Fast Tabu Search Algorithm for Solving Multi-cargo Loading Problem under the Limited Loading Capacity

Chunyu REN[1, a]

[1] School of Information science and technology, Heilongjiang University, Harbin, China

[a]rency2004@163.com

Keywords: Multi-cargo loading Problem; Fast tabu search algorithm; Insertion method; three operations

Abstract. This paper studies the loading problems of Multi-category Goods under the limited loading capacity. According to the characteristics of model, According to the features of the problem, fast taboo search algorithm is used to get the optimization solution from the overall situation. Firstly, it applies newly improved insertion method to construct initial solution in order to improve the feasibility of the solution. Secondly, it centers cubage-weight balance to design three operations for fastening the speed of convergence and efficiency. Finally, the good performance of this algorithm can be proved by experiment calculation and concrete examples for solving practical problems.

Introduction

Bulk cargos have the characteristics of low-volume, multi-variety and multi-batch. Cargo loading problem is a complex optimization problem, which is NP-hard problem. Michael introduced the "block" concept. "Block" is a rectangular structure, which is composed as the same object and there is no space between objects. He also proposed similar "block" algorithm. This algorithm use greedy algorithm to generate an initial solution, and then use branch boundary method to optimize the initial solution, the algorithm select next node by searching methods. This algorithm improves the box space utilization with goods stability, easy to loading and unloading, easy to place object characters [1]. Tobias proposed "composite block" concept based on "block" concept, "composite block" allows a variety of items. The new heuristic algorithm combine the advantages of "composite block" and "free space", use heuristic tree search method to find the optimal solution. This method is the most effective packing algorithm on packing area so far [2]. Leo Ho Wai Yeung use hybrid genetic-based heuristic algorithm based sort method to solve container loading problems [3]. Bortfeldt use hybrid genetic algorithm to solve container loading problems [4]. Pu Lei designed heuristic algorithm to solve cargo loading problem with multi-variety [5]. Cao Hongmei designed ant algorithm to solve cargo loading problem with multi-variety [6]. For the specificity of multi-variety and freight loading problems with priority, traditional algorithm is easy to set the utilization rate of weight and capacity to the opposite side so as to form waste of loading resource and increase transport cost.

Therefore, according to the characteristics of model, fast taboo search algorithm is used to get the optimization solution from overall optimization and fully utilize the weight and capacity of loading tools.

Model

$$Max \quad Z = \sum_{i\in N}\sum_{j\in K}\lambda g_i x_{ij} + \sum_{i\in N}\sum_{j\in K}(1-\lambda)v_i x_{ij} \quad (1)$$

Restraint condition,

$$\lambda G \min{}_j \le \sum_{i\in N} g_i x_{ij} \le \eta_1 G_j, \quad j \in K \quad . (2)$$

$$(1-\lambda)V \min{}_j \le \sum_{i\in N} v_i x_{ij} \le \eta_2 V_j, \quad j \in K \quad (3)$$

Published by Atlantis Press, Paris, France.

$$\sum_{j\in K} x_{ij} \le 1, \quad i \in N - D_0 \qquad (4)$$

In model, all variables can be shown as followings. The maximum loading weight of j tools is G_j. The maximum loading capacity is V_j. The weight of $i(i \in N)$ is g_i. Volume is v_i. $G\min_j$ is the minimum loading weight of j loading tools. $V\min_j$ is the minimum loading capacity of j loading tools. $\eta_2(0 \le \eta_2 \le 1)$ is the elasticity coefficient of loading capacity. $D_i(D_i \subset N)$ is the cargo mustering not mixing cargo. $D_0(D_0 \subset N)$ is the loading cargo muster of priority mandatory installation.

Application in MCLP of Tabu Search Algorithm

Initial Solution Forming. Suppose that $N(i = 1,2,...,n)$ is the muster of loading cargo, $K(j = 1,2,...,m)$ is the muster of loading tool, the maximum weight of j loading tool is G_j, the maximum loading capacity is V_j. The weight of $i(i \in N)$ cargo is g_i, volume is v_i. If all cargos are installed, it needs $n_k = \max(n_g, n_v)$ vehicles, in which $n_g = \left|\sum_{i=1}^{N} g_i / \min(G_1, G_2, ..., G_m)\right|$ and $n_v = \left|\sum_{i=1}^{n} v_i / \min(V_1, V_2, ..., V_m)\right|$. If $n_g > n_v$, $\omega_i = \frac{g_i}{v_i}(i \in N)$. Otherwise, $\omega_i = \frac{v_i}{g_i}(i \in N)$. Taking $n_g > n_v$ as example, the concrete steps are as followings.

Step1: Initial data are supposed $n_k = 0$, $V_{sum}^1 = 0$, $G_{sum}^1 = 0$, $S^0 = \Phi$, $K = m$;

Step 2: Calculate $\omega_i = \frac{g_i}{v_i}(i \in N)$, sorting according to non-increasing order. Take this muster as p^1.

Step 3: Suppose $p = p^1$, calculate muster p. Review it from beginning to ending. The order is $p[1], p[N], P[2], P[N-1], ...$

Step 4: if $G_{sum}^1 + g_{p[1]} \le G_j(j \in m)$ and $V_{sum}^1 + v_{p[1]} \le V_j(j \in m)$, $G_{sum}^1 = G_{sum}^1 + 1$ and $V_{sum}^1 = V_{sum}^1 + 1$.

Step5: if $G_{sum}^1 + g_{p[N]} \le G_j(j \in m)$ and $V_{sum}^1 + v_{p[n]} \le V_j(j \in m)$, $G_{sum}^1 = G_{sum}^1 + 1$ and $V_{sum}^1 = V_{sum}^1 + 1$. Otherwise, turn into step 7.

Step 6: Repeat step 4 and 5 to $G_{sum}^1 > G_j(j \in m)$ and $V_{sum}^1 > V_j(j \in m)$.

Step 7: Record current status of G_{sum}^1, V_{sum}^1, S^1, $n_k = n_k + 1$, $K = K - 1$ and $S^2 = S^1 \cup S^0$.

Step8: Suppose $p^1 = p - S^2$;

Step 9: Repeat step 3 to step 8 until to $K = 0$. And turn into $K = 0$.

Step 10: Output the cargo muster $\{S^1, S^2, ..., S^{nk}\}$, total weight of loading cargo G_{sum}, total volume of loading cargo is V_{sum}.

1-exchange Neighborhood Operation. 1-exchange is to delete two clients in two routes, alternately insert them into their counterpart route, which can effectively boost the local search capability. Its neighborhood structure is the same as 1-move, but its radius can be larger.

2-opt Neighborhood Operation. 2-opt is used to conduct neighborhood search, which was to randomly choose the positions of two client nodes, and then exchange the clients between the two positions. $k(i)$ signified the neighbor point of the client point i in the route l, and $a(i, j)$ signified to change the direction of the route from i to j. That was in the l route, the client points were: $(0,1,2,..., n,0)$, in it, 0 signified distribution centre. The procedures of the $2-opt$ neighborhood operation were as such:

Step1: $i_1 := 1, i := 0$;

Step2: if $i > n - 2$, end; otherwise, turn to Step3;

Step3: revise $i_2 := k(i_i), \quad j_1 := k(i_2), \quad j := i + 2$;

Step4: if $j > n$, turn to Step8, if not, turn to Step5;

Step5: $j_2 := s(j_1)$, change route l as such (1) $a(i_2, j_1)$, (2) alternately used (i_1, j_1) and (i_2, j_2), substitute (i_1, i_2) and (j_1, j_2);

Step6: If the changed route l_1 is feasible, and better than l, revise l, if not, turn to Step7;

Step7: $j_1 := j_2, \quad j := j+1$, return to Step4;

Step8: $i_1 := i_2, \quad i := i+1$, return to Step2.

2-opt* Neighborhood Operation. 2-opt* operates on the exchange of two edges in different routes, in order to realize optimization between routes. That is in the route l, the client points are $(0,1,2,...,n,0)$, in the route k, the client points are $(0,1,2,...,m,0)$, in it, 0 signifies distribution centre.

Step1: Randomly choose n number of client points in the route l, for each client point i, choose client point j nearby the route k, if exist, exchange chains $(i, i+1), (j, j+1)$;

Step2: Conduct $2-opt$ neighborhood operation in the exchanged routes l^1 and k^1, to obtain feasible solution;

Step3: Calculate the exchanged objective function f^1, if $f^1 > f$, turn to Step4; if not, turn to Step5;

Step4: If the current optimal solution does not exist in the tabu object, update it, input the obtained optimal solution into the object, simultaneously remove out the ban-lifted elements; otherwise, turn to Step5;

Step5: $i = i+1$, turn to Step1;

Step6: repeat Step1- 5, till the current optimal solution can not update.

Contempt regular. The study adopts the regular based on fitness value. If all solutions of candidate muster are tabu solutions, liberate the best solution of candidate muster.

Ending principle. The study adopts iterative times by limited algorithm in advance as the ending principle, which refers to confirm a big enough positive number so as that the total iterative times don't exceed this number. Iterative times in advance can effectively control operation time of algorithm and is easy to operate.

Experimental Calculation and Analysis

The data of study is from reference [5, 6], utilize TBJ10 type container to load bulk cargos of 42 freight invoices. Quality and outer diameter of each cargo can be shown in reference. The maximum loading weight of TBJ10 type container is $G = 10t$. The maximum loading capacity is $V = 16.81m^3$. The cargo of No.1 freight invoice must be loaded firstly. And nature and package of cargo don't occur conflicting. Confirm the scheme of maximum loading rate.

Solution of Tabu Search Algorithm. After many trails, tabu search algorithm adopts the following parameters as part. The maximum iterative times are max_iter =100, tabu length is L =3-7, and candidate solution amount is 10. Randomly get solutions for 10 times. Here, the optimal loading schemes can be shown in Table 1.

Table 1 Optimal results by TS

No.	1 3 10 17 24 26
Loading utilization /%	93.61
Capacity utilization /%	99.70

Solutions by Genetic Algorithm. Reference [5] is adopted genetic algorithm to get the solution. The main parameters are as followings. Group scale is $N = 80$. The maximum iterative times is $K = 300$, crossover operator is $p_c = 0.95$, mutation operator is $p_m = 0.01$, initial temperature is T_0 =250, temperature coefficient $\delta = 0.89$, and randomly get the solutions for 30 times. The optimal loading weight and capacity utilization rate is 83.80% and 91.13%. The concrete loading schemes can be shown in Table 2.

Table 2 Optimal results on solving by GA

No.	1 9 18 23 28 33 40

Loading utilization /%	83.80
Capacity utilization /%	91.13

Solutions by Ant Colony Algorithm. Reference [6] is adopted by ant colony algorithm to get the solution through multi-times solving. The concrete parameters are as followings. $\alpha = 1$, $\beta = 5$, $\rho = 0.9$, $\tau_0 = 0.5$, $\eta_1 = \eta_2 = 1$, $m = 30$, $n_c = 200$. The optimal loading weight and capacity utilization rate are 81.19% and 98.75%. The concrete scheme can be shown in Table 3.

Table 3 Optimal results on solving by CA

No.	1 3 7 18 29 36
Loading utilization /%	81.19
Capacity utilization /%	98.75

Analysis on Three Algorithms. Compared the optimal scheme of reference [5, 6], experiments proved that this algorithm can achieve not only better calculating results, but also better calculation efficiency and quicker convergence rate. The concrete scheme can be shown in Table 4.

Table 4 Comparison among GA, ACS and This Algorithm

	Genetic Algorithm	Ant Colony Algorithm	This Algorithm
Loading utilization /%	83.80	81.19	90.32
Capacity utilization /%	91.13	98.75	99.35

Conclusions

In all, the algorithm has a strong search capability, high computational efficiency and high quality on algorithm solving. At the same time, the proposed algorithm doesn’t set opposition between utilization rate of loading weight and capacity.

Acknowledgment

This research was financially supported by the Ministry of Education of Humanities and Social Science Research Project of China (Grant NO. 12YJC630160).

References

[1] Michael Eley. Solving container loading problem by block arrangement. European Journal of Operational Research, Vol. 141(2002),p.393-409

[2] Tobias Fanslau, Andreas Bortfeldt. A Tree Search Algorithm for Solving the Container Loading Problem. INFORMS Journal on Computing Spring, Vol. 22(2010), p.222-235

[3] Leo Ho Wai Yeung, Wallace K. S. Tang. A Hybrid Genetic Approach for Container Loading in Logistics Industry. IEEE Transactions on Industrial Electronics, Vol. 52(2005), p.617-627

[4] A. Bortfeldt, H. Gehring. A hybrid genetic algorithm for the container problem. European Journal of Operational Research, Vol. 131(2001), p.143-161

[5] BU Lei, YIN Chuan-zhong, PU Yun. Genetic algorithm for optimal arrangement of general piece goods. Journal of Traffic and Transportation Engineering, Vol. 34(2004), P.84-87

[6] CAO Hongmei, GAO Li, ZHANG Tiangong. Ant algorithm for optimal loading of multi category goods. Computer and Communications, Vol. 26(2008), P. 11-14