The Mix Integer Programming Model for Torpedo Car Scheduling in Iron and Steel Industry

Y.Y. Liu, G.S. Wang
The Logistics Institute
Northeastern University
Shenyang, China

***Abstract*-The scheduling for molten iron logistic optimization is to determine the distribution of molten iron from iron-making stage to steelmaking stage, the transportation routes of torpedo cars and locomotives such that the molten iron from blast furnaces can be delivered to the steelmaking shops on schedule and on quality. By taking the molten iron transportation process of an iron and steel enterprise as research background, the paper investigated the torpedo car scheduling problem, which is a key problem in the molten iron logistics optimization and developed a mixed integer programming model. The model was then solved by commercial optimization software CPLEX to obtain the torpedo car scheduling scheme. Meanwhile, another scheduling scheme was obtained by a heuristics based on the nearest neighborhood idea in which the earliest available torpedo car is used to service the un-serviced task whose tapping time is the earliest. Two schemes were compared to validate the efficacy and reasonability of the model.**

Keywords-torpedo car scheduling; mixed integer programming; heuristics; the nearest neighbourhood

I. INTRODUCTION

With the speeding up of global integration, the competitions of iron and steel industry are intensifying; hence it is solved to research a solution of scheduling optimization in production management. The schedule of molten iron logistic optimization is the problem of meeting the balance between supply from the blast furnaces and demand of the steel-making works for a fleet of torpedo car (TPC) to transport molten iron, and the following key elements including the tapping information of blast furnaces, the demand of converters, and the status of preprocess positions such as deslagging, desulfuration and dephosphorization, should be taken into consideration [1].

Torpedo car scheduling problem, as the key problem in the molten iron logistics optimization, calls for the design of a set of routes, all tasks are serviced by a set of identical torpedo car located at a central park with known demands and a set of operating time constrains are satisfied. Figure 1 shows the operation process such as tapping iron, heavy pot transportation and empty pot transportation of a torpedo car in the process of performing transportation tasks.

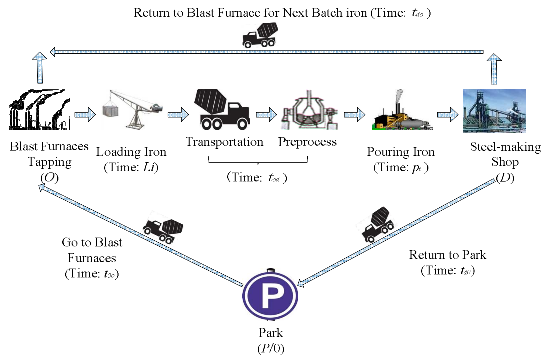

FIGURE I. PROCESS DIAGRAM OF TRANSPORTATION.

Italian scholar Baldacci proposed the mixed mathematical programming, which is a kind of accurate algorithm to solve VRP [2-5]. The VRP in the iron and steel enterprises could be summarized as the pickup and delivery problem with time window, called PDPTW. In the PDPTW, introduced by Savelsbergh and Sol [6], a fleet of identical vehicles has to be optimally routed and each task is to be serviced within a time window in addition to matching restrictions and precedence restrictions.

In this paper, we describe a mixed integer programming formulation based on the torpedo car scheduling problem, according to the site factors such as the supply and demand of molten iron and the distance between each station point.

II. DISCUSSED PROBLEMS

Compared with the molten iron allocation problem which solves the problem when will transport the molten iron from blast furnace to steel-making work, the torpedo car scheduling problem is the problem of designing optimal delivery routes for a fleet of torpedo car to transport molten iron with given demands. The following assumptions are needed before setting up model. There are some assumptions. i) Every route performed by a torpedo car must start and end at the park and the load carried must be less than or equal to the vehicle capacity. ii) Assuming that the information such as iron weight of production and demand, transportation time, loading time and pouring time are known. iii) Assuming that the number of locomotive which drags torpedo car is enough.

The simplified layout diagram of the rail tracks for molten iron transportation between the iron and steel-making plants is shown in figure 2. The problem considered in this paper is described as follows: An undirected graph G = (N, A), as a transportation network, is given where N $=O\cup D\cup\{0\}$ is the set of n+1 nodes and A $=\{(0,o):o\in O\}\cup\{(d,0):d\in D\}\cup\{(o,d):o\in O,d\in D\}\cup\{(d,o):o\in O,d\in D\}$ is the set of arcs. Node 0 represents the park, O denotes a set of origination sites and D denotes a set of destination sites. Each destination requires the molten iron from the origination and a set of identical vehicles are stationed at the park 0. With $o\in O$ and $d\in D$ is associated a travel time t_{od} ($t_{od,}(o,d)\in$ A), including three blast furnaces and two steel-making works and the figure 3 indicates the nodes of transportation task.

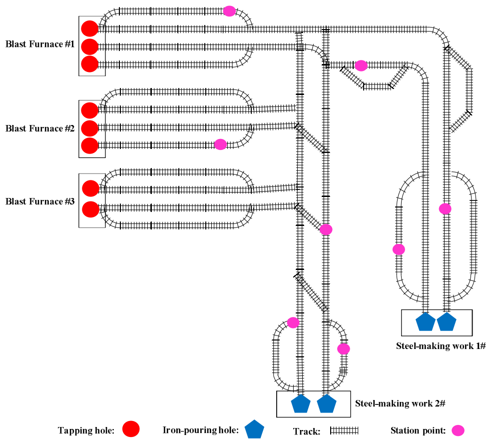

FIGURE II. LAYOUT OF THE RAIL TRACKS FOR TASK.

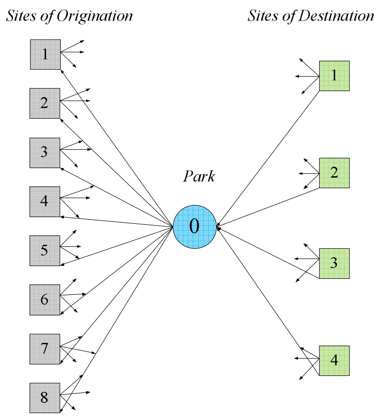

FIGURE III. NODES OF TRANSPORTATION MOLTEN IRON TRANSPORTATION .

The transportation tasks are composed of a set N = {1,…, n}. For each task $i\in N$, it is associated a loading station $o_i\in O$ and a pouring station $d_i\in D$. The time window $[a_i,b_i]$, where a_i and b_i represent the earliest and latest time to implement task i, that is, a_i is the begin of loading iron and b_i is the end of pouring iron. The operation time of loading and pouring are l_i and p_i, respectively. For a subset in each origination state denoted by $SN_o=\{i\in N\mid o_i=o\}$, the tasks must be picked up according to the given order $(i_o^1,i_o^2,...,i_o^{sn_o})$ (Here $sn_o=|SN_o|$) for molten iron loading operation; in the same way, for each subset of tasks in the destination denoted by $EN_d=\{i\in N\mid d_i=d\}$, the tasks must be picked up according to the given order $(i_d^1,i_d^2,...,i_d^{en_d})$ (Here $en_d=|EN_d|$) for molten iron pouring operation.

A fleet of v identical torpedo car of limited capacity stationed at the park has to fulfill transportation task. For each torpedo car $k\in V$ leaves the park 0 at time A_k, visit blast furnace and convert within its time window $[A_k,B_k]$, and return to the park before B_k.

III. MODELLING

A. *Model Parameters*

TABLE I .SYMBOL DESCRIPTION OF MODEL.

Symbol	**Description**
O	Sites of origination, a set of tapping holes located at bottom of blast furnaces
D	Sites of destination, a set of iron pouring holes at steel-making shops
t_{od}	The travel time from site o to site d
N	A set of transportation tasks
o_i	The origination site/tapping hole of task $i\in N$
d_i	The destination site/iron pouring hole of task $i\in N$
SN_o	A subset of tasks $SN_o\subseteq N$ originating from site o
EN_d	A subset of tasks $EN_d\subseteq N$ delivering to site d
$[a_i,b_i]$	The time window for the task $i\in N$
l_i	The loading time for the task i
p_i	The pouring time for the task i
V	A set of torpedo cars
$[A_k,B_k]$	The time window for the torpedo car $k\in V$

B. *Variable Definition*

$$x_{ijk}=\begin{cases}1,\text{if task } j \text{ is picked up directly after task } i \text{ by car } k\\0,\text{Otherwise}\end{cases}$$

$$x_{0jk}=\begin{cases}1,\text{if task } j \text{ is the first one picked up by car } k\\0,\text{Otherwise}\end{cases}$$

$$x_{i0k}=\begin{cases}1,\text{if task } i \text{ is the last one picked up car } k\\0,\text{Otherwise}\end{cases}$$

s_i = the loading start time of task i at its origination

e_i = the pouring end time of task i at its destination

C. Mixed Integer Programming Model

The mathematical formulation of the torpedo car scheduling problem is as follows:

$$\min \sum_{i\in N} e_i \tag{1}$$

$$\text{s.t.} \quad \sum_{k\in V}\sum_{j\in N\cup\{0\}} x_{ijk} = 1, \forall i \in N \tag{2}$$

$$\sum_{i\in N\cup\{0\}} x_{ijk} = \sum_{i\in N\cup\{0\}} x_{jik}, \forall j \in N, k\in V \tag{3}$$

$$\sum_{j\in N} x_{0jk} = \sum_{j\in N} x_{j0k} \le 1, \forall k \in V \tag{4}$$

$$e_i \ge s_i + t_{o_i d_i} + l_i + p_i, \forall i \in N \tag{5}$$

$$e_i + t_{d_i o_j} - M(1-\sum_{k\in V} x_{ijk}) \le s_j, \forall i \ne j \in N \tag{6}$$

$$A_k + t_{0o_j} - M(1-x_{0jk}) \le s_j, \forall j \in N, k \in V \tag{7}$$

$$e_j + t_{d_j 0} - M(1-x_{j0k}) \le B_k, \forall j \in N, k\in V \tag{8}$$

$$s_{i_o^l} + l_{i_o^l} \le s_{i_o^{l+1}}, \forall o \in O, l \in \{1,2,...,n_o\text{-}1\} \tag{9}$$

$$e_{i_d^l} \le e_{i_d^{l+1}} - p_{i_d^{l+1}}, \forall d \in D, l\in\{1,2,...,n_d\text{-}1\} \tag{10}$$

$$s_i \ge a_i, \forall i \in N \tag{11}$$

$$e_i \le b_i, \forall i \in N \tag{12}$$

$$x_{ijk} \in \{0,1\}, \forall i, \ne j \in N\cup\{0\}, k \in V \tag{13}$$

Objective (1) of the model is to minimize the total weighted accomplishment time. Constraints (2) specify that each task $i\in N$ must be serviced exactly once. Constraints (3) are viewed as the flow conservation constraint of net flow problems. Constraints (4) ensure the car must leave and return to the park. Constraints (5) define the time of completing a transportation task and the equation indicate the task is completed without waiting time. Constraints (6) guarantee that a station point can only be visited after the previous one on the same torpedo car completes processing. Constraints (7) and (8) ensure that each torpedo car satisfy the time window restrictions. Constraints (9) indicate that a car loads molten iron after the prior pot has done on the station of origination and constraints (10) indicate that a car pours molten iron after the prior pot has done on the station of destination. The time window restrictions of each transportation task are defined by constraints (11) and (12). Constraints (13) define (0-1) variables.

D. Generation of Random Data

The data associated with the mathematical model were generated at random according to the practical situations in the iron and steel industry. The details are shown as follows:

Setting eight tapping holes (*O*), four pouring-iron holes (*D*) and 20 transportation tasks (*N*) whose related parameters including *ID*, stations of tapping hole *s*(0,1,…,6,7) and pouring iron hole *d*(0,1,2,3), the earliest time a_i and latest time b_i to implement task, the operation time of loading l_i and pouring p_i. The specific data as shown in table 2.

The parameters related to torpedo car is *ID*, the earliest time *A* and latest time *B*, where *V*=5, *A*=0, *B*=2000.

The travel time is classified four classes, called T_1, T_2, T_3, T_4 representing the time from park to tapping hole, pouring iron hole to park, tapping hole to pouring iron hole and pouring iron hole to tapping iron respectively. The specific data are shown in table 3.

TABLE II .PARAMETER OF TRANSPORTATION.

ID	*s*	*d*	*a*	*b*	*l*	*p*
1	1	3	817	1027	53	46
2	5	3	868	1085	36	51
3	4	1	982	1200	40	40
4	7	2	986	1191	38	38
5	1	0	52	283	48	59
6	7	2	182	430	31	55
7	4	3	470	713	47	36
8	4	2	856	1092	51	42
9	2	2	793	1026	54	37
10	1	2	701	929	34	40
11	6	0	52	270	51	30
12	7	1	968	1209	4	48
13	2	1	122	334	44	58
14	0	1	462	681	40	35
15	5	0	645	885	44	44
16	6	0	369	581	42	38
17	0	3	97	338	45	59
18	3	2	188	413	57	44
19	2	1	942	1146	53	53
20	3	0	722	971	51	50

TABLE III .TRAVEL TIME FROM PARK TO TAPPING HOLES BY EMPTY TPC.

Tapping hole	0	1	2	3	4	5	6	7
Time	37	40	38	22	28	26	35	36

TABLE IV .TRAVEL TIME FROM POURING IRON HOLES TO PARK BY EMPTY TPC.

Pouring iron hole	0	1	2	3
Time	32	45	33	37

TABLE V .TRAVEL TIME FROM TAPPING HOLES TO POURING IRON HOLES BY HEAVY TPC.

Tapping / **Pouring**	0	1	2	3	4	5	6	7
0	51	40	56	52	56	48	35	34
1	50	54	50	44	44	49	42	60
2	48	43	55	33	31	48	54	49
3	33	55	34	46	48	57	37	44

TABLE VI .TRAVEL TIME FROM POURING IRON HOLES TO TAPPING HOLES BY EMPTY TPC.

Tapping / Pouring	0	1	2	3	4	5	6	7
0	39	34	34	30	39	27	36	21
1	32	37	35	26	34	24	28	39
2	24	20	31	21	37	22	26	34
3	35	36	30	31	33	20	31	31

IV. COMPUTATIONAL RESULTS

The model is then solved by CPLEX to obtain the scheme as follow:

The first torpedo car: $x_{0,11}=1$, $x_{11,14}=1$, $x_{14,9}=1$, $x_{9,4}=1$, $x_{4,0}=1$; The second torpedo car: $x_{0,17}=1$, $x_{17,20}=1$, $x_{20,19}=1$, $x_{19,0}=1$; The third torpedo car: $x_{0,13}=1$, $x_{13,16}=1$, $x_{16,2}=1$, $x_{2,0}=1$; The fourth torpedo car: $x_{0,5}=1$, $x_{5,18}=1$, $x_{18,15}=1$, $x_{15,1}=1$, $x_{1,12}=1$, $x_{12,0}=1$; The fifth torpedo car: $x_{0,6}=1$, $x_{6,7}=1$, $x_{7,10}=1$, $x_{10,8}=1$, $x_{8,3}=1$, $x_{3,0}=1$.

So the torpedo car scheduling schemes based on the MIP model is shown in figure 4.

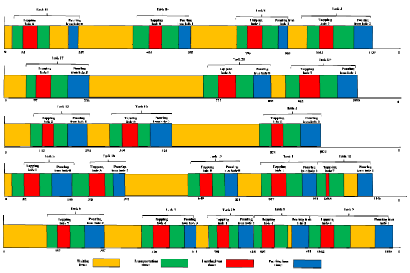

FIGURE IV. SCHEME OF THE TORPEDO CAR SCHEDULING MODEL.

V. HEURISTIC ALGORITHM

Academic for solving large-scale VRP generally adopts approximate algorithm based on heuristic [7-10]. In order to validate the efficacy of the model, another scheduling scheme is obtained by a heuristics based on the nearest neighborhood idea in which the earliest available torpedo car is used to service the un-serviced task whose tapping time is the earliest. The description of procedure heuristic as follows: ***Step 1*****:** Select a task whose tapping time is the earliest. Then arrange available torpedo car whose time window within the earliest time to implement the task and add the completing time to lower limit of time window. ***Step 2*****:** Then select an un-serviced task whose tapping time is the earliest and schedule a available torpedo car to implement the task according to step 1. ***Step 3*****:** Repeat the step 2 until all the tasks are serviced. ***Step 4*****:** The early car loads/pours iron firstly on the tapping/ pouring iron hole.

The torpedo car scheduling schemes based on the nearest neighborhood idea is shown in figure 5.

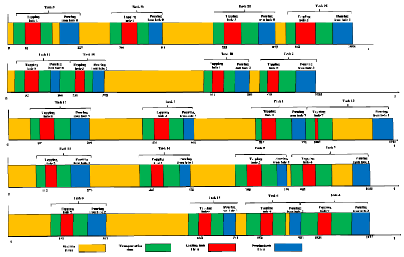

FIGURE V. SCHEME OF THE TORPEDO CAR SCHEDULING BASED ON THE HEURISTIC.

VI. CONCLUSIONS

In this paper, we have proposed two torpedo car scheduling schemes. The value of total weighted accomplishment time based on MIP model and heuristic method is 14251 and 14265, respectively. The results reveal the model efficiency.

ACKNOWLEDGEMENT

This research is partly supported by National Natural Science Foundation of China (Grant No.71202151).

REFERENCES

[1] Lixin Tang, Yue Zhao and Jiyin Liu. An Improved Differential Evolution Algorithm for Practical Dynamic Scheduling in Steelmaking-continuous Casting Production. IEEE Transactions on Evolutionary Computation, 2014, 18(2): 209-225.

[2] Baldacci R. Christofides N. and Mingozzi A., An Exact Algorithm for the Vehicle Routing Problem Based on the Set Partitioning Formulation with Additional Cuts. Mathematical Programming, 2008, 115(2):351-385.

[3] Baldacci R. and Maniezzo V., An Exact Method for the Car Pooling Problem Based on Lagrangean Column Generation. Operations Research, 2004, 52(3), 424-439.

[4] Baldacci R. and Mingozzi A., A Unified Exact Method for Solving Different Classes of Vehicle Routing Problems. Mathematical Programming, 2009, 120: 347-380.

[5] Baldacci R., Mingozzi A. and Calvo R., An Exact Method for the Capacitated Location-Routing Problem, Operations Research, 2011, 59(5), 1284-1296.

[6] Savelsbergh M. and Sol M., The General Pickup and Delivery Problem, Transportation Science, 1995, 29(1): 17-29.

[7] Niaz A. and Gábor N., A Heuristic Method for the Vehicle Routing Problem with Mixed Deliveries and Pickups. Journal of Scheduling, 2008, 11(2), 149-161.

[8] Cordone R, Calvo R., A Heuristic for the Vehicle Routing Problem with Time Windows. Journal of Heuristics, 2001, 7(2), 107-129.

[9] Osman I., Meta-strategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem. Annals of Operations Research, 1993, 41(4), 421-451.

[10] Ombuki B. and Nakamura M., A Hybrid Search Based on Genetic Algorithms and Tabu Search for Vehicle Routing. Brock University Technical Report, 2002, 5: 1-7.