



Memetic clonal selection algorithm with EDA vaccination for unconstrained binary quadratic programming problems

Yiqiao Cai^a, Jiahai Wang^{a,*}, Jian Yin^a, Yalan Zhou^b

^a Department of Computer Science, Sun Yat-sen University, No. 132, Waihuan East Road, Guangzhou Higher Education Mega Center 510006, PR China

^b Information Science School, Guangdong University of Business Studies, No. 21, Chisha Road, Guangzhou 510320, PR China

ARTICLE INFO

Keywords:

Memetic clonal selection algorithm
Estimation of distribution
Fitness uniform selection
Tabu search
Unconstrained binary quadratic programming problem

ABSTRACT

This paper presents a memetic clonal selection algorithm (MCSA) with estimation of distribution algorithm (EDA) vaccination, named MCSA-EDA, for the unconstrained binary quadratic programming problem (UBQP). In order to improve the performance of the conventional clonal selection algorithm (CSA), three components are adopted in MCSA-EDA. First, to compensate for the absence of recombination among different antibodies, an EDA vaccination is designed and incorporated into CSA. Second, to keep the diversity of the population, a fitness uniform selection scheme (FUSS) is adopted as a selection operator. Third, to enhance the exploitation ability of CSA, an adaptive tabu search (TS) with feedback mechanism is introduced. Thus, MCSA-EDA can overcome the deficiencies of CSA and further search better solutions. MCSA-EDA is tested on a series of UBQP with size up to 7000 variables. Simulation results show that MCSA-EDA is effective for improving the performance of the conventional CSA and is better than or at least competitive with other existing metaheuristic algorithms.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Given a symmetric rational $n \times n$ matrix $Q = (q_{ij})$, and a binary vector $X = (x_1, \dots, x_i, \dots, x_n)$, where $x_i \in \{0, 1\}$, the objective of the unconstrained binary quadratic programming problem (UBQP) is to maximize the following function:

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

This problem is also known as the (unconstrained) quadratic bivalent programming problem, the (unconstrained) quadratic zero-one programming problem, or the (unconstrained) quadratic (pseudo-) Boolean programming problem (Beasley, 1998; Merz & Katayama, 2004).

UBQP is known to be NP-hard (Michael & David, 1979). It is a unified model for a variety of combinatorial optimization problems and has many important applications (Kochenberger, Glover, Alidaee, & Rego, 2004). There are numerous hard problems in many diverse areas have been formulated as UBQP, such as VLSI design, computer aided design, statistical mechanics, and so on. Moreover, many graph problems can be converted to UBQP, including the maximum clique problem, maximum cut problem, maximum

vertex packing problem, minimum covering problem, and graph coloring problem (Kochenberger et al., 2004). Algorithms for UBQP also can be utilized to solve these problems. Therefore, solution approaches for UBQP have been an active research area for many years due to the computational challenge and the vast potential applications (Beasley, 1998; Kochenberger et al., 2004; Michael & David, 1979).

Several exact methods have been developed for UBQP in the literature (Helmberg & Rendl, 1998, 1995). However, since UBQP is known to be NP-hard, the problems with large size are proved to be intractable for these exact approaches and only the problems with small size can be solved by them. For UBQP with medium and large size, different kinds of metaheuristic algorithms have been proposed and have found good solutions, such as tabu search (TS) (Beasley, 1998; Glover, Kochenberger, & Alidaee, 1998), simulate annealing (Alkhamis, Hasan, & Ahmed, 1998; Beasley, 1998; Katayama & Narihisa, 2001), scatter search (Amini, Alidaee, & Kochenberger, 1999), hybrid genetic algorithm (Merz & Freisleben, 1999), evolutionary heuristic (Lodi, Allemand, & Liebling, 1999) and so on.

In order to solve UBQP with larger size and higher density, some sophisticated approaches are proposed recently. Merz and Katayama (2004) proposed a powerful memetic algorithm (MA-MK) for UBQP with up to 2500 variables. Palubeckis (2004) proposed five different multistart tabu search (MST1-5) algorithms for UBQP with up to 6000 variables. After that, Palubeckis (2006) also

* Corresponding author.

E-mail address: wjiahai@hotmail.com (J. Wang).

proposed an iterated tabu search (ITS) for UBQP with up to 7000 variables. Most recently, Pardalos, Prokopyev, Shylo, and Shylo (2008) proposed a heuristic method based on the global equilibrium search framework for UBQP.

In recent years, artificial immune systems (AISs) have received a large amount of interest from researchers and industrial sponsors, and have been applied to solve the problems from many areas, such as mathematics, engineering, and information technology (Cutello, Nicosia, Pavone, & Timmis, 2007; Hart & Timmis, 2008). AISs are adaptive systems inspired by the theoretical immunology and observed immune functions, principles and models (de Castro & Timmis, 2002). They utilize the mechanism of vertebrate immune system in terms of the model of information processing and design new intelligent algorithms to solve problems. Most immune system-inspired optimization algorithms are based on the applications of the clonal selection principle (Cutello et al., 2007; Hart & Timmis, 2008).

This paper focuses on the study of the clonal selection algorithms (CSAs), which are a special class of AISs. In most of CSAs, especially the pure CSAs, there is only a simple mechanism to carry out the search through cloning, hypermutation and selection operator. However, there are some deficiencies in CSA to make its search mechanism inefficient when searching in complex problem spaces. Firstly, there is no any recombination operator in CSA, which will lead to the absence of any cooperation and communication among different antibodies. Secondly, the hypermutation operator in CSA is always implemented in a random way. The random mutation is lack of the capability of capturing the hits of finding optimum solution from the global statistical information. As a result, the hypermutation operator may be not effective for guiding the search when dealing with complex problems. Thirdly, the selection operator in CSA is carried out in a deterministic and greedy way. It will lead to the decreasing of the genetic diversity and get stuck in a local optimum.

Motivated by these observations above, a memetic CSA (MCSA) with estimation of distribution algorithm (EDA) vaccination, named MCSA-EDA, is proposed to improve the performance of CSA. In MCSA-EDA, three components are adopted. Firstly, an EDA vaccination is incorporated to compensate for the absence of recombination operator in most of CSAs. Secondly, a fitness uniform selection scheme (FUSS) (Hutter & Legg, 2006) is adopted as a selection operator to keep the diversity of the population. Finally, an adaptive TS with feedback mechanism is introduced into CSA to enhance the exploitation ability. The performance of MCSA-EDA is tested on a series of benchmark instances of UBQP. The comparisons are firstly carried out on the variants of MCSA to validate the effectiveness of the three components in MCSA-EDA. The results demonstrate that each component of MCSA-EDA is effective for improving the performance of CSA. Then, in order to show the effectiveness and robustness of MCSA-EDA, the comparisons of MCSA-EDA with other existing metaheuristic approaches, such as MA-MK, MST2 and ITS, are carried out. The simulation results show that MCSA-EDA is superior to MA-MK, and is competitive with ITS and MST2.

The main contributions of this paper can be summarized as follows. Firstly, the EDA vaccination, FUSS and adaptive TS are effectively incorporated into CSA, and thus a new clonal selection based algorithm, named MCSA-EDA, is proposed. Secondly, the simulation results on UBQP show that MCSA-EDA can effectively improve the performance of CSA and is a competitive approach for UBQP.

The remaining sections of this paper are organized as follows. Section 2 describes the related background of clonal selection, including the clonal selection theory and CSA. The proposed MCSA-EDA is described in detail in Section 3. In Section 4, the simulation results are presented. The conclusions and future research are given in Section 5.

2. Related background of clonal selection

2.1. Clonal selection theory

Clonal selection theory, developed by Burnet (1959, 1978), is the foundational principle of modern immunology. It interprets how an immune response is triggered when confronting an antigenic stimulus. The clonal selection theory states two main aspects: (1) only the lymphocytes specific to an activating antigen can be selected and cloned by the immune system, while others are selected against; (2) only antigen from a pathogen might stimulate a lymphocyte to proliferated and thus cause a destructive adaptive immune response (Burnet, 1959). Until now, there are a large part of AIS works based on the clonal selection theory.

The clonal selection principle is derived from the clonal selection theory. And it is used to explain the way of the immune system reacting to pathogens and improving the capability of recognizing and eliminating pathogens (Ada & Nossal, 1987). The clonal selection principle (De Castro & Von Zuben, 2002) is shown in Fig. 1 and the details are described as follows.

When a *B* cell's antigen receptor (antibody) recognizes an antigen with a certain affinity, it will be selected and become activated to proliferate. Proliferation of the *B* cell is an asexual and mitotic process. During this process, the *B* cells divide themselves into many clones, which are identical to the parent. Then, all the clones undergo the somatic hypermutation process. The hypermutation alters the characteristics of antibodies by introducing random changes to the genes of them. The hypermutation is inversely proportional to the affinity of the antibody. That is, the antibody with higher affinity will suffer less hypermutation. After hypermutation, the clones differentiate into plasma or memory cells. Plasma cells produce a large number of antigen-specific antibodies, and memory cells are circulating through the blood, lymph, and tissues. The memory cells can promote a rapid secondary response upon a subsequent encounter with the same or similar antigen (Burnet, 1959). The more comprehensive presentation of the clonal selection theory can be found in Burnet (1959, 1978) and De Castro and Von Zuben (2002).

2.2. CSA

CSAs are a special class of immune algorithms, which are inspired by the clonal selection principle (Burnet, 1959, 1978) of the human immune system. Recently, CSAs are very popular in the AIS community and bring about large numbers of research, such as optimization, learning, clustering and so on (de Castro & Timmis, 2002; De Castro & Von Zuben, 2002; Timmis, Andrews, Owens, & Clark, 2008). To date, numerous of population-based CSAs in the literature have been developed. In (Brownlee, 2008), there is a taxonomy of CSAs that divides the scope of such algorithms into six groups, including the artificial immune recognition system, the *B*-cell algorithm, the clonal selection algorithm (CLONALG), the immunological algorithm family, multi-objective immune system algorithm, and other for unclassified works. The details of the algorithms' classification can be found in (Brownlee, 2008).

The most popular CSA is CLONALG (De Castro & Von Zuben, 2002), which is proposed by Castro and Von Zuben. CLONALG is inspired by two important features of the affinity maturation in *B* cells (de Castro & Timmis, 2002). One is that the proliferation of each antibody is proportional to its affinity. That is, the higher the affinity is, the more clones are produced. Another is that the mutation suffered of an antibody is inversely proportional to its affinity. That is, the higher the affinity is, the lower mutation is suffered.

Most of CSAs are based on CLONALG, and a flowchart of the conventional CSA is shown in Fig. 2a. In the conventional CSA, the evolution process includes three main operators: cloning, hypermutation and selection. The brief description of CSA is as follows:

- (1) Related terms: *antigen*, *antibody* and *affinity*. In CSA, the antigen represents the objective function that to be optimized and the antibody represents the solution candidate. The affinity of an antibody is used to measure the fitness of the solution relative to the objective function.

- (2) Clonal selection operators.

- (a) *Cloning*: The cloning operator clones offspring identical to the parent for the antibody with high affinity in the current population. The cloning number for the antibody can be determined in two ways: static cloning and proportional cloning.
- (b) *Hypermutation*: The hypermutation operator generates the matured clone population by performing the mutation that inversely proportional to the affinity of the antibodies. The higher the affinity is, the lower the mutation is. Additional, the hypermutation operator is always

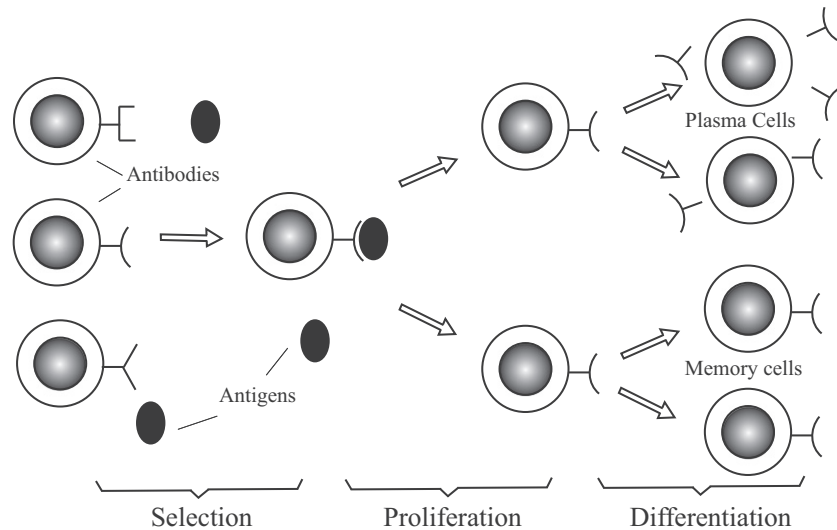


Fig. 1. Clonal selection principle.

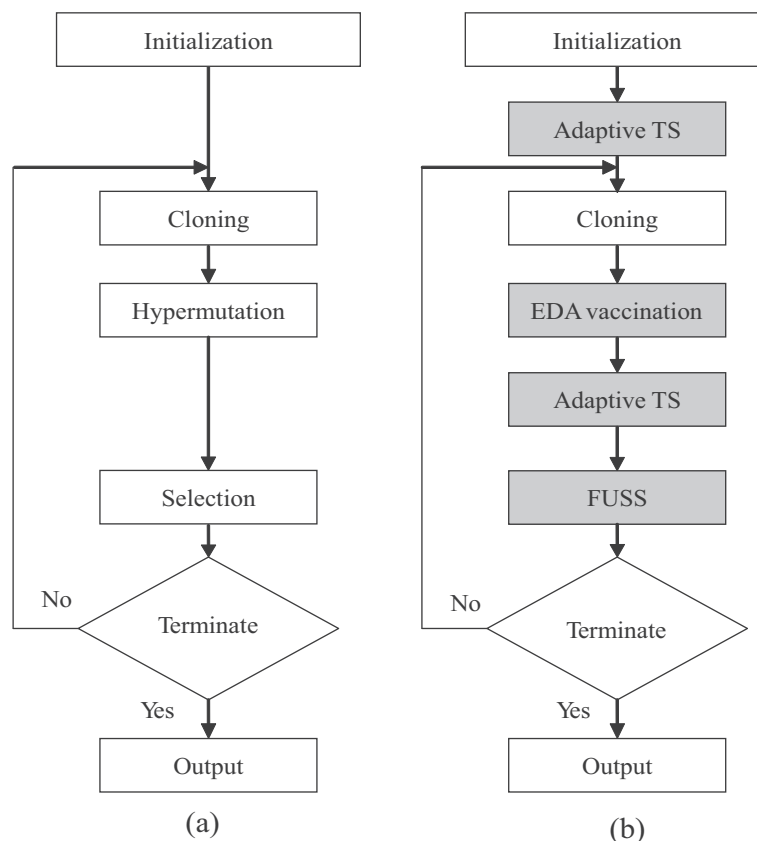


Fig. 2. The flowchart of (a) the conventional CSA and (b) MCSA-EDA.

carried out in a random way that introduces the random changes into the genes to expect to increase the affinity of the clonal antibody.

- (c) *Selection*: After hypermutation, the affinity of each antibody is recalculated. Then the antibodies with higher affinity in the combined population of parent and offspring are selected to constitute the new population. The selection operator in most of the CSAs is carried out in the deterministic and greedy way.

From the flowchart of the conventional CSA shown in Fig. 2a, CSA is characterized as an evolutionary-like algorithm, but it has the distinct characteristics from the evolutionary algorithms (EAs). CSA has the proportionate cloning operator and inversely proportional hypermutation operator according to the affinity of the antibody. It also has the ability of locating multiple optimal and keeping a diverse set of local optimal solutions. In essential, CSA and EAs are similar in the coding schemes and evaluation functions, but different in the evolutionary search from the viewpoint of inspiration, vocabulary, and fundamentals (De Castro & Von Zuben, 2002).

3. MCSA-EDA

In this section, a new method, named MCSA-EDA, is proposed to improve the performance of CSA. The flowchart of MCSA-EDA is shown in Fig. 2b. Compared with the conventional CSA, MCSA-EDA is different from it in three aspects: (1) an EDA vaccination is adopted as a mutation and recombination operator to replace the hypermutation operator; (2) a new FUSS selection operator is introduced to replace the deterministic selection operator; (3) an adaptive TS is incorporated as a neighborhood search approach to enhance the exploitation ability.

In the following subsections, the three components are detailed and discussed from the following issues: (1) how to utilize the global statistical information and knowledge from the searching history; (2) how to keep the diversity of the population; (3) how to balance the exploration and exploitation during the search.

3.1. EDA vaccination

Vaccination is a way to stimulate the immune system develops resistance against a pathogen (Burnet, 1959, 1987). During the process of vaccination, vaccine is injected into the body to cause the immune system with false infection. The immune system can learn and memorize these threatening antigens after vaccinating. In AIS, vaccines can be regarded as *a priori* knowledge about a certain disease.

Inspired by the mechanism of vaccination, an EDA vaccination is proposed. In the EDA vaccination, the global statistical information obtained from the previous search is used to extract vaccines and then vaccinate the clone antibodies, which is similar to the techniques of vaccine inoculation. By means of the EDA vaccination, the antibodies can fall in or close to a promising area with a higher probability.

- (1) *EDA*: EDA is a population-based probabilistic search algorithm framework based on modeling promising solutions by estimating their probability distribution. After that, it uses the constructed model to guide the exploration of the search space (Pelikan, Goldberg, & Lobo, 2002). In EDAs, there is a different way to generate offspring. They have neither crossover nor mutation operator. Instead, new individuals are generated by sampling the probability distribution, which is estimated from the selected promising solutions

from the searching history. Thus, these EDAs have a theoretical foundation in probability theory. The main loop of EDA consists of three principal stages: (a) selecting the best individuals according to some criteria from current population; (b) using the selected individuals in the first stage to update or recreate the solution distribution model; (c) sampling the distribution model to generate new offspring. Recently, the use of EDAs in genetic and evolutionary computation has become very popular. Especially, in our previous work, the idea of EDA is incorporated into the particle swarm optimization (Wang, Kuang, Xu, & Zhou, 2009; Wang, Zhang, Zhou, & Yin, 2008) and Hopfield neural network (Wang, 2008; Wang, Zhou, Yin, & Zhang, 2009), respectively. The simulation results have demonstrated the effectiveness of these proposed algorithms for solving the combinatorial optimization problems. A more comprehensive presentation of the EDAs field can be found in (Larranaga & Lozano, 2002; Pelikan et al., 2002).

- (2) *EDA vaccination*: The EDA vaccination guides the search through a probability distribution model and is implemented through two operators: *updating vaccines* and *vaccination*. The former is used for updating the probability distribution model, and the latter is used for vaccinating population according to the probability distribution model. In the EDA vaccination, the vaccines are produced by extracting the global statistical information from the searching history, and then represented as a probability model. The vaccines use a probability vector $P = (p_1, \dots, p_j, \dots, p_n)$ to characterize the distribution of promising solutions in the search space, where p_j is the probability that the value of the j th position of a promising solution is 1. And each antibody in current population is represented as $X^i = \{x_1^i, \dots, x_j^i, \dots, x_n^i\}$, where $i = 1, 2, \dots, N$.

The procedure of the EDA vaccination is described below:

- (a) *Updating vaccines operator*: Updating vaccines begins with extracting information and knowledge from the most promising antibodies. The promising antibodies are selected from the current population based on a certain criterion. Then the extracted information is used to update the probability model. The details of updating vaccines operator are described as follows:

Updating vaccines operator

- Step 1*: Select the antibodies satisfied the *selection criterion*, the number of the selected antibodies is denoted as SN ;
- Step 2*: Use Eq. (2) to update the probability model with the probability vector of the selected antibodies.

$$p_j = (1 - \lambda)p_j + \lambda \frac{\sum_{i=1}^{SN} x_j^i}{SN}, \quad (2)$$

where $j = 1, 2, \dots, n$, $\lambda \in (0, 1]$ is the learning rate.

Noted that the updating operator of the probability vector P is similar to the PBIL algorithm (Baluja, 1994). And the probability vector is initialized as $P = (0.5, 0.5, \dots, 0.5)$ in MCSA-EDA.

The validity of vaccines is critical to the performance of the algorithm. If the selection of a vaccine is wrong or inaccuracy, the vaccination will hold back the searching actions, and even exert a negative influence. Therefore, the *selection criterion* in the updating vaccines operator should correctly reflect the information of local optimum and guide the search. In MCSA-EDA, the *selection criterion* is based on the average hamming distance in current population.

Once the average distance of an antibody to other antibodies is greater than the average distance of the whole population, the antibody will be selected. By means of this *selection criterion*, the diversity of the clones can be kept after the vaccination operator.

(b) *Vaccination operator*: The vaccination operator is used to generate offspring by modifying some genes of the antibody according to the updated probability vector P . The process of the vaccination operator for X^i in binary 0–1 search space is described as follows:

Vaccination operator

Set $distance = 0, j = 1$

Do

If $x_j = 1$ then

If $rand() > p_j$ then

$x_j = 0; distance = distance + 1;$

End if

Else

If $rand() < p_j$ then

$x_j = 1; distance = distance + 1;$

End if

End if

$j = (j + 1) \bmod n;$

Until $distance > D^i$

In the vaccination operator, D^i means the hamming distance between the antibody X^i before and after vaccinating. D^i is calculated as follows:

$$D^i = n \times Rate^i \quad (3)$$

where $Rate^i$ means the mutation rate for X^i .

Each antibody is vaccinated based on the updated vaccines (i.e., the probability vector P). And the amount of vaccines for each antibody is inversely proportional to its affinity. The higher the affinity is, the smaller the amount of vaccines is. In other word, if the affinity of X^i is higher, the $Rate^i$ is smaller, and verse vice. The mutation rate for X^i is calculated by Eq. (4).

$$Rate^i = \alpha_{\min} + \frac{f_{\max} - f_i}{f_{\max} - f_{\min}} \times (\alpha_{\max} - \alpha_{\min}) \quad (4)$$

where $\alpha_{\max}, \alpha_{\min}$ are the maximum and minimum value of the mutation rate respectively, f_{\max}, f_{\min} are the maximum and minimum affinity value in current population respectively, and f_i is the affinity value of X^i .

In the EDA vaccination, the vaccination operator is realized by combining the global statistical information and location information of solutions found so far. This operator is akin to the guided mutation (GM) operator proposed in (Zhang, Sun, & Tsang, 2005; Zhang, Sun, Xiao, & Tsan, 2007), but there are several differences between them. Firstly, in the EDA vaccination, the mutation rate is inversely proportional to the affinity of the antibody. That is, the antibody with different affinity has different mutation rate. But in GM, the mutation rate is fixed for every individual. Secondly, the distance between the offspring and the parent can be calculated in advance according to Eq. (3) in the EDA vaccination, but uncertain in the GM before mutating.

Remark 1. The hypermutation in the conventional CSA is carried out in a random way and does not use the global statistical information. It will lead to that the hypermutation lacks the capability of capturing the hits of finding optimum solution from the global information. However, in the EDA vaccination, the mutation is guided by the probability vector. The probability vector

is learned and updated at each generation for modeling the distribution of promising solutions. And more importantly, in the EDA vaccination, on the one hand, the information of the antibodies can be cooperated and communicated through the updating vaccines operator, on the other hand, the information from the individual solution and the global information extracted from the searching history can be combined through the vaccination operator to generate the offspring. Therefore, the EDA vaccination can be regarded as a recombination operator. In this sense, the EDA vaccination is a complement for CSAs, especially some pure CSAs, which are lack of recombination mechanism. Recently, the recombination operator is also adopted in some clonal selection based memetic algorithms (MAs), such as clonal selection based memetic algorithm (Yang, Sun, Lee, Qian, & Liang, 2008), Lamarckian clonal selection algorithm (Yang, Gong, Jiao, & Zhang, 2008). However, the recombination operators in these existing works do not use the global information.

Remark 2. The mechanism of vaccination is employed in several previous researches (Jiao & Wang, 2000; Qi, 2007; Woldemariam & Yen, 2009). In (Jiao & Wang, 2000), a novel genetic algorithm based on immunity (IGA) is proposed. In IGA, the vaccines are extracted from the prior knowledge of the pending problem for the traveling salesman problem. And for the function optimization, the vaccines are produced from the optimal solution in the current population. In (Woldemariam & Yen, 2009), a vaccine-enhanced artificial immune system (Vaccine-AIS) is proposed. The vaccines in Vaccine-AIS are produced by generating random points that cover the whole decision space. A pheromone based dynamic vaccination for immune algorithm (PHDV-CSA) is also propose in (Qi, 2007). PHDV-CSA employs a dynamic vaccine updated based on pheromone deposition and volatilization mechanism of ant colonies. Compared with IGA and Vaccine-AIS, the vaccines in the EDA vaccination are produced by extracting the global statistical information from the previous search. That is, the EDA vaccination uses the global information to guide the search, whereas IGA and Vaccine-AIS only use the local characteristics and knowledge. And compared with PHDV-CSA, the EDA vaccination is different from it in some aspects. Firstly, PHDV-CSA is proposed for the function optimization, and the EDA vaccination is proposed for the combinatorial optimization problems. Secondly, PHDV-CSA is based on the ant colony algorithm, which ignores the locations of the best individual solutions found so far, while the EDA vaccination directly takes advantage of the solutions in the current population. Thirdly, the vaccination operator of PHDV-CSA and the EDA vaccination are distinct in the extracting the vaccines, updating vaccines and vaccination operator.

3.2. FUSS

Motivated by a universal similarity relation on the individuals, Hutter and Legg (2006) proposed a new fitness uniform selection scheme (FUSS), which generates selection pressure toward sparsely populated fitness regions, not necessarily toward higher fitness. FUSS can automatically create a suitable selection pressure and preserve genetic diversity better than other selection schemes, such as proportionate, truncation, ranking, and tournament selection. The more details of FUSS can be found in (Hutter & Legg, 2006).

In most of CSAs, the selection operator always selects the best antibody from the combined population of parent and offspring. This deterministic and greedy way will lead to the decreasing of the genetic diversity in the population after the selection. Furthermore, it often prevents the algorithm from escaping local minima at the early stages of search and leads to a premature convergence.

In order to overcome the deficiency of the deterministic and greedy selection operator, FUSS is adopted as a selection operator in MCSA-EDA. FUSS can generate selection pressure toward the sparsely populated fitness regions, which bounds the number of similar antibodies in the population after selection (Hutter & Legg, 2006).

FUSS in MCSA-EDA is described as follows:

FUSS
<p>Step 1: Identify the lowest and highest affinity in the current population, denoted as f_{low} and f_{high} respectively, and set the elite population $Pop = \phi$;</p> <p>Step 2: Copy the best antibody in current population to Pop;</p> <p>Step 3: Generate an affinity value f uniformly from the interval $[f_{low}, f_{high}]$;</p> <p>Step 4: If the antibody X^i in current population with the affinity nearest to f and does not have a copy in Pop, then copy X^i to Pop; else go to step 3;</p> <p>Step 5: If the Pop is not filled, go to step 3, else return Pop.</p>

3.3. Adaptive TS

TS is a metaheuristic algorithm and has been shown to be an effective and efficient scheme for combinatorial optimization. It combines a hill climbing search strategy and a heuristics to avoid the stops at local optima points and the revisiting of the previous points (Glover & Laguna, 1997). Up to the present, a larger number of applications like job shop scheduling, graph coloring, TSP have been successfully solved by TS. The more comprehensive review of TS can be found in (Glover & Laguna, 1997).

In MCSA-EDA, an adaptive TS is incorporated into CSA to enhance the exploitation ability. Thus, CSA can be used to maintain a diverse set of local optimal solutions and explore the local optimum, while the adaptive TS can be used for exploitation. Therefore, the combination of CSA and TS can balance the exploration and exploitation of MCSA-EDA.

The adaptive TS adopted in MCSA-EDA is modified from a simple TS proposed by Beasley (1998), which has been proven to be a very effective approach for UBQP (Palubeckis, 2004, 2006). Moreover, a feedback mechanism is adopted into TS to automatically adjust the tenure time. From the fitness landscape analysis and local search escape analysis in Merz and Katayama (2004) and Merz (2004), the local optima of the UBQP instances are very close in terms of the hamming distance. And after the simple recombination schemes and local search, there are very large basins of attractions to cause the revisit of the local optima very often (Merz & Katayama, 2004; Merz, 2004). Therefore, the feedback mechanism in the adaptive TS is adopted to reduce the probability of rediscovery the local optimum by adjusting the tenure time T .

The adaptive TS consists of two main components: *TS procedure* and *feedback mechanism*. The complete procedure of the adaptive TS is described as follows: firstly, all the antibodies in current population are improved by the TS procedure; then, the T is adjusted through the feedback mechanism. The details of the TS procedure and feedback mechanism are described below respectively.

- (1) *TS procedure*: In order to accelerate the evaluation of the solutions after flipping, two special formulas for UBQP are adopted in the TS procedure (Merz & Katayama, 2004). First, the gain of flipping the k th bit in the current solution is calculated in linear time using the following formula (Merz & Katayama, 2004):

$$gain_k(X) = q_{kk}(1 - 2x_k) + 2 \sum_{i=1, i \neq k}^n q_{ik}x_i(1 - 2x_k) \quad (5)$$

The $gain_k(X)$ means the gain value in the objective function after the k th bit has been flipped in the current solution. All the gains for the current solution are calculated by Eq. (5) and stored during the search process. Then, find the bit which has not been taboo (or satisfies with the aspiration criteria) and with the maximum gain value, and flip the bit, assumed to be the k th bit. After flipping the k th bit, the new gain for each bit of the solution is updated efficiently by the following formula (Merz & Katayama, 2004):

$$gain_i(X) = \begin{cases} -gain_i(X), & \text{if } i = k \\ gain_i(X) + 2q_{ik}(1 - 2x_i)(2x_k - 1), & \text{otherwise} \end{cases} \quad (6)$$

The TS procedure is shown as follows:

TS procedure
<p>Set $count = 0$, $t_i = 0$, $i = 1, 2, \dots, n$;</p> <p>Set T and $Maxcount$;</p> <p>$f_{current} = f_{best} = f(X)$; $X_{best} = X$;</p> <p>Calculate the gain for each bit of X using Eq. (5);</p> <p>While (True)</p> <p>{</p> <p>$v = \arg \max_i \{gain_i(X) \mid T_i = 0 \text{ or } f_{current} + gain_i(X) > f_{best},$</p> <p>$i = 1, 2, \dots, n\}$;</p> <p>$x_v = 1 - x_v$; $t_v = T$;</p> <p>$f_{current} = f_{current} + gain_v(X)$;</p> <p>Update $gain_i(X)$ for each bit of X using Eq. (6);</p> <p>If $f_{current} > f_{best}$ then</p> <p>$count = 0$;</p> <p>Invoke Local Search procedure;</p> <p>$X_{best} = X$; $f_{best} = f_{current}$;</p> <p>Else</p> <p>$count = count + 1$;</p> <p>End if</p> <p>For $i = 1$ to n then</p> <p>If $t_i > 0$ then $t_i = t_i - 1$;</p> <p>End for</p> <p>If $count > Maxcount$ then</p> <p>Break;</p> <p>End if</p> <p>};</p> <p>Return f_{best} and X_{best};</p>

In TS, the local search (LS) procedure is used to attempt to improve the solution further by examining all possible moves regardless of their tabu status and making any improvement of the solution (Beasley, 1998). The details of the LS procedure can be found in Beasley (1998) and Palubeckis (2006).

- (2) *Feedback mechanism*: In TS, the setting of T is a difficult problem. On the one hand, if the T is set too small, the search trajectory will revisit the previously points very often. On the other hand, if the T is set too large, there is no move be allowed to come back after the initial phase. Therefore, we adopt a feedback mechanism to adjust the value of T in TS to reduce the probability of revisiting a local optimum. When the offspring rediscover the local optimum of the parent after executing the TS procedure, the feedback mechanism tries to improve differentiation by increasing T . The feedback mechanism tunes T in the following way: firstly, calculate the number of offspring with the equal affinity to the parent after executing the TS procedure, denoted as N_{redis} ; then, adjust T using Eq. (7).

$$T = T + T \times \frac{N_{redis}}{\sum_{i=1}^N N_{child}^i} \quad (7)$$

3.4. Complete procedure of MCSA-EDA

The complete procedure of MCSA-EDA can be summarized as follows:

MCSA-EDA
Step 1: Randomly initialize the population Pop with N antibodies in the n -dimensional decision space, denoted as $Pop = (X^1, X^2, \dots, X^i, \dots, X^N)$;
Step 2: Apply the Adaptive TS to each antibody X^i in Pop , and evaluate the affinity $f(X^i)$, where $i = 1, 2, \dots, N$.
Step 3: Clone N_{child}^i offspring for each antibody X^i in Pop , and the size of the clone population C is proliferated to $\sum_{i=1}^N N_{child}^i$;
Step 4: Inject the vaccines into all the clones in C with the Vaccination operator of the EDA vaccination , and thus produce the vaccinated population C^V ;
Step 5: Apply the Adaptive TS to every antibody of C^V , and obtain the new population C^{ATS} ;
Step 6: Apply the Updating vaccines operator of the EDA vaccination with C^{ATS} to update the probability model;
Step 7: Use the FUSS to select N antibodies from $C^{ATS} \cup Pop$ to form the new population Pop ;
Step 8: If the terminate condition is met, return and output the best antibody; otherwise, go to step 3.

4. Simulation results and discussions

In order to evaluate the performance of MCSA-EDA, MCSA-EDA is implemented to solve a series of benchmark problems of UBQP. In the following subsections, the benchmark datasets are firstly described. Then, the parameter settings for MCSA-EDA are discussed. Simulation results are presented finally. The simulation experiment includes two parts. Firstly, the variants of MCSA are compared to validate the effectiveness of each component of MCSA-EDA. Secondly, a comparison of MCSA-EDA with other metaheuristics, such as ITS (Palubeckis, 2006), MST2 (Palubeckis, 2004) and MA-MK (Merz & Katayama, 2004), is carried out to test the effectiveness and robustness of MCSA-EDA. All the simulations are carried out on the same PC environment (Intel Core2 2.4 GHz, 2 GB RAM).

4.1. Benchmark datasets

In the simulation, there are two groups of UBQP to be tested. The first group has 10 problems, which are the largest problem instances in the OR-Library. These 10 problems, firstly studied by Beasley (1998), are with 2500 variables and have density 10%, named b2500-1...b2500-10 in this paper. The second group has 21 problems, which are firstly introduced by Palubeckis (2006). These 21 problems are randomly generated with larger size from 3000 to 7000 and higher density from 50% to 100%. All nonzero coefficients of the objective function are drawn uniformly at random from the interval $[-100, 100]$ for all the problems. The main characteristics of the benchmark problems are shown in Table 1. The best known solutions for the problems in first group can be found in Palubeckis (2006). And the sources of the generator and input files to reproduce the second group problems can be obtained at the website (<http://www.soften.ktu.lt/~gintaras>).

4.2. Parameter settings

In MCSA-EDA, there are some parameters to be tuned, including: (1) the population size N , the number of clones for each

Table 1

Main characteristics of benchmark problems for UBQP.

Problem	Density	Linear and quadratic coefficients	Best known solution
b2500-1 to b2500-10	0.1	$[-100, 100]$	Refer to (Palubeckis, 2006)
p3000-1	0.5		3931583
p3000-2	0.8		5193073
p3000-3	0.8		5111533
p3000-4	1.0		5761822
p3000-5	1.0		5675625
p4000-1	0.5		6181830
p4000-2	0.8		7801355
p4000-3	0.8		7741685
p4000-4	1.0		8711822
p4000-5	1.0		8908979
p5000-1	0.5		8567067
p5000-2	0.8		10836019
p5000-3	0.8		10489137
p5000-4	1.0		12256474
p5000-5	1.0		12731803
p6000-1	0.5		11385007
p6000-2	0.8		14333855
p6000-3	1.0		16138892
p7000-1	0.5		14478676
p7000-2	0.8		18249948
p7000-3	1.0		20571136

antibody N_{child}^i ; (2) λ , α_{min} and α_{max} in the EDA vaccination; (3) T_{init} and $Maxcount$ in the adaptive TS.

In order to balance the computational cost and the performance of MCSA-EDA, a modest population size is set, that is, $N = 7$ and $N_{child}^i = 3$ for each antibody is adopted in this paper. Hence, there are $|C| = \sum_{i=1}^N N_{child}^i = 21$ offspring to be improved with the adaptive TS at each generation.

In the EDA vaccination, the mutation rate directly controls the distance between the parent and the offspring, and the learning rate λ balances the contributions between the old statistical information extracting from historical local optimal and the information of the local optimal in current population. Due to the similarity between the local optima and the large basins of attractions of the local optima in UBQP (Merz, 2004; Merz & Katayama, 2004), $\alpha_{min} = 0.1$ and $\alpha_{max} = 0.3$ are set for all the benchmark problems to keep adequate mutation strength for escaping from the local optima. And the learning rate λ is set to 0.1.

The setting of T_{init} is important for preventing cycling to revisit a previously encountered solution. In the adaptive TS, the T_{init} is set to 20, which is found to be good by Beasley (1998). And the $Maxcount$ is set to the size of the problem (i.e., n).

4.3. Comparison with variants of MCSA

To validate the effectiveness of each component in MCSA-EDA, three variants of MCSA are compared. A representative subset with different size and density are selected from the benchmark problems for comparison, including b2500-2, b2500-3, p3000-2, p3000-5, p4000-1, p4000-5, p5000-2, p6000-2 and p7000-1. A stopping rule based on the CPU clock is adopted. The time allocation for a run is set to 300, 800, 1500, 2500, 3500 and 4500 s for the problem with 2500, 3000, 4000, 5000, 6000 and 7000 variables, respectively. Each algorithm is independently run 10 times for each problem. The simulation results are shown in Tables 2–4. In Tables 2–4, “Best” means the deviation of the best solution value found by the algorithms from the best known solution, “Av.” means the deviation of the average solution value found by the algorithms from the best known solution, and “SR” means the

Table 2

Comparison of the results obtained by MCSA-R and MCSA-EDA.

Problem	Best known solution	MCSA-R		MCSA-EDA	
		Av.	SR	Av.	SR
b2500-2	1471392	0	1	0	1
b2500-3	1414192	0	1	0	1
p3000-2	5193073	0	1	0	1
p3000-5	5675625	257	0.3	68	0.8
p4000-1	6181830	0	1	0	1
p4000-5	8908979	904	0.5	402	0.8
p5000-2	10836019	726	0.2	350	0.4
p6000-2	14333855	557	0.3	487	0.3
p7000-1	14478676	1245	0.1	945	0.1
Average		409.89	0.60	250.22	0.71

Table 3

Comparison of the results obtained by MCSA-D and MCSA-EDA.

Problem	Best known solution	MCSA-D		MCSA-EDA	
		Av.	SR	Av.	SR
b2500-2	1471392	0	1	0	1
b2500-3	1414192	2	0.9	0	1
p3000-2	5193073	0	1	0	1
p3000-5	5675625	234	0.6	68	0.8
p4000-1	6181830	0	1	0	1
p4000-5	8908979	260	0.8	402	0.8
p5000-2	10836019	589	0.0	350	0.4
p6000-2	14333855	300	0.1	487	0.3
p7000-1	14478676	1000	0.1	945	0.1
Average		265	0.61	250.22	0.71

frequency of hitting the best known solution (success rate) measured in terms of the number of runs of an algorithm for each problem within the allocated time.

The following abbreviations represent the variants of MCSA considered:

- MCSA-R: MCSA-EDA with random hypermutation replacing the EDA vaccination.
- MCSA-D: MCSA-EDA with deterministic selection operator replacing FUSS.
- MCSA-RKLS: MCSA-EDA with randomized k -opt local search replacing the adaptive TS.

(1) *Comparison of MCSA-EDA and MCSA-R*: In order to validate the effectiveness of the EDA vaccination, MCSA-EDA is compared with MCSA-R. MCSA-R adopts a random hypermutation, which is used in most of the CSAs. The hypermutation in MCSA-R is carried out by flipping the bits randomly. The distance between the parent and offspring in MCSA-R is also inversely proportional to the affinity of the antibody. The results are shown in Table 2. It is interesting to find that the success rate obtained by MCSA-EDA is superior to or at least equal to that of MCSA-R for all the selected problems. Specifically, for p3000-5, p4000-5 and p5000-2, MCSA-EDA is better than MCSA-R in terms of the success rate. For the average value and the success rate on the whole, MCSA-EDA is also better than MCSA-R. It is due that the EDA vaccination can use the global statistical information to guide the mutation so that the offspring will fall in or close to the promising area, whereas the hypermutation only uses the stochastic mechanism to carry out the mutation randomly. The results demonstrate that the EDA vaccination is more effectively and powerful than the random hypermutation for MCSA.

(2) *Comparison of MCSA-EDA and MCSA-D*: FUSS is used to keep the diversity of the population in MCSA-EDA. In order to evaluate the effectiveness of FUSS, a comparison between MCSA-EDA and MCSA-D is carried out. The results are shown in Table 3. From Table 3, it shows that the performance of MCSA-EDA is competitive with that of MCSA-D for the smaller problems, but is better than that of MCSA-D for the larger problems, especially for p5000-2 and p6000-2. Moreover, MCSA-EDA is better than MCSA-D in terms of the average value and the success rate on the whole. The reason lies in that FUSS in MCSA-EDA can maintain genetic diversity and guide the search toward the sparsely populated fitness regions for exploration, whereas the selection operator in MCSA-D in a deterministic and greedy way may prevent the algorithm from escaping local minima at the early stages of search and lead to a premature convergence. In a word, the results in Table 3 indicate the effectiveness of FUSS for MCSA-EDA as compared to the deterministic and greedy selection operator.

(3) *Comparison of MCSA-EDA and MCSA-RKLS*: Recent studies have shown that the choice of the local search method significantly affects the performance of memetic algorithms (MAs) (Ong, 2006; Ong & Keane, 2004). For UBQP, the randomized k -opt local search (RKLS) is a very effective local search proposed by Merz and Katayama (2004) and has been used in many previous researches (Katayama & Narihisa, 2001; Merz & Freisleben, 2002; Merz & Katayama, 2004). In order to evaluate the effectiveness of the adaptive TS in MCSA-EDA, the MCSA with RKLS (named MCSA-RKLS) is proposed for comparison. The results are shown in Table 4. From Table 4, one can clearly find that MCSA-EDA obtains the best known solution for all the selected problems, whereas MCSA-RKLS only finds the best known solution for a few relatively simpler problems, such as b2500-2, b2500-3 and p4000-1. In addition, MCSA-EDA is much better than MCSA-RKLS in terms of the average value and the success rate on the whole. The results indicate that the adaptive TS is clearly and consistently superior to RKLS as a local search in MCSA-EDA for UBQP.

(4) *Conclusion remarks for the comparisons of MCSA's variants*: From the comparisons above, we can observe that MCSA-EDA outperforms all the variants of MCSA. In order to clearly demonstrate the different performances of the variants of MCSA for the selected problems, the success rate of each MCSA's variant for the selected problems in Tables 2–4 are summarized and plotted in Fig. 3. From Fig. 3, one can clearly note that the success rate obtained by MCSA-EDA is consistently better than or at least equal to that of all the variants of MCSA for all the selected problems. Moreover, a close inspection of Tables 2–4 finds that MCSA-EDA is better than all the variants of MCSA in terms of the average value on the whole. These results clearly demonstrate the effectiveness of each component (i.e., the EDA vaccination, FUSS and adaptive TS) adopted in MCSA-EDA. As a whole, MCSA-EDA can effectively improve the performance of CSA.

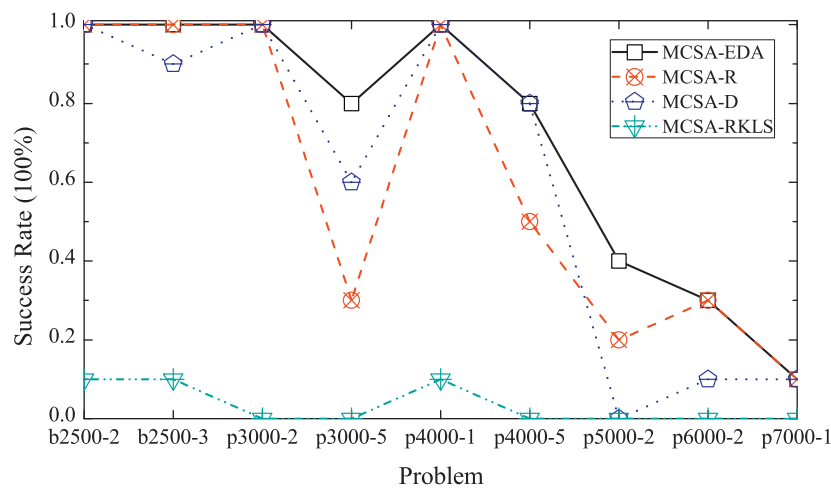
4.4. Comparison with other metaheuristic algorithms

In order to show the effectiveness and robustness of MCSA-EDA, we present a comparative performance against some metaheuristic algorithms specially designed for UBQP, i.e., MA-MK (Merz & Katayama, 2004), MST2 (Palubeckis, 2004) and ITS (Palubeckis, 2006). These algorithms have been proposed as one of the best heuristic approaches for UBQP. MA-MK is a population based algorithm. It uses RKLS as the neighborhood search approach and an innovative variation to generate a population of offspring solution.

Table 4

Comparison of the results obtained by MCSA-RKLS and MCSA-EDA.

Problem	Best known solution	MCSA-RKLS			MCSA-EDA		
		Best	Av.	SR	Best	Av.	SR
b2500-2	1471392	0	510	0.1	0	0	1
b2500-3	1414192	0	287	0.1	0	0	1
p3000-2	5193073	483	1025	0	0	0	1
p3000-5	5675625	1158	2922	0	0	68	0.8
p4000-1	6181830	0	3386	0.1	0	0	1
p4000-5	8908979	5781	8206	0	0	402	0.8
p5000-2	10836019	2531	7351	0	0	350	0.4
p6000-2	14333855	7895	17,031	0	0	487	0.3
p7000-1	14478676	11,844	15,456	0	0	945	0.1
Average		3299	6241.56	0.03	0	250.22	0.71

**Fig. 3.** The success rate of variants of MCSA for the selected problems.

MST2 and ITS belong to the class of multistart TS approaches. They both use a simple TS as the local search, but use different mechanisms to construct the starting solution. MST2 applies a constructive heuristic to certain instance of UBQP obtained by fixing the values of a subset of the variables. And ITS incorporates a solution perturbation operator by flipping a number of variables according to the coefficients of the linear part of the transformed problem instance.

For the comparison, MA-MK is rewritten in the C programming language, and the source codes of MST2 and ITS are downloaded from <http://www.soften.ktu.lt/~gintaras>. The settings of the experiments in this section are the same as that described in Section 4.3. And the parameters of MA-MK, MST2 and ITS are set to the values suggested in the original papers.

For all of the problems in the first group, the four compared algorithms can find the best known solution in all the runs. That is, the success rate of the compared algorithms for each problem is always 100%. Therefore, the results are not shown here because they cannot provide sufficiently strong comparisons for the algorithms. For the problems in the second group, the simulation results are shown in Tables 5 and 6. The meanings of “Best”, “Av.” and “SR” in Tables 5 and 6 are the same as those in Tables 2–4.

(1) *Comparison of quality of the final solution:* Table 5 shows the best and average values produced by MCSA-EDA, ITS, MST2 and MA-MK for the benchmark problems in the second group. From Table 5, one may note that for a few relatively

simpler problems like p3000-1, p3000-3 and p4000-1, all of the compared algorithms can obtain the best known solutions although differences in terms of average value. Substantial performance differences however, are noted for the rest of the more challenging problems with larger size and higher density. For these more difficult problems (e.g., p5000, p6000 and p7000), MCSA-EDA, ITS and MST2 significantly outperform MA-MK. MCSA-EDA is equal to ITS in terms of the best value, and is slightly inferior to ITS in terms of the average value on the whole. Compared with MST2, MCSA-EDA is better than MST2 in terms of the best value, while slightly worse than MST2 in terms of the average value on the whole.

It is interesting to find that out of the 21 benchmark instances, MCSA-EDA can obtain the best known solutions for 16 cases, and for the other 5 cases, MCSA-EDA also can obtain the best solutions reported in (Palubeckis, 2006). It suggests that MCSA-EDA is effective for solving UBQP with large size and high density.

The close inspection of Table 5 indicates that MCSA-EDA clearly and consistently outperforms MA-MK, and is competitive with ITS and MST2 in terms of the quality of the final solutions.

(2) *Comparison of success rate:* Table 6 shows the success rate obtained by the compared algorithms. One may note that the success rate decreases when the size and density of the problems increases from Table 6. This is expected

Table 5

Comparison of the results obtained by MCSA-EDA, ITS, MST2 and MA-MK for the 21 benchmark problems in the second group.

Problem	Best known solution	MCSA-EDA		ITS		MST2		MA-MK	
		Best	Av.	Best	Av.	Best	Av.	Best	Av.
p3000-1	3931583	0	0	0	0	0	0	0	1914
p3000-2	5193073	0	0	0	0	0	0	226	1746
p3000-3	5111533	0	0	0	143	0	0	0	2251
p3000-4	5761822	0	0	0	0	0	0	385	1236
p3000-5	5675625	0	68	0	143	0	151	1346	3051
p4000-1	6181830	0	0	0	0	0	0	0	764
p4000-2	7801355	0	51	0	702.8	0	145.5	2071	5916
p4000-3	7741685	0	6	0	10.8	0	0	54	3614
p4000-4	8711822	0	0	0	0	0	0	2455	3783
p4000-5	8908979	0	402	0	488.7	0	249.9	45	7225
p5000-1	8567067	7712	7827	7712	7771	7712	7817	13,532	17,718
p5000-2	10836019	0	350	0	337.7	582	582	9630	13,359
p5000-3	10489137	0	182	0	117.6	0	87	13,223	17,647
p5000-4	12256474	4156	4950	4156	4777	4799	5052	20,215	29,197
p5000-5	12731803	0	325	0	410	0	500	8634	14,445
p6000-1	11385007	31	91.2	31	31	31	31	10,223	16901.6
p6000-2	14333855	0	487	0	79.2	0	173.5	17,301	24728.4
p6000-3	16138892	5977	7971	5977	6916	5977	7770	22,900	30184.2
p7000-1	14478676	0	945	0	234	1461	1872	9523	19083.3
p7000-2	18249948	0	5744.9	0	1748.3	104	2438	29,418	34988.6
p7000-3	20571136	124,729	124,729	124,729	128,880	124,729	124,729	165,650	172547.8
Average		6791	7339	6791	7276	6924	7219	15,563	20,110

Table 6

Comparison of the success rate obtained by MCSA-EDA, ITS, MST2 and MA-MK for the 21 benchmark problems in the second group.

Problem	SR (100%)			
	MCSA-EDA	ITS	MST2	MA-MK
p3000-1	1.0	1.0	1.0	0.1
p3000-2	1.0	1.0	1.0	0.0
p3000-3	1.0	0.8	1.0	0.1
p3000-4	1.0	1.0	1.0	0.0
p3000-5	0.8	0.7	0.4	0.0
p4000-1	1.0	1.0	1.0	0.4
p4000-2	0.9	0.6	0.9	0.0
p4000-3	0.9	0.8	1.0	0.0
p4000-4	1.0	1.0	1.0	0.0
p4000-5	0.7	0.8	0.8	0.0
p5000-1	0.0	0.0	0.0	0.0
p5000-2	0.4	0.4	0.0	0.0
p5000-3	0.4	0.7	0.6	0.0
p5000-4	0.0	0.0	0.0	0.0
p5000-5	0.6	0.6	0.4	0.0
p6000-1	0.0	0.0	0.0	0.0
p6000-2	0.3	0.1	0.1	0.0
p6000-3	0.0	0.0	0.0	0.0
p7000-1	0.1	0.2	0.0	0.0
p7000-2	0.1	0.1	0.0	0.0
p7000-3	0.0	0.0	0.0	0.0
Average	0.53	0.51	0.49	0.03

because the problems with larger size and higher density become difficult to be solved from the fitness distance correlation analysis (Merz & Katayama, 2004). From Table 6, we can observe that MCSA-EDA, ITS and MST2 are much superior to MA-MK for all the problems. It is interesting to find that MCSA-EDA obtains the same success rate with ITS and MST2 for most of the problems except for some problems, such as p3000-5, p4000-2, p5000-2, p5000-3, p6000-2 and p7000-1. And the success rate obtained by MCSA-EDA is better than that of ITS and MST2 on the whole. The results also indicate that MA-MK for the problems with larger size and higher density is not as effective as for the problems with smaller size and lower density

(e.g., the high success rate for the first group problems), whereas MCSA-EDA, ITS and MST2 are seems to be more reliable and robust than MA-MK.

In summary, the results in Tables 5 and 6 clearly show that in terms of the average value and the success rate, MCSA-EDA is much superior to MA-MK and is competitive with ITS and MST2, which are two of the best heuristics for UBQP.

5. Conclusion and future research

In this paper, we present a novel MCSA-EDA to improve the performance of CSA and apply it to solve UBQP. In order to overcome the deficiencies in most of CSAs, three components are adopted in MCSA-EDA. Firstly, an EDA vaccination is proposed to compensate for the absence of recombination operator in most of CSAs. The EDA vaccination combines the global statistical information and location information of solutions found so far to guide the search. Secondly, a FUSS is adopted as a selection operator to keep the population diversity by overcoming the weaknesses of the deterministic selection mechanism. Finally, an adaptive TS with feedback mechanism is introduced to enhance the exploitation ability of CSA. Simulation results on UBQP show that MCSA-EDA is effective to improve the performance of CSA, and is better than or competitive with other existing metaheuristic algorithms, such as MA-MK, ITS and MST2.

MCSA-EDA has been shown as a powerful and effective combinatorial optimization framework. It should be pointed out that MCSA-EDA is not the best metaheuristic for UBQP. The main purpose of this paper is to study the effectiveness of the three components (i.e., the EDA vaccination, FUSS and adaptive TS) adopted in MCSA-EDA and show how they can improve the performance of CSA. In the future, we intend to further improve MCSA-EDA and apply it to other combinatorial optimization problems.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (60805026, 60905038, 61070076,

61033010), the Specialized Research Fund for the Doctoral Program of Higher Education (20070558052), and the Fundamental Research Funds for the Central Universities (10lgpy32).

References

- Ada, G., & Nossal, G. (1987). The clonal-selection theory. *Scientific American*, 257, 50–57.
- Alkhamis, T., Hasan, M., & Ahmed, M. (1998). Simulated annealing for the unconstrained quadratic pseudo-boolean function. *European Journal of Operational Research*, 108, 641–652.
- Amini, M., Alidaee, B., & Kochenberger, G. (1999). A scatter search approach to unconstrained quadratic binary programs. In D. Corne, M. Dorigo, & F. Glover (Eds.), *New ideas in optimization* (pp. 317–329). London, England: McGraw-Hill.
- Baluja, S. (1994). *Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning*. Pittsburgh, PA: School of Computer Science, Carnegie Mellon Univ.
- Beasley, J. (1998). *Heuristic algorithms for the unconstrained binary quadratic programming problem*. London: The Management School, Imperial College.
- Brownlee, J. (2008). Clonal selection as an inspiration for adaptive and distributed information processing. Ph.D. Dissertation. Swinburne University of Technology.
- Burnet, F. (1959). *The clonal selection theory of acquired immunity*. Cambridge University Press.
- Burnet, F. (1978). Clonal selection and after. In G. I. Bell, A. S. Perelson, & G. H. Pimbley, Jr. (Eds.), *Theoretical immunology* (pp. 63–85). New York: Marcel Dekker.
- Cutello, V., Nicosia, G., Pavone, M., & Timmis, J. (2007). An immune algorithm for protein structure prediction on lattice models. *IEEE Transactions on Evolutionary Computation*, 11, 101–117.
- De Castro, L., & Timmis, J. (2002). *Artificial immune systems: A new computational intelligence paradigm*. London, UK: Springer-Verlag.
- De Castro, L., & Von Zuben, F. (2002). Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6, 239–251.
- Glover, F., Kochenberger, G., & Alidaee, B. (1998). Adaptive memory tabu search for binary quadratic programs. *Management Science*, 44, 336–345.
- Glover, F., & Laguna, M. (1997). *Tabu search*. Springer.
- Hart, E., & Timmis, J. (2008). Application areas of AIS: The past, the present and the future. *Applied Soft Computing Journal*, 8, 191–201.
- Helmberg, C., & Rendl, F. (1998). Solving quadratic (0,1)-problems by semidefinite programs and cutting planes. *Mathematical Programming*, 82, 291–315.
- Hutter, M., & Legg, S. (2006). Fitness uniform optimization. *IEEE Transactions on Evolutionary Computation*, 10, 568–589.
- Jiao, L., & Wang, L. (2000). A novel genetic algorithm based on immunity. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 30, 552–561.
- Katayama, K., & Narihisa, H. (2001). On fundamental design of parthenogenetic algorithm for the binaryquadratic programming problem. In *Evolutionary computation 2001. Proceedings of the 2001 congress* (pp. 356–363).
- Katayama, K., & Narihisa, H. (2001). Performance of simulated annealing-based heuristic for the unconstrained binary quadratic programming problem. *European Journal of Operational Research*, 134, 103–119.
- Kochenberger, G., Glover, F., Alidaee, B., & Rego, C. (2004). A unified modeling and solution framework for combinatorial optimization problems. *OR Spectrum*, 26, 237–250.
- Larranaga, P., & Lozano, J. (2002). *Estimation of distribution algorithms: A new tool for evolutionary computation*. Kluwer Academic Publishers.
- Lodi, A., Allemand, K., & Liebling, T. (1999). An evolutionary heuristic for quadratic 0–1 programming. *European Journal of Operational Research*, 119, 662–670.
- Merz, P. (2004). Advanced fitness landscape analysis and the performance of memetic algorithms. *Evolutionary Computation*, 12, 303–325.
- Merz, P., & Freisleben, B. (1999). Genetic algorithms for binary quadratic programming. In *Proceedings of the 1999 international genetic and evolutionary computation conference (GECCO'99)* (pp. 417–424).
- Merz, P., & Freisleben, B. (2002). Greedy and local search heuristics for unconstrained binary quadratic programming. *Journal of Heuristics*, 8, 197–213.
- Merz, P., & Katayama, K. (2004). Memetic algorithms for the unconstrained binary quadratic programming problem. *BioSystems*, 78, 99–118.
- Michael, R., & David, S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. San Francisco: WH Freeman & Co.
- Ong, Y., Lim, M., Zhu, N., & Wong, K. (2006). Classification of adaptive memetic algorithms: A comparative study. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 36, 141–152.
- Ong, Y., & Keane, A. (2004). Meta-Lamarckian learning in memetic algorithms. *IEEE Transactions on Evolutionary Computation*, 8, 99–110.
- Palubeckis, G. (1995). A heuristic-based branch and bound algorithm for unconstrained quadratic zero-one programming. *Computing*, 54, 283–301.
- Palubeckis, G. (2004). Multistart tabu search strategies for the unconstrained binary quadratic optimization problem. *Annals of Operations Research*, 131, 259–282.
- Palubeckis, G. (2006). Iterated tabu search for the unconstrained binary quadratic optimization problem. *Informatica*, 17, 279–296.
- Pardalos, P., Prokopyev, O., Shylo, O., & Shylo, V. (2008). Global equilibrium search applied to the unconstrained binary quadratic optimization problem. *Optimization Methods & Software*, 23, 129–140.
- Pelikan, M., Goldberg, D., & Lobo, F. (2002). A survey of optimization by building and using probabilistic models. *Computational optimization and applications*, 21, 5–20.
- Qi, Y., Liu, F., & Jiao, L. (2007). Pheromone based dynamic vaccination for immune algorithms. *Lecture Notes in Computer Science*, 4490, 1053–1060.
- Timmis, J., Andrews, P., Owens, N., & Clark, E. (2008). An interdisciplinary perspective on artificial immune systems. *Evolutionary Intelligence*, 1, 5–26.
- Wang, J. (2008). Hopfield neural network based on estimation of distribution for two-page crossing number problem. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 55, 797–801.
- Wang, J., Kuang, Z., Xu, X., & Zhou, Y. (2009). Discrete particle swarm optimization based on estimation of distribution for polygonal approximation problems. *Expert Systems with Applications*, 36, 9398–9408.
- Wang, J., Zhou, Y., Yin, J., & Zhang, Y. (2009). Competitive hopfield network combined with estimation of distribution for maximum diversity problems. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, 39, 1048–1066.
- Wang, J., Zhang, Y., Zhou, Y., & Yin, J. (2008). Discrete quantum-behaved particle swarm optimization based on estimation of distribution for combinatorial optimization. In *Proceeding of IEEE congress on evolutionary computation* (pp. 897–904).
- Woldemariam, K., & Yen, G. (2009). Vaccine enhanced artificial immune system for multimodal function optimization. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, 40, 218–228.
- Yang, J., Sun, L., Lee, H., Qian, Y., & Liang, Y. (2008). Clonal selection based memetic algorithm for job shop scheduling problems. *Journal of Bionic Engineering*, 5, 111–119.
- Yang, J., Gong, M., Jiao, L., & Zhang, L. (2008). Improved clonal selection algorithm based on Lamarckian local search technique. In *Proceeding of IEEE congress on evolutionary computation* pp. 535–541.
- Zhang, Q., Sun, J., & Tsang, E. (2005). An evolutionary algorithm with guided mutation for the maximum clique problem. *IEEE Transactions on Evolutionary Computation*, 9, 192–200.
- Zhang, Q., Sun, J., Xiao, G., & Tsan, E. (2007). Evolutionary algorithms refining a heuristic: A hybrid method for shared-path protections in WDM networks under SRLG constraints. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, 37, 51–61.