http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 10 (1): 175-181, 2011
ISSN 1812-5638 / DOI: 10.3923/itj.2011.175.181

Bi-evolutionary Algorithm Simulating the Mechanism of Human Evolution and its Application in Knapsack Problem

Yan Taishan and Cui Duwu
School of Computer Science and Engineering,
Xi'an University of Technology, 710048, Xi'an, Shaanxi, People's Republic of China

Abstract: This study presents a Bi-Evolutionary Algorithm (BEA) simulating the mechanism of human evolution. Bi-Evolutionary Algorithm system includes knowledge subsystem and biological subsystem. They are mutually independent and interactive. Biological subsystem is used to solve the specific problems and provide knowledge source for knowledge subsystem. Knowledge subsystem is used to optimize the solving knowledge for a certain type of problem and provide guidance on solving method for biological subsystem. The proposed algorithm is used to solve knapsack problems; its work process is explained in the experiments. As knowledge subsystem provides the optimal solving method, such that the solution quality of BEA for the knapsack problem is much better than that of many other optimization algorithms that simulate the nature evolution course of general creature and its solving time is minimized. BEA is an optimization algorithm with high solving efficiency.

Key words: Biological evolution, knowledge evolution, bi-evolutionary algorithm, inherit operator, innovation operator, update operator

INTRODUCTION

The traditional evolution algorithms simulate the evolution course of general creature. In the course of solving a problem by them, the superior individuals survive and the inferior individuals are washed out by reproducing-competition-further reproducing-further competition and the optimal solution is drawn up length by length. As a typical representative of evolutionary computation, genetic algorithm only simulates the nature evolution course of general creature too, it is formed based on Darwin's natural selection theory. Genetic algorithm has its universal applicability, but it holds the inherent random blindness at the same time. In practice, genetic algorithm inevitably falls into the problem of premature convergence and low convergence speed as it depends on traversal searching. In order to speed up the global convergence of genetic algorithm, many scholars have done a lot of research work (Zhang *et al*., 2008; Tareeq *et al*., 2007; Ze-Su *et al*., 2007; Rong-Chang *et al*., 2009; Liu and Yan, 2007; Zhang and Zhang, 2006; Xiong *et al*., 2005; Cai and Li, 2006; Guo and Liu, 2002; Vijayakumar and Kumudinidevi, 2008; Hormwichian *et al*., 2009) and the performance of genetic algorithm is improved to a certain extent. However, the two fundamental limitations of premature convergence and low convergence speed of evolution algorithms simulating general biological are not solved. Because they are inherent from the evolution algorithms simulating general creature.

Human evolution is the most successful evolution in the world of creature, it takes on a character of accelerated rise. In the course of human evolution, knowledge system plays a vital role. Knowledge system is formed gradually when human accept the natural laws and it affects biological evolution greatly. Of course, biological system also affects the knowledge system; it makes knowledge system perfect increasingly. So, when human evolve in physiology, knowledge is improved synchronously. It is obvious that biological system and knowledge system affect human evolution together. Thus, if we can simulate human evolution and find out the mechanism of optimization algorithm, then a novel evolutionary algorithm may be formed. The performance of this novel evolutionary algorithm must be better then that of the evolutionary algorithms only simulate general creature.

Bi-evolutionary algorithm proposed in this study is a novel algorithm system simulating the mechanism of human evolution.

Corresponding Author: Yan Taishan, School of Computer Science and Engineering, Xi'an University of Technology, 710048, Xi'an, Shaanxi, People's Republic of China

THE BASIC IDEA OF BI-EVOLUTIONARY ALGORITHM

The theory foundation of Bi-Evolutionary algorithm: The history of human society tells us that human evolution has its natural characteristics and social characteristics. Look from the natural perspective, human evolution obeys biological evolution. Look from the social perspective, human have the knowledge system, we not only can summarize and accumulate previous knowledge, but also can find and create new knowledge. This is knowledge evolution, which is unique to human. Therefore, human evolution is a typical bi-evolutionary process.

Knowledge evolution and biological evolution interact in the course of human evolution. Their interaction is mainly represented in the following aspects. Firstly, our cognitive abilities are closely related to human knowledge; Secondly, after the beginning of knowledge evolution, biological evolution does not terminate. Knowledge evolution react on biological evolution and accelerate the process of biological evolution; Thirdly, knowledge evolution functions based on biological facts and physiological enginery, it can not work without biological premise; Fourthly, there are certain similar rules between biological evolution and knowledge evolution, such as selection, adaptation rules.

In terms of the evolutionary mechanism, human knowledge evolution is very similar to the simple biological evolution. Biological evolution aims to seek better species by attempt and correction. The mechanism of biological evolution includes the species creation mechanism and species selection mechanism. Knowledge evolution has the similar form with biological evolution. At any time, our knowledge is composed by the proved adaptive hypothesizes and the proved inappropriate hypothesizes were eliminated. Therefore, knowledge evolution also includes two mechanisms, the one is knowledge generation mechanism and the other is the natural selection mechanism of survival of the fittest.

The basic framework of Bi-evolutionary algorithm: Bi-evolutionary algorithm (inamed BEA) is formed by simulating the mechanism of human evolution. The basic framework of BEA is shown as Fig. 1.

Bi-evolutionary algorithm system includes knowledge subsystem and biological subsystem. They are mutually independent and interactive. The evolution of biological subsystem goes along in the solution space of the problems for optimization, its optimization is for the specific problems and its goal is to obtain the optimal solutions of the specific problems. The evolution of knowledge subsystem goes along in the knowledge space of a certain types of problem for optimization, its optimization is for the solution knowledge of a certain type of problem and its goal is to obtain the knowledge for the optimal solution of a certain type of problem. Knowledge subsystem and biological subsystem interact in the whole evolution process. Knowledge Subsystem provides guidance for biological subsystem and biological subsystem provides knowledge source for knowledge subsystem. In the application stage, for a new problem to be solved, the first work is to search the most appropriate method and parameters from the knowledge base, then solve the problem in the biological subsystem efficiently and get the optimal solution with high quality.

Bi-evolutionary algorithm based on knowledge evolution algorithm and genetic algorithm is an example of BEA, the biological subsystem of which is realized by genetic algorithm.

THE REALIZATION OF BI-EVOLUTIONARY ALGORITHM BASED ON KNOWLEDGE EVOLUTION ALGORITHM AND GENETIC ALGORITHM

Code method: In biological subsystem, the code method of genetic algorithm usually includes binary code, real code and symbol code etc. In this study, binary code is selected as the code method of genetic algorithm. In

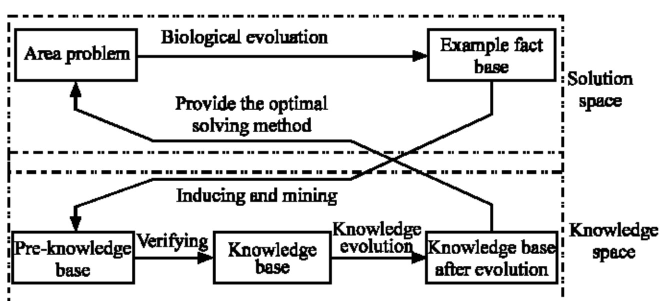

Fig. 1: The basic framework of BEA

knowledge subsystem, a suitable code method may be adopted in knowledge evolution algorithm according to the actual situation. Generally, a hybrid code method may be used in knowledge evolution algorithm. Binary code method and real code method are used for different variables with different properties and then cascade their codes together to form the code of knowledge individuals.

Evolution operators: In biological subsystem, the genetic algorithm simulating human reproduction mode (Yan *et al*., 2007) is adopted. The key operators of this genetic algorithm include selection operator, help operator, crossover operator and mutation operator.

In knowledge subsystem, the key operators of knowledge evolution algorithm (Yan and Cui, 2008) include inheritance operator, innovation operator and update operator. With the help of their united action, the knowledge for a certain type of problem evolves continuously and the knowledge base can be updated continually.

Fitness evaluation method: In biological subsystem, the general method to evaluate the individual fitness is: Firstly, decode the individual code string and get the individual phenotype; secondly, calculate the value of the corresponding objective function through the individual phenotype; Thirdly, calculate the individual fitness by the objective function according to the determinate transformation rules and the type of optimization problem.

In knowledge subsystem, the linear combination method (Yan and Cui, 2008) is adopted to evaluate the fitness of knowledge individuals. This method can be described as follows:

$$f(x_j) = g_1 p_{j1} + g_2 p_{j2} + \cdots + g_n p_{jn}$$

where, p_{j1}, p_{j2},....., p_{jn}, respectively denotes the value of the evaluated object i (i = 1, 2, ..., n) of number j knowledge individual; g_1, g_2,......, g_n, respectively denotes the weight of the evaluated object.

The work process of bi-evolutionary algorithm: After a certain problem area is selected, bi-evolutionary algorithm will go through the following stages:

- **Stage 1: Formation of example fact base:** Solve the optimization problems of the selected area in biological subsystem and gain their optimal solutions. Then put the related solution data and methods into example fact base. Repeat this process until there are sufficient data in the example fact base
- **Stage 2: Formation of pre-knowledge base:** Obtain knowledge for solving the problems of the selected area by means of inducing and mining etc. From the example fact base. Put this knowledge and the existing prior knowledge into pre-knowledge base
- **Stage 3: Formation of knowledge base:** Knowledge in pre-knowledge base are used to solve certain specific problems. Judge this knowledge according to the solution results. Reserve the excellent knowledge and discard the inferior knowledge. So, the knowledge base with guidance value for a certain type of problem is formed
- **Stage 4: Knowledge evolution:** In knowledge subsystem, inherit operator and innovation operator are used to generate new knowledge. After tested, the new knowledge is put into knowledge base by update operator. The knowledge base after evolution is formed
- **Stage 5: Solving new problems:** When bi-evolutionary algorithm is used to solve a new problem, the first work is to obtain the optimal knowledge about solving method from the knowledge base after evolution. Then, the problem is solved in biological subsystem with the help of obtained knowledge. If there is no corresponding knowledge in the knowledge base after evolution, the problem is solved directly in biological subsystem and the new solving knowledge is put into knowledge base

THE APPLICATION OF BI-EVOLUTIONARY ALGORITHM IN KNAPSACK PROBLEMS

Description of knapsack problems: Knapsack problem (Sinnamon and Andrews, 1997; Wang and Cao, 2002) is a well-known NP problem in combinatorial optimization. It has a wide range of applications, for example network planning, network routing, parallel scheduling and budgeting etc. Mathematically the 0/1 knapsack problem may be formulated as:

$$\text{Maximize: } f(x_1, x_2, \cdots, x_n) = \sum_{i=1}^{n} v_i x_i$$

$$\text{Subject to: } \sum_{i=1}^{n} w_i x_i \le c$$

where, $x_i \in \{0,1\}$, i = 1,2,...,n, $x_i = 1$ denotes that object i will be put into the knapsack and $x_i = 0$ denotes that object i will not be put into the knapsack. c Denotes the capacity of knapsack, v_i denotes the value of object I and w_i denotes the weight of object i. Generally, we assume that w_i, v_i and c are all positive integers.

In order to illustrate the working process of bi-evolutionary algorithm proposed in this paper and verify its performance, it is used to solve knapsack problems here. Four classical examples (Ma and Wang, 2001; Zhang and Zhang, 2001; Miao and Gao, 2009; Xu, 2004) are selected. Suppose the size, value and weight of goods is n, V and W, respectively and the maximum capacity of knapsack is c. The examples are described as:

- **Example 1:** n = 10, V = {55, 10, 47, 5, 4, 50, 8, 61, 85, 87}, W = {95, 4, 60, 32, 23, 72, 80, 62, 65, 46}, c = 269
- **Example 2:** n = 50, V = {72, 490, 651, 833, 883, 489, 359, 337, 267, 441, 70, 934, 467, 661, 220, 329, 440, 774, 595, 98, 424, 37, 807, 320, 501, 309, 834, 851, 34, 459, 111, 253, 159, 858, 793, 145, 651, 856, 400, 285, 405, 95, 391, 19, 96, 273, 152, 473, 448, 231}, W = {438, 754, 699, 587, 789, 912, 819, 347, 511, 287, 541, 784, 676, 198, 572, 914, 988, 4, 355, 569, 144, 272, 531, 556, 741, 489, 321, 84, 194, 483, 205, 607, 399, 747, 118, 651, 806, 9, 607, 121, 370, 999, 494, 743, 967, 718, 397, 589, 193, 369}, c = 11258
- **Example 3:** n = 100, V = {597, 596, 593, 586, 581, 568, 567, 560, 549, 548, 547, 529, 529, 527, 520, 491, 482, 478, 475, 475, 466, 462, 459, 458, 454, 451, 449, 443, 442, 421, 410, 409, 395, 394, 390, 377, 375, 366, 361, 347, 334, 322, 315, 313, 311, 309, 296, 295, 294, 289, 285, 279, 277, 276, 272, 248, 246, 245, 238, 237, 232, 231, 230, 225, 192, 184, 183, 176, 174, 171, 169, 165, 165, 154, 153, 150, 149, 147, 143, 140, 138, 134, 132, 127, 124, 123, 114, 111, 104, 89, 74, 63, 62, 58, 55, 48, 27, 22, 12, 6}, W={54, 183, 106, 82, 30, 58, 71, 166, 117, 190, 90, 191, 205, 128, 110, 89, 63, 6, 140, 86, 30, 91, 156, 31, 70, 199, 142, 98, 178, 16, 140, 31, 24, 197, 101, 73, 169, 73, 92, 159, 71, 102, 144, 151, 27, 131, 209, 164, 177, 177, 129, 146, 17, 53, 164, 146, 43, 170, 180, 171, 130, 183, 5, 113, 207, 57, 13, 163, 20, 63, 12, 24, 9, 42, 6, 109, 170, 108, 46, 69, 43, 175, 81, 5, 34, 146, 148, 114, 160, 174, 156, 82, 47, 126, 102, 83, 58, 34, 21, 14}, c = 6718
- **Example 4:** n = 50, V = {220, 208, 198, 192, 185, 180, 165, 162, 160, 158, 155, 130, 125, 122, 120, 118, 115, 110, 105, 101, 100, 100, 98, 96, 95, 90, 88, 82, 80, 77, 75, 73, 72, 70, 69, 66, 65, 63, 60, 58, 56, 50, 30, 25, 15, 10, 8, 5, 3, 1}, W = {80, 82, 85, 70, 72, 70, 66, 50, 55, 25, 50, 55, 40, 48, 50, 32, 22, 60, 30, 32, 40, 38, 35, 32, 25, 28, 30, 22, 50, 30, 45, 30, 60, 50, 20, 65, 20, 25, 30, 10, 20, 25, 15, 10, 10, 10, 4, 4, 2, 1}, c = 1000

The evolution of biological subsystem: In this study, the biological subsystem of bi-evolutionary algorithm is realized by the genetic algorithm simulating human reproduction mode (Yan *et al*., 2007). Example 1, 2 and 3 are solved by the evolution of biological subsystem, namely by the genetic algorithm simulating human reproduction mode. The solution results are saved in the example fact base, as Table 1 shows. Where, L denotes the scale of the problem; G denotes the size of population; T denotes the maximal evolution generation; P_h denotes the help probability; P_c denotes the crossover probability; P_m denotes the mutation probability; T_v denotes the total value in the knapsack; T_w denotes the total weight in the knapsack; t denotes the running time of algorithm, it refers to the average running time of 20 times, its units is second.

Table 1: The example fact base

L	G	T	P_h	P_c	P_m	T_v	T_w	t
10	50	200	0.02	0.80	0.10	295	269	0.319
10	50	200	0.05	0.80	0.05	295	269	0.299
10	50	200	0.02	0.70	0.10	295	269	0.345
10	50	200	0.05	0.70	0.05	295	269	0.383
50	100	400	0.02	0.80	0.10	16102	11231	8.059
50	100	400	0.05	0.80	0.05	16102	11231	6.775
50	100	400	0.02	0.70	0.10	15955	11160	7.939
50	100	400	0.05	0.70	0.05	15844	11178	7.496
100	200	600	0.02	0.80	0.10	26559	6717	22.219
100	200	600	0.05	0.80	0.05	26559	6717	24.565
100	200	600	0.02	0.70	0.10	26487	6718	26.955
100	200	600	0.05	0.70	0.05	24864	6651	25.726

The evolution of knowledge subsystem: The solving knowledge for knapsack problems is optimized by the evolution of knowledge subsystem. The purpose of knowledge evolution is to obtain better knowledge for a certain type of problem and provide more effective guidance for biological subsystem.

Structure of knowledge base: In the knowledge base designed in this section, the knowledge individual contains the scale of knapsack problem (namely, the number of objects), the parameter setting of algorithm and the evaluation of knowledge (namely, the fitness of knowledge individual) etc. There are three parts in structure which may be expressed as $\langle S, P, F\rangle$. The meaning of S, P and F can be explained as follows:

- $S = \{s^t_1, s^t_2, ..., s^t_n\}$ is the set of the scale of knapsack problems. Where, G is the scale of knowledge base, s^t_i denotes the scale of knapsack problem in number i (i = 1,2,...,G) knowledge individual in the knowledge base of generation t
- $P = \langle P_{size}, T, P_h, P_c, P_m\rangle$ denotes the parameter setting of algorithm. $P_{size} = \{P_{size1}^t, P_{size2}^t, ..., P_{sizeG}^t\}$ is the set of the scale of population, $T = \{T_1^t, T_2^t, ..., T_G^t\}$ is the set of the maximal generation, $P_h = \{p_{h1}^t, p_{h2}^t, ..., p_{hG}^t)$ is the set of help probability, $P_c = \{p_{c1}^t, p_{c2}^t, ..., p_{cG}^t)$ is the set of

crossover probability, $P_m = \{p_{m1}^t, p_{m2}^t, ..., p_{mG}^t\}$ is the set of mutation probability, where, P_{sizei}^t, T_i^t, p_{hi}^t, p_{ci}^t and p_{mi}^t separately denotes the scale of population, maximal generation, help probability, crossover probability and mutation probability of number i knowledge individual in the knowledge base of generation t

- $F = \{f_1^t, f_2^t, ..., f_G^t\}$ is the set of the fitness of knowledge individuals for knapsack problems. Where, f_i^t, denotes the fitness of number i knowledge individual in the knowledge base of generation t

Code method of knowledge: Here, real code method is adopted to describe the knowledge individuals. It may be regarded that there are G real vectors $Xi = (x_{i,1}, x_{i,2}, ..., x_{i,w})$ in the knowledge base. Where, w is the number of attributes of knowledge individuals, i = 1,2,...,G; $x_{i,j}$ denotes the value of number j attribute of number i knowledge individual.

Fitness evaluation method of knowledge: When we evaluate the fitness of knowledge for knapsack problems, there are three factors to be considered, they are the total value, total weight and running time of algorithm. The evaluating method can be described as:

$$f(i) = \frac{v_i}{v_{max}} g_1 + \frac{c - w_i}{c - w_{min} + 1} g_2 + \frac{t_{min}}{t_i} g_3$$

where, f(i) denotes the fitness of number i knowledge individual; v_{max}, w_{min} and t_{min}, separately denotes the known maximal total value, minimal total weight and shortest running time of algorithm for a certain type of knapsack problem; v_i, w_i and t_i separately denotes the total value, total weight and running time of algorithm of number i knowledge; c is the capacity of knapsack; g_1, g_2 and g_3 are the coefficients of evaluation objectives, their values are 0.5, 0.1 and 0.4 separately.

Formation of knowledge base: The knowledge for knapsack problems is gained from the fact base of the examples by inducing and mining. After verified and evaluated, the knowledge individuals are put into the knowledge base (suppose that, for every type of knapsack problems, 4 excellent knowledge individuals are saved in the knowledge base), as Table 2 shows. Where, F denotes the fitness of knowledge.

Knowledge inherit and knowledge innovation: For the knowledge populations with different scale of knapsack problem, suppose the scale of evolution is 2. After knowledge inherit and innovation operation, new knowledge individuals are generated. Table 3 shows the results after one evolution. The fitness of these new knowledge individuals is shown in Table 3.

Table 2: Knowledge base

ID	L	G	T	P_h	P_c	P_m	F
1	10	50	200	0.02	0.80	0.10	0.927
2	10	50	200	0.05	0.80	0.05	0.950
3	10	50	200	0.02	0.70	0.10	0.906
4	10	50	200	0.05	0.70	0.05	0.881
5	50	100	400	0.02	0.80	0.10	0.916
6	50	100	400	0.05	0.80	0.05	0.964
7	50	100	400	0.02	0.70	0.10	0.949
8	50	100	400	0.05	0.70	0.05	0.951
9	100	200	600	0.02	0.80	0.10	0.950
10	100	200	600	0.05	0.80	0.05	0.922
11	100	200	600	0.02	0.70	0.10	0.896
12	100	200	600	0.05	0.70	0.05	0.917

Table 3: The set of the new knowledge

ID	L	G	T	P_h	P_c	P_m	F
13	10	40	200	0.01	0.65	0.04	0.852
14	10	50	240	0.06	0.75	0.11	0.932
15	50	120	480	0.02	0.65	0.25	0.910
16	50	120	400	0.10	0.85	0.11	0.965
17	100	240	720	0.01	0.60	0.05	0.869
18	100	240	720	0.02	0.65	0.09	0.886

Table 4: The knowledge base after evolution

ID	L	G	T	P_h	P_c	P_m	F
2	10	50	200	0.05	0.80	0.05	0.950
14	10	50	240	0.06	0.75	0.11	0.932
1	10	50	200	0.02	0.80	0.10	0.927
3	10	50	200	0.02	0.70	0.10	0.906
16	50	120	400	0.10	0.85	0.11	0.965
6	50	100	400	0.05	0.80	0.05	0.964
8	50	100	400	0.05	0.70	0.05	0.951
7	50	100	400	0.02	0.70	0.10	0.949
9	100	200	600	0.02	0.80	0.10	0.950
10	100	200	600	0.05	0.80	0.05	0.921
12	100	200	600	0.05	0.70	0.05	0.917
11	100	200	600	0.02	0.70	0.10	0.895

Update of knowledge base: After the generation of new knowledge individuals, the update operator is used to add the excellent new knowledge individuals to knowledge base in order to update it. By comparison with the original knowledge base, we can easily find: in Table 3, number 14 knowledge individual is the more excellent knowledge individual for the knapsack problems whose scale is 10; number 16 knowledge individual is the more excellent knowledge individual for the knapsack problems whose scale is 50 and there is no more excellent knowledge individual for the knapsack problems whose scale is 100. The above excellent new knowledge individuals will be added to knowledge base. In order to maintain a certain scale of knowledge base, for every type of knapsack problems, it is necessary to delete the knowledge individuals with lower fitness, the number of which is equal to the added new knowledge individuals. In this way, the knowledge base after evolution is gained as shown in Table 4.

The interaction between biological subsystem and knowledge subsystem: When bi-evolutionary algorithm

Table 5: Comparison of the solution results of example 4

Algorithm	Total value	Total weight	t
BEA	3117	1000	6.515
GDA	2994	960	7.773
GDPSO	3033	993	45.910
GDGA	3077	999	11.4825
GDDEA	3095	1000	11.4116

is used to solve a new knapsack problem, the first work is to search corresponding knowledge (solving method) in the knowledge base after evolution. If there is corresponding knowledge, solve the new knapsack problem by the optimal knowledge; If there is no corresponding knowledge, solve the new knapsack problem by other feasible method and add the solution results to the example fact base. When the accumulated data is sufficient, the optimized knowledge for the new knapsack problem may be gained by inducing, mining and verifying.

We solve example 4 now. Example 4 is a new knapsack problem to be solved whose scale is 50. When it is solved by bi-evolutionary algorithm, we can see that there is corresponding knowledge in the knowledge base after evolution. The optimal knowledge 16 is selected to solve this new knapsack problem. The optimal solution is gained, in which the total value is 3117 and the total weight 1000.

The solution result of BEA for example 4 is compared with other existed algorithms, they are greedy algorithm (GDA) (Miao and Gao, 2009), PSO combined with greedy transform (GDPSO) (Miao and Gao, 2009), genetic algorithm combined with greedy transform (GDGA) (Xu, 2004) and discrete differential evolution algorithm combined with greedy transform (GDDEA) (Miao and Gao, 2009). The comparison result is shown in Table 5. Where, t denotes the running time of algorithms, it refers to the average running time of 50 times, its units is second.

DISCUSSION

From the comparison of the solution results of example 4 (Table 5), it is very obvious that Bi-Evolutionary Algorithm (BEA) proposed in this study provides the optimal solution whose total value is 3117 and tatal weight is 1000. On the other hand, the solving time of BEA is minimized, its running time is only 6.515.

For greedy algorithm (GDA) (Miao and Gao, 2009), although its running time is closer to that of BEA, its solution result is clearly worse, the total value in its optimal solution is only 2994 and its running time is longer then that of BEA. For PSO combined with greedy transform (GDPSO) (Miao and Gao, 2009), genetic algorithm combined with greedy transform (GDGA) (Xu, 2004) and discrete differential evolution algorithm combined with greedy transform (GDDEA) (Miao and Gao, 2009), their running time is all longer then that of BEA, especially the running time of GDPSO is 6 times longer then that of BEA and their solution results all have a certain gap with that of BEA.

In the above algorithms, GDA, GDPSO, GDGA and GDDEA are optimization algorithms that simulate the nature evolution course of general creature, they have no knowledge accumulation and optimization mechanism. They solve different types of problems by the same method, thus the solving efficiency must be influenced. However, for BEA, the knowledge subsystem provides the optimized knowledge (the optimal solving method) for a certain type of problem, different types of problem are solved by different methods, so the solving efficiency is guaranteed.

CONCLUSION

Bi-evolutionary algorithm is essentially established by simulating human evolution mechanism. It contains biological evolution subsystem and knowledge evolution subsystem. The biological evolution subsystem is used to solve the specific problems and gain their optimal solutions. The knowledge evolution subsystem is used to optimize the solving knowledge and gain the optimization knowledge for a certain type of problem, so as to provide guidance for future problem solving. When a new problem is to be solved, it is easy for the algorithm to search the most suitable optimization method for the problem with very little price on knowledge matching. So, not only the solving time is shortened but also the quality of the solution is increased. Experiments were taken on knapsack problems, the experimental results show that bi-evolutionary algorithm is more efficient compared with many other optimization algorithms.

ACKNOWLEDGMENTS

Support from the National Natural Science Foundation of China under Grant No. 60873035 is gratefully acknowledged.

REFERENCES

Cai, L.W. and X. Li, 2006. Improvement on crossover operation of genetic algorithms. Syst. Eng. Electr., 28: 925-928.

Guo, D.W. and D.Y. Liu, 2002. Dynamic analysis of gas constringency and its application. J. Comput. Res. Develop., 39: 225-230.

Hormwichian, R., A. Kangrang and A. Lamom, 2009. A conditional genetic algorithm model for searching optimal reservoir rule curves. J. Applied Sci., 9: 3575-3580.

Liu, M. and J.W. Yan, 2007. An adaptively annealing genetic algorithm based scheduling method of workshop daily operating planning. Chinese J. Comput., 30: 1164-1172.

Ma, L. and L.D. Wang, 2001. Ant optimization algorithm for knapsack problem. Comput. Appl., 21: 4-5.

Miao, S.Q. and Y.L. Gao, 2009. Discrete differential evolution algorithm for solving 0/1 knapsack problem. J. Chin. Comput. Syst., 9: 1828-1830.

Rong-Chang, C., C. Ting-Tsuen and F. Wei-Luen, 2009. Assignment of external off-the-job training courses to employees using genetic algorithm. Inform. Technol. J., 8: 147-155.

Sinnamon, R.M. and J.D. Andrews, 1997. New approaches to evaluating fault trees. Reliability Eng. Syst. Safety, 58: 89-96.

Tareeq, S.M., R. Parveen, L.J. Rozario and M. Al-Amin-Bhuiyan, 2007. Robust face detection using genetic algorithm. Inform. Technol. J., 6: 142-147.

Vijayakumar, K. and R.P. Kumudinidevi, 2008. A hybrid genetic algorithm for optimal power flow incorporating FACTS devices. Asian J. Scientific Res., 1: 403-411.

Wang, X.P. and L.M. Cao, 2002. Genetic Algorithm-Theory, Application and Software Realization. 1st Edn., Xi'an Jiao tong University Press, Xi'an. ISBN:7-5605-1448-0.

Xiong, W.Q., M.D. Liu and S.Y. Zhang, 2005. A genetic algorithm with sex character. Comput. Eng., 31: 165-166.

Xu, Z.B., 2004. Computation Intelligence- Simulated Evolutionary Computation. 1st Edn., High Education Press, Beijing.

Yan, T.S., Y.Q. Tao and D.W. Cui, 2007. Research on genetic algorithm based on human reproduction phenomenon. Comput. Eng. Appl., 42: 78-81.

Yan, T.S. and D.W. Cui, 2008. Research on knowledge evolution algorithm. Comput. Eng. Appl., 44: 8-11.

Ze-Su, C., H. Bing-Rong and L. Hong, 2007. An improved polar scan matching using genetic algorithm. Inform. Technol. J., 6: 89-95.

Zhang, L. and B. Zhang, 2001. Good point set based genetic algorithm. Chin. J. Comput., 24: 917-922.

Zhang, S.C. and F.X. Zhang, 2006. A modified method to fitness function of genetic algorithms. Comput. Appl. Software, 23: 108-110.

Zhang, Q., B. Wang, R. Zhang and C.X. Xu, 2008. Genetic algorithm-based design for DNA sequences sets. Chinese J. Comput., 31: 2193-2199.