

Full Length Research Paper

Using simulated annealing algorithm for optimization of quay cranes and automated guided vehicles scheduling

S. M. Homayouni^{1,2*}, S. H. Tang², N. Ismail² and M. K. A. Ariffin²

¹Department of Industrial Engineering, Lenjan Branch, Islamic Azad University, Isfahan, Iran.

²Mechanical and Manufacturing Engineering Department, Engineering Faculty, Universiti Putra Malaysia, UPM Serdang, 43400, Malaysia.

Accepted 24 August, 2011

Sea port container terminals play an important role in the national and regional economy. The integrated scheduling of handling equipment has been recently investigated in literature to improve the performance of the container terminals. In this paper, an integrated scheduling of quay cranes and automated guided vehicles is formulated as a mixed integer linear programming model. This model minimizes the makespan of all the loading and unloading tasks for a set of cranes in a scheduling problem. Based on the simulated annealing (SA) algorithm, a scheduling method is proposed to solve the problem in a relatively short period of time. Comparison of the respective results of the mathematical model and the SA algorithm evidently shows acceptable performance of the proposed SA algorithm in finding good solutions for practical scheduling problems. Moreover, the effects of three cooling processes and two sets of control parameters on the best solution of the SA are investigated.

Key words: Quay cranes, automated guided vehicles, simulated annealing, integrated scheduling.

INTRODUCTION

As a consequence of global economic crises, world merchandise trade experienced a sharp decline in 2009 after a persistent growth rate for the past 25 years. Consequently, international seaborne trade volume contracted by 4.5% in 2009, and world container terminal throughput reached 457.3 million twenty equivalent units (TEUs). However, the United Nations conference on trade and development (UNCTAD) stated that on January

2010, there were more than 4,600 vessels with a total capacity of 12.8 million TEUs in the world containership fleet, a 5.6% growth against 2009 statistics (UNCTAD, 2010). From all the data revealed in containerization, it can be concluded that increasing the throughput of the ports is unavoidable for almost all the major ports in the world. Ding and Chou (2011) defined a container terminal (CT) as a nodal point used to handle containers among international trade and logistics systems.

Improvements in CTs, along with more efficient scheduling methods highly influence the performance of the sea ports. According to Notteboom (2006), 86% of uncertainties in containerships schedules are due to their unexpected waiting times before berthing and during loading/unloading tasks at CTs. Therefore, the integrated scheduling methods have been developed in order to coordinate various types of equipment at CTs (Bierwirth and Meisel, 2010; Zeng and Yang, 2009; Nguyen and Kim, 2009). Moreover, the integrated scheduling has been investigated for flexible manufacturing systems (Subbaiah et al., 2009; Reddy and Rao, 2006).

A typical CT serves the containerships in quay area in unloading the importing containers and loading the exporting ones. The huge cranes used in quay area are

*Corresponding author. E-mail: homayouni@iauln.ac.ir. Tel: +98-913-3259168.

Abbreviations: AGV, Automated guided vehicle; ALV: automated lifting vehicle; GA, genetic algorithm; SA, simulated annealing; CT, container terminal; YC, yard crane; QC, Quay crane; TEU, twenty equivalent units; UNCTAD, United Nations conference on trade and development; MIP, mixed integer programming; L/U, load/unload station; OQ, operational time for QCs; TQ, transferring time for QCs; PC, proposed cooling process; EC, exponential cooling process; LC, linear cooling process; M, the number of iterations in each temperature level; T_i, initial Temperature; T_f, final temperature; T_c, Current temperature.

called quay cranes (QCs). UNCTAD (2010) reported that tandem and triple lift QCs are implemented in pioneer container terminals. The containers are stored in a storage area temporarily. The common equipment in storage area are the gantry cranes, yard cranes (YCs), straddle carriers, and automated stacking crane (Stahlbock and Voß, 2008). Additionally, automated storage/retrieval systems are proposed to enhance the performance of the storage yards (Vasili et al., 2008). A transport area connects quay and yard areas of the CTs, where the vehicles transfer the containers from the containership to the storage yard and vice versa. In addition to the conventional yard trucks and rail-based handling equipment, automated guided vehicles (AGVs) and automated lifting vehicles (ALVs) are implemented in CTs.

The scheduling problem in CTs is one of the most interesting topics in the literature. Bierwirth and Meisel (2010) surveyed the literature regarding the main scheduling and control problems in CTs. They affirmed that a new stream on simultaneous and integrated scheduling of various types of equipment has been started in the last few years. Lee et al. (2008) considered integrated scheduling of QCs and YTs. They formulated the problem as a mixed integer programming (MIP) model aiming to minimize the makespan of all the jobs assigned to QCs and YTs. Furthermore, they developed a genetic algorithm (GA) to obtain near optimal solution of the problem. In a recent research, Nguyen and Kim (2009) proposed a dispatching method for ALVs in container terminals. An MIP model for the problem and a heuristic algorithm to solve it has been developed in their paper.

Integrated scheduling of yard trucks and YCs has been investigated by Cao et al. (2010) who proposed the Benders' decomposition method to solve the problem. They stated that the proposed method could be applied in most of the integrated models in CTs. Among other studies in this regard, Zeng and Yang (2009) proposed a simulation-based optimization method for scheduling of QCs, yard trucks, and YCs, which enhances the coordination among different types of handling equipment. A scheduling method for AGVs, automated YCs, and QCs in container terminals was also proposed by Lau and Zhao (2008) in which study they developed an MIP model and solved it by the use of GA.

The simulated annealing algorithm was widely applied to optimize the scheduling problems. Kim and Moon (2003) formulated berth allocation problem as an MIP model. They developed an SA method to obtain near optimal solution for the problem while decreasing the computation time. Another SA was proposed by Lee et al. (2007) for solving the two-transtainer scheduling problem for loading containers in CTs. They claimed that the performance of the proposed SA is irrelevant to the number of containers loaded. On the other hand, the SA has been proposed to enhance the performance of the

GA. Mak and Zhang (2009) proposed a genetic algorithm to optimize the handling operations for loading and unloading tasks in CTs. An SA is proposed as the selection process of the GA. Notably, this hybrid algorithm needs much less computation time than canonical GA. Norozi et al. (2011) proposed the same method to optimize multi-objective scheduling problem. Moreover, Abdullah et al. (2011) developed an SA for the course scheduling in universities.

In this paper, an integrated scheduling of quay cranes and AGVs in container terminals is presented. This problem is formulated as a mixed integer programming model. The integrated scheduling is an NP-hard problem, and thus no efficient algorithm would obtain global optimum solutions for real world applications in this case (Meersmans and Wagelmans, 2001). Consequently, based on simulated annealing algorithm, a scheduling method is proposed to find the near optimal solutions for the problem. The performance of SAs is highly affected by their selected control parameters. Also presented in this paper is the sensitivity analysis on various control parameters and procedures of the SA algorithm.

RESEARCH METHOD

In container terminals, operations include loading the exported containers to, and unloading imported containers from the containerships. The stowage planning determines the location of each container in the ship. On the other hand, the QC scheduling problem proposes the preceding relations for loading and unloading tasks assigned to a QC in a bay of the containership. The whole program is divided into some short term scheduling periods for better performance of the scheduling process. In each period, the sequence of tasks and their preceding relations are pre-determined. In this study's proposed scheduling method, the tasks for all the assigned QCs are scheduled regarding availability of the AGVs.

In loading tasks, the vehicle moves from its dwell point to the assigned load/unload (L/U) station in the yard area; receives the container, and moves to the crane. If the crane is free, the container is loaded to the ship; otherwise, the vehicle has to wait for it. During the unloading tasks, the vehicle moves from its dwell point to the assigned crane. Concurrent to this movement, the crane handles the container from its storage location in the ship to the quay area. Once the vehicle reaches the crane, it receives the container and moves to the assigned L/U station in the yard area. The vehicle stops at L/U station, delivers the container to YC, and waits for the next assigned operation. It should be noted that the AGVs are unable to pick up the containers by them. Therefore, they should be served directly by the QCs or YCs, that is, there is no buffer space in between the QCs or the YCs and the AGVs. If the crane has finished the assigned operation prior to the vehicle arriving, the crane has to wait for the vehicle. On the contrary, if the vehicle arrives before finishing the crane's operation, the vehicle has to wait for the crane.

Scheduling problem formulation

In the formulation of the problem, it is assumed that the vehicles can be served by the YCs without any delay, that is, the YC is able to pick up the imported containers immediately, and the exported containers are available to be delivered to the vehicle as soon as it

reaches the L/U station. These are the same as the assumptions made by Nguyen and Kim (2009) who declared that yard cranes are not recognized as a bottleneck in container terminals. The vehicles start their initial journey from predetermined L/U stations, and they finish their last journey by moving to the initial positions. In the scheduling problem, the vehicles are similar to each other, thus they are neither assigned to a specific kind of container nor to a crane. It is assumed that handling times between the vehicles and the cranes are small enough to be ignored, and the operational time of cranes (OQ), its travel time between the ship and the quay area (TQ) and travel time of the vehicles are deterministic.

In the proposed MIP model, K represents the set of cranes, the number of vehicles is represented by V , and S / F define the initial and final positions of the vehicles, respectively. The set of loading and unloading tasks are illustrated by “ L ” and “ U ”, correspondingly. 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 the two assigned T_{ki} and T_{lj} is defined by t_{kij} , and χ_{kij} is the decision variable for this journey which is a binary variable. Moreover, t_{kij} indicates the travel time between QC_k and QC_l , including the operations required for T_{ki} and T_{lj} . The required calculation for the travel time is illustrated in Figure 1. Furthermore, y_{ki} is the real completion time of crane operations in T_{ki} . The model is formulated as:

$$\text{Minimize } Z = \text{Makespan} \quad (1)$$

Subject to:

$$C_j - \bigcirc_{ki} - TQ - OQ + t_{kij} \geq M(\chi_{kij} - 1) \quad \forall k \in K, \forall j \in V, \forall i = 1, \dots, m_k, \text{ and } T_{ki} \in L, \quad (2)$$

$$C_j - \bigcirc_{ki} + t_{kij} \geq M(\chi_{kij} - 1) \quad \forall k \in K \cup S, \forall j \in V, \forall i = 1, \dots, m_k, \text{ and } T_{ki} \in U, \quad (3)$$

$$\text{Makespan} \geq C_j \quad \forall j \in V, \quad (4)$$

$$\sum_{l=1}^{K \cup F} \sum_{j=1}^{m_l} \chi_{kij} = 1 \quad \forall k \in K \cup S, \forall i = 1, \dots, m_k, \quad (5)$$

$$\sum_{k=1}^{K \cup S} \sum_{i=1}^{m_k} \chi_{kij} = 1 \quad \forall l \in K \cup F, \forall j = 1, \dots, m_l, \quad (6)$$

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

$$\chi_{kij} = 0 \text{ or } 1, \quad \forall k \in K \cup S, \forall l \in K \cup F, \forall i = 1, \dots, m_k, \forall j = 1, \dots, m_l, l = k \Rightarrow j > i. \quad (8)$$

The objective function (Equation 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 vehicles to the final positions. By minimizing the makespan of the vehicles, the crane's completion time is decreased as well. Equations (2) and (3) define the cycle time for the vehicles, comprising the time that the vehicle receives or delivers the last container to the crane, and the travel time of last journey of the vehicle to the assigned final position. In loading tasks, the crane continues its operation after receiving the container. However, the vehicle is free to continue its journey (that is, the completion time of crane's operation is greater than the release time of the vehicle by its TQ and OQ). On the other hand, in unloading tasks, the completion time of the crane and release time of the vehicle are equal. The makespan is the largest cycle time of the vehicles calculated through Equation (4) to (6) implying that there should be

a one to one relation between two consecutive tasks including the initial and final journeys of the vehicles. Equation (7) indicates that the completion time of T_{lj} on QC_l (y_{lj}) is related to y_{lj-1} and y_{ki} . 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, Equation (8) defines χ_{kij} as a binary decision variable.

Simulated annealing algorithm

The simulated annealing algorithm, which is a compact and robust technique in single and multi-objective optimization problems, is proposed to solve the integrated scheduling problem in a reasonable computation time. The SA mimics the concept of thermodynamics with the way metals cool and are annealed. The liquid metal has a high level of energy and its atoms are free to move around, and to change the structure of the metal. 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 (Suman and Kumar, 2006). The algorithm simulates the cooling process until the system converges to a steady (frozen) state.

The SA algorithm differs from iterative algorithms in that it has a mechanism serving it to escape from local optimum and rather to reach global optimum (Dereli and Sena Das, 2010). Typically, the algorithm searches for the solutions M times in each temperature level. Therefore, the initial, and the final temperatures in addition to the number of generated solutions (trials) in each temperature are the control parameters of the SA. The SA accepts better solutions in its procedure in addition to worse solutions with an acceptance probability. The acceptance probability decreases during the process of the SA. This means that as the temperature decreases, the probability of accepting worse neighborhood solutions reduces. The steps of the SA are presented in Table 2, in which, T_i , T_f , and T_r are the initial, final, and current temperatures, respectively.

Decrement of the temperature or the cooling process is the most important procedure to ensure that the SA converges to a near optimal solution. Three various cooling processes are implemented in this paper. Naderi et al. (2009) described linear cooling (LC) process, in which the temperature decreased by a linear function illustrated in Equation (9). Moreover, they proposed exponential cooling (EC) process, accessible in Equations (10) and (11). Equation (12) shows the proposed cooling (PC) process presented by Chen and Shahandashti (2009), in which α is the cooling ratio. The higher the cooling ratio, the faster the temperature cools down. In Equations (9) to (12), R and r are the total number of decrements and the current number of times the temperature has been decreased, respectively.

$$T_r = T_i - r \frac{T_i - T_f}{R} \quad r = 1, 2, \dots, R \quad (9)$$

$$T_r = \frac{A}{r+1} + B \quad r = 1, 2, \dots, R \quad (10)$$

$$A = \frac{(T_i - T_f)(R+1)}{R} \quad \text{and} \quad B = T_i - A \quad (11)$$

$$T_r = T_i e^{(-\alpha r)} \quad r = 1, 2, \dots, R \quad (12)$$

In the integrated scheduling problem, a feasible solution can be proposed as a string of the QC tasks observing the preceding relations of the tasks. The tasks are numbered 1 to N (total number of tasks) which can be calculated using Equation (13). To construct

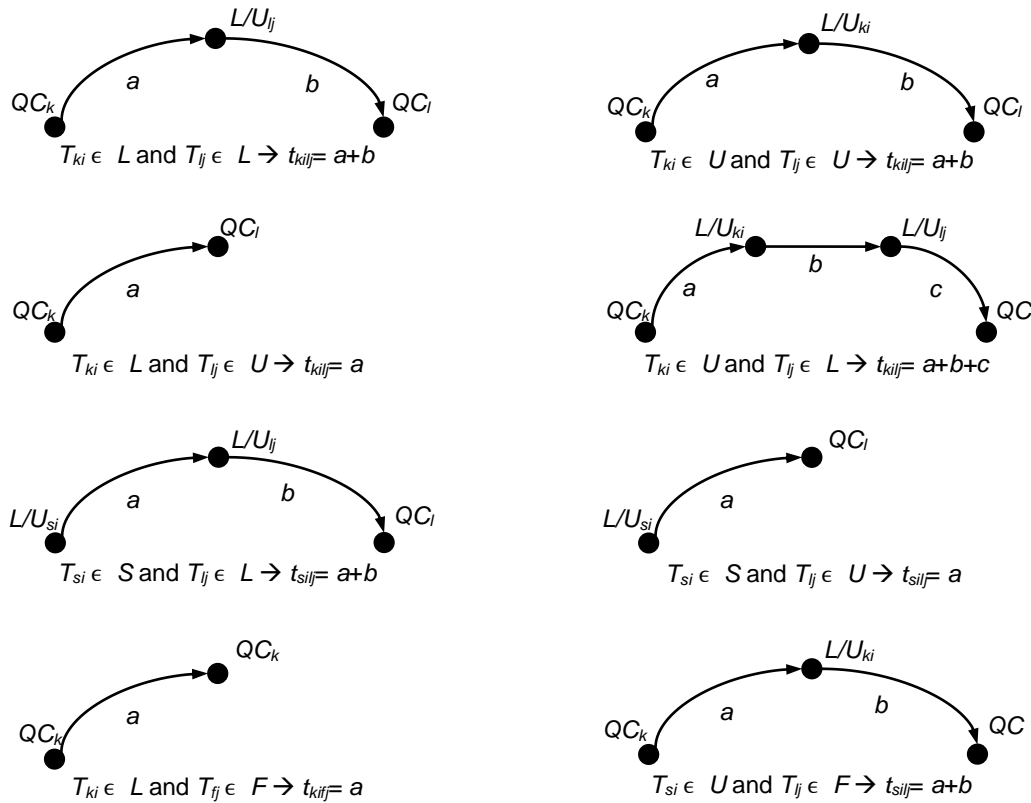


Figure 1. An illustration for AGV travel time calculations.

Table 1. Calculations for constraint (7) on $y_{ij} - A \geq M(\chi_{kij} - 1)$.

| T_{ij} | $T_{(i-1)}$ | T_{ki} | A |
|----------|-------------|----------|---|
| L | L | L | $\text{Max}(y_{ij-1} + TQ, y_{ki} - TQ - OQ + t_{kij}) + TQ + OQ$ |
| L | U | L | $\text{Max}(y_{ij-1}, y_{ki} - TQ - OQ + t_{kij}) + TQ + OQ$ |
| L | L | U | $\text{Max}(y_{ij-1} + TQ, y_{ki} + t_{kij}) + TQ + OQ$ |
| L | U | U | $\text{Max}(y_{ij-1}, y_{ki} + t_{kij}) + TQ + OQ$ |
| U | U | U | $\text{Max}(y_{ij-1} + 2TQ + OQ, y_{ki} + t_{kij})$ |
| U | L | U | $\text{Max}(y_{ij-1} + TQ + OQ, y_{ki} + t_{kij})$ |
| U | U | L | $\text{Max}(y_{ij-1} + 2TQ + OQ, y_{ki} - TQ - OQ + t_{kij})$ |
| U | L | L | $\text{Max}(y_{ij-1} + TQ + OQ, y_{ki} - TQ - OQ + t_{kij})$ |

an initial solution, a random string of tasks is produced, and the tasks for each crane are sorted considering preceding relations of the tasks.

$$N = \sum_{i=1}^K m_i \quad (13)$$

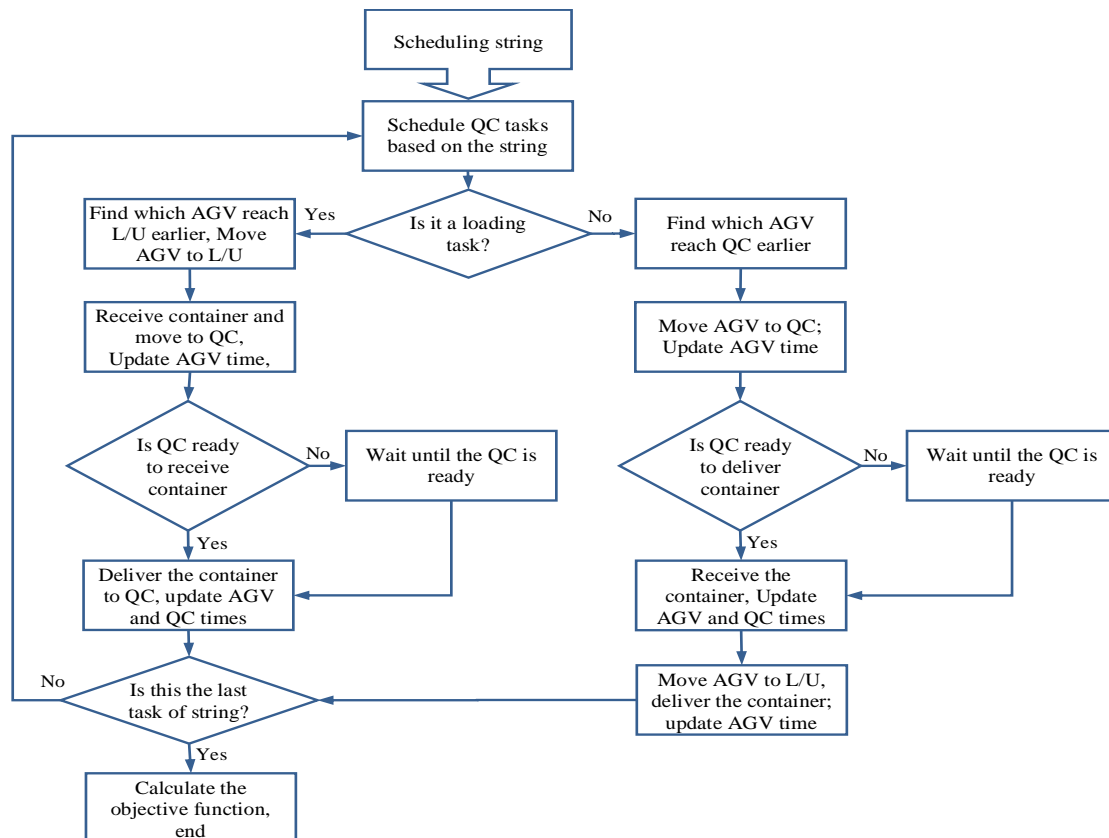
A heuristic algorithm is applied to assign the vehicles to the tasks, which calculates the objective function of the problem. The algorithm starts with a string of QC tasks proposed by the SA; and will assign AGVs to all the QC tasks. With reference to Figure 2, in loading tasks, the vehicle should pick up the container from the L/U station. Therefore, the algorithm selects the nearest vehicle to

the desired L/U station. On the other hand, in unloading tasks, the vehicle should receive the container in the QC and deliver it to the assigned L/U station. In such cases, the algorithm searches for the nearest vehicle to the QC. The AGVs need to be loaded/ unloaded by a crane. Therefore, in QC or L/U station, each of the equipment that arrives at the place earlier should wait for the other one.

To generate the new solution in each trial of the SA a "swap" neighborhood search structure is applied. In this structure, two tasks belonging to two different QCs are selected randomly. They are sorted according to their position in the solution string. If the task in the first position is the last task of a crane, or its successor task is located after the second position, it can be substituted into the second position. On the other hand, if the task in the second

Table 2. The proposed simulated annealing algorithm.

1. Initialize control parameters: T_i , T_f and M ,
2. Generate the initial solution, x , and evaluate the objective function ($E(x)$),
3. Initialize the inner loop, $r=0$,
 - 3.1 Generate a new solution y in neighborhood of x , and evaluate the objective function ($E(y)$),
 - 3.2 Calculate $\Delta E = E(x) - E(y)$; If $\Delta E < 0$ then accept the new solution and, make it as the current solution by setting $x = y$. Update the existing optimal solution, go to step 3.4
 - 3.3 If $\Delta E \geq 0$ then, $P = \exp(-\Delta E/T_r)$, If $P \geq \text{rand}(0,1)$ then accept the new solution and, make it the current solution by setting $x = y$.
 - 3.4 $r = r + 1$; If $r > M$ then go to step 4, else go to step 3.1
- 4 Cool down the current temperature (e.g. $T_r = T_i - r \frac{T_i - T_f}{R}$)
- 5 If $T_r > T_f$ then go to step 3, else, go to step 6.
- 6 Terminate the algorithm, print the best solution

**Figure 2.** A heuristic algorithm to dispatch AGVs among QC tasks.

position is the first task of a crane, or its predecessor is located before the first position, it can be substituted into a former position. The tasks are swapped if the abovementioned rules are true for both selected tasks.

NUMERICAL EXPERIMENTS AND DISCUSSION

Two sets of numerical test cases were designed to

evaluate the performance of the proposed SA algorithm. The test cases were planned in a typical container terminal containing six cranes and six L/U stations to serve the container ships. The layout and the travel time between cranes and L/Us were described by Lau and Zhao (2008). The performance of the SA is compared with the optimal solutions obtained by the MIP model in five small-size test cases. In these cases, two or three

Table 3. Specifications of five numerical small size test cases.

| No. | Task*QC* L/U*AGV | QC no. | L/U station | AGV ini pos | 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 4. Specifications of five numerical medium size test cases.

| No. | Task*QC* L/U*AGV | QC no. | L/U station | AGV ini pos | 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,U; U,U,L,U,L,U,L,U,U,L; L,L,U,U,L,U,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,L,U; U,L,U,L,L,U,L,U,U,L,U,L,U; L,U,U,L,U,U,L,U,L,U,L,U,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,U,L; L,U,L,L,U,L,U,U,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,U; L,U,U,L,U,L,L,U,U,L,U; U,L,L,U,U,L,L,U,L,U,U; 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,U,U; U,L,L,U,U,L,L,U,U,L,U,L,U; L,L,U,U,L,U,L,U,L,U,L,U; L,U,L,U,L,U,L,U,L,U,L,U. |

cranes are assigned to a containership and less than six tasks are allocated to each crane. Table 3 presents characteristics of the small-size test cases. In medium-size test cases, three to four cranes are assigned to a containership, and more than 10 tasks assigned to each crane. The medium-size test cases were designed to evaluate the performance of various cooling processes and control parameters. Table 4 describes details of the medium-size test cases. In all the test cases, the OQ for loading and unloading tasks is set to 20 s. Moreover, the TQ is equal to 10 s for both loaded and empty journeys. The MIP model was solved by using branch and bound (B&B) solver of *Lingo*[®], while the SA codes were programmed in *MATLAB*[®]. The software packages were executed on an Intel Pentium 2.13 GHz computer, holding 4 GB of RAM and running under Windows 7[®].

The first set of experiments is to determine the best control parameters for each cooling processes of the SA. The test case number M5 was selected to perform the analysis. The effects of various numbers of trials in each temperature, and the initial temperatures were examined for linear and exponential cooling processes. Moreover, the effects of various α , and T_i , on the final solution for PC were tested. The final temperature in all the experiments was set to 1. According to a trial and error process, α was selected in a way that the total number of generations for PC under various T_i s is set to be equal to those for LC and EC. Thus, for $T_i = 500, 1000, 2000$ and 5000 , α was selected as 0.0124, 0.0061, 0.0038, and 0.0017, respectively. For every combination of T_i and M (that is, 84 combinations for all the cooling processes), the SA was repeated five times; and the mean of the runs

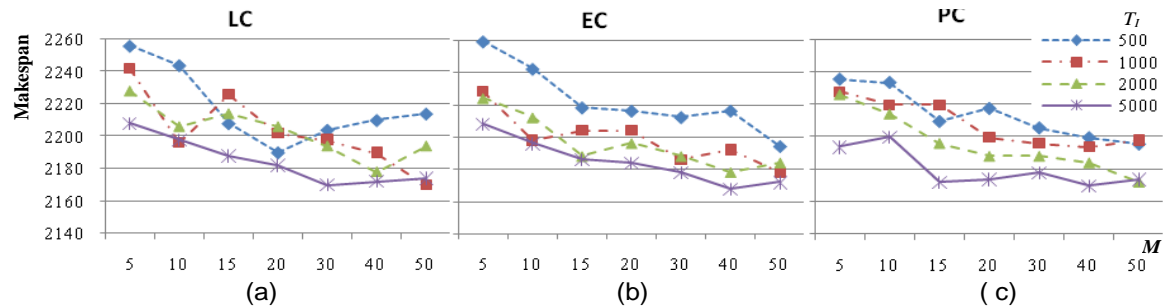


Figure 3. Effects of T_i and M on mean of the best solutions for various cooling processes for the test case M5.

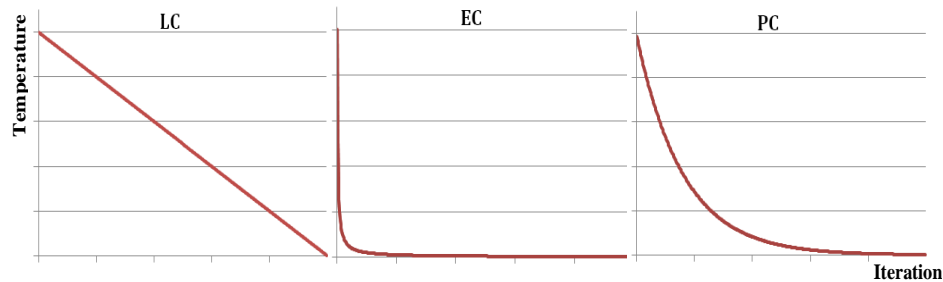


Figure 4. Decrement of temperature under various cooling processes.

was compared in Figure 3a to c for LC, EC, and PC, respectively. Furthermore, Figure 4 shows how to decrease the temperature by LC, EC and PC.

The main trend of the outcomes shows that as the total number of generated solutions increases, the mean of the best solution decreases. However, the number of trials in each temperature, M , influences the final result more than the number of generated solutions. For example, both $T_i = 1000$, $M = 50$, and $T_i = 5000$, $M = 10$ generate 50,000 solutions in the SA based on linear cooling process. Incidentally, the mean of solutions in the former is less than that of the later (Figure 3a). Table 5 presents the best solution obtained under various combinations of T_i and M . The results in Table 5 show that all the cooling processes are able to find the best solutions for M5 (2130).

The experiment shows that the CPU computation time for the SA is directly related to the total number of generations; and the type of cooling process does not affect it. To select the best control parameters for cooling processes, the authors considered the mean, and the best solutions, in addition to the computation time. As a result, $T_i = 1000$, $M = 50$ for LC, $T_i = 5000$, $M = 40$ for EC, and $T_i = 2000$, $M = 50$ for PC were selected for the remainder of the experiments. The results show that the mean of the obtained solutions for the problem, under any of the cooling processes and various control parameters, is only 1.8% worse than the best available solution. This indicates the acceptable performance of the algorithm to converge to a good solution.

The second set of experiments involves evaluating the performance of various cooling processes in the medium-size test cases. The test cases were solved by using the LC, EC, and PC processes under the control parameters determined in the previous set of experiments. The mean of five runs of the SA and CPU computation time for every test case as well as their best solution are presented in Table 6. The asterisk signs specify the cooling processes found the best solution for the test cases. However, the performance of the three cooling processes is very close. The results illustrate that EC and PC outperform the LC. Since the CPU time required for PC is less than what is required for EC, the proposed cooling process by Chen and Shahandashti (2009) is introduced as the best cooling process for the problem.

The last set of experiments is to demonstrate that the proposed SA algorithm is able to find near optimal solutions for the integrated scheduling problem in a reasonable computation time. The small size test cases were solved by using the MILP model and the proposed SA. The optimal solution was found by the MILP, the mean of five runs and the best solution for the SA, and the CPU computation time are tabulated in Table 7.

Furthermore, the optimality gap between the solutions of the MIP model and the SA algorithm is defined as:

$$\%Optimality\,Gap = \frac{Best-Optimal}{Optimal} \times 100 \quad (14)$$

The results indicate that the SA can find good solutions in

Table 5. The best solution for test case M5 for various T_i and M under LC, EC and PC.

| M | LC | | | | | | | EC | | | | | | | PC | | | | | | |
|-------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| T_i | 5 | 10 | 15 | 20 | 30 | 40 | 50 | 5 | 10 | 15 | 20 | 30 | 40 | 50 | 5 | 10 | 15 | 20 | 30 | 40 | 50 |
| 500 | 2210 | 2200 | 2150 | 2150 | 2170 | 2190 | 2210 | 2270 | 2160 | 2170 | 2180 | 2140 | 2190 | 2140 | 2190 | 2210 | 2190 | 2150 | 2180 | 2180 | 2160 |
| 1000 | 2220 | 2170 | 2220 | 2150 | 2180 | 2160 | 2140 | 2180 | 2170 | 2190 | 2170 | 2160 | 2150 | 2160 | 2190 | 2190 | 2160 | 2170 | 2150 | 2170 | 2190 |
| 2000 | 2200 | 2180 | 2190 | 2200 | 2180 | 2160 | 2180 | 2200 | 2170 | 2150 | 2140 | 2170 | 2160 | 2170 | 2190 | 2180 | 2170 | 2170 | 2170 | 2150 | 2150 |
| 5000 | 2190 | 2180 | 2160 | 2130 | 2140 | 2140 | 2140 | 2190 | 2170 | 2150 | 2160 | 2160 | 2130 | 2140 | 2150 | 2180 | 2150 | 2130 | 2170 | 2160 | 2160 |

Table 6. Comparative results for various cooling processes in medium size test cases.

| Test case | LC | | EC | | PC | | The best |
|-----------|-------|----------|-------|----------|-------|----------|----------|
| | Mean | CPU time | Mean | CPU time | Mean | CPU time | |
| M1 | 1206 | 39.4 | 1202 | 340.2 | 1192* | 102.7 | 1170 |
| M2 | 1326 | 54.8 | 1312* | 398.3 | 1318 | 135.5 | 1300 |
| M3 | 1956* | 46.3 | 1962 | 389.8 | 1962 | 116.1 | 1930 |
| M4 | 1708 | 55.3 | 1678* | 405.7 | 1688* | 133.5 | 1660 |
| M5 | 2170 | 66.6 | 2168* | 478.4 | 2164* | 161.2 | 2130 |

Table 7. Comparative results of small size test cases.

| Test case | MINILP | | MINean | PC | | Optimality gap % |
|-----------|---------|-----------------|--------|----------|----------|------------------|
| | Optimal | CPU time | | The best | CPU time | |
| S1 | 510 | 27 min 38 s | 524 | 510 | 65.9 s | 0 |
| S2 | 650 | 2 h 20 min 31 s | 694 | 670 | 70.3 s | 3.1 |
| S3* | 750 | 5 h 23 min 2 s | 734 | 710 | 73.1 s | -5.3 |
| S4* | 680 | 7 h 35 min 40 s | 700 | 690 | 76.4 s | 1.4 |
| S5* | 900 | 7 h 32 min 20 s | 924 | 900 | 86.4 s | 0 |

comparison with the MIP model. In some cases, even the mean of the solutions obtained by the SA is less than the solution reached by the mixed integer programming (MIP) model. In such cases, the MIP model is not able to find the global

optimum in a reasonable CPU computation time. Therefore, the authors marked these test cases with an asterisk. The negative optimality gap in one case means that the MIP model is not able to find a better solution than that of the SA method in

its computation time.

Conclusion

A simulated annealing algorithm was proposed in

this paper to solve the scheduling of quay cranes tasks considering the availability of the automated guided vehicles. The problem was formulated as a mixed integer linear programming model in order 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 from the initial to the final positions. This objective function decreases both the travel time of AGVs and the completion time of the QC tasks. For the sake of the problem statement, three cooling processes were proposed for the SA algorithm, and the effects of various control parameters namely the initial temperature, and number of trials in each temperature were examined through two sets of test cases. Moreover, the small-size test cases were solved by the mathematical model and the SA algorithm. The comparative results indicate that the SA is able to find near optimal solutions in low CPU computation time.

Moreover, the SA can also find the near optimal solution for medium-size cases (30 to 60 tasks in a scheduling horizon) in reasonable CPU computation time. Therefore, the proposed SA is applicable for real cases. For future researches, the most interested topic is to investigate on the automated lifting vehicles. The integrated scheduling of all the types of handling equipment in the CTs is the other topic for future researches. Moreover, it is recommended to analyze the uncertainty in operational and travel time of QCs. The failure of the AGVs and their varying velocity may affect the final solutions of the problem.

REFERENCES

- Abdullah S, Shaker K, Shaker H (2011). Investigating a round robin strategy over multi algorithms in optimising the quality of university course timetables. *Int. J. Phys. Sci.*, 6(6): 1452-1462.
- Bierwirth C, Meisel F (2010). A survey of berth allocation and quay crane scheduling problems in container terminals. *Eur. J. Oper. Res.*, 202(3): 615-627.
- Cao JX, Lee D-H, Chen JH, Shi Q (2010). The integrated yard truck and yard crane scheduling problem: Benders' decomposition-based methods. *Transport. Res. E-Log.*, 46(3): 344-353.
- Chen P-H, Shahandasthi SM (2009). Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints. *Automat. Constr.*, 18(4): 434-443.
- Dereli T, Sena Das G (2010). A hybrid simulated annealing algorithm for solving multi-objective container-loading problems. *Appl. Artif. Intell.*, 24(5): 463-486.
- Ding JF, Chou CC (2011). A fuzzy MCDM model of service performance for container ports. *Sci. Res. Essays*, 6(3): 559-566.
- Kim KH, Moon KC (2003). Berth scheduling by simulated annealing. *Transport. Res. B-Meth.*, 37(6): 541-560.
- Lau HYK, Zhao Y (2008). Integrated scheduling of handling equipment at automated container terminals. *Ann. Oper. Res.*, 159(1): 373-394.
- Lee DH, Cao JX, Shi QX (2008). Integrated quay crane and yard truck schedule for inbound containers. Paper presented at the IEEE International Conference on Industrial Engineering and Engineering Management, IEEM 2008, Singapore, pp. 1219-1223.
- Lee DH, Cao Z, Meng Q (2007). Scheduling of two-transtainer systems for loading outbound containers in port container terminals with simulated annealing algorithm. *Int. J. Prod. Econ.*, 107(1): 115-124.
- Mak K, Zhang L (2009). Simultaneous Scheduling of Import and Export Containers Handling in Container Terminals. Paper presented at the World Congress on Engineering, London, UK, pp. 604-609.
- Meersmans PJM, Wagelmans APM. (2001). Dynamic scheduling of handling equipment at automated container terminals. Rotterdam, The Netherlands: Erasmus University.
- Naderi B, Zandieh M, Khaleghi Ghoshe Balagh A, Roshanaei V (2009). An improved simulated annealing for hybrid flowshops with sequence-dependent setup and transportation times to minimize total completion time and total tardiness. *Expert Syst. Appl.*, 36(6): 9625-9633.
- Nguyen VD, Kim KH (2009). A dispatching method for automated lifting vehicles in automated port container terminals. *Comput. Ind. Eng.*, 56(3): 1002-1020.
- Norozi A, Ariffin M, Ismail N, Mustapha F (2011). An optimization technique using hybrid GA-SA algorithm for multi-objective scheduling problem. *Sci. Res. Essays*, 6(8): 1720-1731.
- Notteboom TE (2006). The time factor in liner shipping services. *Mar. Econ. Log.*, 8(1): 19-39.
- Reddy B, Rao C (2006). A hybrid multi-objective GA for simultaneous scheduling of machines and AGVs in FMS. *Int. J. Adv. Manuf. Tech.*, 31(5): 602-613.
- Stahlbock R, Voß S (2008). Operations research at container terminals: a literature update. *OR Spectrum*, 30(1): 1-52.
- Subbaiah K, Rao MN, Rao KN (2009). Scheduling of AGVs and machines in FMS with makespan criteria using sheep flock heredity algorithm. *Int. J. Phys. Sci.*, 4(2): 139-148.
- Suman B, Kumar P (2006). A survey of simulated annealing as a tool for single and multi-objective optimization. *J. Oper. Res. Soc.*, 57: 1143-1160.
- UNCTAD. (2010). Review of maritime transport. New York and Geneva: United Nations.
- Vasili M, Tang SH, Homayouni SM, Ismail N (2008). A statistical model for expected cycle time of SP-AS/RS: an application of Monte Carlo simulation. *Appl. Artif. Intell.*, 22(7): 824-840.
- Zeng Q, Yang Z (2009). Integrating simulation and optimization to schedule loading operations in container terminals. *Comput. Oper. Res.*, 36(6): 1935-1944.