J. Pokojski et al. (Eds.)

doi:10.3233/ATDE200089

A Hybrid Heuristic Optimization Approach for Green Flatcar Transportation Scheduling in Shipbuilding

Yini CHEN [a,b,1] , Zuhua JIANG [a,b] and Baihe LI [a,b]
[a] *School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai China, 200240, PR China*
[b] *Collaborative Innovation Center for Advanced Ship and Deep-Sea Exploration, Shanghai Jiao Tong University, Shanghai China,200240, PR China*

Abstract. To increase efficiency and decrease energy in fierce competition, higher standard of transportation scheduling mode for shipbuilding is necessary and urgent. By analyzing the "one-vehicle and one-cargo" transportation scheduling problem in shipbuilding, this paper proposes a bi-objective mathematical model and design a Multi-Objective Tabu Search algorithm(MOTS) to minimize total carbon emission and transportation time cost. Further, to improve the computation performance of the solution method, we combined NSGA-II and MOTS to design a hybrid heuristic algorithm. Computational experiments compare three optimizing approaches and reveal that MOTS and NSGAII-MOTS have certain advantages in terms of solution effect and convergence speed in large-scale instances. The case shows the proposed optimization approach can reduce carbon emissions by 61.22% for daily transportation.

Keywords. Transportation scheduling, green transportation, multi-objective optimization, hybrid heuristic optimization

Introduction

Shipbuilding adopts a typical pulling production method. "Segments" are basic operating units for ship construction process, which need to pre-process, assemble, outfit in different yards before general assembly and loading. Due to the heavy load, segment logistics is important for organizing the ship construction process flows. The transportation of ship blocks mainly depends on heavy flatcars, which are scarce resources in shipyards. The daily fuel consumption and carbon emission of flatcars are high. A large-scale shipyard in Shanghai produces 36 ships per year, with an average of 200 segments per ship, the basic operation of the flatcar is 160 times per day. Based on the average workload of 2km per time, about 2172kg CO_2 emission per day will be generated by flatcar transportation.

The value proposition of this paper in transdisciplinary systems engineering is embodied in energy, logistic, and computer science domains to achieve successful diffusion(10) . In particular, it is demonstrated in terms of time for traditional metrics and quantifying transport energy consumption, and a bi-objective mathematical model is

[1] Corresponding Author, Mail: ynchen@sjtu.edu.cn.

proposed to further explore a high efficiency and low energy consumption transportation scheduling mode.

Figure 1. Transport ship block by a flatcar.

1. Previous studies and techniques

Flatcar transportation scheduling problem in shipyards is different from traditional VRP. Nowadays, the research on the ship block transportation scheduling is systematizing at home and abroad. Meanwhile, the optimization algorithm are becoming mature, such as GA(10), greedy algorithm(10), meta-heuristic algorithm(10), ACO(10), etc.

Transportation is one of the main sources of greenhouse gas emissions, and green vehicle transportation problem has become a research hotspot. Table 1 summarizes and compares GVRP-related research in terms of types and solution methods. With the intensification of environmental pollution and scarcity of resources, green transportation has become an inevitable development trend of ship block transportation scheduling. Flatcars generate a large amount of carbon emissions. However, there are few researches on the green flatcar transportation scheduling in shipyards at home and abroad, and only a few scholars in China have studied green scheduling problem in container terminals.

Table 1. The comparison of GVRP-related research literature.

Autor	Problem types		Vehicle speed		Solution method
	Scheduling	Route Planning	Yes	No	
Li et al. [7] (2013)	√			√	Fuzzy multi-objective optimization algorithm
Guo Zhaoxia et al.[8] (2016)	√			√	Novel memetic algorithm
Salehi et al.[9](2017)	√		√		Novel constructive heuristic
Zhang et al. [10](2018)	√	√		√	ACO
Wang Yong et al. [11] (2019)	√	√	√		Multi-objective particle swarm optimization

2. Problem statement

2.1. Problem description

The problem of flatcars scheduling in shipyard based on OVOC mode can be described as follows: there are n transportation tasks and m flatcars. Each transportation task includes: task number, segment number, segment weight, start location, destination, and time window (when the task can be started). Each heavy flatcar includes: flatcar number, flatcar ID, load-bearing capacity. Each transport task must be performed within a time window by a heavy flatcar that meets its segmented weight requirements. The following Figure 2 is the distribution map of road junctions, the location of the yards, and parking location of a shipyard in Shanghai, China.

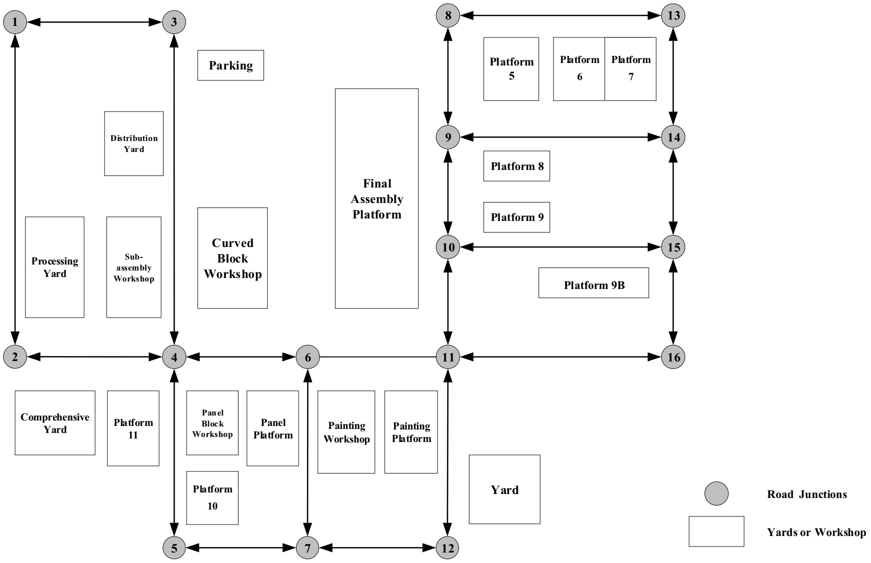


Figure 2. Distribution map of intersections and yards.

The basic assumptions of the problem are as follows:

1. All segments in the task table meet the load-bearing requirements of flatcars, and the order of tasks can be changed.
2. The flatcar can't be interrupted during the mission.
3. A single flatcar cannot transport multiple segments at the same time.
4. Without considering the road factors, such as the interference.
5. Set the loading and unloading time of the flatcar to a fixed value and add it to the task execution time.
6. The target yard must be able to accommodate the transported mission segments.

Based on the above assumptions, the problems to be solved are as follows:

1. Transport task sequence on each flatcar.
2. The optimal route for each flatcar.
3. The actual start time of each task.

2.2. Mathematical model

The factors that evaluate the quality of the flatcar driving route are: 1) the length of the route; 2) the number of turns during load driving. We use the depth-first search algorithm([6]) to traverse all feasible paths and consider the factors of turning.The definitions of model-related parameters and decision variables are shown in Table 2.

Table 2. The indices, parameters and decision variables.

Notation	Meaning
x_P	Abscissa of intersection p
y_P	Ordinate of intersection p

m_i	Number of turns when task i is performed
T	Task number set T=$\{1,2,\ldots,n\}$; $i,j \in T$, where $i \neq j$
F	Flatcar number set F=$\{1,2,\ldots,m\}$; $f \in F$
L	Flatcar speed set L={3,6}, the unit is m/s
$lsp, nlsp$	lsp represents load speed; $nlsp$ represents no-load speed of the flatcars
LT_i	Execution time of the task i
NLT_{ij}	No-load driving time to the starting point of task j after performing task i; where $i = 0$ or $j = 0$ represents that the flatcar departs from or returns to the parking lot
M	Infinite positive number
$[es_i, ls_i]$	The time window of task i; es_i is the time starting point at which the task can start executing; ls_i is the time ending point at which the task must start
$[E, L]$	Time window of the opening and closing of the parking lot
w_i	Segment weight of task i
cw_f	Load-bearing capacity of flatcar f
λ_1	The weight of no-load travel time for all tasks performed by the flatcar
λ_2	The weight of waiting time for the flatcar
efc	Transport efficiency of flatcar, set $efc = 0.8$
β	Specific constants of flatcar, set $\beta = 2.1072$
α	Specific constant of road ($\alpha \in [0.09, 0.2]$), set $\alpha = 0.0981$
f_e	The fuel emissions for diesel fuel, set $f_e = 2.621$
v	Conversion factor for fuel consumption per Joule energy ($v = \frac{1}{3.6\times10^6\times8.8}$)
dis_i	Distance between start and end of task i
dis_{ij}	Distance from the end of task i to the start of task j
TC	The time flatcar takes to make a turn, the unit is min / time
Decision Variables	
x_{oif}	0-1 decision variable for flatcar f to perform its first task i
x_{ijf}	0-1 decision variable for task i,j execution order on flatcar f
x_{iof}	0-1 decision variable for flatcar f to perform its final task i
y_{if}	0-1 decision variable for whether platform f performs task i
s_i	Actual start time of task i

The mathematical model to minimize total non-value-added transportation time and total carbon emission of the flatcars is presented as follows:

$$Min\ f = \lambda_1 \cdot (f_1 + f_2) + \lambda_2 \cdot f_3 \tag{2.1}$$

$$Min\ e = e_1 + e_2 \tag{2.2}$$

$$f_1 = \sum_{f\in F} \sum_{j\in T\setminus i} \sum_{i\in T} x_{ijf} \cdot NLT_{ij} \tag{2.3}$$

$$f_2 = \sum_{f\in F} \sum_{i\in T} x_{oif} \cdot NLT_{oi} + \sum_{f\in F} \sum_{i\in T} x_{iof} \cdot NLT_{io} \tag{2.4}$$

$$f_3 = \sum_{f\in F} \sum_{j\in T\setminus i} \sum_{i\in T} x_{ijf} \cdot (s_j - s_i - LT_i - NLT_{ij}) \tag{2.5}$$

$$e_1 = \sum_{f\in F} \sum_{j\in T\setminus i} \sum_{i\in T} \frac{f_e}{efc} \times x_{ijf} \times (\beta \times nlsp^2 + \alpha \times w_f) \times v \times dis_{ij} \tag{2.6}$$

$$e_2 = \sum_{f\in F} \sum_{i\in T} \frac{f_e}{efc} \times y_{if} \times [\beta \times lsp^2 + \alpha \times (w_f + w_i)] \times v \times dis_i \tag{2.7}$$

$$\sum_{i\in T} x_{oif} = 1, \forall f \in F \tag{2.8}$$

$$\sum_{i\in T} x_{iof} = 1, \forall f \in F \tag{2.9}$$

$$s_i + LT_i + x_{ijf} \cdot NLT_{ij} - s_j \leq (1 - x_{ijf}) \cdot M, \forall i,j \in T, i \neq j, f \in F \tag{2.10}$$

$$es_i \leq s_i \leq ls_i, \forall i \in T \tag{2.11}$$

$$s_i + LT_i + NLT_{io} \leq L, \forall i \in T \tag{2.12}$$

$$E \leq s_i - NLT_{oi}, \forall i \in T \tag{2.13}$$

$$w_i \leq \sum_{f\in F} y_{if} \cdot cw_f, \forall i \in T, f \in F \tag{2.14}$$

$$\sum_{j\in T\setminus i} x_{jif} + x_{oif} = y_{if}, \forall i \in T, f \in F \tag{2.15}$$

$$\sum_{j\in T\setminus i} x_{ijf} + x_{iof} = y_{if}, \forall i \in T, f \in F \tag{2.16}$$

$$x_{ijf} + x_{jif} \leq 1, \forall i,j \in T, i \neq j, f \in F \tag{2.17}$$

$$\sum_{f\in F} y_{if} = 1, \forall i \in T \tag{2.18}$$

Objective functions are presented in (2-1) and (2-2). Objective (2-1) represents the minimizing total non-value-added transportation time of the flatcars and (2-2) represents the total carbon emission of the flatcars. Objective (2-1) contains three parts. It includes the no-load travel timef_1, f_2 and total waiting time f_3.The formula of f_1 is shown in (2-3). It represents the no-load travel time between two adjacent tasks performed by the flatcar. In (2-4), f_2 represents the no-load travel time for the flatcar exiting and returning the parking lot. In (2-5), f_3 represents the waiting time of the flatcar arriving earlier than the time window. Objective (2-2) contains two parts. Carbon emissions of no-load flatcar traveling between two adjacent tasks is show in (2.6). In (2.7), e_2 represents carbon emissions of load driving in tasks.

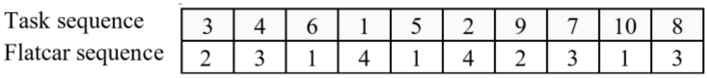
Constraint (2-8), (2-9) ensure that each flatcar only has one first and final task and (2.18) ensures that each task is performed by one flatcar. Constraint on the start time between the adjacent tasks is done by (2-10). Constraint (2-11) represents the task time window constraint. Constraint (2-12), (2-13) limit the time window of the parking lot. Constraint (2-14) ensures the task segment weight and flatcar load-bearing capacity. Constraint (2-15), (2-16) put limitation on the number of times a task appears in each flatcar. Constraint (2-17) indicates that the adjacent tasks cannot be repeatedly executed.

3. NSGAII-MOTS Hybrid Optimizing Approach

The MILP model established in this paper is based on the research of Li Baihe et al([6]). The increasing green goal of carbon emission determines that we need to solve the multi-objective optimization problem. This paper designs a multi-objective tabu search (MOTS) algorithm, and proposes a hybrid optimization algorithm combining NSGA-II and MOTS. In NSGAII-MOTS algorithm, NSGA-II is used to obtain a better solution set, and the optimal solution is input into MOTS algorithm to continue solving.

The chromosome is designed as two one-dimensional arrays based on positive integers, which respectively represent the task sequence and the flatcar sequence. During decoding, each task is assigned to corresponding flatcar according to chromosome coding, and the task order on the flatcar means the execution order.

We propose two methods of constructing neighborhood solutions: (1) Local search for task sequence. The neighborhood solution is obtained by exchanging the position of two tasks, which are performed by a randomly selected flatcar. (2) Local search for flatcar sequence. A task is randomly selected, and the flatcar corresponding to it is replaced with a newly generated flatcar that meet the weight requirement. In each iteration, when generating neighborhood solutions, the above two methods are selected according to the fixed ratio.



Task sequence	3	4	6	1	5	2	9	7	10	8
Flatcar sequence	2	3	1	4	1	4	2	3	1	3

Figure 3. Chromosome coding.

In our hybrid optimizing approach, NSGA-II is used to get a high-quality solution, and then MOTS is used to continue the search. The specific steps are shown in Figure 4.

Step1: Input the information of the task, flatcar, and coordinates of yards, workshop, platform and intersection. Initialization parameters: population size ($popsize$), single-point crossover rate (p_c), exchange mutation rate (p_m), elitism preservation rate (p_r), maximum iterations, unimproved times and iteration times.

Step2: Randomly generate the task sequence, once for each task, and the flatcar meeting the weight constraint is generated randomly for each task. The population size is popsize, and these random individuals formed the initial population.

Step4: Calculate non-dominated rank of each individual in the population.

Step5: Calculate crowding distance of each individual in the population.

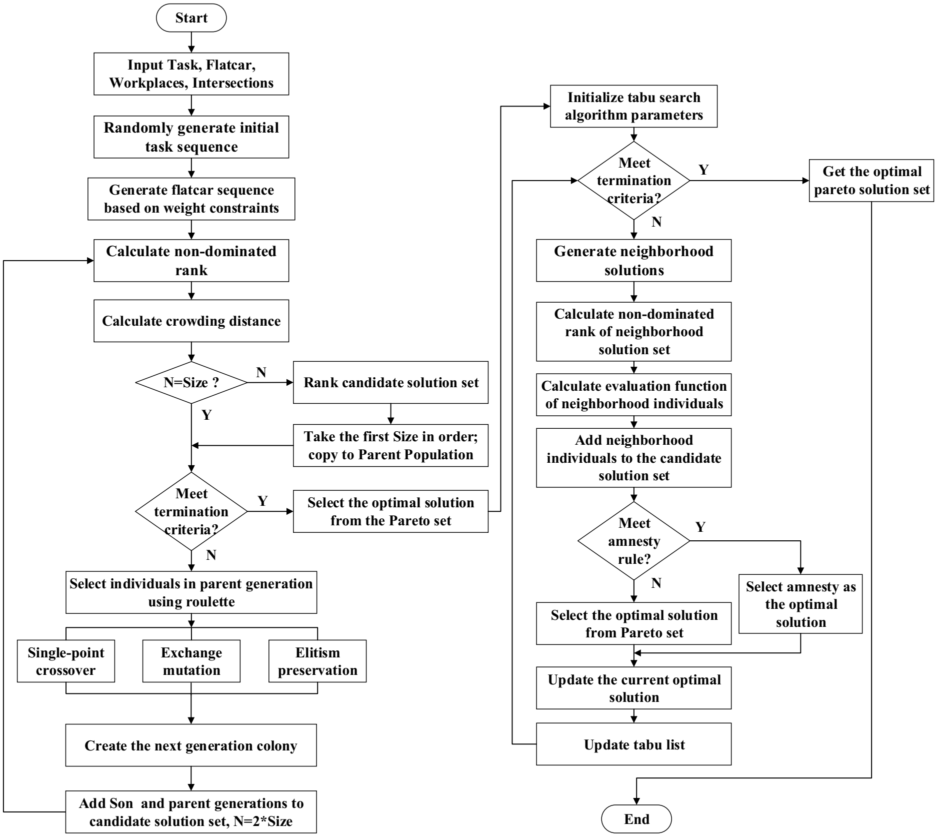

Start
Input Task, Flatcar, Workplaces, Intersections
Randomly generate initial task sequence
Generate flatcar sequence based on weight constraints
Calculate non-dominated rank
Calculate crowding distance
N=Size ?
N
Rank candidate solution set
Take the first Size in order; copy to Parent Population
Y
Meet termination criteria?
Y
Select the optimal solution from the Pareto set
N
Select individuals in parent generation using roulette
Single-point crossover
Exchange mutation
Elitism preservation
Create the next generation colony
Add Son and parent generations to candidate solution set, N=2*Size
Initialize tabu search algorithm parameters
Meet termination criteria?
Y
Get the optimal pareto solution set
N
Generate neighborhood solutions
Calculate non-dominated rank of neighborhood solution set
Calculate evaluation function of neighborhood individuals
Add neighborhood individuals to the candidate solution set
Meet amnesty rule?
Y
Select amnesty as the optimal solution
N
Select the optimal solution from Pareto set
Update the current optimal solution
Update tabu list
End

Figure 4. Hybrid optimization algorithm flow.

Step6: If the population number $N = popsize$, proceed to step 7; otherwise, go directly to step 11.

Step7: If the times meet the termination criteria, go directly to step 12. If not, proceed to step 8.

Step8: Use roulette method to select individuals in the parents.

Step9: Use single-point crossover to generate $popsize \cdot p_c$ offspring, and use exchange mutation to generate $popsize \cdot p_m$ offspring, and then use elitism preservation strategy to generate $popsize \cdot p_r$ offspring, where $p_c + p_r + p_m = 1$.

Step10: The offspring and parent are mixed into the candidate population. The iteration time is increased by 1, and go back to step 4.

Step11: The candidate population is sorted in ascending order of non-dominant rank, and descending order of crowding distance. Take the first $popsize$ in order to substitute the parent population.

Step12: Use NSGA-II to find the optimal solution as the initial solution of MOTS algorithm (current iterative solution), and initialize the parameters: neighborhood space n_size, tabu length, maximum iteration times, and iteration times.

Step13: If the times meet the termination criteria, the algorithm solution ends. If not, proceed to step 14.

Step14: Use the current iterative solution to generate n_size neighborhood solutions, and calculate the non-dominated rank and the evaluation function of each individual.

Step15: If the solution meets amnesty rule, update the current optimal solution with amnesty.

Step16: Update the tabu list, if the current solution is feasible, then update the optimal solution; otherwise, do not update. The iteration time is increased by 1 and go back to step13.

4. Numerical experiments and discussions

4.1. Algorithm comparison analysis

This section describes computational results to compare three optimization algorithms which are implemented in Java. The computer running the test is configured with Intel (R) -64 Core (TM) I5-7200U CPU @ 2.50GHz 2.71GHz.

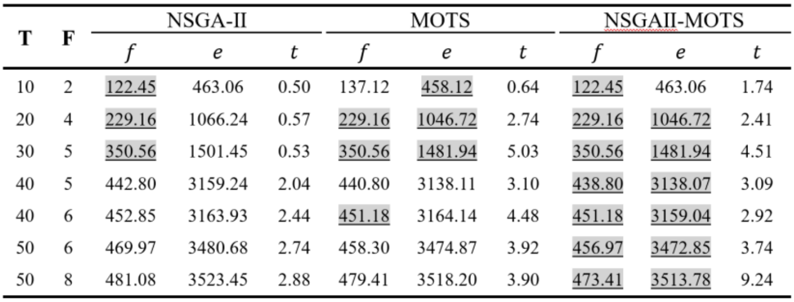
The test datasets are encoded as"$TxFy$", where"x" is the number of tasks, and"y" is the number of flatcars. By setting different combinations of task numbers (T=10,20,30,40,50) and flatcar numbers (F=2,3,4,5,6,7,8), the solving effects of three different algorithms are compared. The main parameter settings of NSGA-II and MOTS are shown in the Table 3. The result is the average value of 5 times for each test.

Table 3. Default parameter settings of the NSGA-II.

Parameters	Values
Population size	100
Maximal generation/iteration	NSGA-II (300), MOTS (300), NSGAII-MOTS (50+300)
Crossover / Mutation rate	0.4/0.4
Retain Elite probability	0.2
Neighborhood space size	500
Tabu length	300

The test results are shown in Table 4, where f represents the total non-value-added transportation time of the flatcars (unit: min), and e represents the total carbon emission of the flatcars (unit: g). The results will be analyzed in next section.

Table 4. Results about different tasks and flatcars.



T	F	NSGA-II			MOTS			NSGAII-MOTS		
		f	e	t	f	e	t	f	e	t
10	2	122.45	463.06	0.50	137.12	458.12	0.64	122.45	463.06	1.74
20	4	229.16	1066.24	0.57	229.16	1046.72	2.74	229.16	1046.72	2.41
30	5	350.56	1501.45	0.53	350.56	1481.94	5.03	350.56	1481.94	4.51
40	5	442.80	3159.24	2.04	440.80	3138.11	3.10	438.80	3138.07	3.09
40	6	452.85	3163.93	2.44	451.18	3164.14	4.48	451.18	3159.04	2.92
50	6	469.97	3480.68	2.74	458.30	3474.87	3.92	456.97	3472.85	3.74
50	8	481.08	3523.45	2.88	479.41	3518.20	3.90	473.41	3513.78	9.24

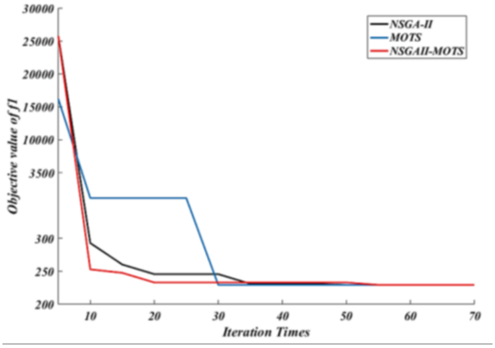
NSGA-II has a strong global search ability, but it is easy to fall into a local solution. The tabu search algorithm relies on the tabu list, and has the ability to jump out of local solutions. Meanwhile, different methods of constructing neighborhood also increase the diversity of neighborhood space, which has a good solution effect and a fast convergence speed.

The comparison of results in Table 4 can draw the following conclusions:

a) With the increasing of problem size, MOTS can obtain a better Pareto solution set than NSGA-II in the same iteration times;

b) When the scale is small (T= 10,20), both MOTS and NSGAII-MOTS can obtain the better results. When the scale is large, such as T = 40, the solution effect of the hybrid algorithm is significantly better than NSGA-II.

From the perspective of algorithm convergence, further compare the solution quality of the three algorithms. Taking $T20F4$ as examples, the relationship between the two objective function values and the iteration times is shown in Figure 5,6. It is clearly observed that NSGA-II has not converged in the set iteration times for 20 tasks, while for MOTS both two targets have converged at the 30th generation, and the hybrid algorithm has converged at the 60th generation. Within the iteration times set by the algorithm, both MOTS and hybrid algorithms can obtain better objective function values.



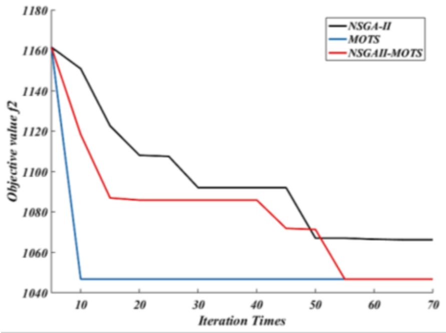


Figure 5. Convergence analysis of objective f.

Figure 6. Convergence analysis of objective e.

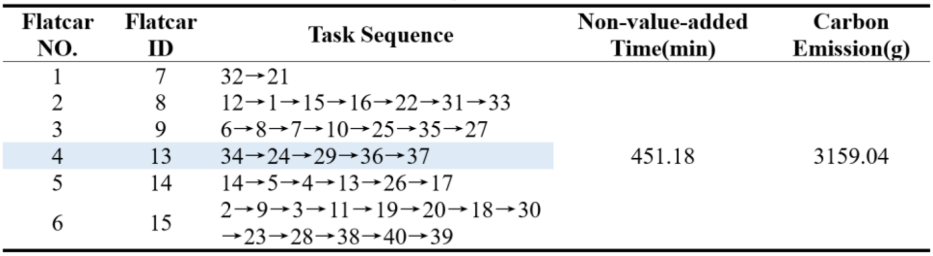
4.2. Real-word case study of emission reduction benefits

Taking the actual scheduling task of a shipyard in Shanghai as an example, to verify the feasibility and effectiveness of the MILP model and hybrid algorithm proposed in this paper. The number of tasks is 40.We chose $T40F6$ dataset, while the detail of NO.7,8,9,13,14,15 flatcars is in Table 5. Randomly chosen one in Pareto set for example, the scheduling results is shown in Table 6,7.

Table 5. Task list for T20 and Flatcars information.

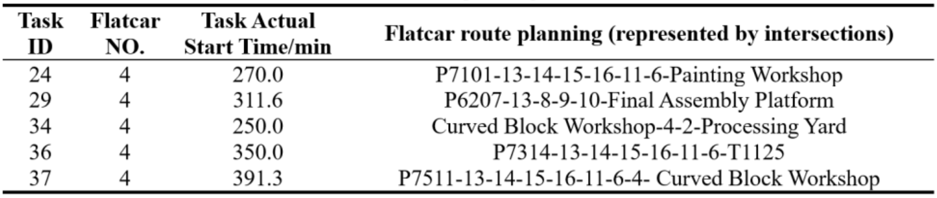
Task ID	Start	End	Time Window		Weight	Task ID	Start	End	Time Window		Weight	Flatcar ID	Load-Capacity
			Earliest	Latest					Earliest	Latest			
1	P6207	T1101	0	120	238	11	P6207	T1125	50	200	220	1	200
2	T1805	T1301	0	120	244	12	CBW	FAP	50	200	222	3	200
3	P6207	PW	0	120	264	13	P6207	PW	50	200	226	5	200
4	P7101	PW	0	120	276	14	CBW	SAW	50	200	226	6	200
5	P7107	FAP	0	120	280	15	P7107	FAP	50	200	242	7	250
6	T1805	SAW	0	120	283	16	P7314	T1125	100	220	258	8	270
7	P6207	SAW	0	120	283	17	P7511	CBW	100	220	276	9	325
8	P6207	PW	0	120	288	18	P7314	Yard	100	220	400	13	380
9	P6207	FAP	0	120	400	19	P6207	T1101	100	220	319	14	380
10	P6207	FAP	0	120	220	20	T1805	P7314	100	220	326	15	425

Table 6. Obtained task sequence results for each flatcar.



Flatcar NO.	Flatcar ID	Task Sequence	Non-value-added Time(min)	Carbon Emission(g)
1	7	32→21	451.18	3159.04
2	8	12→1→15→16→22→31→33		
3	9	6→8→7→10→25→35→27		
4	13	34→24→29→36→37		
5	14	14→5→4→13→26→17		
6	15	2→9→3→11→19→20→18→30 →23→28→38→40→39		

Table 7. Obtained route planning for flatcar NO.4.



Task ID	Flatcar NO.	Task Actual Start Time/min	Flatcar route planning (represented by intersections)
24	4	270.0	P7101-13-14-15-16-11-6-Painting Workshop
29	4	311.6	P6207-13-8-9-10-Final Assembly Platform
34	4	250.0	Curved Block Workshop-4-2-Processing Yard
36	4	350.0	P7314-13-14-15-16-11-6-T1125
37	4	391.3	P7511-13-14-15-16-11-6-4- Curved Block Workshop

According to the latest EU emission standards in early 2019, heavy cargo trucks are required to reduce CO2 emissions by 30% by 2030. Calculated based on the average workload of 2km per time, heavy fuel truck fuel consumption of 25.9L / 100km, and fuel emission factor of 2.621kg / L, the comparison of the optimal scheduling emission obtained in this paper and standard emission is shown in Table 8. We can conclude that the optimization of carbon emissions can meet the latest emission reduction standards.

Table 8. Carbon emissions comparison.

Standard Emission(g)	Generated Emission(g)	Decrease Percentage
8146.07	3159.04	61.22%

5. Conclusion

In order to actively respond to the call of green shipbuilding in China and achieve a high efficiency and low energy consumption transportation scheduling mode, we propose a bi-objective mathematical model for OVOC transportation scheduling problem, and design a NSGAII-MOTS algorithm. The NSGA-II and NSGAII-MOTS have different advantages according to the numerical results. The NSGA-II is competitive in computation time and can find Pareto solution for small-scale instances. However, the NSGAII-MOTS is absolutely competitive in terms of solution effect and convergence speed, and is suitable for optimizing large-scale and complex instances. The non-value-added time of 6 flatcars is reduced to 8h/day and carbon emission reduction benefits are obvious with the decrease percentage of 61.22%.

Based on the developed methods, the following practical features can be further studied to improve the applicability of the algorithms. (1) Two synchronizing flatcars transport one overweight segment can be considered, the corresponding coding method and time update mechanism are adjusted accordingly. (2) The road interference exists during the transportation of flatcars. Some roads can only travel with one flatcar at the same time. (3) Advanced computing technologies, e.g. cloud computing, can be used to improve the computation speed. (4) In order to highlight the application value of transdisciplinary engineering, we can consider introducing knowledge-based scheduling method, which can store knowledge in different fields to assist in scheduling decisions.

References

[1] A.M. Madni, Transdisciplinary systems engineering || looking to the future, Springer Nature Switzerland, 2018.

[2] C. Wang, Y. Mao and J. Shin, Ship block transportation scheduling approach based on genetic algorithm. *Journal of Shanghai Jiaotong University*, 2017, Vol. 51(003), pp. 338-343.

[3] C. Wang, Y. Mao, B. Hu, Z. Deng and J.G. Shin, Ship block transportation scheduling problem based on greedy algorithm. *Journal of Engineering Science & Technology Review*, 2016, Vol. 9(2), pp. 93-98.

[4] C.M. Joo and B.S. Kim,. Block transportation scheduling under delivery restriction in shipyard using meta-heuristic algorithms. *Expert Systems with Applications*, 2014, Vol. 41(6), pp. 2851-2858.

[5] B.S. Kim and C.M. Joo, Ant colony optimisation with random selection for block transportation scheduling with heterogeneous transporters in a shipyard. *International Journal of Production Research*, 2012, Vol. 50(24), pp. 7229-7241.

[6] B. Li, Z. Jiang, N. Tao and L. Meng, Research on dispatch of blocks between stockyards based on hybrid optimization algorithm. *Journal of Harbin Engineering University*, 2018, Vol. 39(12), pp. 157-164.

[7] X. Li, D. Wang, K. Li and Z. Gao, A green train scheduling model and fuzzy multiobjective optimization algorithm. *Applied Mathematical Modelling,* 2013, Vol. 37(4), pp. 2063–2073.

[8] Z. Guo, D. Zhang, H. Liu, Z. He and L. Shi, Green transportation scheduling with pickup time and transport mode selections using a novel multi-objective memetic optimization approach. *Transportation Research Part D: Transport and Environment*, 2016, S1361920916000146.

[9] M. Salehi, M. Jalalian and M.M.V. Siar, Green transportation scheduling with speed control: trade-off between total transportation cost and carbon emission. *Computers & Industrial Engineering*, 2017, S0360835217304370.

[10] S. Zhang, W. Zhang, Y. Gajpal and S.S. Appadoo, Ant Colony Algorithm for Routing Alternate Fuel Vehicles in Multi-depot Vehicle Routing Problem, In: K. Deep, M. Jain, S. Salhi (eds) *Decision Science in Action*, 2018, Springer, Singapore, pp. 251-260.

[11] W. Wang and W. Yun, Scheduling for inland container truck and train transportation, *Int. J. Prod. Econ.* , 2013, Vol. 143 (2), pp. 349–356.

[12] M. Eskandarpour, E. Nikbakhsh, S.H. Zegordi, Variable neighborhood search for the bi-objective post-sales network design problem: a fitness landscape analysis approach. *Comput. Oper. Res.*, 2014, Vol. 52, pp. 300–314.