An Inherited Tabu Search Algorithm for the Truck and Trailer Vehicle Scheduling Problem in Iron and Steel Industry

Lixin TANG and Kun LI

The Logistics Institute, Northeastern University, Shenyang, Liaoning Province, 110004, P. R. China.
E-mail: qhjytlx@mail.neu.edu.cn (L. Tang); icedewl@163.com (K. Li)

(Received on February 14, 2008; accepted on October 16, 2008)

The truck and trailer vehicle scheduling problem (TTVSP) involves vehicles and products allocation when the products are transported from warehouse to dock yard. The TTVSP with multiple types of vehicles which can be assigned to different tasks more than one time is considered and the vehicle can also transport the products that don’t absolutely suit it. For this TTVSP, the continuity of task and truck are also under the consideration based on the practical requirement. This problem is formulated as a mixed integer programming model to minimize the total cost by considering some practical issues. In this paper, the problem how to transport the last operation in appointed task is also analyzed and the property about the scheduling of last operation is put forward to optimize algorithm. An inherited composite neighborhood tabu search algorithm is developed to find a near-optimal scheduling where an initial solution is obtained based on the know-how knowledge. The results of experiment show that the proposed method could resolve the problem more effectively than current manual method.

KEY WORDS: TTVSP; iron and steel industry; inherited tabu search; composite neighborhood.

1. Introduction

In today’s iron and steel industry, the pressure is how to reduce the production cost and logistics cost efficiently in order to survive in vigorous market competition. According to statistic, the logistic cost is almost about 30% in whole cost of iron and steel industry in China. Due to the special character of the iron and steel industry, the transportation in factory is characterized as follows: 1) The gross of material is large and the output of inner transportation is five times than production; 2) There are more especial requirements for the means of delivery and technology by reason of high transportation frequency of each working procedure and high temperature of semi-manufactured goods. For example, the molten iron needs to be transported from blast furnace to electric cooeker by torpedo car.

According to the particularity of carrying material and special technology for the connection between working procedures, the scheduling of material conveyance is embarked in order to reduce the logistics cost. All kinds of products and semi-manufactured products have corresponding terminal warehouse and the products need to be transported from warehouse to stack yard for batch lading ship. Owing to the differences of product figuration, it needs to be used different carriages to load distinct products for ensuring the transportation safety. Figure 1 indicates the transportation of products from production warehouse to dock yard.

Before going into more details about the organization and content of this paper, let us provide an overview of the related existing literature. Vehicle scheduling problem (VSP) is firstly put forward by Dantzig and Ramser\(^1\) in 1959, but little research has addressed the truck and trailer vehicle scheduling problem as yet. The radical difference of them is the existing of complicated matching problem between truck and trailer. As an iron and steel scheduling problem, Tang et al.\(^2\) had studied the molten iron allocation problem (MIAP). For this kind of problem, researchers first formulated the molten iron allocation problem as an integer programming model and then reformulated it as a set partitioning model by applying the Dantzig–Wolfe decomposi-
tion. For this MIAP, the researchers applied an optimal algorithm to solve it and most test problems can be found the optimal solutions, but as a real time scheduling problem, it is rigorous for the algorithm time.

Carpaneto et al.\textsuperscript{3,4} introduced the Multiple Depot Vehicle Scheduling Problem (MD-VSP) in 1989, and two or more depots were specified and the only duty feasibility constraint requires that a crew starts and ends at the same depot. Forbes et al.\textsuperscript{3,4} and Löbel\textsuperscript{5,6} used a "multi-commodity" model for solving the MD-VSP. The main drawback of these two models is the difficulty of representing real-world constraints and costs. Funke\textsuperscript{7} provided a review of both classical and modern local search neighborhoods for vehicle routing and scheduling problem (VRSP). Their analysis shows how the properties of the partial moves and the constraints of VRSP influence the choice of an appropriate search technique. The last conclusion is that researchers can robust solve both ‘classical’ and more complex real-world problems applying efficient local search algorithm. The similar problem\textsuperscript{8} can be seen in 1994. They described a similar vehicle scheduling problem but didn’t consider the requirement on both truck continuity and task continuity, an underload case for a task at the last operation and effect of the number of operations for tasks on logistics cost. However, these features have been considered in our problem. Though they don’t present an integrated math model for their problem, but a branch and bound algorithm is used to obtain an optimal solution for small-scale problem.

In this paper, we develop a mixed math integer programming model by considering the practical constraints for the TTVSP and present the inherited tabu search algorithm with composite neighborhoods to solve it. In course of iterations, the inherited tabu search algorithm can remain the last scheduling partly to fast get to the better scheduling. The rest of the paper is organized as follows: In Sec. 1, we present the backdrop of the vehicle scheduling problem in iron and steel industry. In Sec. 2, we detailedly describe the characters of special vehicle scheduling problem in transportation. In Sec. 3, we formulate integer programming model according to the characters of the problem. In Sec. 4, we discuss the process of getting initial solution and then present details of inherited composite neighborhood tabu search algorithm. In Sec. 5, we present experiment results about different composite neighborhood and the conclusions.

2. Problem Description

In iron and steel industry, different products have different shapes such as slab, coil and steel tube and so on. All of the products need be transported from warehouse to dock yard for exporting. Each transportation task is characterized as follows:

1) Each task is restricted by quantity and time window.
2) The class of products and carriages. The appropriate carriage of the truck is used to transport different products in order to improve efficiency of whole system of transportation.

The number of available carriage is more than the number of trucks. A truck must be accompanied with a carriage when transporting products. The truck and carriage is one-to-one correspondence and can be freely joined and disjoined. In order to ensure the safety of the transportation, different products need relevant carriage for matching. This matching problem has three instances as follow:

1) The products match the carriage absolutely.
2) The products and the carriage are not entire match. Products can be stowed with the carriage which needs a setup for transportation convenience and safety.
3) The carriage cannot stow the products due to the shape of the products.

In order to reduce the premium of the setup and improve transportation efficiency, the carriage that entirely matches the products should be made the most of in scheduling, and for the same carriage with truck, it is best to avoid the change of the kinds of stowing products at every turn.

There are some special illustrations (such as Fig. 2) for practical TTVSP problem in iron and steel industry:

1) Because of the shape variety of products such as plate, coil and tube, we must consider the suitability of the products and the carriage.
2) It is all fully loaded transportation for the sake of improving transportation efficiency and reducing transportation cost except of the last time in a transport task.
3) Many of trucks can transport the products of the same task simultaneously.
4) It is a large-scale problem. If there is 30 tasks, 60 trucks and 96 time periods (a quarter hour is a time period), the potential instance is $2^{30 \times 60 \times 96} = 2^{172,800}$.
5) When a truck with carriage is assigned to a task, it can not serve for another task as far as the assigned task is fulfilled, furthermore the serving course is consecutive. This important feature can be defined as truck continuity. It exists in factory to employ the truck effectively and avoid changing carriage frequently.
6) If a task begins to process, it can not be intermittend until all products are transported to destination.
7) The truck has three kinds of carrying capacity and each has a fixed transport cost.

3. Mixed Integer Linear Program (MILP)

The problem in this paper can be come down to a fully loaded TTVSP problem of multi-vehicle, multi-type and multi-object. In order to explain clearly, we assume the matching problem of the truck and products to substitute for the relation of carriage and products. The following symbols are used to define the problem parameters and decision variables.
3.1. Description of Model

Known parameters:

- $\Omega$: The set of tasks, where $N$ is the number of tasks.
- $i$: Task identifier.
- $i_t$: The $i$th task, $i_t \in \Omega$.
- $i_z$: The $z$th task, $i_z \in \Omega$, where $i_z < i_t$.
- $j$: Truck identifier, $j = 1, 2, \ldots, J$, and $J$ is the number of trucks.
- $V$: The set of trucks.
- $T$: The total task time.
- $a_i$: The products coefficient of the $i$th task, $\forall i \in \Omega$.
- $D_i$: The number of operations of the $i$th task is processed by the $j$th truck, $\forall i \in \Omega, \forall j \in V$.
- $N_T$: The set of unused trucks, $N_T = V - \bigcup_{1}^{\Omega} V_{\text{ij}}$.
- $d_i$: The distance of the $i$th task, $\forall i \in \Omega$.
- $C_j$: The fixed cost of one operation by the $j$th truck, $\forall j \in V$.
- $p_j$: The matching coefficient of the $i$th task and the $j$th truck, $\forall i \in \Omega, \forall j \in V$.
- $p_i$: The number of time periods needed by the truck that processes one operation of the $i$th task, $\forall i \in \Omega$.
- $q_j$: The carrying capacity of the $j$th truck, $\forall j \in V$.
- $Q_i$: The transport quantity of the $i$th task, $\forall i \in \Omega$.
- $t_i^{\text{max}}$: The upper limit of time window for $i$th task, $\forall i \in \Omega$.
- $t_i^{\text{min}}$: The lower limit of time window for $i$th task, $\forall i \in \Omega$.
- $\omega_i$: The cost coefficients, $i = 1, 2, 3, 4, 5$.

Decision variables:

- $x_{ij}^t$: $\begin{cases} 1 & \text{if the } j \text{th task is assigned to the } i \text{th task in } t \text{ time period,} \\ 0 & \text{otherwise.} \end{cases}$
- $Z_j = \begin{cases} 1 & \text{if the } j \text{th truck is assigned}, \\ 0 & \text{otherwise.} \end{cases}$

Auxiliary decision variables:

- $t_i^L$: The start time of the $i$th task, $\forall i \in \Omega$.
- $t_i^U$: The end time of the $i$th task, $\forall i \in \Omega$.
- $t_j^L$: The start time of the $j$th truck that processes the $i$th task, $\forall i \in \Omega, \forall j \in V$.
- $t_j^U$: The end time of the $j$th truck that processes the $i$th task, $\forall i \in \Omega, \forall j \in V$.

Using the above notation, the TTVSP problem can be formulated as the following mixed integer programming model:

\[
\min \omega_i \sum_{j} x_{ij}^t + \omega_2 \sum_{i} \sum_{j} x_{ij}^t + \omega_3 \sum_{i} \sum_{j} \sum_{t} \rho_{ij} x_{ij}^t + \omega_4 \sum_{i} \sum_{j} \sum_{t} \rho_{ij} x_{ij}^t + \omega_5 \sum_{i} \sum_{j} \sum_{t} \rho_{ij} x_{ij}^t + \omega_6 \sum_{i} \sum_{j} \sum_{t} \rho_{ij} x_{ij}^t,
\]

subject to:

\begin{align*}
\sum_{j} x_{ij}^t & \leq Gp_j, \quad G \text{ is a great count.} \\
p_i & \text{The number of time periods needed by the truck that processes one operation of the } i \text{th task, } \forall i \in \Omega. \\
q_j & \text{The carrying capacity of the } j \text{th truck, } \forall j \in V. \\
Q_i & \text{The transport quantity of the } i \text{th task, } \forall i \in \Omega. \\
t_i^{\text{max}} & \text{The upper limit of time window for } i \text{th task, } \forall i \in \Omega. \\
t_i^{\text{min}} & \text{The lower limit of time window for } i \text{th task, } \forall i \in \Omega. \\
\omega_i & \text{The cost coefficients, } i = 1, 2, 3, 4, 5.
\end{align*}

Objective (1) of the model is composed of five parts. The first part is the number of the used truck; the second part is the total transportation distance; the third part is the total matching degree between trucks and tasks; the fourth part is the total task changing for the truck; the fifth part is the total operations for the trucks.

Constraints (2) ensure that each truck can process not more than one task in each time period. Constraints (3) indicate the bound of the number of truck. Constraints (4) explain that each task must be processed by at least one truck. Constraints (5) guarantee that each task must be completed by the trucks serving this task. Constraints (6) and (7) allow that different trucks can process the same task at different start time. Constraints (8) define the continuity of the truck. Constraints (9) explain the relation of two auxiliary decision variables. Constraints (10) and (11) are time window constraints of the task. There is a cryptic constraint of continuity for task that can be incarnated by constraints (8). For constraints (8), if a truck is assigned to a task, the assigned truck can not process other tasks until it completes this task, that is to say, it is impossible for intermitting the processing task until it has been completed.

In our model, the constraints on both truck continuity and task continuity are considered which are different from the approach in Ref. 8). Also the logistics cost concerning the number of operations of tasks is an important consideration for practical requirement in the objective function.

In order to reduce the logistics cost of objective function
effectively, it is significant for assigning which truck that has been assigned to this task to process the last operation of a task because of the different fixed cost of them. The next section has an analysis for specifying which truck is used to process the last operation of a task.

3.2. Dealing with the Last Operation of a Task

Supposing all of the trucks are the best for transporting products that is the instance of full-loading and the maximal underload operation number is one. We particularly analyze the diversity of $C_j$ induced by different value of $q_j$. For researching conveniently, a hypothesis is given that the underload instance just happens at the last operation of the task. A property about the last operation of last task is deduced by problem peculiarity as follows:

Property. For a random task, if the last operation is not full loading, the truck for processing this operation is decided by $j = \{j \mid \min \{\rho_j + C_j\}, j \in V_i\}$.

Proof. From $V_1$ to $V_5$ is the cost of every part in objective function by (1). For $V_1, V_2$ and $V_4$, assigning the truck for processing at the last operation of the $i$th task does not alter the cost value because of $j \in V_i$, however the cost value of $V_3$ and $V_5$ would be changed directly because of the different truck, so find a truck that has a minimal sum of $\rho_j$ and $C_j$ to process the last operation of a random task.

4. The Inherited Tabu Search Algorithm

Tabu search is proposed by Glover[16] in 1986 and has been proven to be an effective method to solve complicated combinatorial optimization problems. It is based on a neighborhood or local search technique and its principle is to make the recent moves tabu to avoid being wrapped at local optimum. The local search of this algorithm starts with an initial solution and stepwise updates it until reaches the stopping criteria.

4.1. Heuristics Algorithm for Initial Solution

For the transport department of the iron and steel industry, the cost of permanent assets of conveyance is a big one, thus the most important aim is how to make the most of the lesser truck number to accomplish the most transport task. We use the greedy heuristics algorithm based on know-how knowledge to generate an initial solution in order to minimize the used number of truck to obtain the preferable solutions. The detailed heuristic steps are as follows:

Step 1. Permute the ascending order of task based on $t_j^{\min}$ ($\forall i \in \Omega$) and obtain a task order. Set $V_i = \phi$ ($V_i \subseteq V$) for the $i$th task from scratch ($\forall i \in \Omega$). Denote $U$ is a set of used trucks and $W$ is a set of unused trucks, where $V = U \cup W$. Set $i = 1$.

Step 2. If $U = \phi$, go to step 4. If $\sum_{i \in V_i} \{q_i(t_i^{\max} - t_i^{\min})/\rho_i\} > Q$, $i = i + 1$, go to step 6; otherwise set $j_0 = J + 1$ and find $j_0 = \min_{j \in \{1, \ldots, J\} \cup \{0\}} \{j \mid \rho_j = 0\}$. If $j_0 < J$ and set $V_i = V_i \cup \{j_0\}$, go to step 2; otherwise, go to step 3.

Step 3. If $\sum_{i \in V_i} \{q_i(t_i^{\max} - t_i^{\min})/\rho_i\} \leq Q$, $i = i + 1$, go to step 6; otherwise set $j_0 = J + 1$ and find $j_0 = \min_{j \in \{1, \ldots, J\} \cup \{0\}} \{j \mid \rho_j = 0\}$. If $j_0 < J$, set $V_i = V_i \cup \{j_0\}$, go to step 3; otherwise, go to step 4.

Step 4. If $\sum_{i \in V_i} \{q_i(t_i^{\max} - t_i^{\min})/\rho_i\} > Q$, $i = i + 1$, go to step 6; otherwise set $j_0 = J + 1$ and find $j_0 = \min_{j \in \{1, \ldots, J\} \cup \{0\}} \{j \mid \rho_j = 0\}$.

5. If $\sum_{i \in V_i} \{q_i(t_i^{\max} - t_i^{\min})/\rho_i\} \geq Q$, $i = i + 1$, go to step 6; otherwise set $j_0 = J + 1$ and find $j_0 = \min_{j \in \{1, \ldots, J\} \cup \{0\}} \{j \mid \rho_j = 0\}$.

4.2. Neighborhood Structure of the Algorithm

In this problem, the scheduled tasks have strict time window constraint, and the truck must accomplish the assigned task at the beginning of the next task, therefore we construct three kinds of neighborhood structures and take into account the usability on processing time of the truck. Three kinds of neighborhood structures are defined as follows:

1) $\alpha$–$\alpha$ Swap Neighborhood

$V_i$ and $V_j$ are the sets of trucks processing the $i$th task and $i$th task respectively ($i \neq j$). If two trucks $j$ and $j'$ can swap, for the sake of decreasing computation time and improving efficiency, we use some rules to reduce the neighborhood size for accelerating the search process. These tasks have been ranked in non-descending order and $t_{ij}^{\min} \leq t_{ij}^{\max}$.

(1) $x_i^j = x_i^{j'} = 1$, $x_i^{j''} = x_i^{j'''} = 1$, if $t_i = t_{ij}$, then truck $j$ does not swap with truck $j'$.

(2) $x_i^j = x_i^{j'} = 1$, $x_i^{j''} = x_i^{j'''} = 1$, if $\rho_j < 0 < \rho_{j'}$, then truck $j$ does not swap with truck $j'$.

(3) $x_i^j = x_i^{j'} = 1$, $x_i^{j''} = x_i^{j'''} = 1$, if the $i$th task can not be completed after truck $j$ swaps with truck $j'$, then $j$ does not swap with $j'$.

(4) $x_i^j = x_i^{j'} = 1$, $x_i^{j''} = x_i^{j'''} = 1$, if $t_{ij} < t_{ij'}$ or $t_{ij'} < t_{ij''}$, then truck $j$ does not swap with truck $j'$.

Then we illustrate the rules as follows:

The number in rectangle figures represents the different trucks that are assigned to corresponding task; the number in parentheses represents the type of the relevant task.

For the first rule, the truck 1, 3 and 4 that are assigned to task 1 do not exchange each other because this kind of operation does not affect the objective function value. For the second rule, the truck 4 that is assigned to task 1 does not exchange with the truck 7 that is distributed to task 4 because the truck does not match task after the exchange ($\rho_{4,7} = 0$). For the third rule, it can not be exchanged if the truck 1 exchanges truck 6 that is assigned to task 3 which induces task 1 can not be completed within time window. For the fourth rule, the truck 3 can not exchange with truck 2 and 8 that are assigned to task 2 if $t_{i,j}^{\max} - t_{i,j}^{\min} > t_{i,j}^{\max}$. $t_{i,j}^{\min} - t_{i,j}^{\min}$.

2) $\alpha$–$NT$ Swap Neighborhood

The commutative objects belong to set $V_i$ and set $NT$ in this neighborhood structure. The relation of these two sets is $NT = V_i \cup V_2 \cup \cdots \cup V_N$. Such as the neighborhood
structure being interpreted above, we also apply the following rules to cut neighborhood size in order to accelerate the search process.

1. \( j \in V, j' \in NT \) and \( V \subseteq V' \), if truck \( j \) and \( j' \) are the same class including the same carrying capacity, the truck \( j \) does not swap with \( j' \).

2. \( j \in V, j' \in NT \) and \( V \subseteq V' \), if \( \rho_{i,j} < 0 \), then truck \( j \) does not swap with truck \( j' \).

3. \( j \in V, j' \in NT \) and \( V \subseteq V' \), if \( \rho_{i,j} > 0 \), then truck \( j \) does not swap with truck \( j' \).

2) RH Neighborhood

For the assigned truck \( j \) \( (j \in V) \) of the task \( i \) \( (i \in \Omega) \), the number of operation can be given by

\[
D_j = \frac{\sum_{t \in I^j} \lambda_{j,t} p_t}{\sum_{t \in I^j} x_{j,t}}
\]

\( p_t \) is the number of time periods needed by the truck that processes one operation of the \( t \)th task. In this neighborhood, we modify the operation number of the used truck to influence the objective function value. Judging the relation of \( t_{i,j}^0 < t_{i,j}^1 \), the operation number of truck \( j \) can be increased or decreased; if \( t_{i,j}^0 = t_{i,j}^1 \), the operation number of truck \( j \) only can be decreased. For minimizing objective function value, we only consider increasing the operation number about the instance of \( t_{i,j}^0 < t_{i,j}^1 \).

4.3. Inherited Speciality in Algorithm

1) \( \alpha-\alpha \) Swap Neighborhood

In the course of the local search, \( i_1, i_2 \in \Omega \) and \( i_1 < i_2 \), let \( j \in V, j' \in V', f(x) \) be current objective function value. When \( j \) and \( j' \) swaps each other, \( f(x) \rightarrow f(x^*) \). If \( f(x^*) < f(x) \), the new scheduling will be accepted. The change of \( t_{i,j}^1 \) will affect the scheduling of the truck \( j \), furthermore the influence can spread to all of the truck scheduling for the task after the \( i \)th. As a result, the scheduling before task \( i_2 \) can be remained and remainder tasks should be scheduled.

2) \( \alpha-NT \) Swap Neighborhood

It is similar to the instance of \( \alpha-\alpha \) swap neighborhood. Let \( j \in V, j' \in NT \), where \( i \in \Omega, f(x) \) is current objective function value. The swap move leads to the change of objective function value from \( f(x) \) to \( f(x^*) \). When the new scheduling is accepted, the scheduling before task \( i \) can be remained and remainder tasks should be scheduled.

3) RH Neighborhood

Like above, let \( j \in V, i \in \Omega, f(x) \) is current objective function value. When \( D_j \) is changed, the scheduling before task \( i \) can be remained and the rest tasks should be scheduled.

4.4. Tabu List/Tabu Object/Stopping Criteria

Tabu list: The size of the tabu list is a very important parameter of tabu search algorithm. The tabu list might be either fixed or variable. The size of the tabu list may be fixed to some values. This means that the tabu list contains fixed number of prohibited moves. Other kinds of tabu list are variable that stochastic adapts to the change of the iterated number in a known bound. This dynamic change is a peculiar disturbance for avoiding the algorithm gets in local minimums. The key is how to confirm the bound of list. In this paper the variable list is adopted to try to produce better results.

Tabu object: The selection of tabu object is based on three distinct neighborhood structures. For \( \alpha-\alpha \) swap neighborhood, the tabu object is \((i_1, j_1, i_2, j_2)\), where \( j_1 \in V \) and \( j_2 \in V \). \((i_1, j_2, i_2, j_1)\) is a tabu object of \( \alpha-NT \) swap neighborhood, where \( j_2 \in V \) and \( j_1 \in NT \). \((i, j)\) is another tabu object for RH neighborhood and \( j \in V \).

Stopping criteria: There are two common stopping criteria used to terminate the search procedure: 1) the maximum number of iterations \( t_{\text{max}} \) has been reached; 2) the maximum number of continuous iterations without improvement \( t_{\text{max}} \) on the value of the objective function has been reached.

4.5. The Detailed Steps of Inherited Tabu Search Algorithm

With the above description of the framework, the steps of inherited tabu search are described below:

- \( IN \) is the number of iterations, \( NIN \) is the number of iterations without improvement, \( t_{\text{max}} \) is the maximal number of iterations and \( t_{\text{max}} \) is the maximal number of iterations without improvement. The initial set of tabu list is \( \phi \).

Step 1. Create an initial solution \( S_0 \), using the heuristic algorithm represented above, \( S^* \) is the historical best solution.

Step 2. If \( IN < t_{\text{max}} \) and \( NIN < t_{\text{max}} \), go to step 3.

Step 3. \( S^* \) is a local optimization solution without tabu status. The solution \( S^* \) is a new adjacent solution of \( S^* \), and it can be obtained using a move. If \( S' < S^* \), \( S' \) can remain part of solution \( S^* \) and other part of it can be renewed using the heuristic algorithm.

Step 4. Judge \( S^* \) and \( S' \), if \( S' \) is better than \( S^* \), update \( S^* \) and tabu list, \( NIN = 0 \), \( IN = IN + 1 \), go to step 2; otherwise \( NIN = NIN + 1 \), go to step 2.

5. Computation Results

To test the performance of the inherited tabu search algorithm, it was employed to test algorithm using nine instances and every of it involved ten groups of data. Different single neighborhood structure and composite neighborhood are all used in experiments. For example, a composite neighborhood such as \((\alpha-\alpha)-(\alpha-\alpha-\alpha)\). \( \alpha-\alpha \) neighborhood structure is first used in local search and the second neighborhood of \( \alpha-\alpha-\alpha \) is applied when the search of first neighborhood is over. Since it is very difficult for human schedulers to obtain reasonable schedules within a patient horizon, the manual scheduling is simulated in the experiment. The current manual scheduling method is essentially a procedure assigning the unused trucks to the next task and resulting in more used trucks. It is a lack of assigning the used trucks to the appropriate task in order to reduce the
cost of the unused truck. All experiments implementation is used by C++ and run by PIV-3.00-Ghz PC.

5.1. Parameter Setup

The parameter of algorithm affects the result of computation intensively because it is delicacy such as in GA algorithm or SA algorithm. For solving this problem, a mass of experiments are done to obtain the appropriate parameters. In this research, the taboo list length is T_{list} \sim \text{rand}(5, 9); the cost coefficients of the object function are \alpha_1 = 0.4, \alpha_2 = 0.1, \alpha_3 = 0.1, \alpha_4 = 0.3; the maximal number of iteration without improvement is t_{max} = 1000; the maximal number of iterations without improvement is t_{max} = 200; the matching coefficient is defined of a matrix such as follows. If \rho_{ij} = 0, it means that the truck \ j matches for the task \ i; it is a constrained matching for them when \rho_{ij} = m (m > 0) and the number \ m implies the additional cost; if \rho_{ij} = n \ (n < 0), the truck \ j does not match the task \ i. In our experiments, the value of these parameters are \ m = 5 \text{ and } n = -1.

\[
\rho_{ij} = \begin{cases}
0 & m \ n
\end{cases}
\]

5.2. Experiment Data

Two kinds of data are used in experiments including task data and truck data. All of these are real data from transportation scheduling department of steel industry. Table 1 and Table 2 are examples of them.

5.3. Results

The computational results are shown in Table 3 and Table 4. Table 3 is the result of three kinds of single neighborhood algorithm and Table 4 shows the result of three kinds of composite neighborhood algorithm. The column \ J \ represents the number of available trucks; the column \ N \ represents the number of total tasks; the column \ RM \ represents the number of used trucks with respect to different neighborhoods; the column \ Man \ represents the values obtained by the simulated manual scheduling method; the column \ Obj \ represents the values of objective function of relevant neighborhood; the column \ CPU \ represents the computation time measured in second. The columns of \ RM, Obj, \text{ and CPU} \ represent the average value of ten groups of data with respect to each column.

Based on the results presented in Table 3, Table 4, Table 5 and Table 6, the following conclusions are summarized:

| Task code | Task quantity (ton) | Time window (time period) | Task type |
|-----------|---------------------|---------------------------|-----------|
| 1         | 2000                | 4-20                      | 0         |
| 2         | 3600                | 0-40                      | 1         |
| 3         | 3000                | 0-24                      | 2         |
| 4         | 1500                | 12-44                     | 0         |
| 5         | 3600                | 8-49                      | 1         |

| Truck code | Carrying capacity (ton) | Truck type | Cost (operation) |
|------------|-------------------------|------------|-----------------|
| 1          | 80                      | 0          | 50              |
| 2          | 80                      | 1          | 50              |
| 3          | 100                     | 2          | 55              |
| 4          | 100                     | 0          | 55              |
| 5          | 120                     | 1          | 60              |
| 6          | 120                     | 2          | 60              |

| N | J | \( (\alpha - \alpha) \) | \( (\alpha - \alpha) \) | \( (\alpha - \alpha) \) | \( (\alpha - \alpha) \) |
|---|---|----------------|----------------|----------------|----------------|
| 5 | 10 | 37909.4 | 5 | 36806.3 | 3 | 10297.9 | 2.772 | 4 | 21562.1 | 2.438 |
| 10 | 20 | 99967.1 | 12 | 92371.7 | 2.734 | 17 | 86748.2 | 2.297 | 13 | 83604.6 | 2.219 |
| 15 | 30 | 119738 | 14 | 110510 | 2.453 | 17 | 102228 | 3.438 | 14 | 104820 | 3.203 |
| 20 | 40 | 150948 | 18 | 143408 | 4.641 | 20 | 141632 | 4.859 | 19 | 138546 | 2.281 |
| 25 | 50 | 268142 | 25 | 188154 | 8.531 | 28 | 179589 | 5.216 | 24 | 164759 | 2.765 |
| 30 | 60 | 231609 | 28 | 223145 | 17.265 | 29 | 213083 | 15.656 | 25 | 208322 | 2.938 |
| 35 | 70 | 258993 | 35 | 253680 | 26.532 | 32 | 241271 | 27.063 | 32 | 240469 | 2.578 |
| 40 | 80 | 294237 | 37 | 289947 | 55.250 | 39 | 283847 | 65.875 | 35 | 271300 | 2.984 |
| 45 | 90 | 326470 | 41 | 316687 | 115.875 | 41 | 304829 | 158.734 | 40 | 294928 | 3.843 |

| N | J | \( (\alpha - \alpha) \) | \( (\alpha - \alpha) \) | \( (\alpha - \alpha) \) | \( (\alpha - \alpha) \) |
|---|---|----------------|----------------|----------------|----------------|
| 5 | 10 | 37909.4 | 5 | 36806.3 | 3 | 10297.9 | 2.772 | 4 | 21562.1 | 2.438 |
| 10 | 20 | 99967.1 | 12 | 92371.7 | 2.734 | 17 | 86748.2 | 2.297 | 13 | 83604.6 | 2.219 |
| 15 | 30 | 119738 | 14 | 110510 | 2.453 | 17 | 102228 | 3.438 | 14 | 104820 | 3.203 |
| 20 | 40 | 150948 | 18 | 143408 | 4.641 | 20 | 141632 | 4.859 | 19 | 138546 | 2.281 |
| 25 | 50 | 268142 | 25 | 188154 | 8.531 | 28 | 179589 | 5.216 | 24 | 164759 | 2.765 |
| 30 | 60 | 231609 | 28 | 223145 | 17.265 | 29 | 213083 | 15.656 | 25 | 208322 | 2.938 |
| 35 | 70 | 258993 | 35 | 253680 | 26.532 | 32 | 241271 | 27.063 | 32 | 240469 | 2.578 |
| 40 | 80 | 294237 | 37 | 289947 | 55.250 | 39 | 283847 | 65.875 | 35 | 271300 | 2.984 |
| 45 | 90 | 326470 | 41 | 316687 | 115.875 | 41 | 304829 | 158.734 | 40 | 294928 | 3.843 |

| Neighborhood name | \( (\alpha - \alpha) \) | \( (\alpha - \alpha) \) | \( (\alpha - \alpha) \) | \( (\alpha - \alpha) \) |
|-------------------|----------------|----------------|----------------|----------------|
| RM                | 4.92%          | 4.24%          | -3.72%         | 9.59%          |
| Obj               | 9.67%          | 14.46%         |               |                |

* means the increment of the used truck number than manual scheduling

Table 5. The average decrement of different single neighborhood on RM and Obj.

Table 6. The average decrement of objective function value from different neighborhoods.

\[
\begin{cases}
(\alpha - \alpha) & 4.24% \\
(\alpha - \alpha) & 4.24% \\
(RM) & 14.46% \\
(RM) & 11.09% \\
(RM) & 16.32% \\
(RM) & 14.76%
\end{cases}
\]
from them.

(1) The inherited tabu search algorithm with different neighborhood structures incorporating the proposed mathematical model can obtain much better scheduling than the current manual scheduling method on used truck number and the value of the objective function. The objective function value reflecting the total cost has much reduced such as the average decrement is 14.06% and the average reduced of the used truck number is 9.87%.

(2) In the three kinds of single neighborhood inherited tabu search algorithms, RH neighborhood can obtain the more perfect results than other neighborhoods for RM, Obj and CPU, for example of Table 5 as a result it can be made as the foundation neighborhood firstly used in composite neighborhood algorithm.

(3) The results of composite neighborhood algorithm are better than single neighborhood algorithm in all aspects except computation time. For composite neighborhood algorithm, (RH)-(α–α) and (RH)-(α–NT) are better than (α–α)-(α–NT) in all aspects because of the foundation neighborhood of RH is used in local search. The number of used truck of (RH)-(α–α) is fewer than (RH)-(α–NT).

(4) In the Table 6, we have obtained the decrement of Obj of inherited TS algorithm with different neighborhoods. It is obvious that the algorithm with (RH)-(α–α) neighborhood can find a better solution than others.

(5) Based on the results of the test problem, Fig. 4 shows the computational times of the inherited TS algorithm with different neighborhood change as the number of tasks. It can be observed that the computational time increases as the problem size gets larger. From the viewpoint of computational efficiency, the inherited TS algorithm with RH neighborhood is great faster than other neighborhoods just about no increment with problem size in definite confine.

6. Conclusion

Regarding the iron and steel industry transportation problem as context in this paper, a heuristic algorithm based on minimizing truck number is designed to obtain an initial solution for inherited tabu search algorithm. Aiming at this special TTVSP problem, the inherited tabu search algorithm with diverse composite neighborhoods is proposed. Experiment results show that the proposed inherited TS algorithm is better than the current manual scheduling method, in terms of both the quality and the efficiency of the solution, and the composite neighborhood based on RH neighborhood is better than others. It is useful to solve this kind of problem effectively.

Acknowledgement

This research is partly supported by National Natural Science Foundation for Distinguished Young Scholars of China (Grant No. 70425003), National 863 High-Tech Research and Development Program of China through approved No. 2006AA04Z174 and National Natural Science Foundation of China (Grant No. 60674084).

REFERENCES

1) G. B. Dantzig and J. H. Ramser: *Management Science*, 6 (1959), 80.
2) L. Tang, G. Wang and J. Liu: “A branch-and-price algorithm to solve the molten iron allocation problem in iron and steel industry”, Computers & Operations Research: Available online, 20 December 2005.
3) G. Carpaneto, M. Dell’Amico, M. Fischetti and P. Toth: *Networks*, 19 (1989), 531.
4) M. A. Forbes, J. N. Holt and A. M. Watts: *European Journal of Operational Research*, 72 (1994), 115.
5) A. Löbel: Ph. D. Dissertation, TU Berlin, (1998).
6) A. Löbel: *Management Science*, 44 (1998), 1637.
7) B. Funke and T. Grunert: *Journal of Heuristics*, 11 (2005), 267.
8) Y. Nakagawa, M. Ishizuka, H. Isogai and H. Hatano: “The Method of Vehicle Scheduling Problem in Iron and Steel Industry”, Japan Operation Research, 3, (1994).
9) F. Glover: *ORSJ J. Comput.*, 1 (1989), 190.