

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



An Intelligent Marshaling Based on Transfer Distance of Containers Using a New Reinforcement Learning for Logistics

Yoichi Hirashima
Osaka Institute of Technology
Japan

1. Introduction

Recent shipping amount in maritime transportation keeps growing, and efficient material handling operations at marine ports becomes important issue. In many cases, containers are used for transportation of cargos, and thus the growth of shipping amount leads to the growth of the number of containers. In a marine port, containers are shifted between seaborne and landside transportation at container yard terminal. Especially, shifting containers from landside into a vessel is highly complex, including many constraints and sensitive parameters. In addition, the complexity grows at an exponential rate according to the linear growth of the number of containers. Thus, the material handling operation occupies a large part of the total run time of shipping at container terminals.

This chapter addresses to improve throughput of the material handling operations for loading container into a vessel by using reinforcement learning. Commonly, each container in a vessel has its own position determined by the destination, weight, owner, and so on (Günther & Kim, 2005). Thus, the containers have to be loaded into a vessel in a certain desired order because they cannot be rearranged in the ship. Therefore, containers must be rearranged before loading if the initial layout is different from the desired layout. Containers carried into the terminal are stacked randomly in a certain area called bay and a set of bays are called yard. The rearrangement process conducted within a bay is called marshaling.

In the problem, the number of stacks in each bay is predetermined and the maximum number of containers in a stack is limited. Containers are moved by a transfer crane and the destination stack for the container in a bay is selected from the stacks being in the same bay. In this case, a long series of container movements is often required to achieve a desired layout, and results that are derived from similar initial layouts can be quite different. Problems of this type have been solved by using techniques of optimization, such as genetic algorithm (GA) and multi agent method (Koza, 1992; Minagawa & Kakazu, 1997). These methods can successfully yield some solutions for block stacking problems. However, they adopt the environmental model different from the marshaling process, and cannot be applied directly to generate marshaling plan to obtain the desired layout of containers.

Another candidate for solving the problem is the reinforcement learning (Watkins & Dayan, 1992), which is known to be effective for learning under unknown environment that has the Markov Property. The Q-learning, one of the realization algorithm for the reinforcement learning, can be applied to generate marshaling plan, with evaluation-values for pairs of the

layout and movement of container. These values are called Q-value. The optimal series of container movements can be obtained by selecting the movement that has the best evaluation for each layout. However, conventional Q-learning has to store evaluation-values for all the layout-movement pairs. Therefore, the conventional Q-learning has great difficulties for solving the marshaling problem, due to its huge number of learning iterations required to obtain admissible plan. Recently, a Q-learning method that can generate marshaling plan has been proposed (Motoyama et al., 2001). Although these methods were effective for several cases, the desired layout was not achievable for every trial so that the early-phase performances of learning process can be spoiled. Modified methods (Hirashima et al., 2005; 2006) has been proposed to improve marshaling plan, and the environmental model considering groups of containers is shown to be effective to reduce the total number of movements of containers.

This chapter introduces a new environmental model integrated in reinforcement learning method for marshaling plan in order to minimize the transfer distance of container-movements. A container transfer process consists of 4 elements: 1. holding a container by a crane, 2. removing the container, 3. placing the container, and 4. releasing it from the crane. In the proposed method, elements 1., 3. and 4. are evaluated by the number of container-movements, and the transfer distance of container-movements in the element 2. is considered by using a weighted cost of a container-movement. Then, evaluation values reflect the total transfer-distance of containermovements, and the distance is minimized by using the reinforcement learningmethod. Consequently, selecting the best evaluation values leads the best series of container movements required to obtain a desired layout. Moreover, each rearranged container is placed into the desired position so that every trial can achieve one of desired layouts. In addition, in the proposed method, each container has several desired positions in the final layout, and the feature is considered in the learning algorithm. Thus, the early-phase performances of the learning process can be improved.

The remainder of the chapter is organized as follows. The marshaling process in container yard terminals is elaborated in section 2, followed by problem description. Also, in this section, desired positions and transfer distance of containers are newly explained. In section 3, a new Q-Learning algorithmbased on the transfer distance of containers is detailed. Computer simulations are conducted for several cases and proposed method is compared to conventional ones in section 4. Finally, concluding remarks are given in section 5.

2. Problem description

Figure 1 shows an example of container yard terminal. The terminal consists of containers, yard areas, yard transfer cranes, and port crane. Containers are carried by trucks and each container is stacked in a corresponding area called bay and a set of bays constitutes a yard area. k containers, c_i ($i = 1, \dots, k$), are assumed to be exist in a bay, and each bay has n_y stacks that m_y containers can be laden. A position of each container is discriminated by using discrete position numbers, $1, \dots, n_y \cdot m_y$. Then, the position of the container c_i is described by x_i ($1 \leq i \leq k, 1 \leq x_i \leq m_y \cdot n_y$), and the state of a bay is determined by the vector, $x = [x_1, \dots, x_k]$. Figure 2 shows an example of initial layout of container and the desired layout for $k = 8, m_y = n_y = 4$. In the figure, the state vector for the initial layout is $[13, 10, 9, 11, 14, 15, 16, 12]$, and $[13, 14, 15, 16, 9, 10, 11, 12]$ for the desired layout.

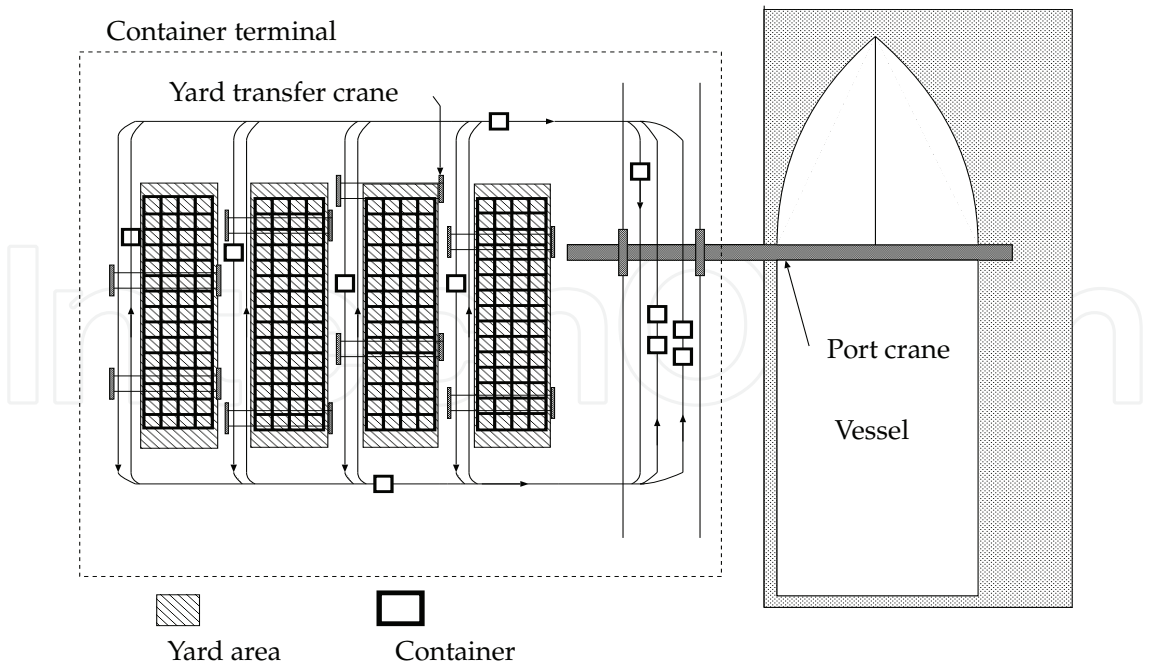


Fig. 1. Container terminal

2.1 Grouping

The desired layout in a bay is generated based on the loading order of containers that are moved from the bay to a vessel. The container to be loaded into the vessel can be located at any stack if it is on top of the stack. This feature yields several desired layouts for the bay.

2.1.1 Horizontal group

In the addressed problem, when containers on different stacks are placed at the same height in the desired layout of a bay, it is assumed that the positions of such containers can be exchanged. Figure 3 shows an example of desired layouts, where $m_y = n_y = 3, k = 9$. In the figure, containers are loaded into the vessel in the descendent order. Then, containers c_7, c_8, c_9 are in the same group (group₁), and their positions are exchanged because the loading order can be kept unchanged after the exchange of positions. In the same way, c_4, c_5, c_6 are in the group₂, and c_1, c_2, c_3 are in the group₃ where positions of containers can be exchanged. Consequently several candidates for desired layout of the bay are generated from the original

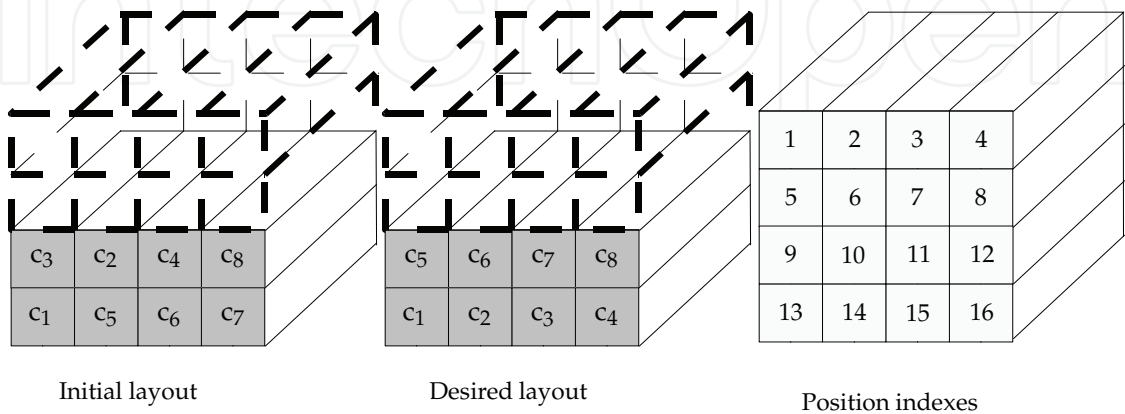


Fig. 2. Example of container layouts in a Bay

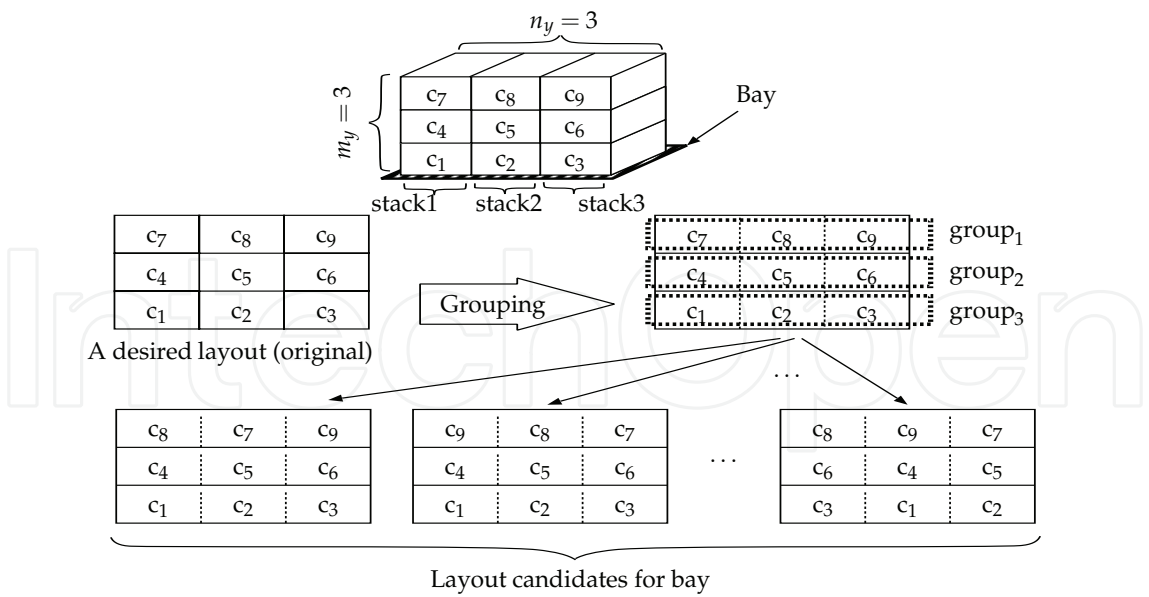


Fig. 3. Horizontal group

desired-layout.

2.1.2 Heap group

In addition to the horizontal grouping, a “heap group” for n_y containers at the top of stacks in the original desired-layout (group₁) is generated as follows:

- 1. n_y containers in group₁ can be placed at any stacks if their height is same as the original one.
- 2. Each of them can be stacked on other $n_y - 1$ containers when both of followings are satisfied:
 - (a) They are placed at the top of each stack in the original disired-layout,
 - (b) The container to be stacked is loaded into the ship before other containers being under the container.

Other groups are the same as ones in the horizontal group, so that the heap group contains all the desired layout in the horizontal group.

Figure 4 depicts an example of heap group for $k = 9, n_y = 3$. In the figure, containers are loaded into a vessel by the order c_9, c_8, c_7, \dots . Then, c_9 can be placed on c_7 and c_8 , c_8 can be placed on c_7 , so that the number of desired layouts is increased.

2.2 Marshaling process

The marshaling process consists of 2 stages: ① selection of a container to be rearranged, and ② removal of the containers on the selected container in ①. After these stages, rearrangement of the selected container is conducted. In the stage ②, the removed container is placed on the destination stack selected from stacks being in the same bay. When a container is rearranged, n_y positions that are at the same height in a bay can be candidates for the destination. In addition, n_y containers can be placed for each candidate of the destination. Then, defining t as the time step, $c_a(t)$ denotes the container to be rearranged at t in the stage ①. $c_a(t)$ is selected from candidates $c_{y_{i_1}}$ ($i_1 = 1, \dots, n_y^2$) that are at the same height in a desired layout.

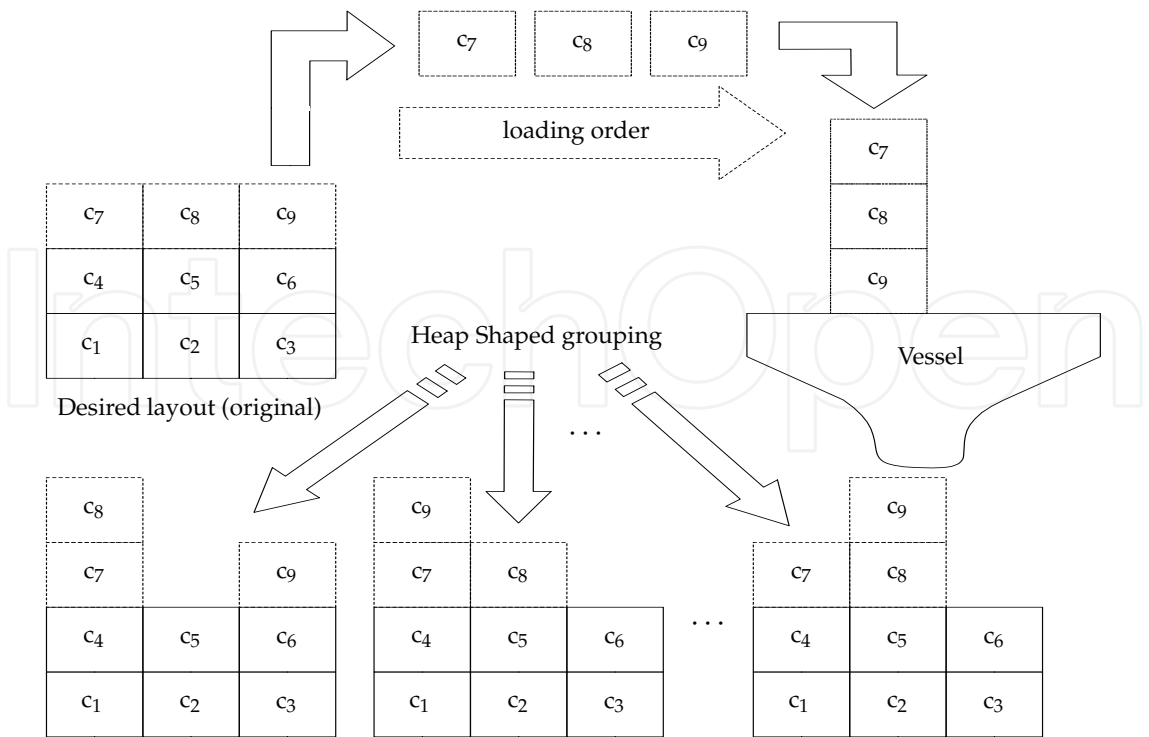


Fig. 4. Heap group

A candidate of destination exists at a bottom position that has undesired container in each corresponding stack. The maximum number of such stacks is n_y , and they can have n_y containers as candidates, since the proposed method considers groups in the desired position. The number of candidates of $c_a(t)$ is thus $n_y \times n_y$. In the stage ②, the container to be removed at t is $c_b(t)$ and is selected from two containers $c_{y_{i_2}}$ ($i_2 = 1, 2$) on the top of stacks. c_{y_1} is on the $c_a(t)$ and c_{y_2} is on the destination of $c_a(t)$. Then, in the stage ②, $c_b(t)$ is removed to one of the other stacks in the same bay, and the destination stack $u(t)$ at time t is selected from the candidates u_j ($j = 1, \dots, n_y - 2$). $c_a(t)$ is rearranged to its desired position after all the $c_{y_{i_2}}$ s are removed. Thus, a state transition of the bay is described as follows:

$$x_{t+1} = \begin{cases} f(x_t, c_a(t)) & \text{(stage ①)} \\ f(x_t, c_b(t), u(t)) & \text{(stage ②)} \end{cases} \tag{1}$$

where $f(\cdot)$ denotes that removal is processed and x_{t+1} is the state determined only by $c_a(t), c_b(t)$ and $u(t)$ at the previous state x_t . Therefore, the marshaling plan can be treated as the Markov Decision Process.

The objective of the problem is to find the best series of movements which transfers every container from an initial position to the goal position. The goal state is generated from the shipping order that is predetermined according to destinations of containers. A series of movements that leads a initial state into the goal state is defined as an episode. The best episode is the series of movements having the minimum transfer distance of containers to achieve the goal state.

3. Reinforcement learning for marshaling plan

In the selection of c_a , the container to be rearranged, an evaluation value is used for each candidate $c_{y_{i_1}}$ ($i_1 = 1, \dots, r$), where r is the number of candidates. In the same way, evaluation

values are used in the selection of the container to be removed c_b and its destination u_j ($j = 1, \dots, n_y - 2$). Candidates of c_b is $c_{y_{i_2}}$ ($i_2 = 1, \dots, n_y$). The evaluation value for the selection of $c_{y_{i_1}}$, $c_{y_{i_2}}$ and u_j at the state x are called Q-values, and a set of Q-values is called Q-table. At the l th episode, the Q-value for selecting $c_{y_{i_1}}$ is defined as $Q_1(l, x, c_{y_{i_1}})$, the Q-value for selecting $c_{y_{i_2}}$ is defined as $Q_2(l, x, c_{y_{i_1}}, c_{y_{i_2}})$ and the Q-value for selecting u_j is defined as $Q_3(l, x, c_{y_{i_1}}, c_{y_{i_2}}, u_j)$. The initial value for Q_1, Q_2, Q_3 is assumed to be 0. Then, Q_3 is updated by the following equation:

$$Q_3(l, x_t, c_a(t), c_b(t), u(t)) = (1 - \alpha)Q_3(l - 1, x_t, c_a(t), c_b(t), u(t)) + \alpha[R + V_{t+1}]$$

$$V_t = \begin{cases} \gamma \max_{y_{i_1}} Q_1(l, x_t, c_{y_{i_1}}) & \text{(stage ①)} \\ \gamma \max_{y_{i_2}} Q_2(l, x_t, c_a(t), c_{y_{i_2}}) & \text{(stage ②)} \end{cases} \quad (2)$$

where γ , ($0 < \gamma < 1$) denotes the discount factor and α is the learning rate. Reward R is given only when the desired layout has been achieved.

In the selection of $c_b(t)$, the evaluation value $Q_3(l, x, c_a(t), c_b(t), u_j)$ can be referred for all the u_j ($j = 1 \dots n_y - 2$), and the state x does not change. Thus, the maximum value of $Q_3(l, x, c_a(t), c_b(t), u_j)$ is copied to $Q_2(l, x, c_a(t), c_b(t))$, that is,

$$Q_2(l, x, c_a(t), c_b(t)) = \max_j Q_3(l, x, c_a(t), c_b(t), u_j). \quad (3)$$

In the selection of $c_a(t)$, the evaluation value $Q_1(l, x, c_a(t))$ is updated by the following equations:

$$Q_1(l, x_t, c_a(t)) = (1 - \alpha)Q_1(l - 1, x_t, c_a(t)) + \alpha[R + V_{t+1}]. \quad (4)$$

In order to reflect the transfer distance of the removed container into the corresponding evaluation value, the discount factor γ is used. In the proposed method, γ has smaller value for larger transfer distance, so that smaller transfer distance has better evaluation. In the following, the calculation method of γ is explained.

Define D as the transfer distance of a removed container. For simplicity, D is set as 1 for a removal between neighboring positions in the horizontal or the vertical direction. Then, by assuming each container is moved either horizontal or vertical direction, the maximum value of D satisfies $\max D = 2m_y + n_y - 1 \stackrel{\text{def}}{=} D_{\max}$. Then, $1 \leq D \leq D_{\max}$. Figure 5 shows an examples of D_{\max} for $m_y = n_y = 6$. Using D and D_{\max} , γ is calculated as follows:

$$\gamma = \frac{D_{\max} - (\beta D - 1)}{D_{\max}}. \quad (5)$$

The size of γ is determined by putting $0 < \beta < 1$, where larger transfer distance yealds larger discount so that the smaller transfer distance obtains the better evaluation value.

$c_a(t)$, $c_b(t)$, $u(j)$ is determined by the soft-max action selection method(Sutton & Barto, 1999). Probability P for selection of each candidate is calculated by

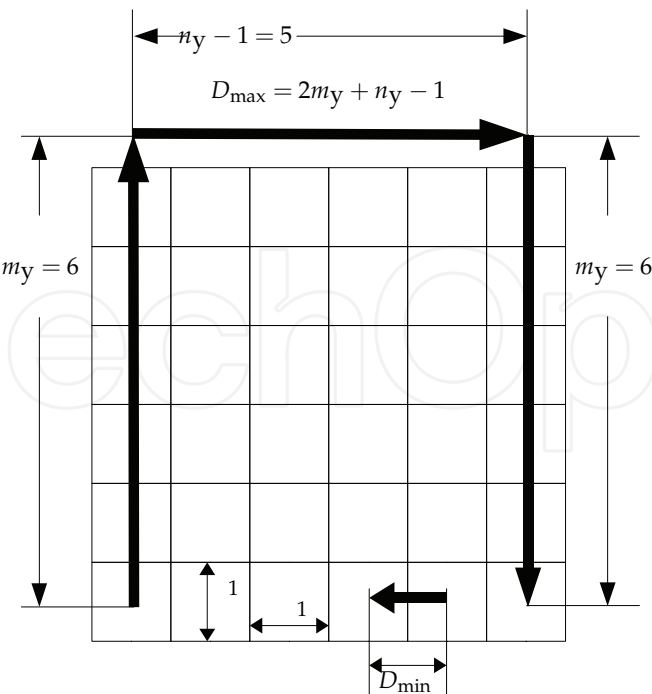


Fig. 5. Transfer distance of a container

$$\tilde{Q}(\tilde{x}, u_L) = \frac{Q(l, \tilde{x}, u_l) - \min_u Q(l, \tilde{x}, u_L)}{\max_u Q(l, \tilde{x}, u) - \min_u Q(l, \tilde{x}, u)} \tag{6}$$

$$P(\tilde{x}, u_L) = \frac{\exp(\tilde{Q}(\tilde{x}, u_L)/T)}{\sum_u \exp(\tilde{Q}(\tilde{x}, u_L)/T)} \tag{7}$$

where $\tilde{x} = x_t, u_L = c_{y_{i_1}}$ for selecting $c_a(t)$, $\tilde{x} = (x_t, c_a(t)), u_L = c_{y_{i_2}}$ for selecting $c_b(t)$ and $\tilde{x} = (x_t, c_a(t), c_b(t)), u_L = u_j$ for selecting $u(t)$. Also, T is the thermo constant.
By using the update rule, restricted movements and goal states explained above, the learning process is described as follows:

- [1]. Rearrange $c_a(t)$ if possible, and count the number of containers being in the goal positions and store it as n
- [2]. If $n = k$, go to [10]
- [3]. Select $c_a(t)$ to be rearranged
- [4]. Store $(x, c_a(t))$
- [5]. Select $c_b(t)$ to be removed
- [6]. Store $(x, c_a(t), c_b(t))$
- [7]. Select destination position u_j for $c_b(t)$
- [8]. Store $(x, c_a(t), c_b(t), u_j)$
- [9]. Remove $c_b(t)$ and go to [5] if another $c_b(t)$ exists, otherwise go to [1]
- [10]. Update Q-values referred in [3],[5],[7].

4. Simulations

Computer simulations are conducted for $k = 18, m_y = n_y = 6$, and learning performances of following 3 methods are compared:

- (A) proposed method using horizontal grouping and heap grouping,
- (B) conventional method only using horizontal grouping (Hirashima et al., 2006),
- (C) conventional method without grouping (Hirashima et al., 2005).

Figure 6 shows the initial state of the yard and an example of desired layout. The desired layout is fixed for method (C), and is extended for methods (A),(B) by grouping. Parameters used in the proposed method are set as $\alpha = 0.8, \beta = 0.8, T = 0.1, R = 1.0$. A trial starts from a initial state and ends when all the containers are rearranged to the buffer. Containers, from c_1 to c_{18} , are loaded into a vessel by ascending order.

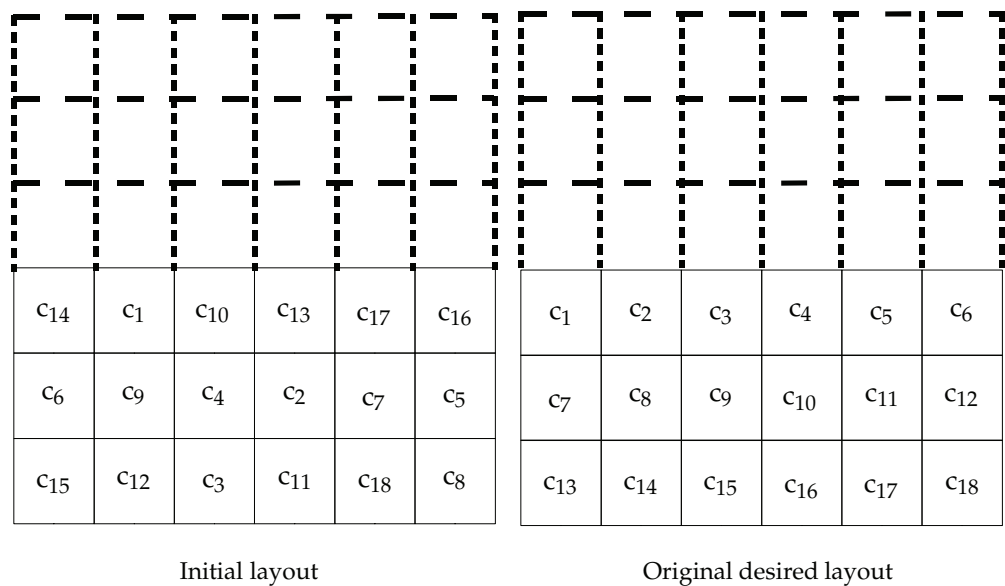


Fig. 6. Initial layout and original desired layout

Figure 7 shows the results, where the horizontal axis expresses the number of trials. The vertical axis expresses the minimum transfer-distance of removed containers to achieve a desired layout found in the past trials. Each result is averaged over 20 independent simulations. Among these simulations, dispersions of averaged data are indicated by error bars on some typical data points at 1000th, 5000th, 10000th, 50000th, 150000th, 20000th and 250000th trials. Method (A) could derive solutions with smaller transfer-distance of container as compared to methods (B),(C). Moreover, in early stages, learning performances of method (A) is much better than that of methods (B),(C), because method (A) has augmented desired layouts including the original ones in methods (B),(C), so that the method (A) has obtained the layout that can reduce the total transfer-distance of container. Figure 8 depicts final layouts generated by methods (A),(B). The final layout of method (C) is identical to the one depicted in Figure 6. In the figure, positions of containers in the same height are exchanged. In the layout obtained by method (A), c_1, c_2, c_3 are located on other containers in the same group, whereas this layout does not regarded as desired layout in methods (B),(C), so that the total distance of movements of containers is reduced.

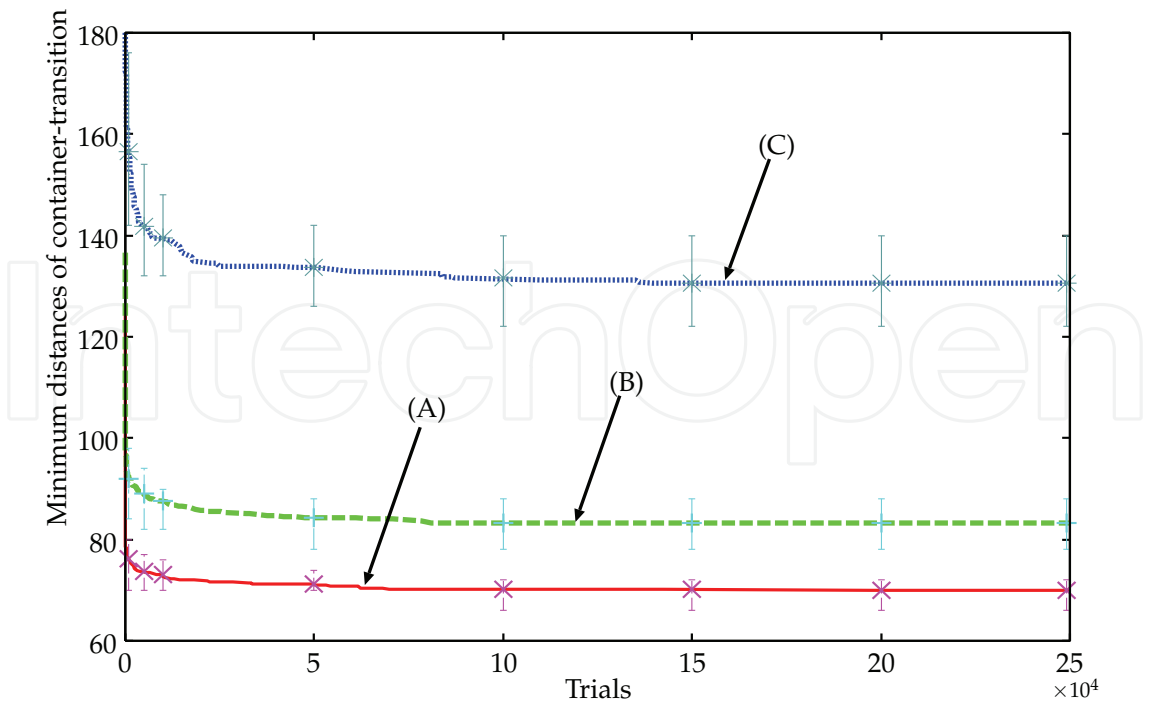


Fig. 7. Performance comparison

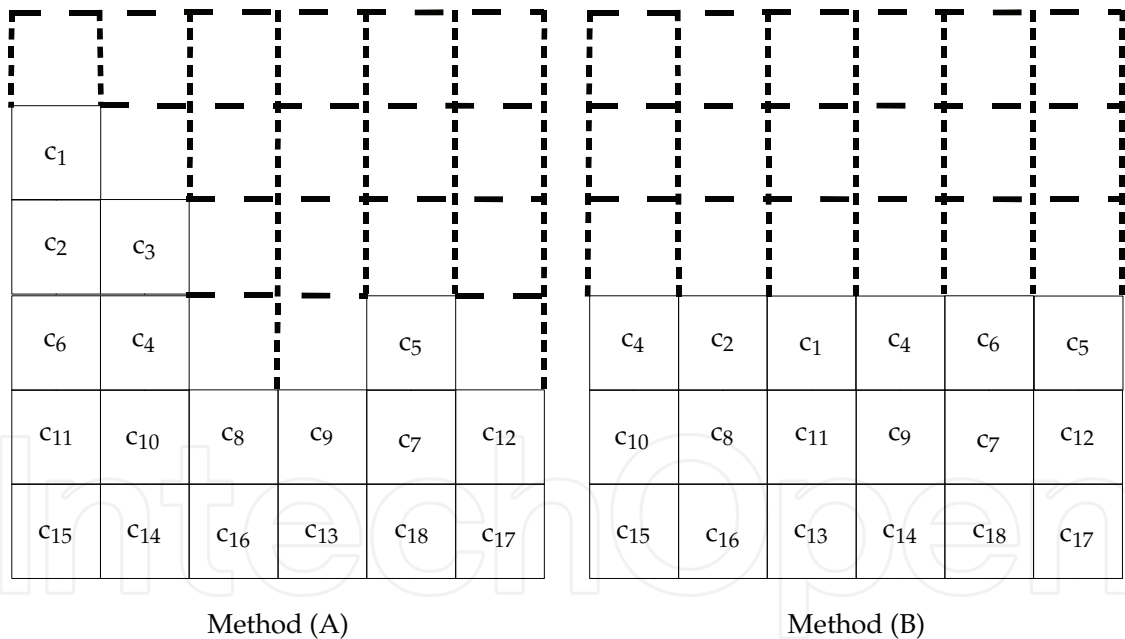


Fig. 8. Final layouts

5. Conclusions

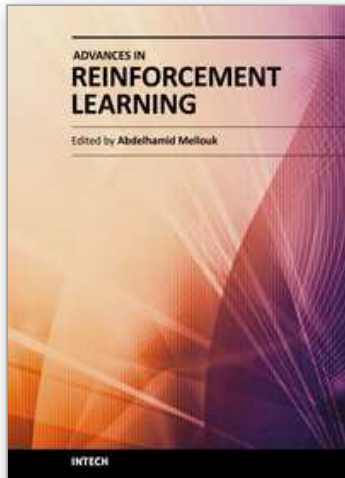
A new marshaling method for containers at marine terminals has been proposed. In the proposed method, Q-Learning is used to derive marshaling plan considering the transfer distance of container that is required to achieve desired layout of the bay. As a consequent, the layout of container, rearrange order of containers, destinations of removed containers are simultaneously optimized, so that the total transfer distance of containers is minimized. The

proposed method applied for realistic scale problems, and showed much better performance for improving solutions as compared to conventional methods.

6. References

- Günther, H. O. & Kim, K. H. (2005). *Container Terminals and Automated Transport Systems*, Springer.
- Hirashima, Y., Ishikawa, N. & Takeda, K. (2006). A new reinforcement learning for group-based marshaling plan considering desired layout of containers in port terminals, *International Conference on Networking, Sensing, and Control* pp. 670–675.
- Hirashima, Y., Takeda, K., Furuya, O., Inoue, A. & Deng, M. (2005). A new method for marshaling plan using a reinforcement learning considering desired layout of containers in terminals, *Preprint of 16th IFAC World Congress We-E16-TO/2*.
- Koza, J. R. (1992). *Genetic Programming : On Programming Computers by means of Natural Selection and Genetics*, MIT Press.
- Minagawa, M. & Kakazu, Y. (1997). An approach to the block stacking problem by multi agent cooperation, *Trans. Jpn. Soc. Mech. Eng. (in Japanese)* C-63(608): 231–240.
- Motoyama, S., Hirashima, Y., Takeda, K. & Inoue, A. (2001). A marshaling plan for container terminals based on reinforcement learning, *Proc. of Inter. Sympo. on Advanced Control of Industrial Processes* pp. 631–636.
- Sutton, R. & Barto, A. (1999). *Reinforcement Learning*, MIT Press, Cambridge.
- Watkins, C. J. C. H. & Dayan, P. (1992). Q-learning, *Machine Learning* 8: 279–292.

IntechOpen



Advances in Reinforcement Learning

Edited by Prof. Abdelhamid Mellouk

ISBN 978-953-307-369-9

Hard cover, 470 pages

Publisher InTech

Published online 14, January, 2011

Published in print edition January, 2011

Reinforcement Learning (RL) is a very dynamic area in terms of theory and application. This book brings together many different aspects of the current research on several fields associated to RL which has been growing rapidly, producing a wide variety of learning algorithms for different applications. Based on 24 Chapters, it covers a very broad variety of topics in RL and their application in autonomous systems. A set of chapters in this book provide a general overview of RL while other chapters focus mostly on the applications of RL paradigms: Game Theory, Multi-Agent Theory, Robotic, Networking Technologies, Vehicular Navigation, Medicine and Industrial Logistic.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Yoichi Hirashima (2011). An Intelligent Marshaling Based on Transfer Distance of Containers Using a New Reinforcement Learning for Logistics, *Advances in Reinforcement Learning*, Prof. Abdelhamid Mellouk (Ed.), ISBN: 978-953-307-369-9, InTech, Available from: <http://www.intechopen.com/books/advances-in-reinforcement-learning/an-intelligent-marshaling-based-on-transfer-distance-of-containers-using-a-new-reinforcement-learnin>

INTech
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen