which employs the

mechanism of combining local and global searches to effectively solve MKPs. However, since the algorithm framework itself is flawed, there is a main bottleneck that the iteration number is too large and the convergence time is too long, which strongly restricts the development of ABC algorithm. In [15], we proposed an artificial bee colony algorithm for the MKP_01, which introduced the pheromone into ABC algorithm and gave the corresponding transition strategy. Though there are still some problems, the preliminary experimental results in [15] are very encouraging, and these results are significant motivations for the present research. This paper conducts a further and thorough investigation along this line. In comparison with our previous work, the main new contributions of this paper include the following.

(1) Based on the researches of entomologists, a new algorithm, combining chemical communication way and behavior communication way for solving MKP_01 (ABCPUD-MKP), has been developed. First, the paper extends our previous work, analyzes, and explicitly presents the pheromone communication mechanism. Second, the paper designs the constructing method of feasible solutions based on the new mechanism. Third, the paper introduces the special updating and diffusing strategies of the inductive pheromone. An important characteristic of the new algorithm is that the collaborations among bees can be strengthened, and the intelligent foraging behavior of honey swarm can be mimicked more faithfully by means of updating and diffusion of the inductive pheromone.

(2) Systematic experiments have been conducted to compare the proposed algorithm with the previous work and some other algorithms proposed recently in respective literatures on many instances of two benchmark data sets. Moreover, the sensitivity to algorithmic parameters and effects of different parameter selections have been experimentally investigated.

The rest of this paper is organized as follows. Section 2 provides an introduction to the artificial bee colony algorithm and the basic idea of the ABC for the MKP_01. In Section 3, we describe our new algorithm for the MKP_01. Computational results are presented in Section 4. Finally, we conclude the paper in Section 5.

2. ABC Algorithm and Its Application in the MKP_01

2.1. The Artificial Bee Colony (ABC) Algorithm. The artificial bee colony algorithm is a population-based metaheuristic approach proposed recently; it finds near-optimal solutions to the difficult optimization problems by simulating the intelligent foraging behavior of a honeybee swarm. There are three groups of the foraging bees, employed ones, onlookers, and scouts. All bees that are currently exploiting food sources are called employed bees. These bees bring nectar from different food sources to their hive. Onlooker bees are those bees who are waiting in the hive for the information on food sources to be shared by the employed bees, and scout bees are those bees that are currently searching for new food sources near the hive. By dancing in a common area of the hive, employed bees share the information on food sources with onlooker bees. The duration of a dance of an employed bee depends on the nectar content of the food source currently being exploited. After watching numerous dances, onlooker bees choose a food source according to the probability proportional to the quality of that food source. Therefore, the good food sources can attract more onlookers than the poor ones. Onlooker bees help the employed bee, who associated with the same food source, to perform the exploiting job within the neighborhood of the food source. Whenever a food source is exploited fully, it is abandoned by the employed bee associated with it. Then, the employed bee becomes a scout. The task of a scout is to look for a new food source around the hive that can be viewed as performing the job of exploration. Once a scout bee finds a new food source in its global expedition, it again becomes an employed one and continue its local exploiting job.

Based on such an intelligent foraging behavior of honey bee swarm, the Artificial Bee Colony (ABC) algorithm is proposed by Karaboga [16] and further developed in [17, 18]. The overall process of the ABC algorithm is given in Algorithm 1.

Each cycle of the search consists of three main steps. The first two steps are moving the employed and onlooker bees onto the food sources and performing local optimization in an exploiting way, and the third step employs a scout bee as an explorer to find a new food source for each food source exploited fully. A food source represents a feasible solution to the problem to be optimized. Due to the nectar amount of a food source corresponds to the quality of the solution represented by that food source, so the employed and onlooker bees make use of such a quantity to perform an exploitive search in a local area. Whenever a solution representing a food source is not improved by a predetermined number of cycles, then that food source is abandoned by its employed bee, and the employed bee becomes a scout. The number of cycles for releasing a food source is called a threshold value of *limit*. Every scout is an explorer who does not have any guidance while looking for a new food. That is, a scout may find any kind of food source. Therefore, sometimes a scout might accidentally discover more rich and entirely unknown food source. As a result of such behavior of scout bees, ABC algorithm can overcome the stagnation phenomenon of solutions which is a general problem of the stochastic search methods for solving a combinatorial optimization problem. It is important to note that exploration and exploitation processes are carried out together in ABC algorithm. More specially, employed and onlooker bees accomplish the exploitation process in the search space, while the scouts control the exploration process. In the ABC algorithm, local and global searches have been incorporated into different behaviors of different bees, and three kinds of bees work together to complete the evolution of solutions.

Since the ABC algorithm was proposed in 2005, it has been applied in many research fields in recent years, such as data clustering [19], flow shop scheduling problem [20, 21], multiobjective optimization [22], neural network training [23] and synthesis [24], image processing [25], generalized

1. Initialization: Initialize parameters **K**, **N**, **Limit**, $C_i = 0$ ($i = 1, \ldots, \mathbf{K}$);
2. Initial solutions: Randomly generated **K** food sources $\{S_i(0) \mid i = 1, \ldots, \mathbf{K}\}$;
3. Loop:
For $t = 1$ to **N** do:
{
(1) For $i = 1$ to **K** do: (employed bees select food sources and perform local searches respectively)
{Associate each employed bee with a food source $S_i(t)$ and compute its nectar amount;
Find a new $S_i'(t)$ in the neighborhood of $S_i(t)$ and compute its nectar amount;
Take the better one in $\{S_i'(t), S_i(t)\}$ as a new location of the employed bee;}
(2) For $j = 1$ to **K** do: (onlooker bees help employed bees to perform further local searches)
{Select a food source $S_j(t)$ from $\{S_i(t)\}$ for every onlooker bee;
Find a new $S_j'(t)$ in the neighborhood of $S_j(t)$ and compute its nectar amount;
Take the better one in $\{S_j'(t), S_j(t)\}$ as a new location of the corresponding bees;}
(3) Exploiting new food sources (scout bees randomly carry out global searches)
For $i = 1$ to **K** do: (food sources)
{If $S_i(t) = S_i(t-1)$ then $C_i = C_i + 1$;
If $C_i =$ **Limit** then
{Abandon the $S_i(t)$ and the associated employed bee becomes a scout;
Randomly generate a new $S_i(t)$ and the scout becomes an employed bee again;
$C_i = 0$;}}
(4) Memorize the best food source S_{best} found so far;
$t = t + 1$;
}
4. Output: Return S_{best} while the predefined end condition is met.

Algorithm 1: ABC.

assignment problem [26], coupled ladder network [27], and nurse Rostering [28]. Studies [17, 29] have indicated that ABC algorithms have high search capability to find good solutions efficiently.

2.2. ABC Algorithm for the MKP_01. Following above the ideas of ABC algorithm, Sundar et al. presented a method which integrates ABC algorithm with the MKP_01, called ABC-MKP [13]. In the algorithm, a food source represents a feasible solution constituted a subset of objects for the MKP_01, so the total profit value of objects in the subset is viewed as the nectar amount to evaluate the quality of food sources. Moreover, the ABC-MKP uses binary tournament selection method for selecting a food source for onlookers, that is, two different food sources are randomly selected from the food sources associated with employed bees, then the food source containing richer nectar amount among these two food sources is selected with a random probability b_t otherwise the poorer one is selected. For the ABC-MKP algorithm, determination of a new solution in the neighborhood of a solution is the most important process where two specific heuristic-based change operators and general local search are incorporated.

To determine a solution in the neighborhood of a solution i, the ABC-MKP algorithm randomly selects another solution s_j ($s_j \neq s_i$), then randomly selects two distinct objects with the maximum profit values from s_j which are not present in s_i and add them to s_i which makes solution s_i infeasible. When this method fails to find even one object in s_j different from the objects of s_i, then it means that s_i and s_j are the same solution [30]. There are two different processes for such a situation. In case of an employed bee, the employed bee abandons its associated solution to become a scout, and this scout is again made employed by looking for a new randomly generated solution. There is no further operations like change operator and local search. However, if the same solution occurs while determining a new neighborhood solution for an onlooker, then another solution s_j is selected randomly. This process is repeated until a solution s_j which is different from the solution s_i is found. And then, the process of making the infeasible solution feasible begins with a change operator. The change operator includes DROP PHASE and ADD PHASE [5]. The DROP PHASE drops objects in a way, which is combined random selection with greedy search, until the infeasible solution becomes feasible. With probability p_d, the objects of solution s_i are dropped in the increasing order of their pseudoutility ratios; otherwise, the objects of the solution s_i are dropped randomly. During ADD PHASE, objects which are not in the solution s_i are sorted in decreasing order in light of their pseudoutility ratios. Then each sorted and unselected object is checked one by one whether it can be added to the solution s_i without violating the feasibility. If so, then the object is added to the solution s_i. This process is repeated until all unselected objects are searched for inclusion.

The ABC-MKP algorithm uses the local search in 1-1 exchange and 2-1 exchange ways to improve the solution quality [31]. That is, the local search tries to repeatedly exchange one or two selected objects with an unselected object if such exchange can increase the total profit while maintaining the feasibility of the solution. In 1-1 exchange

way, the algorithm repeatedly exchanges a selected object in the order appeared in the knapsack with the unselected object of highest profit that will keep the solution feasible after the exchange. In 2-1 exchange way, the algorithm exchanges a pair of selected objects with the first unselected object which will keep the solution feasible after the exchange, and total profit will increase or remain the same. The 2-1 exchange is also repeated till a swapping move has been found or all pairs have been considered. As the local search costs much running time of ABC algorithm, therefore, the number of application of the local search is limited empirically. Moreover, 1-1 exchange and 2-1 exchange are used by a probability P_{ls} in a mutually exclusive manner, that is, a real number p is randomly generated from $[0, 1]$. If the $p \leq P_{ls}$, then 1-1 exchange with maximum five applications is used as the local search; otherwise a single 2-1 exchange is used as the local search.

3. ABC Algorithm Based on Pheromone Communication Mechanism for the MKP_01

Though ABC algorithm has increased tremendously in many research topics, so far the only communication mechanism based on the dancing behaviors is employed to exchange information among bees in almost all applications of the ABC algorithm. From the view of entomologists, there are many ways to transfer information among bees, such as a behavior way (dancing), a chemical way (pheromone), and a physical way (light and sound). In this paper, we will introduce a chemical way with the inductive pheromone into the ABC algorithm for solving the MKP_01.

3.1. Communication Mechanism Based on the Inductive Pheromone. In nature, an important way in which organisms can communicate with each other is through the use of pheromone. Pheromone is produced as a liquid and transmitted by direct contact as a liquid or as a vapor. The pheromone is a chemical messenger secreted by one individual which causes a specific response in other members of the same species. Ants and bees demonstrate two prominent examples of pheromone usage, which acknowledges their incredible capability to organize the behaviors of the whole colony. By means of the communication way of pheromone [32, 33], ant colony optimization (ACO) algorithm becomes one of the most well-known algorithms and has been successfully applied in several real-world problems. The success of the use of pheromone in ACO inspirits us to incorporate the communication mechanism of pheromone into a new ABC algorithm, which is highly possible to improve the performance of the ABC algorithm.

In fact, research on the behavior of real bees has greatly inspired our work. Biologists have discovered that bees are also well known for communicating through the use of pheromone [34, 35]. Like ants, bees use pheromone for a number of different communication and behavior-control purposes. One pheromone may cause many different responses, depending on environmental conditions and pheromone concentration. Honeybee pheromone can be grouped into releaser pheromone with short-term effects and primer pheromone with long-term effects; thereinto, releaser pheromone changes the behavior of the recipient. Releaser pheromones have a short-term effect, and they trigger an almost immediate behavioral response from the receiving bee. The inductive pheromone is a kind of releaser pheromone which is left by bees when they walk and is useful in searching for nectar. To mimic such behavior of real bees in some extent, we establish a new communication mechanism among bees in our new ABC algorithm, which includes some new strategies based on the inductive pheromone.

3.2. Construction of Feasible Solutions Based on the New Mechanism. At each iteration of our algorithm, each candidate solution associated to a scout bee is constructed by means of the inductive pheromone and heuristic information. It first randomly chooses an initial object and then iteratively adds objects chosen from an available set, which contains all the objects that can be selected without violating resource constraints.

Using the ideas of AS_MKP [9], the quantity of pheromone laying on an object o_i is denoted by $\tau_i(t)$ (t is the number of iterations). When $t = 0$, $\tau_i(0) = 1/\sum_{j=1}^{n} p_j$ ($j = 1, \ldots, n$). At each step of the construction of a solution, a scout l randomly selects the next object o_j within the set of *candidates* with respect to a probability $P_j^l(t)$. This probability is defined proportionally to a pheromone factor and a heuristic factor, that is,

$$p_j^l(t) = \begin{cases} \dfrac{\left[\tau_j(t)\right]^{\alpha} \cdot \left[\eta_j\left(S_l(t)\right)\right]^{\beta}}{\sum_{i \in allowed_l(t)} \left[\tau_i(t)\right]^{\alpha} \cdot \left[\eta_i\left(S_l(t)\right)\right]^{\beta}}, & j \in allowed_l(t) \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where $S_l(t)$ is a partial solution set acquired by the l^{-th} scout at the time t, $allowed_l(t) \subseteq J - S_l(t)$ is the candidate set of remaining available objects, $\eta_j(S_l(t))$ represents a local heuristic information, and the parameters α and β determine the relative importance of pheromone trail versus heuristic factor for an object j. Thus, the higher the value of $\tau_j(t)$ and $\eta_j(S_l(t))$, the more profitable it is to select object j in the partial solution.

3.2.1. Heuristic Function. There are many constraints in a MKP, so the heuristic factor depends on not only the set $allowed_l(t)$ of candidate objects but also the whole set $S_l(t)$ of selected objects. Let $u_i(l, t) = \sum_{k \in S_l(t)} r_{ik}$ be the consumed quantity of the resource i when the bee l has selected those objects in $S_l(t)$ at the time t and $\gamma_i(l, t) = c_i - u_i(l, t)$ be the remaining capacity of the resource i, then the tightness of a candidate object j on the resource i can be expressed as

$$\delta_{ij}(l, t) = \frac{r_{ij}}{\gamma_i(l, t)}. \tag{3}$$

The equation represents a ratio between r_{ij} and $\gamma_i(l,t)$. Further, when an object is chosen to be included in $S_l(t)$, the average tightness on all constraints is defined as

$$\delta_j(l,t) = \frac{\sum_{i=1}^{m} \delta_{ij}(l,t)}{m}. \tag{4}$$

From the view of consumption of resources, the lower the tightness ratio, the more the object is profitable to be selected. Moreover, the profits p_j must be taken in account in order to get a pseudoutility measure for each candidate object, Thus, the local heuristic function for the MKP, $\eta_j(S_l(t))$, can be defined as follows:

$$\eta_j(S_l(t)) = \frac{p_j}{\delta_j(l,t)}. \tag{5}$$

3.2.2. Inductive Pheromone Updating. Once each food source (solution) has been optimized at every iteration, pheromone trails of the objects are updated. More specifically, in light of the profit value of every solution associated by employed bees, the algorithm updates the pheromone intensity for objects interrelated, the formula is as follows:

$$\tau_i(t+1) = (1-\rho)\tau_i(t) + \Delta\tau_i(t,t+1), \tag{6}$$

$$\Delta\tau_i(t,t+1) = \sum_{l=1}^{K} \Delta\tau_i^l(t,t+1), \tag{7}$$

$$\Delta\tau_i^l(t,t+1) = \begin{cases} Q \cdot L(S_l), & o_i \in S_l \text{ for } l\text{th bee} \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

where $0 < \rho \leq 1$ is a coefficient which represents pheromone evaporation, the $\Delta\tau_i^l(t,t+1)$ represents the pheromone trail that the lth bee deposited on the object i in the iteration, and the $\Delta\tau_i(t,t+1)$ represents the pheromone increment that all the bees deposited on the object i. Q is a constant parameter for an instance of MKP ($Q = 1/\sum_{j=1}^{n} p_j$), and $L(S_l)$ is the value of the objective function of the solution S_l obtained by the lth bee.

From these descriptions about the heuristic function and pheromone updating, we can know that a value of the transition probability represents a trade-off between pseudoutility and pheromone intensity. That is, those objects which consume less resources and have more profit should be chosen with a high probability. On the other hand, if an object is included in many solutions, then it is highly desirable due to having high pheromone intensity.

3.3. Diffusion Strategy of the Inductive Pheromone. Using the new mechanism for solving MKPs, the key point is how to decide which components of the constructed solutions should be rewarded, and how to exploit these rewards when constructing a new solution. There are two different methods: (1) the first one is to lay pheromone trails on each object selected in J, which considers, respectively, the desirability of each object. The more frequent an object occurs in solutions, the more likely it would be selected when constructing a new solution. (2) The other one is to lay pheromone on each pair (o_i, o_j) of different objects in J, which considers together the desirability of two objects in J. If some objects of J have been contained in a new partial solution, then the other objects that often occurred together with these objects in pairs of some solutions will be more attractive. Our algorithm combines these two ways into pheromone updating and diffusion.

3.3.1. Associated Distance Based on Top-k Strategy. If we take a solution acquired by a bee as a transaction while solving a MKP, then we can get a transaction database in each iteration of a bee colony. Observing the transaction database, it is no difficult to find that some objects occur together with other objects in pairs. In other words, there are some frequent object pairs in the feasible solution space, which implies the relationship between two objects of each pair. Therefore, we give a definition of an associated distance between two objects.

Definition 1 (associated distance). Let D be a database with K transactions; each transaction p ($p = 1, 2, \ldots, K$) represents a feasible solution S_p ($S_p \subseteq J$), and (o_i, o_j) is a pair of objects in J. If the pair of (o_i, o_j) appears f ($f = 0, 1, 2, \ldots, K$) times in D, then we define the associated distance of this pair as: $d_r(o_i, o_j) = 1/(f+1)$.

From this definition, we can observe the two properties:

(1) $d_r(o_i, o_j) = d_r(o_j, o_i)$;

(2) $0 < d_r(o_i, o_j) \leq 1$.

The first property denotes the symmetry of the associated distance, which shows that the associated distance is irrelative with the order of objects. And the second property defines the value range of the associated distance; it indicates that the more frequent an object pair appears in D, the shorter the associated distance. That is, the higher the relation degree between the two objects, the smaller $d_r(o_i, o_j)$ is. When $d_r(o_i, o_j) = 1$, the two objects are absolutely irrelative.

To solve an MKP with the ABC algorithm, the number of a bee colony is usually proportional to the number of objects in J. Thus, every iteration for a bee colony may produce a set of solutions (database), whose size approaches the number of objects in the MKP. Due to the frequency statistic (f) for an object pair needs to scan the database in each iteration, so the computation costs of the associated distances for all object pairs may be very expensive for the large-scale MKP. Moreover, the most important idea in swarming intelligence is to strengthen the effect of good solutions and reduce the effect of bad solutions; thus, our ABC algorithm did not treat equivalently all solutions. Therefore, we adopt the idea of Top-k query widely applied in Web search engine to emphasize the influence of good solutions. More precisely, we take the objective function of MKPs as the Top-k function, select the k solutions which have the higher values of the objective function to form a transaction database $D_{\text{Top-}k}$, and then find the associated distances among objects from the $D_{\text{Top-}k}$. This method has two advantages: (1) reduce the computation complexity. When considering all solutions in each iteration,

mining the associated distances needs $O(K \cdot C_n^2) \approx O(n^3)$. However, If only considering Top-k solutions ($k \ll K$), the computation complexity is $O(k \cdot n^2)$. And (2) give prominence to the preference to the better solutions (i.e., Top-k best solutions), thus strengthen the intellectual insight of the better solutions to the next optimization.

3.3.2. Pheromone Diffusion Based on Associated Distances. In our new mechanism, the chemistry substance called pheromone is an important carrier for a bee colony to implement swarming intelligence. While the time passing, the pheromone gradually volatilizes; that is, the pheromone can diffuse around. To simulate this phenomenon, we present a pheromone diffusion model and the corresponding algorithm for solving MKPs. The basic idea is to take into account the pheromone influences among close objects when a bee constructs a feasible solution. Namely, based on coupling actions among near strength fields of pheromone diffusions, the algorithm performs decoupling compensates for the pheromone of near objects. Thus, the closer the associated distance, the stronger the coupling action is, and yet the further the associated distance, the weaker the coupling action is.

Let $\Delta\tau_i$ be the pheromone trail on an object o_i laid by the bee colony; we can give the pheromone diffusion model based on the associated distance. Figure 1 shows the sketch map of the pheromone diffusion, where the red dot(•) denotes the object o_i as an info fountain, those white dots (◦) denote the other different objects influenced by o_i, and different circles can be interpreted as equipotential lines of the intensity field for different locations. The figure shows that the pheromone trail diffuses around by taking an info fountain as a center of the diffusion intensity field, the closer to the info fountain the location, the stronger the field force of the intensity field is. More precisely, the pheromone influence of an info fountain on other objects will gradually reduce as the associated distance between objects becomes long. Therefore, we can give a simple diffusion model:

$$\Delta\tau_{ij} = \begin{cases} \left(\dfrac{1}{k+1}\right) \cdot \dfrac{\Delta\tau_i}{d_r\left(o_i, o_j\right)}, & \text{if } d_r\left(o_i, o_j\right) < 1, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Based on such a pheromone diffusion model, the behavior of selection of an object for a bee can change the pheromone intensity not only on the selected object but also on other objects related with the selected object. Thus, the pheromone updating of the object o_j in (6) can be revised as follows:

$$\tau_i'(t, t+1) = \tau_i(t, t+1) + \sum_r \Delta\tau_{ri}, \tag{10}$$

where r is one of the objects which are associated with the object o_i. This change more faithfully reflects the volatilization character of pheromone and then can lead bees to select objects in an impersonal way. Thus, the pheromone diffusion model can enhance the collaboration among bees in a colony, improve the effectiveness of the ABC algorithm, and in evidence incarnate the idea of swarm intelligence.

3.4. Algorithm Description and Analysis. Based on pheromone updating and diffusion, we design and achieve an ABC algorithm called ABCPUD for the MKPs. The algorithm holds the basic steps of original ABC algorithms, however adds the new strategies based on the inductive pheromone in the generation process of solutions. The overall process of ABCPUD-MKP algorithm is given in Algorithm 2.

From Algorithm 2, we can see that the ABCPUD-MKP is a hybrid algorithm which merges two communication ways among bees. Based on the behavior communication way, the employed bees and onlooker bees gradually optimize their associated solutions in solution neighbors as the iteration carrying through. On the other hand, the scout bees constructed new solutions in light of the chemical communication way which employs the pheromone accumulated by bees to induct the selection of objects. The important differences between the ABC-MKP and the ABCPUD-MKP focus on two phases, that is, exploring new food sources and generating, diffusing, and updating the pheromone only used by ABCPUD-MKP.

During exploring new food sources, the ABCPUD-MKP employs the transition probability based on pheromone trails and heuristic information to insightfully construct new solutions, which keeps high quality of the new solution. Moreover, the ABCPUD-MKP not only makes use of solutions acquired at every iteration to accumulate pheromone on objects but also emphasizes the pheromone influences among associated objects by means of the pheromone diffusion model which stresses the effect of the best Top-k solutions. Therefore, the ABCPUD-MKP may lead to rapidly select the most profitable nectar source by means of the new communication mechanism.

4. Experimental Evaluation

To assess the performance of our algorithm (ABCPUD-MKP), we conduct a series of experiments on public data sets taken from the OR library, which are the same as those used in [5, 7, 11, 12], and so forth. The experimental platform was a PC with Pentium 4, 2.0 GHz CPU, 1 G RAM and Windows XP. We test our algorithm implemented by MATLAB on some instances of 5.100 and 10.100 and compare the results with that of the ABC_MKP and other algorithms recently proposed on the some data sets. The benchmark set 5.100 has 30 instances with $m = 5$ constraints and $n = 100$ objects. Similarly, the benchmark set 10.100 also has 30 instances with $m = 10$ constraints and $n = 100$ objects. All algorithms are executed 10 times on each instance with a different random seed.

By large numbers of experiments, the main parameters were set as follows: $\alpha = 1$, $\beta = 5$, $N = 10000$, $p_d = 0.3$, $p_{1S} = 0.99$, $b_t = 0.9$, $\mathbf{K} = 80$, $\rho = 0.3$, $k = 25$, and $limit = 80$. Here $\mathbf{K}$, k, ρ, and $limit$ are parameters which mainly affect performances of ABCPUD-MKP. How to select their values will be emphatically discussed in the following subsection.

4.1. Performance Analysis of ABCPUD-MKP. We study the factors that affect the performance of ABCPUD-MKP. Particularly, we wish to investigate the contributions of the

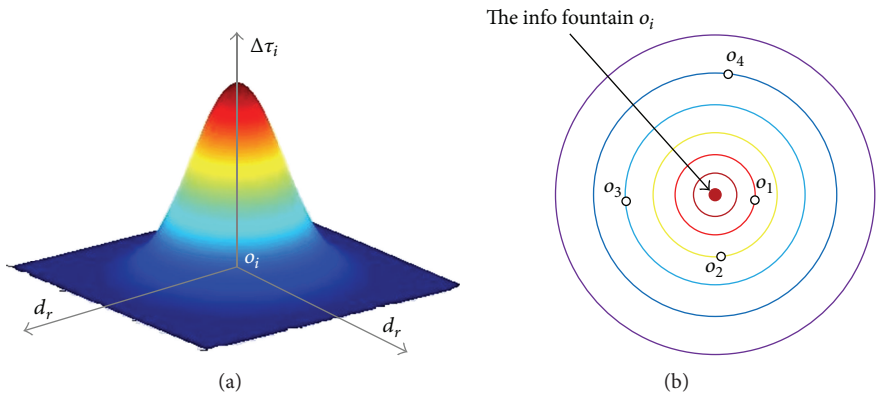

FIGURE 1: The sketch map of the pheromone diffusion of O_i and its intensity field. (a) The relationship between the intensity $\Delta\tau_i$ and the associated distance d_r for the info fountain O_i. (b) The corresponding intensity field and areas of the pheromone diffusion.

pheromone mechanism and the effects of different parameter selection.

(1) *Contributions of the Pheromone Mechanism.* Based on the same running experiment and implemented tools, we developed three algorithms to solve the MKP_01 on some different instances. The three algorithms were called the original ABC-MKP, the ABCPU-MKP (with the pheromone updating), and the ABCPUD-MKP (with the pheromone updating and diffusing), respectively. Table 1 gives the experimental results of three algorithms after running same computational time, where we evaluate the algorithm performance by means of three measures such as *Best*, *Avg.*, and *Num. Best* denotes the profit value of the best solution obtained by the corresponding algorithm over 10 executions. *Avg.* indicates the mean μ and the standard deviation σ of the profit values over 10 executions independently carried out by the corresponding algorithm. *Num.* is the smallest number of the iterations when the best solution was found in 10 trails.

From the table, it is seen that these three algorithms are able to obtain the best solutions known on 10 instances. As for the Avg., ABCPU-MKP produces better results than that of ABC-MKP except for the instance of 5.100.03, and ABCPUD-MKP produces the best results among the three algorithms. From the iteration process, both ABCPU-MKP and ABCPUD-MKP show better performance than ABC-MKP; the latter is more outstanding. Therefore, we can draw the conclusion that the new pheromone mechanism introduced in the paper can evidently improve the ABC-MKP algorithm on the convergence performance while keeping good solution quality. More specifically, (1) the solution construction based on the inductive pheromone not only effectively enhances the convergence performance but also improves the solution quality on a majority of instances tested, which shows that the inductive pheromone is an important media to improve the bee communication in finding better solutions. (2) The new diffusion mechanism can also improve the convergence performance while keeping the solution quality. This suggests that the strategy based on the diffusion model can effectively strengthen the process of solution construction and save iteration times. It is obvious that both of the aforementioned strategies are effective in the improvement of the performance of the ABC-MKP algorithm. This fact encourages us to put both strategies into our algorithm (ABCPUD-MKP) to get even better results.

(2) *Effects of Different Parameter Selection.* In this experiment, we give a solving example to show how to determine the parameter value of ABCPUD-MKP algorithm. The experiment process is that we run ABCPUD-MKP with different parameter settings on the same instance (5.100.06) and acquire the best parameter value by comparing experimental results. During this experimentation, the value of a single parameter is changed while keeping the values of other parameters fixed.

ABC algorithm is a swarm optimization algorithm; the population size of a bee colony determines the number of solutions at each iteration. Figure 2 summarizes the performance of ABCPUD-MKP with 10 different bee colony sizes (**K**). The best value, the worst value, and the average value are, respectively, corresponding the highest, lowest, and mean profit value of solutions obtained in the 10 trails (shown in Figure 2(a)). Figure 2(b) illustrates the results about the running time. The median of each bar is the mean, and the height of the bar is the standard deviation, which are obtained from 10 trails.

A large bee colony means that more initial search points are employed, and then, as reflected by best, worst, and average values in Figure 4, better solutions are obtained than in a smaller bee colony. However, after a sufficient value for the colony size, any increment does not improve the solution performance of algorithms. On the contrary, the search time in each iteration will increase as the size of the bee colony increases. To acquire a balance between getting a better solution and using less time, we recommend a bee colony size of 80 (**K** = 80).

As described in Section 3.2, the coefficient ρ is a value smaller than 1 to avoid unlimited accumulation of pheromone. In the constructing solution process, the value of ρ controls the balance between exploration and exploitation processes. Therefore, it is an important parameter for the new mechanism based on the inductive pheromone. To investigate

1. Initialization: Initialize parameters $K, N, Limit, \alpha, \beta, Q, \tau_0, C_i = 0$ ($i = 1, \ldots,$ **K**) and so on;
2. Initial solutions: Randomly generated K food sources $\{S_i(0) \mid i = 1, \ldots,$ **K**$\}$;
3. Loop:
For $t = 1$ to N do:
{
(1) For $i = 1$ to **K** do: (local searches performed by employed bees)
{Associate each employed bee with a food source $S_i(t)$ and compute its nectar amount;
Find a new $S_i'(t)$ in the neighborhood of $S_i(t)$ and compute its nectar amount;
Take the better one in $\{S_i'(t), S_i(t)\}$ as a new location of the employed bee;}
(2) For $j = 1$ to **K** do: (further local searches performed by onlooker bees)
{Select a food source $S_j(t)$ from $\{S_i(t)\}$ for every onlooker bee;
Find a new $S_j'(t)$ in the neighborhood of $S_j(t)$ and compute its nectar amount;
Take the better one in $\{S_j'(t), S_j(t)\}$ as a new location of the corresponding bees;}
(3) Exploiting new food sources (global searches with a guidance performed by scout bees)
For $l = 1$ to **K** do: (food sources)
{If $S_l(t) = S_l(t-1)$ then $C_l = C_l + 1$;
If $C_l = Limit$ then
{Abandon food source l and the associated employed bee becomes a scout;
$S_l(t) = \{\}$, $allowed_l = J$;
Repeat (constructing a new solution)
Select an object j with $P_j^l(t)$ given by (2);
Add the object into the current solution: $S_l(t) = S_l(t) + \{o_j\}$;
$allowed_l = allowed_l - \{o_j\}$;
Until $allowed_l$ is empty
The scout bee becomes again an employed bee;
$C_l = 0$;}}
(4) Generating, diffusing and updating the pheromone
For each object o_i in J
Calculate the $\Delta\tau_i$ according to (7) and (8).
Select the Top-k solutions from this iteration, and obtain $D_{\text{Top-}k}$;
For each solution $S_w \in D_{\text{Top-}k}$
{For each pair of objects c_{ij} in S_w
c_{ij}.count++;
Calculate the associated distance $d_r(o_i, o_j)$
Calculate $\Delta\tau_{ij}$ according to (9);}
For each object o_i in J
Update the trail level τ_j on all objects according to (6) and (10);
(5) Perform the local optimization in 1-1 or 2-1 exchange ways;
(6) Memorize the best food source S_{best} found so far;
$t = t + 1$;
}
4. Output: Return S_{best} while the predefined end condition is met.

Algorithm 2: ABCPUD-MKP.

Table 1: The contributions of the pheromone mechanism on ABC-MKP.

Instances	Best known	ABC-MKP			ABCPU-MKP			ABCPUD-MKP		
		Best	Avg.	Num.	Best	Avg.	Num.	Best	Avg.	Num.
5.100.00	24381	24381	24381.0 ± 0.0	238	24381	24381.0 ± 0.0	220	24381	24381.0 ± 0.0	**178**
5.100.01	24274	24274	24274.0 ± 0.0	36	24274	24274.0 ± 0.0	32	24274	24274.0 ± 0.0	**20**
5.100.02	23551	23551	23539.3 ± 4.1	4021	23551	23548.0 ± 5.5	2251	23551	23550.0 ± 3.9	**354**
5.100.03	23534	23534	23534.0 ± 0.0	2356	23534	23527.0 ± 3.5	2202	23534	23534.0 ± 0.0	**1538**
5.100.04	23991	23991	23973.2 ± 14.8	1746	23991	23982.4 ± 13.7	1689	23991	23986.0 ± 10.5	**1622**
5.100.05	24613	24613	24613.0 ± 0.0	110	24613	24613.0 ± 0.0	76	24613	24613.0 ± 0.0	**46**
5.100.06	25591	25591	25591.0 ± 0.0	252	25591	25591.0 ± 0.0	224	25591	25591.0 ± 0.0	**65**
5.100.07	23410	23410	23410.0 ± 0.0	409	23410	23410.0 ± 0.0	304	23410	23410.0 ± 0.0	**259**
5.100.08	24216	24216	24216.0 ± 0.0	1051	24216	24216.0 ± 0.0	799	24216	24216.0 ± 0.0	**755**
5.100.09	24411	24411	24411.0 ± 0.0	189	24411	24411.0 ± 0.0	97	24411	24411.0 ± 0.0	**81**

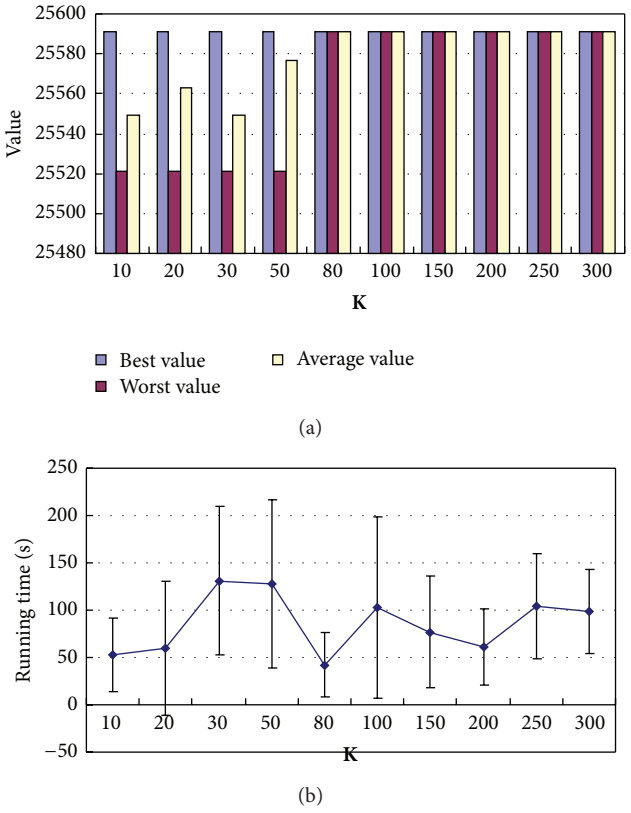

Figure 2: Comparisons of the results for different bee colony sizes (**K**). (a) Best, worst, and average values for different **K**. (b) The runtime for different **K**.

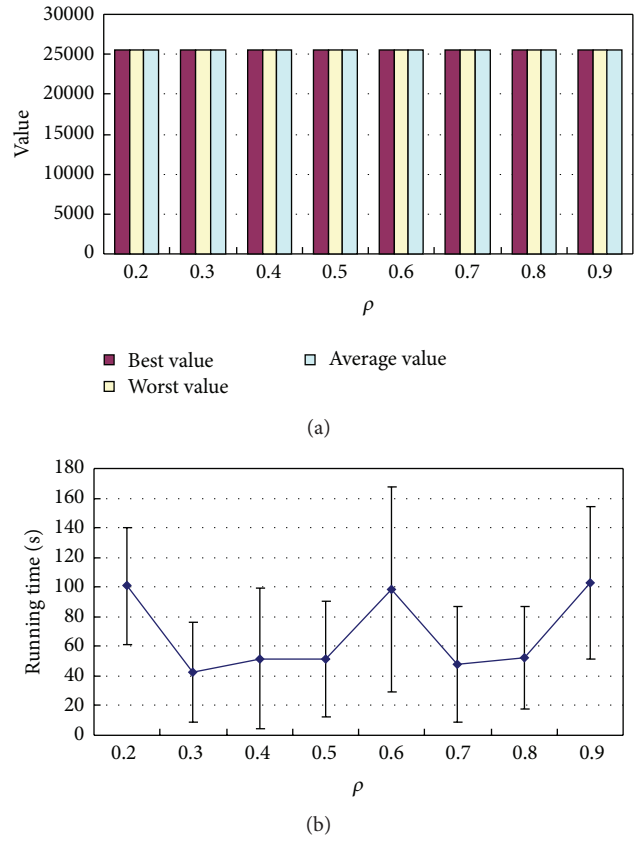

Figure 3: Comparisons of the results for different pheromone parameters (ρ). (a) Best, worst, and average values for different ρ. (b) The runtime for different ρ.

the effect of ρ on ABCPUD-MKP, we perform experiments using different values of ρ. The results are presented in Figure 3.

From Figure 3(a), we notice that the optimum values on best, worse, and average values can be obtained for different parameter values of ρ. That is, the solution of ABCPUD-MKP is insensitive to the parameter ρ. However, the significant difference focuses on the running time shown in Figure 3(b). Because the smallest running time is achieved for $\rho = 0.3$ among all testings, we select a ρ value of 0.3 for our ABCPUD-MKP.

In our algorithm, we introduce a pheromone diffusion model based on associated distances. To save computation costs and strengthen the effect of good solutions, we adopt the Top-k strategy to compute the associated distances of every pair of objects. In this section, we perform experiments using different values of k to investigate the effect of the Top-k strategy on ABCPUD-MKP. Figure 6 shows the experimental results. We observe that there is no difference among the three profit values in Figure 4(a), and only there are some differences in running time for different values of k in Figure 4(b). To quickly get the best result, we can select $k = 25$.

If a solution does not improve for a predetermined number of iterations (*limit*), then the solution will be abandoned by the employed bee associated, and a new solution will be generated. This is a key strategy to deal with the solution stagnation in ABC algorithm. To study the effect of different abandoned frequencies on ABCPUD-MKP, we perform experiments using different values of *limit*. The experimental results are presented in Figure 5.

When *limit* is too small or too large, the results obtained by our algorithm are worse than those produced by using the moderate values of *limit*. This shows that an appropriate frequency of new solution production has useful effect on the running time and the profit values, which can perform some explorations to improve the search ability of the algorithm. However, the balance between exploration and exploitation processes will be broken whether *limit* is too small or too large, which will produce worse solutions and cost much more running time. Thus, we recommend *limit* = 80 after considering effects on factors of the running time and solution quality.

4.2. Comparing ABCPUD-MKP with ABC-MKP. We compare the ABCPUD-MKP with ABC-MKP on many instances. As space is limited, Table 2 only provides a summary of the performance comparison on 6 different instances. Both algorithms are independently executed 10 times for each instance, and the figures are, therefore, the average values of 10 trails.

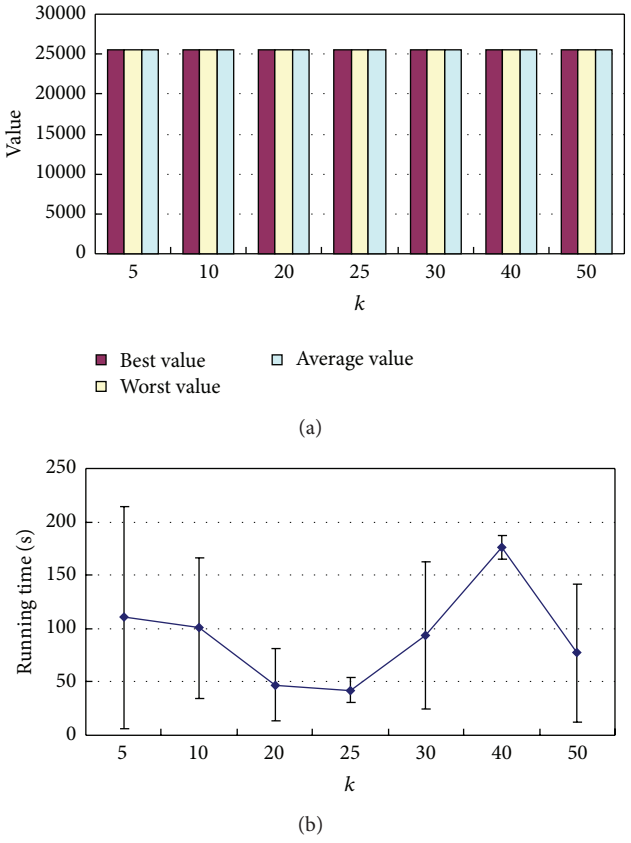

Figure 4: Comparisons of the results for different Top-k parameters (k). (a) Best, worst, and average values for different k. (b) The runtime for different k.

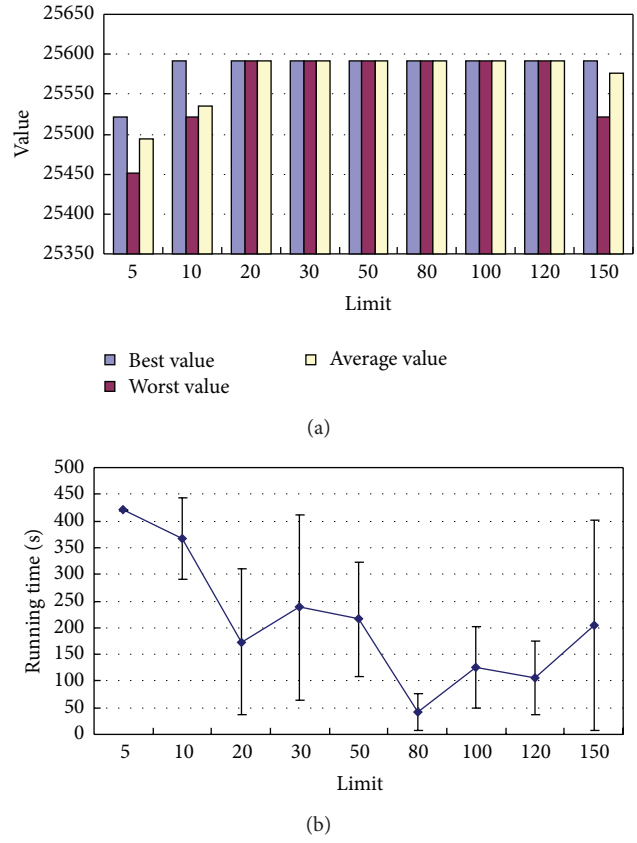

Figure 5: Comparisons of the results for different abandoned frequencies (*limit*). (a) Best, worst, and average values for different *limit*. (b) The runtime for different *limit*.

In Table 2, the meanings of *Best*, *Avg.*, and *Num.* are same as those of Table 1 while the meaning of *Time* is the same as that of Figure 2. *Hit.* represents the number of the best solution obtained in 10 trails. Moreover, numbers in parentheses of the *best* column are the best results known for each corresponding instance, and numbers in parentheses of the *Num.* and *Time* columns are the smallest numbers of the iterations and the shortest running time when the best result was obtained. Compared with ABC-MKP, ABCPUD-MKP can always find better or equally good solutions for all the instances in terms of both profit values. Since the *Num.* is reduced on all instances, hence the *Time* of ABCPUD-MKP is greatly improved. Moreover, the ABCPUD-MKP algorithm also outperforms ABC-MKP algorithm according to the item of *Hit.*

In Figure 6, we compare the iteration numbers and the runtime of an iteration of two algorithms on ten instances of 10.100. For each algorithm, we record the iteration number averaged over 10 runs and compute the runtime of an iteration averaged over 10 runs when obtaining the profit values shown as Table 1 on respective instances. From Figure 6(a), we observe that the iteration numbers of both algorithms vary with different instances. Furthermore, the new pheromone mechanism adopted by ABCPUD-MKP can improve the iteration process for all instances. However, the new pheromone mechanism will increase some computation cost; thus, the runtime per an iteration of ABCPUD-MKP may be longer than that of ABC-MKP on some instances as shown in Figure 6(b).

Figure 7 gives the time performance comparison between two algorithms, which corresponds to Figure 6. We can see that ABCPUD-MKP performs better than ABC-MKP in terms of the runtime on all instances. The main reason is that ABCPUD-MKP can effectively decrease the iteration number. As shown in Figure 7, the advantage in runtime is the most remarkable on almost all instances. This denotes that the pheromone updating and diffusing can accelerate the optimization process by strengthening the communication among bees.

4.3. Comparing ABCPUD-MKP with Other Algorithms. To further evaluate the new algorithm, we compare the solution performance of different algorithms on some large problems. Our main objective in this section is to determine whether ABCPUD-MKP is more efficient and effective on comparable performances than these state-of-the-art approaches which employed various stochastic search schemes in recent years.

We compare the performance of ABCPUD-MKP with that of GA-MKP [5], B&B-EA-MKP [7], Ant-knapsack [11],

Table 2: The results for two algorithms on some instances.

Instance	statistic	Algoritm	
		ABC-MKP	ABCPUD-MKP
5.100.01	Best	**24274 (24274)**	**24274 (24274)**
	Avg.	**24274 ± 0.0**	**24274 ± 0.0**
	Num.	130.25 ± 108.78 (53)	**87.60 ± 74.02 (35)**
	Hit.	**10**	**10**
	Time (s)	12.50 ± 10.87 (4.88)	**7.44 ± 6.56 (2.70)**
5.100.03	Best	**23534 (23534)**	**23534 (23534)**
	Avg.	**23534 ± 0.0**	**23534 ± 0.0**
	Num.	4118.33 ± 782.49 (2356)	**2571.0 ± 551.54 (2181)**
	Hit.	**10**	**10**
	Time (s)	389.45 ± 70.26 (223.03)	**216.21 ± 51.11 (197.92)**
5.100.05	Best	**24613 (24613)**	**24613 (24613)**
	Avg.	**24613 ± 0.0**	**24613 ± 0.0**
	Num.	178.8 ± 63.33 (110)	**96.0 ± 21.74 (74)**
	Hit.	**10**	**10**
	Time (s)	14.24 ± 5.05 (8.67)	**8.03 ± 1.80 (5.89)**
10.100.01	Best	**22801 (22801)**	**22801 (22801)**
	Avg.	**22801 ± 0.0**	**22801 ± 0.0**
	Num.	3824.67 ± 2959.10 (844)	**3161.79 ± 2331.48 (103)**
	Hit.	**10**	**10**
	Time (s)	318.72 ± 304.74 (88.52)	**276.22 ± 244.62 (10.41)**
10.100.03	Best	**22772 (22772)**	**22772 (22772)**
	Avg.	**22772 ± 0.0**	**22772 ± 0.0**
	Num.	5707.0 ± 1035.23 (4483)	**4441.5 ± 1469.34 (2855)**
	Hit.	**10**	**10**
	Time (s)	506.3 ± 97.47 (400.33)	**399.15 ± 133.47 (254.10)**
10.100.05	Best	**22777 (22777)**	**22777 (22777)**
	Avg.	22761.8 ± 29.50	**22771.3 ± 19.30**
	Num.	5887.6 ± 2743.9 (585)	**3706.8 ± 2345.87 (258)**
	Hit.	7	**9**
	Time (s)	633.32 ± 349.37 (65.16)	**338.09 ± 209.85 (23.07)**

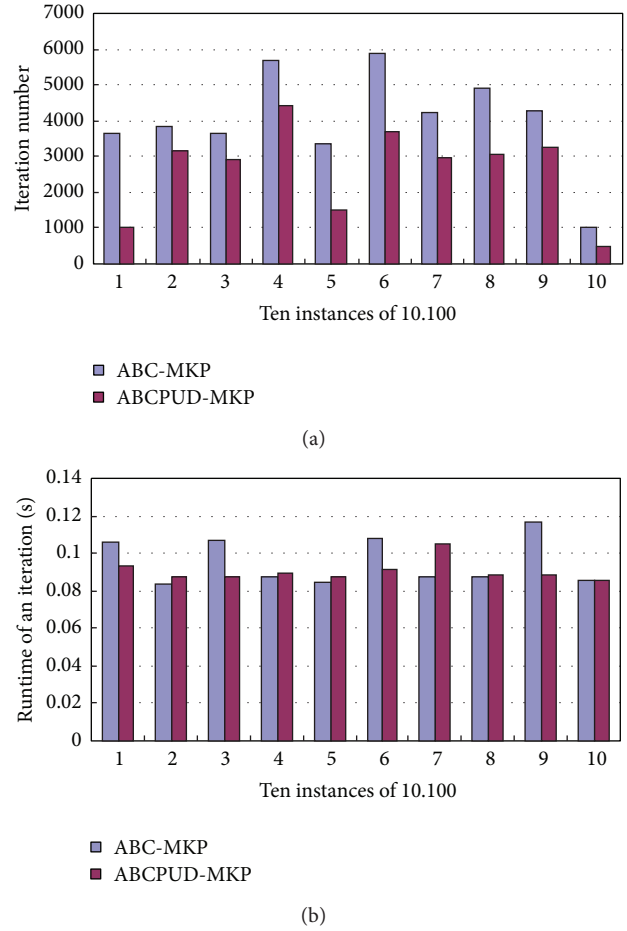

Figure 6: Comparison of the iteration performance on different instances: (a) the iteration numbers, (b) the runtime of an iteration.

ACOMPD-MKP [12], and ABC-MKP. GA-MKP incorporates a heuristic operator which utilizes problem-specific knowledge into the genetic algorithm to solve the multidimensional knapsack problem. B&B-EA-MKP is a hybrid approach which cooperates an evolutionary algorithm (EA) with the branch and bound method (B&B) by exchanging information. Ant-knapsack is an ACO algorithm for the multidimensional knapsack problem, which lays pheromone trails not only on the edges of the visited paths but on all edges connecting any pair of nodes belonging to the solution. ACOMPD-MKP proposed in our prophase research is a novel Ant algorithm, which employs a pheromone diffusion model and a solution mutation strategy to get high quality results. Moreover, as described in Section 2.2, ABC-MKP is a new algorithm which applies ABC algorithm to solve the multidimensional knapsack problem. Therefore, these algorithms are stochastic search approaches, which need to be executed time after time for each testing instance.

In Table 3, we summarize the comparative results, where the results of GA-MKP, B&B-EA-MKP, Ant-knapsack, and ACOMPD-MKP are derived from the respective literature (represents no testing) while the results of ABC-MKP and ABCPUD-MKP algorithms obtained by our experiment over 10 runs.

As it can be seen, ABCPUD-MKP always finds all best solutions on six instances in terms of the profit value while other algorithms only find some best solutions. For instance, ACOMPD-MKP can find the best solutions on five instances except for 5.500.29, B&B-EA-MKP can find best solutions on 5.100.00, 10.100.00, 5.250.00, and 5.500.29 instances. As for the average solutions, ABCPUD-MKP is also competitive with B&B-EA-MKP which provides the highest quality results so far. More specifically, B&B-EA-MKP produces better results than that of ABCPUD-MKP on 5.250.00, 5.250.29, and 5.500.29 instances, and ABCPUD-MKP produces better results than that of B&B-EA-MKP on 5.100.29 and 10.100.00 instances while both algorithms can obtain the best results on 5.100.00 instance. Moreover, the results of average solutions which ABCPUD-MKP produces on 5.250.00, 5.250.29, and 5.500.29 instances are only inferior to that of B&B-EA-MKP. Thus, ABCPUD-MKP is the most outstanding algorithm in terms of the solution quality.

Table 3: Solution comparison among GA [5], B&B-GA [7], Ant-knapsack [11], ACOMPD-MKP [12], ABC-MKP, and ABCPUD-MKP algorithms.

Algorithm	Solution	Instance					
		5.100.00	5.100.29	10.100.00	5.250.00	5.250.29	5.500.29
GA-MKP	Best	**24381**	59960	**23064**	59243	154668	299885
	Avg.	**24381 ± 0.0**	59960 ± 0.0	23050.2 ± 19.2	59211.7 ± 18.0	154626.2 ± 31.7	299842.7 ± 26.9
B&B-EA-MKP	Best	**24381**	59965	**23064**	**59312**	154668	**299904**
	Avg.	**24381 ± 0.0**	59965 ± 0.0	23059.1 ± 3.2	**59305.1 ± 20.7**	**154668 ± 0**	**299902.3 ± 5.1**
Ant-knapsack	Best	**24381**	59965	**23064**	—	—	—
	Avg.	24342 ± 29.3	59958 ± 8.4	23016 ± 42.2	—	—	—
ACOMPD-MKP	Best	**24381**	**60117**	**23064**	**59312**	**154735**	299853
	Avg.	24362.8 ± 25.5	59979.1 ± 34.4	23051.2 ± 16.7	59152 ± 67.3	154622.3 ± 48.3	299817.0 ± 24.5
ABC-MKP	Best	**24381**	59965	**23064**	59243	154668	299885
	Avg.	**24381 ± 0.0**	59961.0 ± 2.1	23059.1 ± 3.9	59215.2 ± 17.3	154637.4 ± 15.5	299804.4 ± 59.1
ABCPUD-MKP	Best	**24381**	**60117**	**23064**	**59312**	**154735**	**299904**
	Avg.	**24381 ± 0.0**	**60015.4 ± 62.6**	**23060.4 ± 4.1**	59222.6 ± 35.5	154649.2 ± 34.2	299848.0 ± 42.4

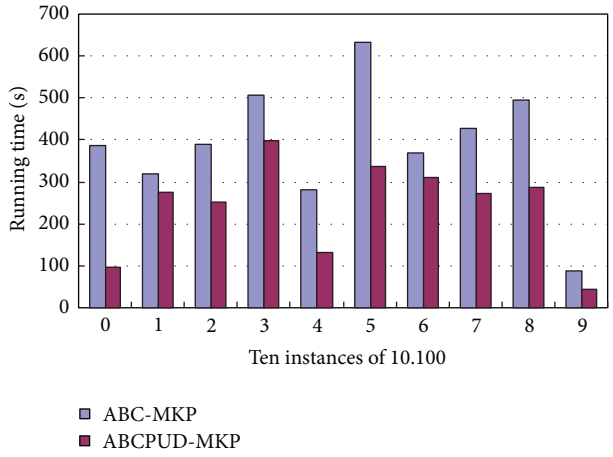

Figure 7: Comparison of the time performance on different instances.

5. Conclusion

In this paper, we propose an ABC algorithm, ABCPUD-MKP, to solve the 0-1 multidimensional knapsack problem effectively and efficiently. This algorithm differs from general ABC algorithms in the fact that a new communication way based on an inductive pheromone is introduced. More specifically, the new algorithm applies the pheromone updating and diffusion to extend the ABC algorithm. The new algorithm has been tested on many instances of the MKP problem with encouraging results: the new algorithm is superior in terms of the computational time on all instances compared to ABC-MKP algorithm, while it can also achieve best solution quality on all instances. We have also presented the performance comparison among GA-MKP, B&B-EA-MKP, Ant-knapsack, ACOMPD-MKP, ABC-MKP, and ABCPUD-MKP on some large problems and found that ABCPUD-MKP can find the best optimization solutions on all problems tested.

The new mechanism employs the chemical communication way to strength the collaboration among bees, which not only can keep the balance between exploitation and exploration, but also can effectively look into promising regions of the search space. Thus, the new mechanism is equally significant for ABC algorithms to tackle difficult combinatorial problems. Our future work is to extend our study to other NP-hard problems such as traveling salesman problem and Bayesian network structure learning.

Acknowledgments

This work is partly supported by the NSFC research program (61033004), 973 program (2011CB302703), and the Beijing Natural Science Foundation (4102010).

References

[1] A. Fréville, "The multidimensional 0-1 knapsack problem: an overview," *European Journal of Operational Research*, vol. 155, no. 1, pp. 1–21, 2004.

[2] A. V. Cabot, "An enumeration algorithm for knapsack problems," *Operations Research*, vol. 18, pp. 306–311, 1970.

[3] W. Shih, "A branch and bound method for the multiconstraint zero-one knapsack problem," *Journal of the Operations Research Society*, vol. 30, pp. 369–378, 1979.

[4] D. Bertsimas and R. Demir, "An approximate dynamic programming approach to multidimensional knapsack problems," *Management Science*, vol. 48, no. 4, pp. 550–565, 2002.

[5] P. C. Chu and J. E. Beasley, "A genetic algorithm for the multidimensional knapsack problem," *Journal of Heuristics*, vol. 4, no. 1, pp. 63–86, 1998.

[6] J.-C. Bai, H.-Y. Chang, and Y. Yi, "An partheno-genetic algorithm for multidimensional knapsack problem," in *Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC '05)*, pp. 2962–2964, August 2005.

[7] J. E. Gallardo, C. Cotta, and A. J. Fernández, "Solving the multidimensional knapsack problem using an evolutionary algorithm hybridized with branch and bound," in *Proceedings of the 1st International Work-Conference on the Interplay Between Natural and Artificial Computation (IWINAC '05)*, pp. 21–30, June 2005.

[8] M. Kong and P. Tian, "Application of the particle swarm optimization to the multidimensional knapsack problem, Artificial Intelligence and Soft Computing," in *Proceedings of the 8th International Conference (ICAISC '06)*, pp. 1140–1149, 2006.

[9] M. Kong, P. Tian, and Y. Kao, "A new ant colony optimization algorithm for the multidimensional Knapsack problem," *Computers and Operations Research*, vol. 35, no. 8, pp. 2672–2683, 2008.

[10] P. H. Rafael and D. Nikitas, "On the Performance of the Ant Colony System for Solving the Multidimensional Knapsack Problem," in *Proceedings of the IEEE Pacific Rim Conference on Communications Computers and Signal Processing (PACRIM '03)*, pp. 338–341, August 2003.

[11] I. Alaya, C. Solnon, and K. Ghedira, "Ant algorithm for the multidimensional knapsack problem," in *Proceedings of International Conference on Bioinspired Methods and their Applications*, pp. 63–72, 2004.

[12] J. Z. Ji, Z. Huang, and C. N. Liu, "An ant colony optimization algorithm based on mutation and pheromone diffusion for the multidimensional knapsack problems," *Jisuanji Yanjiu yu Fazhan/Computer Research and Development*, vol. 46, no. 4, pp. 644–654, 2009.

[13] S. Sundar, A. Singh, and A. Rossi, "An artificial bee colony algorithm for the 0-1 multidimensional knapsack problem," *Communications in Computer and Information Science*, vol. 94, no. 1, pp. 141–151, 2010.

[14] S. Pulikanti and A. Singh, "An artificial bee colony algorithm for the quadratic knapsack problem," in *Proceedings of the International Conference on Neural Information Processing (ICONIP '09)*, vol. 5864 of *Lecture Notes in Computer Science*, pp. 196–205, 2009.

[15] H. K. Wei, J. Z. Ji, Y. F. Qin, Y. M. Wang, and C. N. Liu, "A novel artificial bee colony algorithm based on attraction pheromone for the multidimensional knapsack problems, artificial," in *Proceedings of the Artificial Intelligence and Computational Intelligence (AICI '11) (part II)*, vol. 7003 of *Lecture Notes in Computer Science*, pp. 1–10, 2011.

[16] D. karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. TR06, Computer Engineering Department, Erciyes University, Kayseri, Turkey, 2005.

[17] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied Soft Computing Journal*, vol. 8, no. 1, pp. 687–697, 2008.

[18] W.-F. Gao and S.-Y. Liu, "A modified artificial bee colony algorithm," *Computers and Operations Research*, vol. 39, no. 3, pp. 687–697, 2012.

[19] X. H. Yan, Y. L. Zhu, W. P. Zou, and L. Wang, "A new approach for data clustering using hybrid artificial bee colony algorithm," *Neurocomputing*, vol. 97, pp. 241–250, 2012.

[20] Q.-K. Pan, M. Fatih Tasgetiren, P. N. Suganthan, and T. J. Chua, "A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem," *Information Sciences*, vol. 181, no. 12, pp. 2455–2468, 2011.

[21] Q. K. Pan, L. Wang, K. Mao, J. H. Zhao, and M. Zhang, "An effective artificial bee colony algorithm for a real-world hybrid flowshop problem in steelmaking process," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 2, pp. 307–322, 2013.

[22] S. N. Omkar, J. Senthilnath, R. Khandelwal, G. Narayana Naik, and S. Gopalakrishnan, "Artificial Bee Colony (ABC) for multi-objective design optimization of composite structures," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 489–499, 2011.

[23] D. Karaboga, B. Basturk, and C. Ozturk, "Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks," *Modeling Decisions for Artificial Intelligence*, vol. 4617, pp. 318–329, 2007.

[24] P. Y. Kumbhar and P. S. Krishnan, "Use of Artificial Bee Colony (ABC) algorithm in artificial neural network synthesis," *International Journal of Advanced Engineering Sciences and Technologies*, vol. 11, no. 1, pp. 162–171, 2011.

[25] C. Xu and H. Duan, "Artificial bee colony (ABC) optimized edge potential function (EPF) approach to target recognition for low-altitude aircraft," *Pattern Recognition Letters*, vol. 31, no. 13, pp. 1759–1772, 2010.

[26] B. Adil, Ö. Lale, and T. Pınar, *Artificial bee colony algorithm and its application to generalized assignment problem, swarm intelligence: focus on ant and particle swarm optimization*, Book edited by: F. T. S. Chan and M. K. Tiwari, Itech Education and Publishing, Vienna, Austria, 2007.

[27] P. Mukherjee and L. Satish, "Construction of equivalent circuit of a single and isolated transformer winding from FRA data using the ABC algorithm," *IEEE Transactions on Power Delivery*, vol. 27, no. 2, pp. 963–970, 2012.

[28] N. Todorovic and S. Petrovic, "Bee colony optimization algorithm for nurse rostering," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 43, no. 2, pp. 467–473, 2013.

[29] D. Karaboga and B. Akay, "A comparative study of Artificial Bee Colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.

[30] A. Singh, "An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem," *Applied Soft Computing Journal*, vol. 9, no. 2, pp. 625–631, 2009.

[31] A. Singh and A. K. Gupta, "Two heuristics for the one-dimensional bin-packing problem," *OR Spectrum*, vol. 29, no. 4, pp. 765–781, 2007.

[32] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 26, no. 1, pp. 29–41, 1996.

[33] C. Blum and M. Dorigo, "The Hyper-Cube Framework for Ant Colony Optimization," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 34, no. 2, pp. 1161–1172, 2004.

[34] M. F. Ali and E. D. Morgan, "Chemical communication in insect communities: a guide to insect pheromones with special emphasis on social insects," *Biological Reviews of the Cambridge Philosophical Society*, vol. 65, no. 3, pp. 227–247, 1990.

[35] R. K. VanderMeer, M. D. Breed, K. E. Espelie, and M. L. Winston, *Pheromone Communication in Social Insects*, Westview Press, 1998.