Proceedings of the 2011 International Conference on Industrial Engineering and Operations Management
Kuala Lumpur, Malaysia, January 22 – 24, 2011

Optimization of Integrated Scheduling of Quay Cranes and Automated Guided Vehicles using Simulated Annealing Algorithm

Seyed Mahdi Homayouni, Tang Sai Hong, Napsiah Ismail, Mohd Khairol A.M.K. Ariffin
Department of Mechanical and Manufacturing Engineering, Engineering Faculty,
University Putra Malaysia, Selangor, 43400, Malaysia

Abstract

The integrated scheduling of quay cranes (QCs) and automated guided vehicles (AGVs) is formulated as a mixed integer linear programming (MILP) model. The objective of the model is to minimize the makespan of all the tasks in a planning horizon. The problem is of NP-hard therefore a simulated annealing (SA) algorithm is proposed to solve the problem, in which, a heuristic algorithm assigns the AGVs to the QCs, and the SA finds the best sequence of QC tasks. The problem is solved by using the MILP model and the SA algorithm for some small and medium size test cases. The results show that the SA can solve the problem precisely and accurately.

Keywords
Integrated scheduling, Container terminals, Mixed integer linear programming, Simulated annealing algorithm.

1. Introduction
The economy of developing countries is highly dependent on the efficiency of their sea borne transportation. According to UNCTAD [1] by beginning of 2010, 4,722 vessels were in service to offer total capacity of 13 million twenty-foot equivalent units (TEUs). As an example, Malaysian ports grew 16.5 percent annually during 2000 to 2008, which is the highest growth rate in South-East Asia. Container port traffic in Malaysia grew from a base of 4.6 million TEUs in 2000 to 15.7 million TEUs in 2008. The productivity of port operations to load and unload the vessels is so important to improve the performance of the ports. Notteboom [2] stated that 86% of uncertainties in scheduling of vessel are due to unexpected waiting times of vessels before berthing and during loading and unloading tasks at container terminals.

The vessels are served in the quayside of the container terminals (CTs) to load and unload containers. Various kinds of vehicles are used to transport the containers to/from the quayside to storage side, where the containers are stored temporarily. The current trend in port operations researches focuses on both hardware and software developments to increase the trough put of the CTs. Automated guided vehicles (AGVs), automated lifting vehicles (ALVs), automate storage cranes (ASCs), and double trolley quay cranes (QCs) have been introduced for modern CTs. However many researchers stated that optimization of integrated scheduling of equipments highly influence the performance of the CTs [3, 4]. Lee et al. [5] considered integrated scheduling of QCs and yard trucks (YTs). The problem was formulated as a mixed integer linear programming (MILP) model. The objective was to minimize the makespan of all the jobs. They developed a genetic algorithm (GA) to solve the problem. Nguyen and Kim [6] proposed dispatching method for ALVs in CTs. They developed a heuristic algorithm to solve the problem. Moreover, in a recent research, Cao et al. [7] considered the integrated scheduling of YTs and yard cranes (YCs). In which, Benders' decomposition method was proposed to solve the problem.

Lau and Zhao [3] proposed a scheduling method for AGVs, automated YCs and QCs in container terminals. They developed an MILP model and solved it by a GA. Mak and Zhang [4] have solved the scheduling problem in CTs using a GA. A simulated annealing (SA) algorithm was proposed to select the parents in the GA. This hybrid algorithm needs much less computational time than canonical GA. Bierwirth and Meisel [8] surveyed recent literature on main scheduling and control problems in CTs. They affirmed that a new stream on integrated scheduling has been started in last few years. The SA was used by the researchers to solve the scheduling problems in CTs. Kim and Moon [9] solved berth allocation problem by using an SA algorithm. Furthermore, Lee et al. [10] proposed an SA for two-transtainer scheduling problem for loading containers in CTs. They claimed that the performance of their proposed SA is irrelevant to the number of containers loaded.

In this paper, the integrated scheduling of QCs and AGVs in container terminals is formulated as an MILP model. The problem is an NP-hard; thus, an SA is proposed to solve the problem. The performance of the SA is highly affected by its selected control parameters; thus, the paper present sensitivity analysis on various control parameters and procedures. The paper is organized as following; the problem is defined and formulated in section 2. The proposed SA algorithm for the integrated scheduling is detailed in section 3. The test cases, obtained results and discussions are presented in section 4. In conclusion, the final remarks are presented in section 5.

2. Problem Definition and Formulation

The whole program to unload and load a vessel is divided into some short planning horizons for better performance of scheduling process. In each period, the sequence of tasks and their precedence relations are pre determined. In proposed scheduling method, the tasks for all the assigned QCs are scheduled based on the availability of the AGVs. The AGVs are not able to lift the containers by themselves; hence, they should be loaded/ unloaded by the QCs or YCs. In unloading tasks, the AGV moves from its dwell point to the QC and the QC tranship the container from the vessel to quayside. Once AGV reached the QC, it receives the container and moves to the load/unload (L/U) station in yard area. The AGV stops at L/U, delivers the container to YC, and waits for next operation. In loading tasks, the AGV moves from its dwell point to the assigned L/U station in yard area, receives the container, and moves to the QC. If the QC is free, the container is loaded to the vessel. In cases, that QC is not free, the AGV has to wait for it. It is supposed that all the AGVs start their journeys from L/U stations and return there once accomplishing all the assigned tasks. The AGVs are similar to each other, and thus none of them is assigned to a specific kind of containers or to a QC. It is assumed that the YCs are able to load/unload containers to/from the AGVs as soon as they reach the L/U. It is assumed that handling times between equipments are ignorable, and the operational time of QC (*OQ*), its travel time between the vessel and quayside (*TQ*) and travel time of AGVs are deterministic.

In the proposed MILP model, the set of QCs is denoted by *K*, the number of AGVs is *V*, and *S, F* define the initial and final positions of AGVs. The set of loading and unloading tasks are illustrated by "*L*" and "*U*", respectively. T_{ki} defines the i^{th} task on QC_k, and m_k is the number of tasks determined for QC_k. The travel time of an AGV between two assigned T_{ki} and T_{lj} is defined by t_{kilj}, and χ_{kilj} is the decision variable for the journey that is a binary variable. t_{kilj} is travel time between QC_k and QC_l, including the operations required for T_{ki} and T_{lj}, which is illustrated how to calculate in figure 1. y_{ki} is the real completion time of QC operations in T_{ki}. The model is formulated as:

Minimize Z= *Makespan* (1)

Subject to:

$$C_j - \left(y_{ki} - TQ - OQ + t_{kifj}\right) \geq M(\chi_{kifj} - 1) \qquad \forall k \in K, \forall j \in V, \forall i = 1,..., m_k, \text{and } T_{ki} \in L, \tag{2}$$

$$C_j - \left(y_{ki} + t_{kifj}\right) \geq M(\chi_{kifj} - 1) \qquad \forall k \in S \cup K, \forall j \in V, \forall i = 1,..., m_k, \text{and } T_{ki} \in U, \tag{3}$$

$$Makespan \geq C_j \qquad \forall j \in V, \tag{4}$$

$$\sum_{l=1}^{K \cup F} \sum_{j=1}^{m_l} \chi_{kilj} = 1 \qquad \forall k \in K \cup S, \forall i = 1,..., m_k, \tag{5}$$

$$\sum_{k=1}^{K \cup S} \sum_{i=1}^{m_k} \chi_{kilj} = 1 \qquad \forall l \in K \cup F, \forall j = 1,..., m_l, \tag{6}$$

$$y_{lj} - A \geq M(\chi_{kilj} - 1), \qquad \forall k \in K \cup S, \forall l \in K \cup F, \forall i = 1,..., m_k, \forall j = 1,..., m_l, \tag{7}$$

$$\chi_{kilj} = 0 \text{ or } 1, \qquad \forall k \in K \cup S,\ \forall l \in K \cup F,\ \ \forall i = 1,.., m_k,\ \forall j = 1,.., m_l, l = k \Rightarrow j > i. \tag{8}$$

The objective function (Eq. 1) is to minimize the makespan of the (un)loading tasks in a specific scheduling horizon. The makespan of the tasks is defined as the latest journey of the AGVs to their final position. By minimizing the makespan of AGVs, the QC completion times are decreased, too. Equations (2) and (3) define the cycle time for the AGVs in loading and unloading tasks, respectively. In which the time that AGV receives (delivers) the last container to the QC adds to the travel time of last journey of the AGV to its final position. The makespan is the largest cycle time of the AGVs calculated in equation (4). Equations (5) and (6) imply that there should be one to one relation between two consecutive tasks include the initial and final journeys of the AGVs. The completion time of T_{lj} (y_{lj}) is related to y_{lj-1} and y_{ki}, implied by constraint set (7). This set of constraints is different for various characteristics of the T_{lj}, T_{lj-1}, and T_{ki}, which is detailed in table 1. The *Max* function in this constraint can be decomposed into two inequalities to make a linear set of constraints. Finally, constraint (8) defines χ_{kilj} as a binary decision variable.

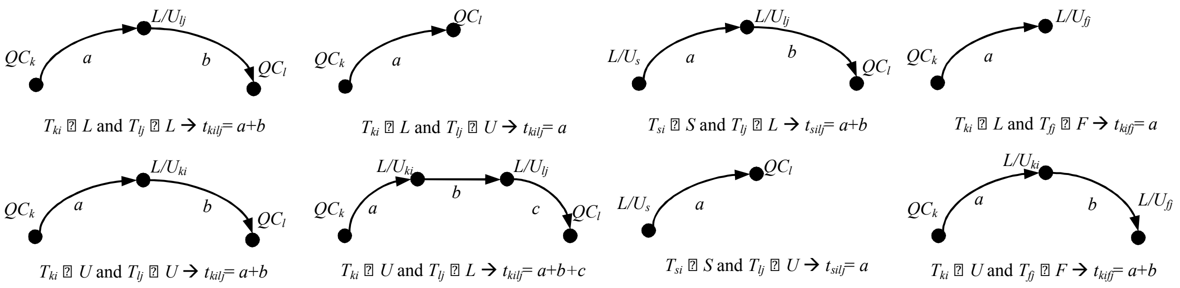

Figure1: An illustration for AGV travel time calculations

Table 1: Calculations for constraint (7) on $y_{lj} - A \geq M\,(\chi_{kilj} - 1)$

T_{li}	$T_{l(i-1)}$	T_{ki}	A	T_{li}	$T_{l(i-1)}$	T_{ki}	A
L	L	L	Max $(y_{lj-1} + TQ\, ,\, y_{ki} - TQ - OQ + t_{kilj}) + TQ + OQ$	U	U	U	Max $(y_{lj-1} + 2TQ + OQ\, ,\, y_{ki} + t_{kilj})$
L	U	L	Max $(y_{lj-1}\, ,\, y_{ki} - TQ - OQ + t_{kilj}) + TQ + OQ$	U	L	U	Max $(y_{lj-1} + TQ + OQ\, ,\, y_{ki} + t_{kilj})$
L	L	U	Max $(y_{lj-1} + TQ\, ,\, y_{ki} + t_{kilj}) + TQ + OQ$	U	U	L	Max $(y_{lj-1} + 2TQ + OQ\, ,\, y_{ki} - TQ - OQ + t_{kilj})$
L	U	U	Max $(y_{lj-1}\, ,\, y_{ki} + t_{kilj}) + TQ + OQ$	U	L	L	Max $(y_{lj-1} + TQ + OQ\, ,\, y_{ki} - TQ - OQ + t_{kilj})$

Simulated Annealing Algorithm

Simulated annealing is an analogy of the cooling process of metals. The liquid metal has a high level of energy and its atoms are free to move around, and change its structure. If the cooling process of a metal is slow enough (annealing), the atoms have the opportunity to find a state with minimum level of energy and to form a pure crystal shape [11]. The SA algorithm is able to search rough spaces and escape from local optima and rather to reach global optimum. The algorithm starts its process from a high initial temperature, searches for the solutions *M* times in each temperature, and decrease the temperature until it reaches a steady state (final temperature). Therefore, initial (T_I), final (T_f) and number of trials (*M*) in each temperature (T_r) are the control parameters for the SA. The SA accepts better solutions in its procedure in addition to worse solution with an acceptance probability. As the temperature decreases, the probability of accepting worse solutions decreases. The cooling process is to ensure that the SA converges to a near optimal solution. Equation (9) is the cooling process proposed by Chen and Shahandashti [12], in which α is the cooling ratio. The higher the cooling ratio, the faster the temperature cools down. In equation (9), r is the current number of times the temperature has been decreased. Figure 2 illustrates the steps of the proposed SA.

$$T_r = T_I\, e^{(-\alpha\, r)} \tag{9}$$

A feasible solution for the scheduling problem is a string of QC tasks observing the precedence relations. The tasks are numbered 1 to $N = \sum_{i=1}^{K} m_i$, total number of tasks. An initial solution is a random string of tasks, which are sorted for each QC to maintain the precedence relations. The following heuristic algorithm is applied to assign the AGVs to the tasks. In case that more than an AGV is available to perform a journey, the nearest AGV to the starting point of the task is selected.

- The sequence of tasks for QCs is considered for scheduling (based on the output of GA),
- For Loading tasks do the following steps,
    - find which AGV reach the L/U earlier, Move the AGV to L/U, and pick up the container,
    - Move the loaded AGV to the QC, wait until QC is free (if it is required),
    - Deliver the container to QC, update the current time of AGV (AGV is ready to perform next task),
    - QC handles the container to the ship, update the current time of QC (QC is ready to perform next task),
- For unloading tasks do the following steps,
    - find which AGV reach the QC earlier, Move the AGV to the desired QC, wait until QC arriving,
    - Concurrently, the QC should pick up the container form the vessel and bring it to the quayside,
    - Pick up the container, update the current time of QC (QC is ready to perform next task),
    - Move the loaded AGV to the L/U, deliver the container, update the current time of the AGV (the AGV is ready to perform next task).
- Check whether all the desired tasks are completed.
- Calculate total travelling time of the AGVs and the delay time of the QCs (if any).

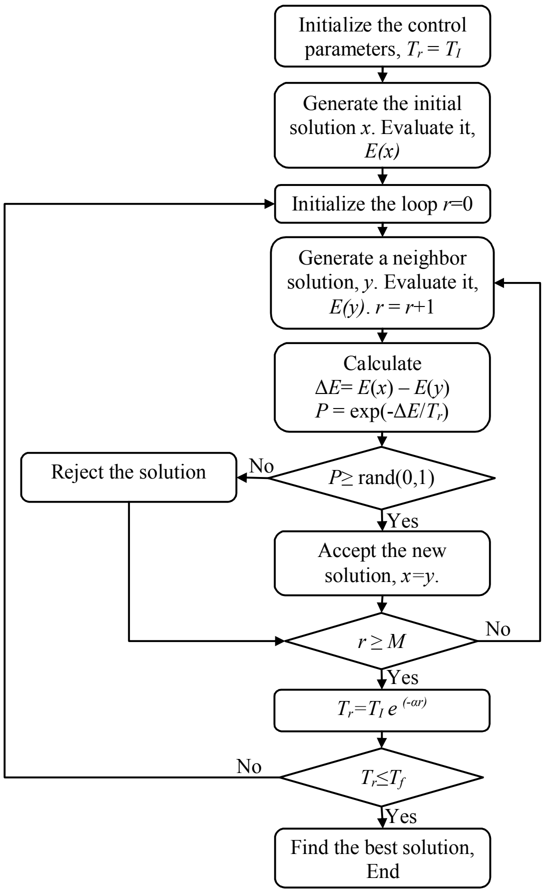

Figure2: The proposed simulated annealing algorithm for integrated scheduling problem

To generate the new solution a "swap" neighborhood search structures (NSS) is applied. In which, two tasks belong to two different QCs are selected randomly. They are sorted according to their position in the string. If the task in first position is the last task of a QC, or its successor is located after the second position it can be substituted to the second position. On the other hand, if the task in second position is the first task of a QC, or its predecessor is located before the first position it can be substituted to a first position. The tasks are swapped if the abovementioned rules are true for both selected tasks.

Numerical Experiments and Discussions

To evaluate the performance of the proposed scheduling method, two set of test cases were designed. The test cases were planned in a CT contains six QCs and six L/U stations. The layout and the travel time of AGVs were described in [3]. The small size test cases presented in table 2, are to compare the results of the SA and the MILP model. Moreover, medium size test cases are described in table 3. In all the test cases, the *OQ* and *TQ* are equal to 20 and 10 seconds, respectively. The MILP model was solved by using branch and bound solver of *Lingo*®, and the SA were programmed in *MATLAB*®. The software packages were executed on Intel Pentium 2.13 GHz computer, holding 4 GB RAM and running under Windows 7®.

The first set of experiments is to determine the best control parameters for the SA. Test case number M5 is selected to perform the analysis. The effects of various T_I and M were tested, and T_f in all the tests is set to 1. According to trial and error process, α was selected to generate equal number of iterations for various T_Is. Thus, for T_I= 500, 1000, 2000, and 5000, α was selected as 0.0124, 0.0061, 0.0038, and 0.0017, respectively. Figure 3.a shows the mean of five runs for each combination of T_I and M. The trend of solutions shows that the mean of the solutions decreases as the total number of iterations increases. However, the number of trials in each temperature, M, influences the final answer more than number of iterations. As a result T_I =2000, and M=50 is selected for the reminder of the experiments. Furthermore, figure 3.b shows how to decrease the temperature in the cooling process. The experiment shows that the CPU time for the SA is directly related to the number of iterations. Selecting the best control parameters for cooling processes, the authors considered mean/ the best of the solutions, and the CPU time.

Table 2: Specifications of small size test cases

No.	Task*QC* L/U*AGV	QC No.	L/U Station	AGV Initial Position	Task type
S1	8*2*2*4	1,2	7,9,9,7; 9,7,9,7.	12,11,12,7	U,U,U,L; U,L,U,U.
S2	10*2*2*5	4,5	10,11,10,10,11; 10,10,10,11,10.	7,12,10,9,11	U,U,L,L,L; L,U,L,U,L.
S3	12*3*2*4	3,4,5	8,9,8,9; 9,8,8,9; 8,9,9,9.	11,11,10,7	L,U,L,U; L,U,L,L; U,L,U,L.
S4	15*3*3*5	2,3,4	7,9,11,7,11; 9,11,7,11,9; 11,7,7,11,9.	12,8,10,9,10	U,U,L,L,U; L,U,L,U,L; U,L,L,U,U.
S5	18*3*4*6	4,5,6	8,10,9,11,10,8; 11,9,8,10,10,8; 9,11,11,8,10,9.	8,8,7,8,10,9	L,L,U,U,L,U; L,U,U,L,L,U; U,U,L,L,U,U.

Table 3: Specifications of medium size test cases

No.	Task*QC* L/U*AGV	QC No.	L/U Station	AGV Initial Position	Task type
M1	30*3*4*6	1,2,3	9,12,10,11,9,11,10,12,9,10; 10,11,9,12,12,11, 9,10,9,11; 12,10,9,11,9,11,12,9,10,11.	12,7,12,8,11,10	L,U,L,U,L,L,U,L,L,U; U,U,L,U,L,U,L,U,U,L; L,L,U,U,L,U,L,L,U,L.
M2	45*3*3*7	2,3,4	8,10,10,8,8,7,10,8,7,7,10,8,7,8,8; 10,8,7,8,7,8,10,8, 8,10,7,7,8,10,10; 8,7,10,7,8,7,8,10,8,7,10,7,10,8,7.	12,7,8,7,11,9,7	L,L,U,U,U,L,U,L,U,L,U,U,L,L,U; U,L,U,L,L,U,L,U, U,L,L,U,L,U,L; L,U,U,L,U,U,L,U,L,L,U,L,U,L,L.
M3	40*4*5*5	3,4,5,6	9,7,10,11,12,11,9,12,7,10; 10,11,7,12,9,11,9,7,9,11; 12,7,9,12,9,11,12,7,10,11; 7,9,12,11,10,7,9,10,12,11.	9,7,10,12,7	U,U,L,U,L,U,L,L,U,L; L,U,L,L,U,L,U,U,L,L; U,L,L,U,U,L,U,L,U,U; L,U,U,L,U,L,L,U,U,L.
M4	48*4*5*6	1,2,3,4	7,11,8,10,9,8,10,7,11,9,8,7; 10,8,9,7,11,9,10,11,7, 8,10,9; 9,10,8,7,11,9,10,7,11,8,9,10; 8,11,9,7,10,8, 7,10,7,9,11,8.	7,10,11,8,9,7	L,U,L,U,L,U,U,L,U,L,L,U; L,U,U,L,U,L, L,U,U,L,U,L; U,L,L,U,U,L,L,U,L,U,U,L; L,L,U,U,L,L,U,L,L,U,L,U.
M5	56*4*6*7	2,3,4,5	7,10,9,8,11,9,12,8,11,10,12,8,7,9; 9,11,12,8,10,7,9, 8,11,12,8,9,7,12; 12,7,8,7,10,7,9,12,11,7,9,11,12,8; 11,7,12,8,9,7,10,12,9,11,7,10,12,9.	10,7,8,9,11,9,7	U,L,L,L,U,U,L,L,U,L,U,L,U,U; U,L,L,U,U,L,L, U,U,L,U,L,U,L; L,L,U,U,L,U,L,U,L,U,L,U,U,L; L,U,L,U,L,U,L,U,L,U,L,L,U,U.

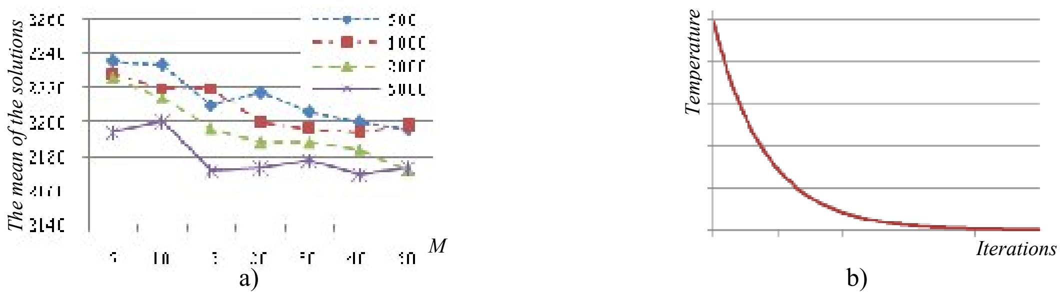

Figure 3: (a) Effects of T_I and M on the mean of the solutions,
(b) How to decrease the temperature in cooling process

The second set of experiments is to evaluate the performance of the SA in medium size test cases. The test cases were solved under the control parameters determined in previous experiment. The mean of five runs, the best solution of the SA, the deviation of the mean and the best solutions, and CPU time for every test case is presented in table 4. The results show that the mean of solutions for the medium size test cases is only 1.2% worse than the best available solution, which indicates the acceptable efficiency of the proposed SA algorithm to find the good solution precisely. The last set of experiments is to demonstrate the accuracy of the proposed SA algorithm. The small size test cases were solved by using the MILP model and the proposed SA. The optimal solution found by the MILP, the mean of five runs for the SA, and their required CPU time are reported in table 5. Furthermore, the optimality gap between the solutions of the MILP model and the SA algorithm is defined in equation (10). The results indicate that the SA can find good solutions in compare to the MILP model. In some cases, even the mean of the solutions obtained by the SA is less than the solution found by the MILP model. In such cases, the MILP model is not able to find the global optimum in a reasonable CPU time. Therefore, the authors marked these test cases with an asterisk.

$$\%\ Optimality\ Gap = 100 \times (Best - Optimal) / Optimal \qquad (10)$$

Table 4: Results for medium size test cases

Test case	Mean	CPU time	The best	Deviation%
M1	1192	102.7	1170	1.9
M2	1318	135.5	1310	0.6
M3	1962	116.1	1960	0.1
M4	1688	133.5	1660	1.7
M5	2164	161.2	2130	1.6

Table 5: Comparative results of small size test cases

Test case	MILP		SA			Optimality Gap %
	Optimal	CPU time	Mean	The Best	CPU time	
S1	510	27M 38S	524	510	65.9S	0
S2	650	2H 20M 31S	694	670	70.3S	3.1
S3*	750	5H 23M 2S	734	710	73.1S	-5.3
S4*	680	7H 35M 40S	700	690	76.4S	1.4
S5*	900	7H 32M 20S	924	900	86.4S	0

Conclusion

The integrated scheduling of quay cranes and AGVs in container terminals was formulated and solved in this paper. A simulated annealing algorithm was developed to the problem. The objective function is to minimize the makespan of all the tasks. The makespan is defined as the largest cycle time among the set of AGVs to perform their assigned journeys. The effects of various control parameters namely the initial temperature, and number of trials in each temperature were examined through two set of numerical test cases. Moreover, the small size test cases were solved by using the mathematical model and the SA. The comparative results indicated that the SA is able to find solutions precisely and accurately in reasonable CPU time. Furthermore, the results of medium size test cases show that the proposed scheduling method can be used in practical application. For further researches, it is recommended to apply other kinds of vehicles such as automated lifting vehicles in scheduling problem.

References

1. UNESCAP, 2009, “Review of Developments in Transport in Asia and the Pacific,” Economic and Social Commission for Asia and Pacific, United Nations, New York.
2. Notteboom, T.E., 2006, “The Time Factor in Liner Shipping Services,” Maritime Economics & Logistics, 8, 19 –39.
3. Lau, H.Y.K., and Zhao, Y., 2008, “Integrated Scheduling of Handling Equipment at Automated Container Terminals,” International Journal of Production Economics, 112, 665 –682.
4. Mak, K.L., and Zhang, L., 2009, “Simultaneous Scheduling of Import and Export Containers Handling in Container Terminals,” In Proceedings of the World Congress on Engineering, WCE 2009, London, UK.
5. Lee, D.H., Cao, J.X., and Shi, Q.X., 2008, “Integrated Quay Crane and Yard Truck Schedule for Inbound Containers,” In Proceedings of IEEE International Conference on Industrial Engineering and Engineering Management, Singapore, 1219 –1223.
6. Nguyen, V.D., and Kim, K.H., 2009, “A Dispatching Method for Automated Lifting Vehicles in Automated Port Container Terminals,” Computers & Industrial Engineering, 56, 1002 –1020.
7. Cao, J.X., Lee, D.H., Chen, J.H., and Shi, Q., 2010, “The Integrated Yard Truck and Yard Crane Scheduling Problem: Benders’ Decomposition-Based Methods,” Transportation Research Part E, 46, 344 –353.
8. Bierwirth, C., and Meisel, F., 2010, “A Survey of Berth Allocation and Quay Crane Scheduling Problems in Container Terminals,” European Journal of Operational Research, 202, 615 –627.
9. Kim, K.H., and Moon, K.C., 2003, “Berth Scheduling by Simulated Annealing,” Transportation Research Part B, 37, 541 –560.
10. Lee, D.H., Cao, Z., and Meng, Q., 2007, “Scheduling of Two-Transtainer Systems for Loading Outbound Containers in Port Container Terminals with Simulated Annealing Algorithm,” International Journal of Production Economics, 107, 115 –124.
11. Suman, B., and Kumar, P., 2006, “A Survey of Simulated Annealing as a Tool for Single and Multi-Objective Optimization,” The Journal of the Operational Research Society, 57, 1143 –1160.
12. Chen, P.H., and Shahandashti, S.M., 2009, “Hybrid of Genetic Algorithm and Simulated Annealing for Multiple Project Scheduling with Multiple Resource Constraints,” Automation in Construction, 18, 434 –443.