

Deciding on Planning Windows by Partitioning Time for Yard Crane Management in Container Terminals

Xi Guo, Shell Ying Huang
School of Computer Engineering
Nanyang Technological University
Block N4, Nanyang Avenue, Singapore 639798
{guox0006@ntu.edu.sg, assyhuang@ntu.edu.sg}

ABSTRACT

This paper proposes a new dynamic Yard Crane (YC) workload partition scheme in container terminals. The main contribution is the time partitioning algorithm which finds a sequence of time windows to match closely the changing workload distribution and get the maximum benefit of the space partition algorithm. Experimental results show that the proposed time partition algorithm plays the most important role in making substantial improvements over basic partition scheme in all tested workload scenarios.

Keywords

Simulation Optimization, Performance, Decision-making, Container terminals

1 INTRODUCTION

Container ports, which serve as hubs of container transshipment, are crucial nodes and play an important role in the marine transportation network. The demand on high quality services from container terminals includes efficiency and reliability in container handling which in turn requires the port to utilize its resources efficiently.

Container terminals are open systems of material flow. Figure 1 shows a large part of a typical layout of a container terminal. In conventional terminals, the storage yard is often divided into several tens of yard blocks in a number of rows in parallel to the quay. Each yard block may have more than 30 slots (bays) of containers stored in length. Vehicles travel along lanes to transfer containers between quay side and yard side. When multiple vessels are loading and unloading, vehicles may arrive at different slot locations of a yard block for storing and retrieving containers. Ex-terminal trucks may also arrive through terminal gates to unload export containers or to load import containers.

One most important performance target of container terminal operation is to minimize vessel turn-around time. It means the Yard Cranes (YC) need to serve vehicle jobs as efficiently as possible to reduce the delay of vehicles at the yard side in order to continuously feed the quay sides to support Quay Crane (QC) operations. The main objective at the yard side is then translated to minimizing the average vehicle job waiting time for YCs service.

2 Error! No text of specified style in document.

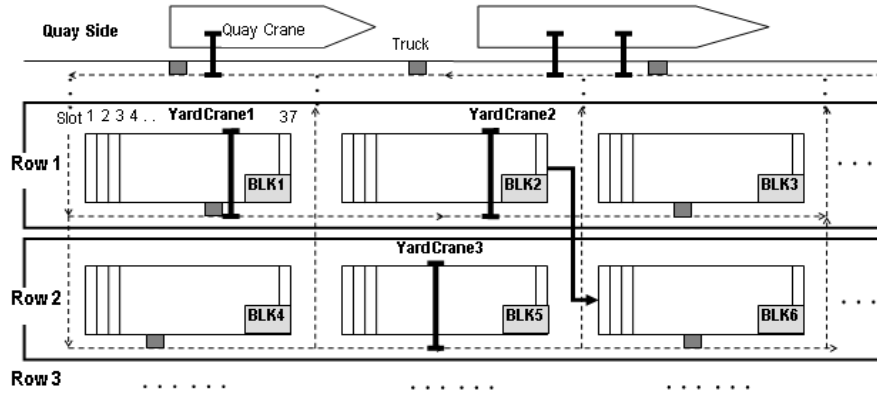


Figure 1. Layout & container flows in a typical terminal

YC deployment is a very complex problem because the workload distribution is uneven and changes dynamically over time. Both the YC and the vehicle must be at the same slot position for the loading or unloading of containers. The time an YC moves from one job location to another is referred to as gantry time. YC gantry times also contribute to vehicle waiting times and may become a significant part of a YC's busy time. The movements of a Rubber Tired Gantry (RTG) Crane include: linear gantry and cross gantry. Cross gantries are movements from one block to another in a different row, e.g. from BLK 2 to BLK 6 in Figure 1. An YC doing cross gantry has to make two 90° turns which take much longer time than linear gantry and may delay the vehicle movements by blocking the lanes.

After equipment ordering at the beginning of a shift, a common practice is to initially assign YCs to various yard blocks. Then a re-distribution of YCs among the yard blocks is done from time to time to match the dynamically changing workload. Workload of a block is commonly estimated by the number of arriving jobs expected in a future time interval or a quantity proportional to the number. These practices may lead to one of the following scenarios: (1) some blocks do not have any YCs in charge; (2) some YC are in charge of two or three blocks; (3) some blocks have more than one YC. In the first situation it will result in long vehicle waiting times. The third scenario needs carefully synchronized YC operations to avoid YC clashes and may result in long YC waiting time. In all scenarios, the workload estimation based on the number of jobs is often not accurate so workload partition usually cannot achieve the best balance among YCs. This results in low productivity of YCs and long waiting time of vehicles.

We propose to use estimated average job waiting time as metrics for workload partitioning and YC dispatching. The best partitioning and dispatching plans should yield the minimum average vehicle job waiting time calculated from all the jobs in the planning window. To reduce vehicle waiting times, we propose a hierarchical scheme for YC operation management which is organized into three levels as shown in Figure 2. Suppose a suitable number of YCs has been assigned to work for the current shift.

Level 1	Distribution of YCs to different Rows
Level 2	Time Partition into planning windows
	Space Partition into YC working zones
Level 3	YC dispatching in individual zones

Figure 2. Hierarchical YC operations management

Level 1 distributes YCs among different rows at suitable times based on predicted future workload. This is done a few times during a shift of 8 hours. **Level 2**, in between two rounds of YC re-distributions at Level 1, distributes YCs to work in various zones in each row. Observing the safety constraints to avoid crane collisions and traffic slowdowns of two neighboring cranes, RTGs are always separated by a safety distances, typically 8-slots. This is to ensure two steams of (55-foot long) vehicles to be independently served by the two YCs without delay (Petering et al. 2009). We follow this industrial concern in our scheme. **Level 3** determines the serving sequences of vehicle jobs for an YC in a service zone over a period of time (e.g. a sub-planning window). The performance of YCs at this level would collectively provide an estimation of overall performance in the upper levels.

The hierarchical scheme aims to minimize overall average vehicle job waiting time through flexible time and space partitioning of workload by re-distributions of YCs to cope with dynamic workload distribution over time at the yard side. A modified A* search algorithm (Guo and Huang 2008) is employed to find the optimal dispatching solution for an YC in a zone over a period of time at Level 3. The modified A* algorithm is able to find the optimal solution from over 2.4×10^{18} possible dispatching sequences within three seconds.

In this paper, we focus on the problem at Level 2 to find the proper time partition of planning windows and the proper space partition of zones in a planning window. To solve the partition problem, we design the search for the suitable partition of both the time and space. Sometimes the space partition of a row of yard blocks into a number of zones does not have a satisfactory solution. The reason could be lack of sufficient partition points which are areas of at least 8-slots separation between adjacent jobs to achieve a well balanced partition plan. It could also be due to the fact that workload distribution to the yard is usually uneven and may be changing by a large margin over time. Therefore, we carefully designed a time partition algorithm to achieve dynamic workload partitioning to cope with changing and uneven workload distribution. A space partition algorithm is proposed which is similar to Ng (2005) algorithm with some modifications. The main contribution is the time partitioning algorithm which finds a sequence of time windows to match closely the changing workload distribution and get the maximum benefit of the space partition algorithm.

In the rest of the paper, we discuss the related work in Section 2. The proposed algorithms are described in Section 3 and then evaluated by simulation experiments in Section 4. Finally, Section 5 concludes the paper.

2 RELATED WORK

The problems of scheduling and dispatching resources in container terminals have been widely studied in recent years (Vis and de Koster, 2003; Steenken et al., 2004; and Stahlbock and Voss, 2008). On the topic of YC management, two main problems are: (i) deciding job servicing sequence for an YC which we refers to as YC dispatching problem in this paper; (ii) allocating YCs to different parts of the yard which we refers to as YC deployment problem in this paper.

Kim and Kim (1999) studied single YC dispatching for loading with a given load plan and a given bay plan. A Mixed Integer Programming (MIP) model is proposed to minimize the total gantry time. Later, Kim and Kim (2003) and Kim et al. (2004) extended the study of this problem by comparing exact optimization, a beam search heuristic and a Genetic Algorithm (GA). For large problems, the MIP has limited applicability due to the excessive computational times, while heuristics cannot guarantee optimal solutions. In addition, the assumption of having dedicated yard cranes just to support vessel loading operations in these works is not always the best for terminals with many berths and more yard blocks than cranes.

Several works studied the dispatching problem with 2 YCs. Due to the problem complexity, commonly MIP models were employed to formulate the problems while heuristic methods to find near-optimal solutions. Jung and Kim (2006) considered 2 YCs working in one shared zone for loadings with a GA and a Simulated Annealing (SA) algorithm to minimize the make-span. Solutions with possible YC interference were considered not feasible in the MIP model. Lee et al. (2007) considered 2 YCs working in 2 non-overlapping zones with a SA algorithm to minimize the make-span. Cao et al. (2008) considered Double-Rail-Mounted gantry (DRMG) crane systems where two YCs can pass through each other along a row of blocks with a combined greedy and SA algorithm to minimize the loading time. Stahlbock and Voß (2010) also studied DRMG operations to maximize productivity. Experiments showed the proposed SA algorithm dominates the priority rule-based heuristics for high workload.

Published works focusing on inter-block level YCs deployment are not abundant. Zhang et al. (2002) studied YCs deployment among blocks. It is formulated as a MIP model to minimize total unfinished workload at the end of each planning period and is solved by a modified Lagrangian relaxation method. However, only one transfer per YC is allowed in the 4 hour planning period, which may not be enough to match the changing workload distribution. Cheung et al. (2002) also studied the problem with a Lagrangian decomposition solution and a successive piecewise-linear approximation approach. Workload in two papers is based on the number of container moves (equivalent to the number of YC jobs), which is not a good metric to partition workload as discussed in section 1.

Ng (2005) considered the problem of multiple YCs sharing a single bi-directional lane and modeled it as an IP to minimize total job completion time. The workload for each YC was estimated by a simple greedy heuristic. However, safety constraints of YC separation of a minimal distance were not considered. Petering et al. (2009) claimed that YCs could only schedule for at most 1.5 container jobs to avoid deadlocks when YCs are in charge of overlapping zones. This reinforces our belief that non-overlapping YC zones should be formed in our YC dispatching schemes to

achieve high performance while maintaining safety constraints and avoiding deadlocks in real-time settings.

3 PARTITION PLANS GENERATION AND SELECTION

YC workload distribution at the row level involves two parts: the space partitioning and the time partitioning. The space partitioning aims to partition a row into a number of non-overlapping zones for individual YCs in a given planning window. The time partitioning aims to partition the time between two rounds of YC re-distribution at Level 1 into a number of sub-planning time windows. Thus, the space partition for various sub-planning time windows may be different to cope with the uneven and changing workload distributions over the yard side while maintain the safety constraint of YC safety separations.

3.1 Space Partition

Let N be the total number of jobs expected to arrive in the row. Let slot locations be indexed by positive integers starting from 1. Obviously each zone for one YC has to be a continuous sequence of slot locations. And for safety concern, YCs need to be separated by at least 8 slots at all times. So the possible boundaries in the partition plan are areas where two neighboring job slot locations satisfy the condition $|L_i - L_{i+1}| \geq 8$. Suppose that there are R such potential boundaries like $P_1, P_2 \dots P_R$ as shown in Figure 3 with P_1 and P_R at the two ends of the row. This means each area between two boundary points is unbreakable in partitions. $R-1$ such unbreakable areas, (P_i, P_{i+1}) for $i=1, 2 \dots R-1$, exist as shown in Figure 3.

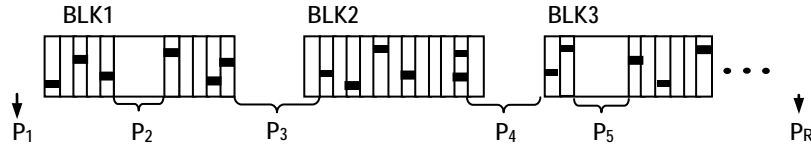


Figure 3. Possible boundary points between zones

Since the arriving job locations could be predicted, the total number of unbreakable areas $R-1$ is known at the beginning of the planning time. The number of YCs allocated to the row, denoted as M , should be no larger than the number of unbreakable areas, i.e. $M \leq R-1$. This ensures that one unbreakable area is covered by at most one YC. The problem of choosing an YC workload partition plan becomes a combinatorial problem. In other words, to allocate M yard cranes to cover the entire row, $(M-1)$ partition points are to be selected from $(R-2)$ candidate P_i , where $1 < i < R$. The total number of possible ways to partition the row into M zones is

$$\binom{R-2}{M-1} = \frac{(R-2)!}{(M-1)!(R-M-1)!}$$

We denote a part of a row starting from P_i to P_j by $Z(P_i, P_j)$ where i, j are integers and $0 < i < j < R+1$. The minimum total job waiting time whose job slot locations are in $Z(P_i, P_j)$ is denoted by $T(Z(P_i, P_j), m)$, where m is the number of YCs allocated to the zone.

We divide the problem of allocating m YCs to the area $Z(P_i, P_j)$ with the objective of minimizing the average job waiting time into smaller subproblems:

$$T(Z(P_i, P_j), m) = \begin{cases} Astar(P_i, P_j), & \text{for } m=1 \\ \min\{T(Z(P_i, P_k), 1) + T(Z(P_k, P_j), m-1)\}, & \text{for } m > 1, \text{ where } i < k \leq j - m + 1 \end{cases}$$

When m is equal to one, all jobs whose slot locations fall in $Z(P_i, P_j)$ will be served by the same YC. In this case, the problem is to find a job servicing sequence by the YC to minimize total waiting time of these jobs. We employ a modified A* search algorithm (Guo and Huang 2008). $Astar(P_i, P_j)$ returns the minimized total job waiting time for all jobs whose job locations are in $Z(P_i, P_j)$.

When m is greater than one, the problem is to find an optimal partition point k among the boundary points such that one YC will be in charge of $Z(P_i, P_k)$ and the remaining $(m-1)$ cranes will be in charge of $Z(P_k, P_j)$. The k value that minimizes the total job waiting time is the correct choice of the partition point. The constraint of $k \leq j - m + 1$ ensures that the number of unbreakable areas in the zone for $m-1$ YCs must be greater than or equal to $m-1$. This is to avoid assigning two or more YCs to one unbreakable area.

We use the dynamic programming technique similar to Ng's (2005) algorithm. A 3-dimensional dictionary DMAP is used where $DMAP[i][j][m]$ stores the total job waiting time for $Z(P_i, P_j)$ with m YCs assigned to it. Three main differences exist from the DP approach in (Ng 2005). Firstly, they did not consider safety constraints for adjacent YCs. Secondly, they estimate the workload in a single YC zone by using a simple greedy heuristics instead of finding the optimal dispatching sequence. Thirdly, they considered every slot position as a potential boundary, which may result in an empty job zone (a, b) , where $a, b \in P_i$ (i.e., there is no job between a and b), or repeated partitioning, e.g. zone (a, b) and zone $(a, b+1)$, where $b, b+1 \in P_i$ (i.e., no job between b and $b+1$).

3.2 Time Partition

The space partition algorithm is able to find the best partition plan in a planning window with static non-overlapping zones. Following the industrial safety requirements, all zones are separated by at least 8 slots so that no collision or slowdown of two neighboring YCs will occur even when noises disturb a pre-planned YC servicing sequence. However, when a planning window is not properly selected, a partition zone may involve a large number of jobs and ends in unbalanced workload partition and poor overall YC performance.

Facing this problem, a time partition algorithm is proposed to find proper planning windows shown in Figure 4. Given current planning window T , the algorithm tries to divide window T into 2 sub-windows of equal planning time $T1$ & $T2$. Dividing is stopped if any of the sub-windows has a smaller number of jobs than the number of YCs in a row. In that case, each YC may have less than one job in average in a sub-window and this would lead to idle YCs in a sub-window.

$$Time_Partition(T) = \begin{cases} Space_Partition(T), & \text{if } |T1| < M \text{ or } |T2| < M \\ \min\{Space_Partition(T), Time_Partition(T1) + Time_Partition(T2)\} \end{cases}$$

Otherwise, the function of $Time_Partition(T)$ returns the minimal value of $Space_Partition(T)$ and the summarization of its two sub-windows. The resulting time partition will be a series of planning time windows of various time lengths which ends in smallest total job waiting time of all combinations.

```

Function Time_Partition ( T )
    Breaking T into 2 planning windows of equal time lengths T1, T2
    TotalJWT_T=Space_Partition (T);
    IF (Number of jobs in T1< Num. of YCs OR Number of jobs in T2 < Num. of YCs)
        Return TotalJWT_T
    ELSE
        TotalJWT_T1=Time_Partition (T1);
        TotalJWT_T2=Time_Partition (T2);
        IF (TotalJWT_T1 + TotalJWT_T2< TotalJWT_T)
            Return TotalJWT_T1 + TotalJWT_T2
        ELSE
            Return TotalJWT_T
END Function Time_Partition (T)

```

Figure 4. Pseudocode of the Time_Partition algorithm

From the view of the original planning window in between two rounds of YC distribution at Level 1, YC workload partition is dynamic since the space partition changes over time. In any of the planning time sub-windows, the space partition maintain the safety constraints of 8-slots separation to keep the advantages that even noise disturbance will not violate the partition plan. In cases of noise disturbance, some vehicles may arrive later than the predicted arrival times. Only the zones affected by these vehicles need to re-plan individual YCs sequencing, while other YCs could keep the pre-planned zones and service sequences. Smaller planning time sub-windows have the advantage of flexibility to match changing workload distributions, while larger planning time sub-windows have the advantage of prospective planning. These two time partition approaches could generate a balanced plan of various window lengths to reduce total job waiting time at the yard side which, in turn, contributes to improvements of overall terminal performance.

4 PERFORMANCE EVALUATION

4.1 Experimental Design

Experiments using parameter settings like terminal layout, YC gantry speed, vehicle arrival patterns and YC handling rate from real world terminal models were carried out. The linear gantry speed of an YC is 7.8km/hour. As reshuffling operation of containers in the yard storage is commonly done separately, we assume containers to be retrieved are already on the top of the slots and containers to be stored in yard will be placed on top of their slot locations. The YC mean process time is thus assumed to be 120s for each container job (loaded or empty), same as in Jung and Kim (2006). The usage of constant YC process time is also seen in Lee et al. (2006) and Lee et al. (2007).

Three algorithms are compared in the experiments: Time Partition Algorithm (TP) as described in Section 3.2, Space Only Partition Algorithm (SoP) as described in Section 3.1, and The Basic Algorithm.

The Basic algorithm selects the partition plan which allocates whole yard blocks to YCs while trying to minimize the difference in number of jobs among them. When the number of YCs is larger than the number of yard blocks in a row, yard blocks would be sorted in descending order by the number of jobs in each block. Extra YCs would be assigned to each block accordingly provided that there is a safety separation of 8 slots in the block. This scheme is quite commonly practiced in terminals and is employed as a benchmark scheme.

Table 1: Means of job inter arrival times under different scenario sets (seconds)

sec	Set A Row	Set B		Set C			Set D							
		T1 Row	T2 Row	Blk 1	Blk 2,3	Blk 4,5	T1				T2			
							Blk 1	Blk 2,3	Blk4	Blk5	Blk 1	Blk 2,3	Blk4	Blk 5
3YC	150	100	300	400	600	400	171	150	171	300	286	500	286	1000
5YC	90	60	180	240	360	240	239	187.5	239	375	471	825	471	1650
7YC	65	45	135	180	270	180	400	300	400	600	857	1500	857	3000

In the experiments, there are 5 yard blocks in a row. Each yard block has 3 yard slots. Inter-block distances between two neighboring yard blocks in a row are about 18 meters which are about 3 yard slots in length. Since YCs may need to be slowed down and may need to wait for instructions before crossing lanes, inter-block gantry time between two neighboring blocks is set as equivalent to linear gantry time of 10 yard slots. We tested scenarios where 3, 5 or 7 YCs are allocated to the row respectively for the planning period of one hour.

It is not the focus of this paper to study how to predict vehicle arrivals by real time data driven simulation or other methods. So we generate vehicle arrival times following an exponential distribution density function for a mixture of storing/retrieval jobs for multiple vessels. Workload of the entire row is adjusted by changing the mean inter-arrival time so that the numbers of job arrivals are proportional to the number of YCs in the row. It reflects the fact that Level 1 of YC operation management will assign a suitable number of YCs to a row in proportion to the expected workload. Data in Table 1 show the mean inter-arrival times to the zones of interest under different workload scenarios e.g. mean inter-arrival time for the entire row is 150s in Scenarios Set A with 3 YCs. The simulation model is programmed using C++ language under Visual C++ 6.0 compiler on Pentium Core2 Quad CPU Q9450 and 3GB RAM.

4.2 Results Analysis

For YC workload partition, the main performance indicator of interests is the average job waiting time (Avg_JWT). Avg_JWT resulted from workload partitioning for 7, 5 and 3 YCs over a row of 5 yard blocks for workload scenarios: Set A, Set B, Set C, and Set D are presented in Table 2. Results presented in all figures show the percentage performance improvement of partition algorithm: SoP and TP over the Basic algorithm.

Table 2: Percentage improvement of average job waiting time of Algorithm TP, SoP over Basic

		7 YCs	5 YCs	3 YCs
Set A	SoP	14.3%	25.7%	36.1%
	TP	51.2%	46.4%	52.6%
Set B	SoP	8.1%	18.7%	32.0%
	TP	48.2%	41.2%	46.7%
Set C	SoP	13.0%	36.8%	20.5%
	TP	44.6%	56.0%	41.3%
Set D	SoP	25.3%	26.6%	39.7%
	TP	49.4%	45.7%	52.3%

Across all scenarios tested in Table 1, both algorithms SoP and TP improve from the basic algorithm in terms of average job waiting time. Both algorithms SoP and TP consider more candidate partition points than the block-based approach in Algorithm Basic. It shows that simply balancing workload by the number of jobs among YCs as in Algorithm Basic will not work well. SoP's results alone show that considering more partition points than the block-based approach improves the performance. SoP is able to improve the average job waiting times over the Basic algorithm by 13% to 39.7% in various scenarios.

Algorithm TP performs best in all tested cases. Its improvement of the average job waiting time over Basic is 12.6% to 40.1% more than what SoP can achieve. TP's results show that time partition helps significantly to better match the changing workload as well as the relative constant workload. Scenario Set B represents different job arrival rates in different time periods within the planning window. Scenarios Set C represents the case of different job arrival rates for different areas of the row. Scenarios Set D represents the combination of Set C and Set B. Under these three workload scenarios, TP performs well as under scenario Set A.

There is no surprise that among the different experimental setups A to D, Set C has the biggest performance improvement in case of 5 YCs. In the case where 5 YCs are assigned to a row of 5 yard blocks, the Basic algorithm will assign each block one YC. It confirms that the proposed algorithms are well suited to generate partition plans when the workloads in different parts of the row are different. Set D has different workload distribution in different parts of the row and during different periods in the planning window, which is usually the case in real terminals. Algorithm TP produces consistently good results with more than 45% performance improvement over the Basic algorithm in all three situations in Set D.

Our experimental results show that our proposed workload partition scheme TP work well in various workload scenarios and is able to make substantial improvements over the usual practice Algorithm Basic with practical concerns of safety constraints.

5 CONCLUSIONS AND FUTURE WORK

This paper proposes a time partitioning algorithm which finds a sequence of time windows to match closely the changing workload distribution and get the maximum benefit of the space partition algorithm. A space partition algorithm is also proposed. Experimental results show that the proposed time partition algorithm works very well with the space partition algorithm in making substantial improvements over basic partition scheme. We will be conducting experiments to compare our algorithms with (Ng 2005).

REFERENCES

- R.K. Cheung, C-L. Li and W. Lin, "Interblock Crane Deployment at Container Terminals", *Transportation Science* 36(1):79-93. (2002).
- R. Dechter and J. Pearl, "Generalized Best-First Search Strategies and the Optimality of A*", *JACM* 32(3): 505-536. (1985).
- X. Guo and S.Y. Huang, "Performing A* Search for Yard Crane Dispatching in Container Terminals", In *Proceedings of the International Conference on Tools with Artificial Intelligence*, (1): 263-267. (2008).
- S.H. Jung and K.H. Kim, "Load Scheduling for Multiple Quay Cranes in Port Container Terminals", *Journal of Intelligent Manufacturing* 17, 479-492. (2006).
- K.H. Kim and K.Y. Kim, "An Optimal Routing Algorithm for a Transfer Crane in Port Container Terminals", *Transportation Science* 33 (1):17-33. (1999).
- K.Y. Kim and K.H. Kim, "Heuristic Algorithms for Routing Yard-Side Equipment for Minimizing Loading Times in Container Terminals", *Naval Research Logistics* 50, 498-514. (2003).
- D-H. Lee, Z. Cao, Q. Meng, "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. (2007).
- A. Narasimhan and U.S. Palekar, "Analysis and Algorithm for the Transtainer Routing Problem in Container Port Operation", *Transportation Science* 36 (1), 63-78. (2002).
- W.C. Ng, "Crane Scheduling in Container Yards with Intercrane Interference", *European Journal of Operational Research* 164, 64-78. (2005).
- M.E.H. Petering, Y. Wu, W. Li, M. Goh, R.d. Souza, "Development and Simulation Analysis of Real-Time Yard Crane Control Systems for Seaport Container Transshipment Terminals", *OR Spectrum* 31:801-835. (2009).
- S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. 2nd Edition. Pearson Education, Inc., Upper Saddle River, NJ. (2003).
- R. Stahlbock, S. Voß, "Operations Research at Container Terminals: A Literature Update", *OR Spectrum*, 30(1):1-52, (2007).
- R. Stahlbock, S. Voß, "Efficiency considerations for sequencing and scheduling of double-rail-mounted gantry cranes at maritime container terminals", *Int. J. Shipping and Transport Logistics*, Vol 2, No. 1, pp.95-123. (2010)
- D. Steenken, S. Voß, R. Stahlbock, "Container Terminal Operation and Operations Research – A Classification and Literature Review", *OR Spectrum* 26, 3-49 (2004).
- I.F.A. Vis, R. De Koster, "Transshipment of Containers at a Container Terminal: An Overview", *European Journal of Operational Research* 147(1), 1-16. (2003).
- C. Zhang, Y-W. Wan, J. Liu, R.J. Linn, "Dynamic Crane Deployment in Container Storage Yard". *Transportation Research Part B: Methodological* 36(6):537-555. (2002).