A Tabu Search Approach with Double Tabu-List for Multidimensional Knapsack Problems

Yi HE[1,2], Yuhui QIU[1] and Guangyuan LIU[1]

[1]School of Management, Chongqing Normal University, Chongqing 400047, P.R. China
[2]Faculty of Computer & Information Science, Southwest-China University, Chongqing 400715, P.R. China

Summary
The knapsack problems are a classic NP-hard problem in the combinational optimization. Inspired by the conclusion of the cognitive psychology about the human memory system, a Tabu Search method based on Double Tabu-List (DTL-TS) has been proposed to solve it. With the addition of the search strategy of intensification and diversification, the excellent experiment results have been gotten. Compared with ImmunoDominance Clone Algorithm, DTL-TS is shown to be an efficient approach of solving complex problems like 0-1 multidimensional knapsack problems.

Key words:
Tabu Search; Double tabu-list; Multidimensional Knapsack Problem

1.Introduction

The 0-1 Multiconstrained Knapsack Problem (0/1 MKP) is a combination optimization problem which has a very simple structure and is easy to understand. Nevertheless, some types of instances can be very hard to solve to proven optimum. A lot of work has been done to develop good heuristics for this problem, using various techniques. These include Branch and Bound Method [1-2], Enumeration Algorithm [3], Two-list algorithm [4], Approximate Dynamic Programming [5], Genetic Algorithm [6-7], Ant Colony System [8], Immune Algorithm [9-11] and Tabu Search (TS) Algorithm [12-13], etc. All these approaches have given good results. We have chosen to focus on tabu search, and base our inspiration on the conclusion of the cognitive psychology about the human memory system. We try to simplify these methods in order to find out if it is possible to get good results while keeping the algorithms simple. Our main contribution is that double tabu-list, namely short tabu-list and long tabu-list are introduced into Tabu Search. In the following section, a brief description of knapsack problem was presented. In section 3, a short introduction of TS was presented. In next section, the Double Tabu-List TS (DTL-TS) approach was detailed. Moreover, the experimental results of DTL-TS were shown, providing also comparisons to other method. Some discussion and a short conclusion follow in section 5.

Manuscript received May 5, 2006.
Manuscript revised May 25, 2006.

2. Problem Descriptions

The 0-1 Multiconstrained Knapsack Problem (0/1 MKP) can be formulated

$$\text{Max} \sum_{j=1}^{n} c_j x_j$$

s.t.

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \forall i \in \{1, 2, \cdots, m\}$$

$$x \in \{0, 1\}$$

n is the number of items, and m is the number of knapsack constraints. The c values can be interpreted as the value of including the different items, the a values can be interpreted as measures of weight or volume for each item, and the b values are limits for each constraint. The x values represent the items, which are given the value *1* if they are included in the solution, *0* otherwise. The problem is then to find the best combination of items to include in the solution, not violating any of the constraints. There are 2^n different solutions to the problem; some of these are infeasible as they violate one or more of the constraints. The problem is known to be NP-hard and a lot of work has been done to find good algorithms to solve it.

3. Tabu Search

Tabu search, proposed by Glover [14], is a meta-heuristic method that can be used to solve combinatorial optimization problems. It has received widespread attention recently. Its flexible control framework and several significant successes in solving NP-hard problems aroused rapid growth of its application. The method of neighborhood exploration and the use of short-term memories distinguish tabu search from local search and other heuristic search methods, and result in lower computational cost and better space exploration.

4. Double Tabu-List based Tabu Search

Tabu Search involves a lot of techniques and strategies, but it mainly comes from the use of short-term memory, namely tabu-list that keep track of recently examined solutions, intending to avoid cycling in the solution exploration. Other prior information about the solution can used to improve the intensification and diversification of the search. Inspired by cognitive psychology, we presented a Double Tabu-List based Tabu Search (DTL-TS) approach for multidimensional 0-1 knapsack problems as follows.

4.1 Main idea

Cognitive psychology considers that the human memory system is constituted by short-term memory and long-term memory, demonstrated with Fig.1. The information received by sense memory is processed by short-term memory and stay in short time, and then enters into long-term memory for long time saving. As same as the human memory system, two tabu-lists are introduced into TS, namely short tabu-list and long tabu-list. The short tabu-list is used to keep the solutions obtained recently. After a period of iteration steps (tabu tenure), the solution whose tabu tenure is zero is released from short tabu-list and then entered into the long tabu-list. The only distinguish between them is that short tabu-list's tabu tenure is set to very short, and yet long tabu-list's tabu tenure is set to long.

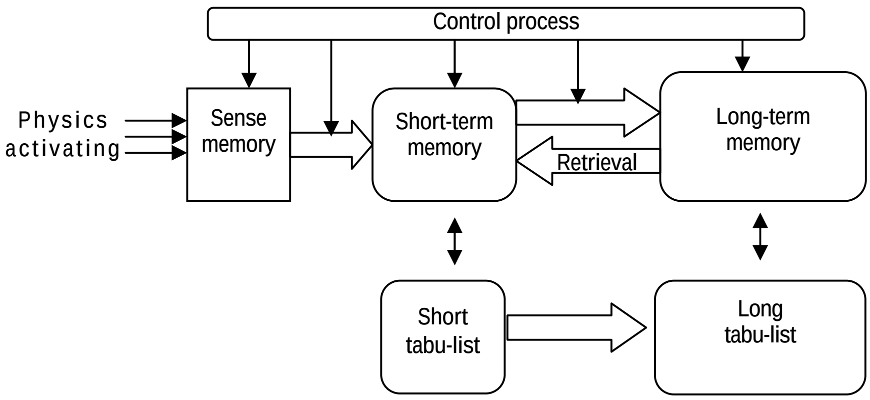


Fig.1. The corresponding relation between human memory system and double tabu-list

4.2 Algorithm design

Aim at the multidimensional knapsack problems, the DTL-TS algorithm was designed as follows:

(1) The definition of solution

$X = (\xi_1,\xi_2, \cdots,\xi_n)$, here, the value of ξ_n is only *1* or *0*.

(2) Starting solution

The starting solution was generated by greedy method, viz. descending sorted by the ratio of profits to consumed costs, and at the same time, the solution must meet all the constraints. As a result, the sequence of loaded items was generated, too.

(3) The definition of the generated rule for neighborhood

$N(x) = \left\{ y \middle| |y - x| = \sum_{i=1}^{n} |y_i - x_i| \le k \right\}$, here, the value of k is positive integral number.

I.e. at first, one or more items are taken out randomly from the sequence of loaded items, then, one or more items are taken into the knapsacks from the sequence of unloaded items at random. According to all the constraints, if the solution is infeasible, it is discarded and then regenerated.

(4) The strategy of intensification and diversification

In TS, intensification strategies are used to encourage TS to search more thoroughly the neighborhood of elite solutions in order to obtain the global optimal, and diversification strategies are used to encourage TS to search solution space as widely as possible, especially the unvisited regions before. They are very important and the coordinative important. To further improve the search efficiency, the elements in neighborhood are divided into two parts, namely intensification elements at about former 1/4 of total and diversification elements at about later 3/4. The difference between them is the taken out region of items from the sequence of loaded items is different when the neighborhood is generated. For intensification elements, the taken out region is about later 20% of the sequence of loaded items, but for diversification elements, it is 100%, demonstrated with Fig. 2.

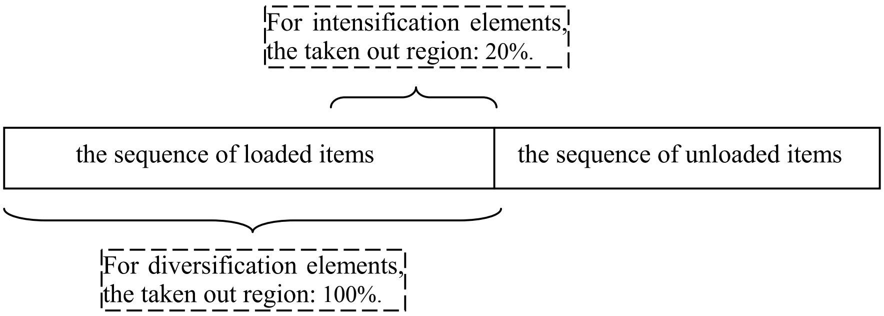


Fig.2 The strategy of intensification and diversification

(5) Short tabu-list, long tabu-list and mutating operation
Short tabu-list saves the recent solutions. Long tabu-list saves the history solution. When update long tabu-list, if the current local optimum solution is equal to one of the solutions in long tabu-list, it can be considered the loop iteration has happened. Of course, proper loop is permitted at certain degree in search process. Viz. the times of each solution being repeated can be set. In our experiments, this repeated times are set to 3. As soon as the third repeating happens, a simple mutating operation is adopted to the current local optimum solution, in order to keep the search from going in loops at next step. The mutating operation is defined as follows. A bit is selected randomly from the current solution, if it is 0, changed to 1; otherwise, it is changed from 1 to 0. The aim of the mutating operation is to further jump out the local optimum.

(6) The basic steps of DTL-TS

Step1 generate the starting solution based on greedy method.

Step2 generate neighborhood.

Step3 generate candidate set.

Step4 obtain the current local optimum.

Step5 update short tabu-list.

Step6 update long tabu-list and implement mutating operation if require.

Step7 if the maximum iteration steps are expired, algorithm terminated, else, take the current solution as the next starting solution, then return to step2.

4.3 simulation experiments

In order to verify the effectiveness and efficiency of this TS technique, the test data were from the literature [15] and there are 55 instances. The algorithm parameters set were detailed on Tab.1. For each knapsack problem instance, DTL-TS had been run for 20 times. The global optimal solution was all obtained for each instance at certain probability. Parts of the experiments were reported in Tab.2 because of the limitation of pages. Fig.3 was the varying process of the objective function for some instances. Tab.3 was the optimal solution and the corresponding sequence of loaded items for some instances. Tab.4 described the comparisons between DTL-TS and Immunodominance Clone Algorithms (IDCA) [9].

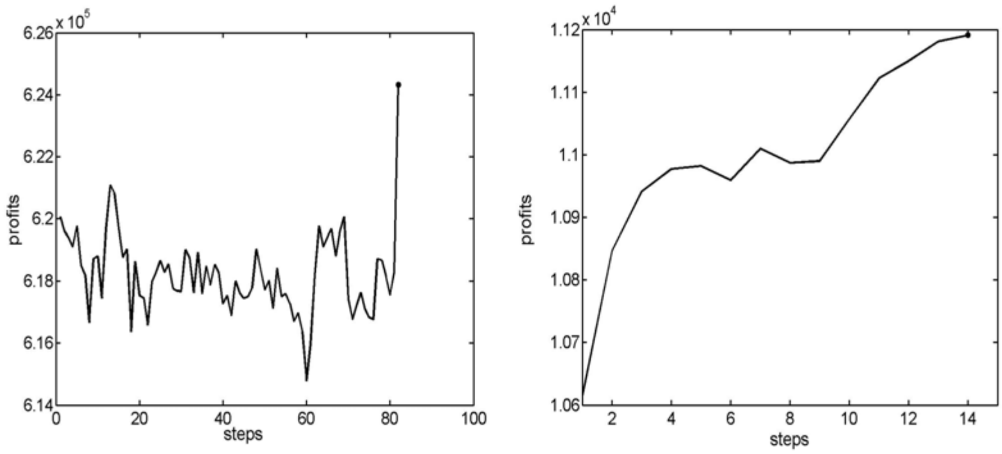
Tab.1 Main parameters set of DTL -TS

Parameters	Value
Length of short tabu-list	$n/2$
Length of long tabu-list	$2n$
Size of neighborhood	$4n$
Size of candidate set	$n/2$
Maximum iteration steps	$4n$

Tab.2 experiment results

Test datas				O_n	Profits			Run time of obtaining optimal (seconds)		
instance	*m*	*n*	Known optimal		mean	max	min	mean	max	min
pb6	30	40	776	20	776	776	776	9.10	14.86	3.33
pb7	30	37	1035	20	1035	1035	1035	8.50	14.72	1.76
pet3	10	15	4015	18	4014	4015	4005	2.0039	6.453	0.00182
pet5	10	28	12400	18	12398	12400	12370	2.9195	30.61	0.00223
pet7	5	50	16537	17	16535	16537	16524	28.459	100.79	8.555
sent01	30	60	7772	16	7769.8	7772	7761	76.951	387.53	1.7239
sent02	30	60	8722	4	8721.2	8722	8721	124.79	247.35	22.983
weing6	2	28	130623	11	130367.7	130623	129272	12.27	18.67	4.10
weing8	2	105	624319	12	623025.8	624319	621086	151.26	210.09	112.88
weish25	5	80	9939	17	9938.55	9939	9936	88.07	349.55	0.65
weish26	5	90	9584	19	9581.9	9584	9542	146.56	249.73	12.37
weish27	5	90	9819	19	9818.85	9819	9816	125.53	606.31	1.15
weish28	5	90	9492	13	9483.95	9492	9469	393.30	1662.63	5.64
weish29	5	90	9410	20	9410	9410	9410	180.73	330.58	24.40
weish30	5	90	11191	19	11190.55	11191	11182	43.48	67.22	11.59

Notes: O_n denotes the times of the optimal solution obtained among 20 times running.



(a) weing8.dat (2x105) (b) weish30.dat (5x90)

Fig.3 the varying process of the objective function

Tab.3 the optimal solution and the corresponding sequence of loaded items

weing8.dat (2x105)	Optimal solution	1 1 0 1 1 1 0 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 0 1 1 0 0 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	sequence of loaded items	14 17 19 21 16 2 34 26 25 45 24 32 15 20 35 1 31 29 4 5 18 13 42 38 40 6 22 30 23 8
weish30.dat (5x90)	Optimal solution	1 0 0 0 1 0 1 0 1 1 1 1 1 0 0 1 0 1 1 0 0 1 1 0 0 1 0 0 0 1 1 0 0 1 1 0 0 0 1 1 0 1 1 0 0 0 0 1 1 1 1 1 0 1 0 1 0 1 1 1 1 1 1 1 0 1 1 0 1 1 0 0 1 1 0 1 0 1 0 0 0 1 1 0 1 1 1 1 0 1
	sequence of loaded items	13 11 12 31 63 1 18 85 50 30 48 70 69 59 49 86 42 67 16 62 54 35 39 26 90 58 9 64 88 7 34 74 61 22 19 10 60 73 76 87 40 5 52 23 82 83 78 43 56 51 66

Tab.4 the comparisons between DTL-TS and IDCA (Immunodominance Clone Algorithms)

Test datas				IDCA		DTL-TS	
instance	*m*	*n*	Known optimal	O_n	Profits (mean)	O_n	Profits (mean)
pet6	5	39	10618	14	10613	16	10615
pb5	10	20	2139	5	2123.3	18	2137.3
pb7	30	37	1035	2	1029.8	20	1035
weing7	2	105	1095445	6	1095382.1	3	1095385.55
weish06	5	40	5557	13	5552	11	5549.1
weish23	5	80	8344	6	8341.9	10	8342.5
weish25	5	80	9939	12	9937.4	17	9938.55

Notes: O_n denotes the times of the optimal solution obtained among 20 times running.

The above experiments were finished on the PC (Celeron(R), 2.6GHz, 512MB RAM, matlab 6.5). The experiment results show that DTL-TS is feasible and efficient.

5 Conclusions

DTL-TS holds the soul of the human memory system instead of its shell. Introducing of long tabu-list is of benefit to judge whether the search traps into loop. The mutation operation further improves the diversification search. As a result, DTL-TS has strong ability to search thoroughly in local space as well as escape the local optimal. According to the comparison in tab.4 the DTL-TS in this work is good at finding high quality solutions and converging at the optimal at quick speed.

Acknowledgments

This work was partly supported by key Project of Chinese Ministry of Education (104262), China.

References

[1] Shih, W., A Branch and Bound Method for the Multiconstraint Zero-One Knapsack Problem, Journal of the Operational Research Society, 30(4): 369-378, 1979.

[2] Gavish B. and H. Pirkul., Efficient Algorithms for Solving Multiconstraint Zero-One Knapsack Problems to Optimality, Mathematical Programming, 31(1): 78-105, 1985.
[3] Cabot A.V., An Enumeration algorithm for Knapsack Problems, Operations Research, 18(2): 306-311, 1970.
[4] Horowitz E. and Sahni S., Computing partitions with applications to the knapsack problem, Journal of ACM, 1974, 21(2): 277-292.
[5] Bertsimas D. and Demir R., An Approximate Dynamic Programming Approach to Multidimensional Knapsack Problems, Management Science, 48(4): 550-565, 2002.
[6] Jun Ye, Xiande Liu, Lu Han, Evolutionary Game Algorithm for Multiple Knapsack Problem, Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology (IAT'03), pp.424-427, 2003.
[7] K. Kosuke and S. Masatoshi, Genetic Algorithms with Decomposition Procedures for Multidimensional 0-1 Knapsack Problems With Block Angular Structures, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, Vol. 33, No. 3, pp.410-419, 2003.
[8] Parra-Hernandez R. and Dimopoulos N., On the performance of the ant colony system for solving the multidimensional knapsack problem, 2003 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM '03), Victoria, Canada, pp. 338-341, 2003.
[9] Du H., Jiao L. and Liu R., Immunodominance Clone Algorithms , Journal of Electronics & Information Technology, 26(12): 1918-1924, 2004. (in Chinese with English abstract)
[10] Castro L.N., Immune, Swarm, and Evolutionary Algorithms Part I: Basic Models, Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'02), Vol.3, pp.1464-1468, 2002.
[11] Castro L.N., Immune, Swarm, and Evolutionary Algorithms Part II: Philosophical Comparisons, Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'02), Vol.3, pp.1469-1473, 2002.
[12] Hanafi S. and Freville A., An efficient tabu search approach for the 0-1 multidimensional knapsack problem, European Journal of Operational Research, 106(2-3): 659-675, 1998.
[13] Li,V.C. Tight oscillations tabu search for multidimensional knapsack problems with generalized upper bound constraints, Computers & Operations Research, 32(11): 2843-2852, 2005.
[14] Glover F. and Laguna M., Tabu Search, Boston, Kluwer Academic Publishers, 1997.
[15] SAC-94 Suite of 0/1-Multiple-Knapsack Problems , http://elib.zib.de/pub/Packages/mp-testdata/ip/sac94-suite/index.html.

Yi HE received the B.S. degree in education technology and M.S. degree in Electrical Engineering from Southwest-China Normal University in 1993 and 2000, respectively, and now is a PH.D candidate in Southwest-China University. He is interested in intelligence optimization algorithms with applications.