http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (18): 4783-4789, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.4783.4789
© 2013 Asian Network for Scientific Information

A Multi-objective Programming Method for Vehicle Dispatching in Container Terminal

[1]Yang Xiaoming, [2]Zhao Ning, [3]Mi Chao, [2]Shu Fan and [2]Liu Haiwei
[1]Logistics Research Center,
[2]Logistics Engineering College,
[3]Container Supply Chain Technology Engineering Research Center,
Shanghai Maritime University, 201306, Shanghai, China

Abstract: It is a fundamental decision making process in container terminals to allocate container transporting works among vehicles. Several categories of methods such as mathematical programming, queuing theory, network models, or Markov decision making as well as heuristics are employed in the research. In this study a multi-objective programming method is proposed to resolve the problem in container terminal. The objective is the minimization of the total working time of the vehicles as well as the associated total cost of the travel. Then a genetic algorithm is developed to resolve the problem. Numerical tests are carried out and the results show the effectiveness and feasibility of the algorithm.

Key words: Multi-objective programming, container, quay side transport, genetic algorithm

INTRODUCTION

The vehicle dispatching is one of the most important decision making process in the container terminal. The efficiency of the travel to some extent determines the total working time of loading and unloading containers from the vessels. The decision-making problem for vehicle dispatching in container terminal has been widely recognized in the previous literature. Different algorithms were applied to enhance the operational efficiency of a container yard.

Zhao *et al.* (2002) presented a dynamic vehicle routing programming model, in which the routing between two objects is changeable and the blocking of travel is taken into account. The shortest path problem is resolved by a greedy algorithm. Mitrovic-Minic *et al.*(2000) focus on the simultaneously vehicles routing programming. The decision making process is divided into several sub-problems. The whole solution is combined with the sub-solutions. Bish (2003) proposes a heuristic method to solve the problem of dispatching vehicles to containers, determining storage location for each discharging container while scheduling loading and discharging operations on QCs. Kim and Bae (2004) propose an extended approach for AGV dispatching with a network-based MIP model. The objective is the minimization of the total idle time of a quay crane resulting from late arrivals of AGVs and the total travel time of the AGV. It is assumed that storage locations of containers and schedules for (un)loading operations are given. The extension considers a pooled dispatching strategy taking multiple quay cranes and dual-cycle operations of AGVs into account. Lv and Zhang (2004) describe an integer programming model to solve the unloading sequence decision making problem. Two heuristic algorithms, look ahead and look backward search algorithms are presented to solve the problem. Bish *et al.* (2005) focus on the quayside process of discharging and uploading containers to and from a single vessel. They discuss the problem of dispatching single load vehicles to the containers for the minimization of the total time for discharging and uploading all containers. It is assumed that a fleet of vehicles is already assigned to the vessel. Several variations of a greedy algorithm are proposed that can be easily implemented in particular for large problem instances. Briskorn *et al.* (2006) focus on the assignment of transportation jobs to AGVs within a terminal control system operating in real time. A rather common problem formulation based on due times for the jobs is described and a greedy priority rule based heuristic with an exact algorithm was developed to solve this problem. Then an alternative formulation of the assignment problem which does not include due times is presented. This formulation is based on a rough analogy to inventory management and is solved using an exact algorithm. The idea behind this alternative formulation is to avoid estimates of driving times, completion times, due times and tardiness because such estimates are often highly unreliable in practice and do not allow for accurate planning. Zheng and Tao (2006) propose a combinatorial optimization method to minimize

Corresponding Author: Liu Haiwei, Logistics Engineering College, Shanghai Maritime University, 201306, Shanghai, China

the idle travel and the vessels serving time. And a simulation model is built to evaluate the method. The result is the average of the vessels' serving time reduce 7%. Nguyen and Kim (2007) discuss the dispatching of ALVs. Information about locations and times of future delivery tasks is utilized in an MIP model aiming at an optimal assignment of delivery tasks to ALVs. The NP-hard problem is formulated as a scheduling problem with precedence and buffer constraints similar to the multiple traveling salesmen problem with precedence constraints and time windows. In Chen *et al.* (2007), the dispatching problem is formulated as a hybrid flow shop scheduling problem with precedence and blocking constraints. A Tabu-search based algorithm is developed to solve this problem. The results show that a good initial solution is important. Angeloudis and Bell (2010) proposed a dispatching model and algorithm structured on a cost/benefit concept, according to unreliable information from terminal operations. An approach as measurement to terminal uncertainties is presented as well and it is proved that the model and algorithm works better in lower uncertainty level. Lee *et al.* (2010) makes similar assumptions as Nguyen and Kim, aside from the one that handling times of cranes are taken into consideration. A mixed integer programming model is proposed to treat the vehicle dispatching for job sequences of multiple cranes and a heuristic to solve the problem. Unfortunately, it could be referred from experiments that, in case the number of cranes and works are large, the computation time willbevery long. Moussi *et al.* (2011) focus on the Lifting AGV scheduling problem at container terminal of Normandy in Le Havre port. Lifting AGVs are vehicles which could lift a container itself while stacking and retrieving containers in storage yards. An integer programming model is proposed, based on assumptions including congestion-free traffic and fixed vehicle speed and travel distance. A generic algorithm which is compared with simulating annealing algorithm is developed to solve the problem. Le *et al.* (2012) formulated the vehicle dispatching problem as a mixed integer programming model. It is proposed an algorithm combining DCA and branch-and-Bound method to solve the model. Skinner *et al.* (2013) presents a GA-based optimization approach to solve the straddle carrier scheduling problem for container handling in the Patrick AutoStrad Terminal. The deterministic mathematical model is extended from a former one and the proposed approach has been fully implemented in terminal operation.

Nowadays, following the trend of jumbo ships, container terminals are changing gradually so that they could serve larger ships. CMA-CGM has brought into uses several ships of 16000 TEUs and Maersk has received even larger container ships, with capacity up to 18000 TEUs. One of the major problems that a container terminal operator is confronted with is how to quickly make a decision for the vehicle dispatching for job sequences of multiple cranes. In summary, many research works have been carried out in the field. But few of them focus on the work to help the operator to quickly make a decision to minimize the total working time and total travel costs. In this study, a multi-objective programming method is proposed. The objective is to minimize the total working time and the related total travel costs has also been carefully taken into account. Then a genetic algorithm is designed to resolve the problem.

The rest of study is structured as follows. Section 2 briefly explains the problem addressed in this study. In section 3 and section 4 the multi-objective programming and the genetic algorithm is discussed which is related to the vehicle dispatching. Then the detailed computational results are given in Section 5. Conclusions are discussed in Section 6.

PROBLEM DESCRIPTION

While unloading and loading containers to or from a vessel, several vehicles will be dispatched to carry out the travel jobs. In traditional, the vehicles are bound to the quay cranes. Which means that a vehicle can only serves for one quay crane. When it finished an unloading job in the block it must travel back to the bounding quay crane for another unloading job. The idle travel always happens in this dispatching method. The bounding model of quay side transports shows in Fig. 1.

A dynamic dispatching model means that the vehicles can serves any quay crane. When a vehicle finished its unloading job in the blocks, it can choose a loading job near it. And it also can choose a unloading job in the quay side when it finished its loading job. A dynamic dispatching model will help to reduce the idle travels. The dynamic dispatching model of quay side transports shows in Fig. 2.

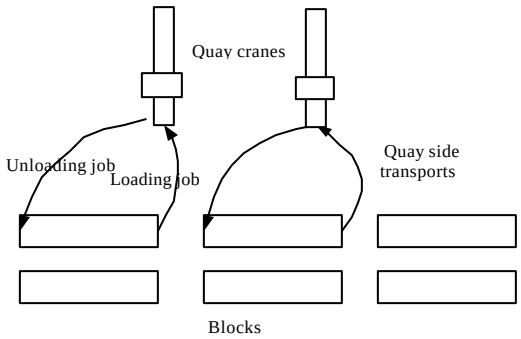

Fig. 1: Bounding model of quay side transports

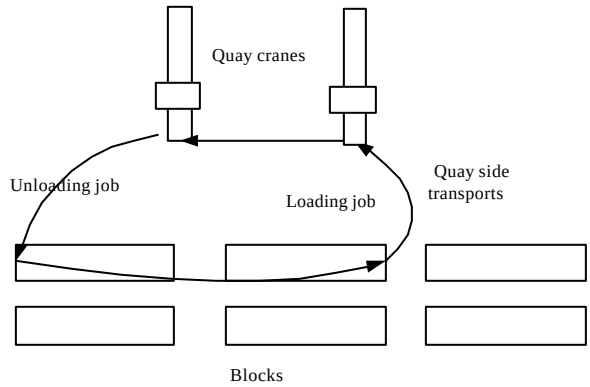

Fig. 2: Dynamic dispatching model

Table 1: Distances between each quay cranes (m)

QC No.	QC1	QC2	QC3	QC4	QC5	QC6
QC1	0	137	274	411	548	685
QC2	137	0	137	274	411	548
QC3	274	137	0	137	274	411
QC4	411	274	137	0	137	274
QC5	548	411	274	137	0	137
QC6	685	548	411	274	137	0

Table 2: Distances between each blocks (m)

B. No.	A	B	C	D	E	F	G	H
A	137	137	411	685	959	1233	1507	1781
B	137	137	137	411	685	959	1233	1507
C	411	137	137	137	411	685	959	1233
D	685	411	137	137	137	411	685	959
E	959	685	411	137	137	137	411	685
F	1233	959	685	411	137	137	137	411
G	1507	1233	959	685	411	137	137	137
H	1781	1507	1233	959	685	411	137	137

Table 3: Distance between blocks and quay cranes (m)

First bay	QC1	QC2	QC3	QC4	QC5	QC6
A01	110.0	493.3	876.6	1259.9	1643.2	2026.5
B01	382.5	219.3	602.6	985.9	1369.2	1752.5
C01	656.5	273.2	328.6	711.9	1095.2	1478.5
D01	930.5	547.2	163.9	437.9	821.2	1204.5
E01	1204.5	821.2	437.9	163.9	547.2	930.5
F01	1478.5	1095.2	711.9	328.6	273.2	656.5

In this study, the problem is to dispatch the loading and unloading jobs among the vehicles and determine the task sequence. There are 6 quay cranes and 8 blocks in the container terminal. The distances between each other are given as fellow Table 1-3.

When a vehicle finished its job in a block and start a new job in the same block, it need to driving from the head of the block to the end of the block, so the distance between the same block is the length of the block.

SOLUTION METHOD

Assumptions: The actual loading and unloading operation in container terminal is complicated. In order to simplify the model some assumption is made:

- The number of quay cranes and yard cranes and vehicles are given
- The processing time of quay crane and yard crane is constant
- The speed of vehicles is constant
- The vehicle can start its next task immediately
- The blocking of the traffic is not taken into account

Notations: The relative symbol definitions contain the set of quay cranes, the set of yard cranes, the starting time of each task and so no which are listed as follows:

- $\{m_1, m_2, m_3, \ldots m_i \ldots m_p\}$ is the set of quay cranes, p is the total number of quay cranes
- $\{n_1, n_2, n_3, \ldots n_w \ldots n_q\}$ is the set of quay cranes, q is the total number of quay cranes
- $\{v_1, v_2, v_3, \ldots v_j \ldots v_k\}$ is the set of vehicles, k is the total number of vehicles
- $\{c_1, c_2, c_3, \ldots c_s \ldots c_h\}$ is the set of loading and unloading task, h is the total number of tasks
- $t^{start}_{v_j,c_s}$ is the start time for v_j to start c_s
- $t^{end}_{v_j,c_s}$ is the end time for v_j to finish c_s
- $\max(t^{end}_{v_j,c_s})$ is the total working time
- $t^{emp}_{v_j,c_s}$ is the idle travel time during v_j serving for c_s
- $t^{trans}_{v_j,c_s}$ is the loaded travel time during v_j serving for c_s
- $T^{emp}_{v_j}$ is the total idle travel time of v_j
- $T^{trans}_{v_j}$ is the total loaded travel time of v_j
- $\max(v_j)$ is the total task number of v_j
- t_{lim} is permitted total working time
- $a_{j,s}$ is the decision variable, when c_s is dispatched to v_j, $a_{j,s}$ is 1,otherwise 0

Objective functions: In order to minimize the total working time of all vehicles, assume $t_{max_{vj}}$ is the total working time of v_j:

$$t_{max_{vj}} = T^{trans}_{v_j} + T^{emp}_{v_j} \tag{1}$$

$$T^{trans}_{v_j} = \sum_{c_s=1}^{\max(v_j)} (t^{end}_{v_j,c_s} - t^{start}_{v_j,c_s}) \tag{2}$$

$$T^{emp}_{v_j} = \sum_{c_s=1}^{\max(v_j)} t^{emp}_{v_j,c_s} \tag{3}$$

The first objective function is to minimize the total working time:

$$\text{Min}(\max\left(t_{max_{vj}}\right), j = 1, 2, 3 \ldots k) \tag{4}$$

In order to minimize the total travel costs all vehicles, assume p_1, p_2 to be the unit cost of loaded travel time and idle travel time. The second objective function is to minimize the total travel costs:

$$\text{Min}(y) = p_1 * \sum_{v_j=1}^{k} \sum_{c_s=1}^{\max(v_j)} (t_{v_j,c_s}^{end} - t_{v_j,c_s}^{start}) + p_2 * \sum_{v_j=1}^{k} \sum_{c_s=1}^{\max(v_j)} t_{v_j,c_s}^{emp} \quad (5)$$

Which can be simplified to:

$$\text{Min}(y) = p_1 * \sum_{v_j=1}^{k} T_{v_j}^{trans} + p_2 * \sum_{v_j=1}^{k} T_{v_j}^{emp} \quad (6)$$

Constraints: The constraints are listed below:

$$t_{v_j,c_s}^{end} \geq t_{v_j,c_s}^{start} \quad (7)$$

Equation 7 indicates the start time of each task must before the end time:

$$t_{v_j,c_s}^{emp} = t_{v_{j+1},c_{s+1}}^{start} - t_{v_j,c_s}^{end} \quad (8)$$

Equation 8 indicates that the idle travel is the first task end time and the second task start time:

$$t_{v_j,c_s}^{start} \geq t_{v_j,c_{s-1}}^{end} \quad (9)$$

Equation 9 is for that the first task end time must before the second task start time:

$$\forall j, \sum_{s=1}^{h} a_{js} = 1 \quad (10)$$

Equation 10 is for that a loading or unloading task must be dispatched to one vehicle:

$$\max\left(t_{v_j,c_s}^{end}\right) \leq t_{lim} \quad (11)$$

Equation 11 indicates that the final task end time must before the permit finish time.

PROPOSED SCHEME

In order to resolve the multi-objective programming, here a genetic algorithm is designed. Genetic algorithm is a common heuristic algorithm which is effective and feasible to get a good solution among the solution space.

Notations: The notations used in the algorithm are listed as follows:

- $\{v_{k1}, v_{k2}, v_{k3}, \ldots v_{ks} \ldots v_{kh}\}$ is the set of the chromosome. v_{ks} means that task c_s is dispatched to v_j
- N = 50 is the total number of the chromosomes in each generation
- GenNUM = 150 is the permit max generation number
- $\{f_1, f_2, f_3, \ldots, f_N\}$ is the set of the fitness
- $P_s = 1.1$ is the genetic operators
- $P_c = 0.7$ is the crossover operators
- $P_m = 0.05$ is the mutation operators

Chromosome representation: In the first population, 50 chromosomes are created. One of the chromosome is:

{5 6 8 13 8 2 3 6 14 15 5 9 8 12 2 5 12 11 11 10 2 2 12 2 14 5 6 9 11 10 1 1 14 14 9 4 7 4 12 4 2 15 9 3 3 3 11 8 8 4 5 14 7 2 11 12 15 12 2 12 6 15 13 7 7 14 11 2 7 13 7 8 8 12 7 3 15 3 15 6 9 4 2 8 7 11 14 7 15 3 15 10 5 15 9 2 11 14 4 4}

In the set, the first number 5 means the first task is dispatched to vehicle 5 and the third number 8 means the third task is dispatched to vehicle 8 and so on.

Parent selection: The fitness function is:

$$c - \text{Min}(y) = c - p_1 * \sum_{v_j=1}^{k} \sum_{c_s=1}^{\max(v_j)} (t_{v_j,c_s}^{end} - t_{v_j,c_s}^{start}) - p_2 * \sum_{v_j=1}^{k} \sum_{c_s=1}^{\max(v_j)} t_{v_j,c_s}^{emp} \quad (12)$$

In which C = 1200. The parent select function is:

$$f_i / \sum_{j=1}^{n} f_j \geq P_s / n \quad (13)$$

$$\frac{t_{max_{vj}}}{\sum (t_{max_{vj}})} < \frac{1}{N} \quad (14)$$

The parent select function indicates that if Eq 13 and 14 is true the solution will be chosen to be the parent chromosomes. In order to ensure that the best individual always survive to the next generation, the parent chromosomes will be selected to the next generation.

Crossover: In the crossover section, the parent chromosomes will be chosen to cross with the other chromosomes. After the crossing operations, the new chromosomes will be selected to the next generation. The crossing operation is to exchange some genes in the chromosomes with the same genes in the parent chromosomes randomly. The exchanged genes will be selected in the probability P_c which is the crossover operator.

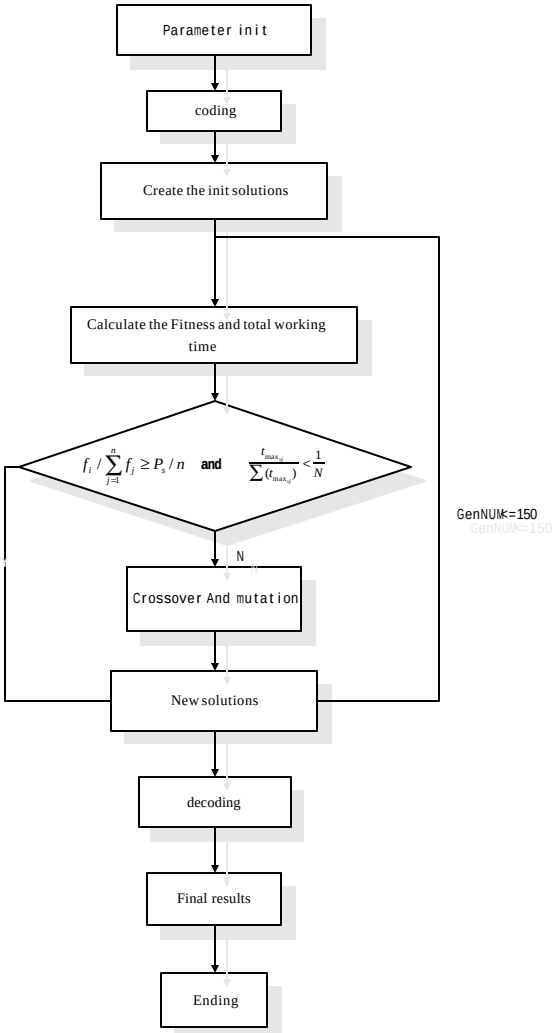

Fig. 3: A framework of proposed algorithm

Mutation: In the mutation section, some genes in the unselected chromosomes will be changed randomly. The changed genes will be selected in the probability P_m which is the mutation operator.

Offspring selection: If the new chromosomes created in crossover and mutation sections have a good fitness value, they will be chosen to the next generation to make sure the number of the populations.

The framework of the proposed genetic algorithm shows in Fig. 3.

NUMERICAL EXPERIMENTS

Here, the numerical experiments are performed by a personal computer with duo CPU @ 1.8 GHz and 1 GB RAM. The loading and unloading information is studied by using the actual data. And moreover, the experimental tests are carried out to evaluate the effectiveness and reliability of the algorithm. The input data are listed as follow:

Table 5: Task list

Task No.	Start points	End points	Task type	Travel distance (m)
1	QC01	A11	unload	414.2
2	QC01	B07	unload	554.7
3	QC01	F08	unload	1686.7
4	QC01	F13	unload	1840.1
5	QC01	E11	unload	1508.7
6	QC01	G10	unload	2015.1
7	QC01	D07	unload	1102.7
8	QC01	A12	unload	437.2
9	QC01	B01	unload	382.5
10	QC01	D02	unload	963.5
11	QC01	H13	unload	2388.1
…	…	…	…	…
88	F18	QC01	load	838.0
89	C02	QC01	load	744.9
90	A06	QC01	load	1407.7
91	B03	QC01	load	1043.3
92	D09	QC01	load	667.5
93	F18	QC01	load	673.3
94	G03	QC02	load	276.7
95	A11	QC02	load	1947.4
96	C03	QC02	load	1152.6
97	E04	QC03	load	1020.9
98	A11	QC03	load	2330.7
99	F18	QC03	load	1165.9
100	E11	3#	load	1234.7

- The distance between each working points has been given in the problem description section
- Fifteen vehicles are given and the speed is assumed to be 36 km h^{-1}, the initial locations are the start points of their first task
- The processing time of quay cranes is assumed to be 2 and 2.5 min $move^{-1}$ for yard cranes
- The unit cost of loaded travel is 240 and 180 h^{-1} for the idle travel
- There are 100 tasks comprise 30 loading tasks and 70 unloading tasks. The task list is show in Table 5

In the section 10 experimental tests are carried out. One of the results is show following:

- The optimal solution is got in the last generation which has a good fitness, low travel cost and less total working time
- The fitness of the solution is 267.14, the total cost of travel is 932 and the total working time is 21.35 min
- The average fitness of the 150 generations is show in Fig. 4. And the lowest cost of the 150 generations is show in Fig. 5

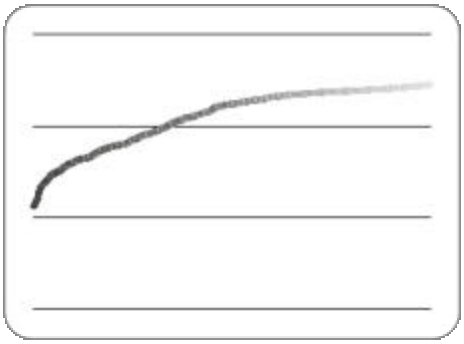

Fig. 4: Average fitness of the 150 generations

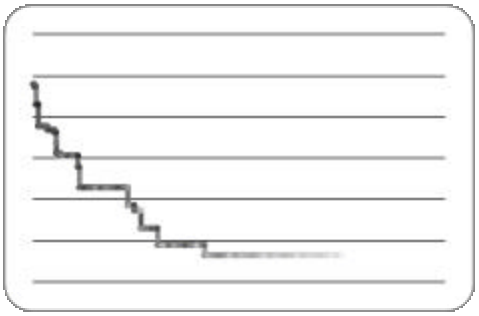

Fig. 5: Lowest cost of the 150 generations

Table 6: Optimal solution

T.	V.	T.	V.	T.	V.	T.	V.	T.	V.
1	1	21	2	23	1	25	11	27	10
2	15	22	11	24	13	26	5	28	5
3	8	23	14	25	7	27	3	29	1
4	4	24	4	26	1	28	12	30	8
5	14	25	5	27	14	29	14	31	6
6	8	26	15	28	6	30	12	32	9
7	5	27	2	29	3	31	6	33	12
8	15	28	5	30	5	32	14	34	3
9	9	29	13	31	2	33	10	35	8
10	4	30	7	32	14	34	1	36	5
11	15	31	9	33	7	35	11	37	15
12	8	32	1	34	15	36	9	38	9
13	11	33	14	35	9	37	2	39	8
14	12	34	10	36	2	38	10	40	12
15	6	35	13	37	12	39	6	41	4
16	9	36	13	38	8	40	4	42	3

t: Task No., v: Vehicle No.

Table 7: List of numerical experiments' results

ID	Yard cranes	Block No.	Task No.	Computational time (sec)
1	3	8	100	23.87
2	3	8	100	21.96
3	4	7	100	23.98
4	4	7	100	27.12
5	4	8	100	24.65
6	5	8	100	34.61
7	5	7	100	29.16
8	5	7	100	35.15
9	6	7	100	35.43
10	6	8	100	35.41

The optimal solution is show in Table 6.

In this study 10 numerical experiments are performed. The results are shown in Table 7.

ACKNOWLEDGMENTS

This study sponsored by National Natural Science Foundation (71201099), Shanghai Municipal Education Commission Project (14ZZ140), Local University Capacity Promotion Special Programs (13510501800) of Science and Technology Commission of Shanghai Municipality and Shanghai Maritime University Foundation Project (20120104). We also thank anonymous referees and the editor-in-chief for their ideas and suggestions.

REFERENCES

Angeloudis, P. and M.G.H. Bell, 2010. An uncertainty-aware AGV assignment algorithm for automated container terminals. Transp. Res. Part E: Logistics Transp. Rev., 46: 354-366.

Bish, E.K., 2003. A multiple-crane-constrained scheduling problem in a container terminal. Eur. J. Oper. Res., 144: 83-107.

Bish, E.K., F.Y. Chen, Y.T. Leong, B.L. Nelson, J.W.C. Ng and D. Simchi-Levi, 2005. Dispatching vehicles in a mega container terminal. OR Spectrum, 27: 491-506.

Briskorn, D., A. Drexl and S. Hartmann, 2006. Inventory-based dispatching of automated guided vehicles on container terminals. OR Spectrum, 28: 611-630.

Chen, L., N. Bostel, P. Dejax, J. Cai and L. Xi, 2007. A tabu search algorithm for the integrated scheduling problem of container handling systems in a maritime terminal. Eur. J. Oper. Res., 181: 40-58.

Kim, K.H. and J.W. Bae, 2004. A look-ahead dispatching method for automated guided vehicles in automated port container terminals. Transp. Sci., 38: 224-234.

Le, H.M., A. Yassine and R. Moussi, 2012. DCA for solving the scheduling of lifting vehicle in an automated port container terminal. Comput. Manage. Sci., 9: 273-286.

Lee, L.H., E.P. Chew, K.C. Tan and Y. Wang, 2010. Vehicle dispatching algorithms for container transshipment hubs. OR Spectrum, 32: 663-685.

Lv, X.Q. and H.W. Zhang, 2004. The integer programming modol for distributing vehides in container terminal. J. Dalian Fisher. Univ., 2: 105-109.

Mitrovic-Minic, S., R. Krishnamurti and G. Laporte, 2000. Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. Transp. Res. B: Methodol., 38: 669-685.

Moussi, R., A. Yassine, A. Kansou and T. Galinho, 2011. Scheduling of lifting vehicles with time windows in an automated port container terminal. Proceedings of the 4th International Conference on Logistics, May 31-June 1, 2011, Hammamet, Tunisia, pp: 55-61.

Nguyen, V.D. and K.H. Kim, 2007. A dispatching method for automated lifting vehicles in automated port container terminals. Proceedings of the International Conference on Intelligent Manufacturing and Logistics Systems, February 26-28, 2007, Kitakyushu, Japan.

Skinner, B., S. Yuan, S. Huang, D. Liu and B. Cai *et al*., 2013. Optimisation for job scheduling at automated container terminals using genetic algorithm. Comput. Ind. Eng., 64: 511-523.

Zhao, Q.H., S.Y. Wang, K.K. Lai and G.P. Xia, 2002. A vehicle routing problem with multiple use of vehicles. Adv. Model. Optimization, 4: 21-40.

Zheng, Y. and D.X. Tao, 2006. A scheduling technique in Container Terminals based on combination of dynamic optimization. J. WUT., 28: 26-29.