A Case Study of Genetic Algorithms for Quay Crane Scheduling

Yi Wang, Yun Chen, and Kesheng Wang

Abstract. In the operations of container terminals, a proper organized quay-crane-scheduling is critical to the operational efficiency. The aim of this paper is to develop a two-quay-crane schedule with non-interference constraints for the port container terminal of Narvik. First, a mathematical formulation of the problem is provided, and then a Genetic Algorithm (GA) approach is developed to obtain near optimal solutions. Finally, computational experiments on GA approach with different parameters are conducted.

1 Introduction

A container terminal is a facility where cargo containers are transshipped between different transport vehicles, for onward transportation. Containerization has enabled globalization [6] and allowed for economy of scale driving the vessels to be bigger and bigger. However, this has led to new challenges concerning terminals management and container handling operations for ports in order to keep up the pace with the supply and demand. The handling speed at the container terminal is a prerequisite for ships to achieve this economy of scale [6]. Hence the port competitiveness relies heavily on minimizing the *makespan* or transshipment time of a vessel at the terminal [5, 6]. *Makespan* means the latest completion time of all handling tasks concerning a given container vessel.

Kim and Park [4] discussed the quay crane scheduling problem with non-interference constraints in which only single container vessel was considered.

Yi Wang
Nottingham Business School, NTU, UK
e-mail: yi.wang@ntu.ac.uk

Yun Chen
School of Public Economics & Administration, SHUFE, China
e-mail: chenyun@mail.shufe.edu.cn

Kesheng Wang
Department of Production and Quality Engineering, NTNU, Norway
e-mail: kesheng.wang@ntnu.no

B.-C. Chien and T.-P. Hong (Eds.): Opportunities and Challenges, SCI 214, pp. 119–125.
springerlink.com

They established a mixed integer programming model for the problem and proposed a branch and bound method and a heuristic algorithm called "greedy randomized adaptive search procedure (GRASP)" for the solution of the quay crane scheduling problem. Park and Kim [7] also studied the problem using two stages approaches. Lee et al. [5] provided a mixed integer programming model for quay scheduling problem with non-interference constraints. Hybrid intelligent system was discussed by Sun [8] for solving quay crane scheduling problem.

The objective of this project is to focus on the quay crane scheduling with non-interface constraints for any one single container vessel in the port of Narvik . This work was stimulated from Kim and Park [4) and Lee el al. [5], which will be adapted and simplified for the practical case of Narvik contain terminal berth. A Genetic Algorithm was used to find near optimization solution for the problem.

2 Problem Description

Because both cranes are on the same rail, the Port-Authority is faced with the problem of scheduling them so that they avoid interference, respect the physical constraints, let at least one hold free between them at any given time, and yield the minimum total *makespan* for any given ship at the berth. The vessel's stability is not taken into consideration here. As shown in Figure 1, the container vessel is divided into up to ten holds and the goal of this paper is to provide a schedule that determines a handling sequence for these two cranes while avoiding interferences between them. A hold is assumed to be served by only one crane at any given time. The transition time from one hold to another is around 1 minute. Both the speed of the cranes and the vessel's capacity can be updated thus obtaining new solutions without influencing the robustness of the algorithm.

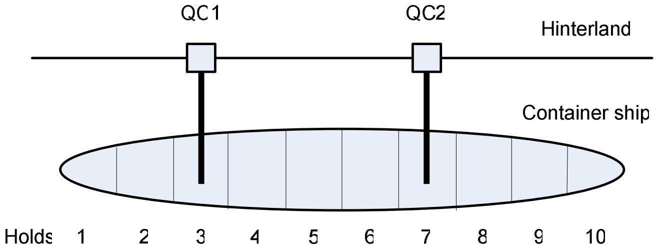


Fig. 1 A schema of the container vessel and the two quay cranes

3 Problem Formulation

For Narvik crane terminal berth, a model of this problem is described as a mixed integer programming and is given below, which is adapted from Kim and Park [4] and Lee et al [5] to the case of two cranes:

Parameters: H the number of holds (= 10); p_h the processing time of hold h by a quay crane ($1 \le h \le H$); M a sufficiently large positive and constant number.

Decision variables: $X_{h,1}$ 1, if hold h is handled by QC 1; 0, otherwise; $X_{h,2}$ 1, if hold h is handled by QC 2; 0, otherwise; $Y_{h,h'}$ 1, if hold h finishes no later than hold h'; C_h the completion time of hold h

Objective function:

(1.1) Minimize $[\underset{h}{Max}\, C_h]$ / minimize the *makespan* of handling one single container ship

Subject to:

(1.2) $C_h - p_h \ge 0$; $(\forall\ 1 \le h \le H)$/ property of C_h

(1.3) $X_{h,1} + X_{h,2} = 1$; $(\forall\ 1 \le h \le H)$ / every hold must be handled by only one QC

(1.4) $C_h - (C_{h'} - p_{h'}) + Y_{h,h'}M > 0$; $(\forall\ 1 \le h, h' \le H)$/ property of $Y_{h,h'}$

(1.5) $C_h - (C_{h'} - p_{h'}) - (1 - Y_{h,h'})M \le 0$; $(\forall\ 1 \le h, h' \le H)$/ property of $Y_{h,h'}$

(1.6) $M(Y_{h,h'} + Y_{h',h}) \ge X_{h,1} - (X_{h',1} + 2X_{h',2}) + 1$; $1 \le h < h' \le H$ / it must avoid interference between QCs

(1.7) $X_{h,1}, X_{h,2}, Y_{h,h'} = 0$ *or 1;* $(\forall\ 1 \le h, h' \le H)$ / binary decision variables

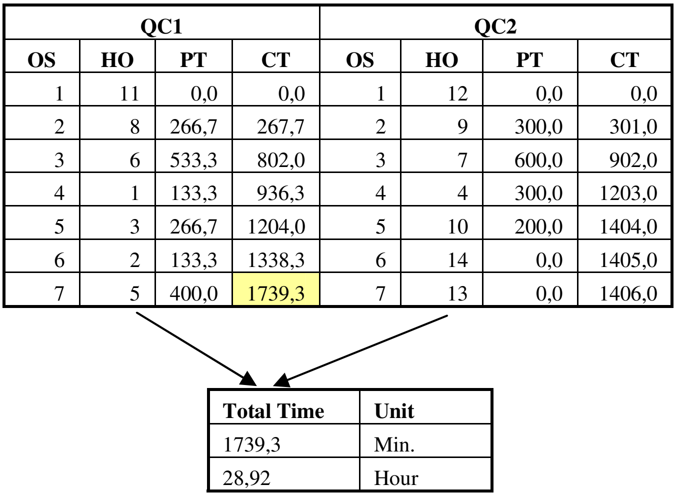
For the general version of this formulation, please refer to Lee et al [5] where a proof of NP-completeness is given. This means that there is no polynomial time algorithm for the exact solution. Therefore, a Genetic Algorithm is adopted to obtain near optimal solutions. The proposed Genetic Algorithm approach is effective, efficient and robust in solving the considered quay cranes scheduling problem.

4 Methodology

The concept of GAs was developed by Holland and his colleagues in the 1960s and 1970s [3]. A GA is a stochastic optimization method based on the mechanisms of natural selection and evolution. In GAs, searches are performed based on a population of chromosomes representing solutions to the problem. A population starts from random values and then evolves through succeeding generations. During each generation a new population is generated by propagating a good solution to replace a bad one and by crossing over or mutating existing solutions to construct new solutions. GAs have been theoretically and empirically proven robust for identifying solutions to combinatorial optimization problems. This is due to GAs ability to conduct an efficient parallel exploration of the search space, while only requiring minimum information on the function to be optimized. [9]

As a development tool, GeneHunter [2], is used as an add-in (solver) in Microsoft Excel. The population size is the size of the genetic breeding pool and is initialized to 50 individuals. The fitness function is the same as the objective function and the goal is to minimize total completion time (equation 1.1). The *generation gap* is set at 0.98 meaning that only 2 percent of individuals will go directly into the next generation without having to go through crossover and mutation. For the population of size 50, one individual will be sent directly to the next generation. As *elitism* is the chosen selection operator, then this one individual sent directly to the next generation is the fittest one. The *crossover* and *mutation rates* are set to 0.90 and 0.01 respectively.

Quay Crane Scheduling is a combinatorial problem where a sequence of holds is needed to be found so as it yields a near optimum solution. This sequence should be constituted of integer and unique values from 1 to 10. Thus, *enumerated chromosomes* with *unique genes* will be used. With enumerated chromosome, genes can have more allele values (integers) than just 0 and 1 while the unique genes property does not allow the chromosome to contain duplicate genes, or else the hold may be handled more than once which is absurd. The GA will strive to find the optimum order of these values so that the total completion time is minimized. The GA will go through the selection, crossover and mutation creating generations and seeking the optimum until a stopping criterion is met. Here, the GA will stop once the best fitness value remains unchanged for 50 generations. These GA parameter values are the one that give the best results from a practical point of view. Below are summarized the main feature of this GA: (1) Tool: GeneHunter v 2.4 embedded into Microsoft Excel 2003, (2) Genetic Algorithm (Fitness (objective) function: *(1.1);* Population size: 50; Chromosome type: Enumerated with unique genes; Crossover: crossover with probability: 0.90; Mutation rate: 0.01; Generation gap: 0.98; Selection strategy: Elitist; Stopping criterion: Best fitness value unchanged after 50 generations).



QC1				QC2			
OS	**HO**	**PT**	**CT**	**OS**	**HO**	**PT**	**CT**
1	11	0,0	0,0	1	12	0,0	0,0
2	8	266,7	267,7	2	9	300,0	301,0
3	6	533,3	802,0	3	7	600,0	902,0
4	1	133,3	936,3	4	4	300,0	1203,0
5	3	266,7	1204,0	5	10	200,0	1404,0
6	2	133,3	1338,3	6	14	0,0	1405,0
7	5	400,0	1739,3	7	13	0,0	1406,0

Total Time	Unit
1739,3	Min.
28,92	Hour

Fig. 2 Implementation of interference check, two-list chromosome, and dummy holds. (OS: Operation Sequence; HO: Hold Order; PT: Processing Time; CT: Completion Time)

A chromosome is modeled as a list of adjustable cells (variables) corresponding to genes with 1 as a minimum value and 10 as maximum mapping obviously the ten holds of the container vessel. The order of these genes is important as it represents the sequence in which the holds will be handled by the two quay cranes. For implementation purposes, having one list of ten adjustable cells proved to be a poor modeling technique because the resulting chromosome has to be assigned to the two QCs offline, i.e. after the optimization. This means that the interference between QCs is not resolved during optimization time. As a good alternative, first, the chromosome is split into two lists with 5 cells each and corresponding to the two QCs. This allows for dynamically checking the interference between the cranes for the considered potential solution. Second, the interference check is implemented in the business logic of MS Excel as tests that return 0 if no interference and 1 otherwise. Then, in GeneHunter, constraints are forcing these cells to always return 0, thus avoiding interference in the adopted solutions. A problem arising from this design choice is how to split these two lists, is it logical to have five for each list when on crane is much faster than the other and might handle more holds than five? To overcome this, the original chromosome is extended from 10 to 14, hence introducing four dummy holds, from 11 to 14, with zero volume. The choice of 4 dummy holds stems from comparing real world QC throughputs. These dummy holds let the optimization be much flexible in finding the near optimal solution and assigning holds to cranes. At the end, these dummy holds can be discarded from the solution. The total *makespan* will be adjusted accordingly (by subtracting four times the transfer time) as shown in Figure 2. The solution here will be holds 8, 6, 1, 3, 2 and 5 for QC1 while holds 9, 7, 4 and 10 will be assigned to QC2 with a total *makespan* of 28.92 hours

5 Computational Results

GAs with different paremeters are tried on different QC throughputs. It is clear that the strategy adopted in the previous section, is the most suitable one as it gives the best results. It could be argued that the population size could be decreased to 40 or perhaps 30, this will decrease the processing time, but a population of 50 is good enough. Another observation is that when the elitist strategy is not used, the search worsens and can not stabilize around a solution and thus the quality of the final solution is not as good as with other strategies.

In solutions given by GeneHunter, knowledge about the distribution of Holds' volume is not exploited. This distribution tends to be of the same nature for different container vessels and this could be used to the operator's advantage when using either GeneHunter or developing a manual heuristic. For example, considering the distribution given in the Excel sheet (200, 200, 400, 300, 600, 800, 600, 400, 300, 200), one could proceed as follows:

- take the maximum *(800 at the 6th position)* assign it to the fastest crane *(QC1)*,
- take the second greatest volume *(600 at the 7th position which is physically possible)* assign it to the other crane *(QC1),*

- with QC1, go left assigning as many as possible
- with QC2, go right assigning as many as possible
- cross (go beyond the 7th position for *QC1* and beyond the 6th position for *QC2*) when the steps 3 and 4 are not feasible (or not optimal) anymore.
- for the manual operator, make sure that hold 1 is handled by *QC1* and hold 10 by *QC2*

This heuristic was tried and proved to yield good results. For instance, for throughputs *(QC1, QC2) = (90, 60)* with Strategy C in GeneHunter, this heuristic gives the following result:

Makespan of *20 hr 02 min* obtained at generation *99* with a processing time of *18 sec*. *QC1* will handle holds *6, 1,(11), 3, 4, 2, and 8* while holds *7,10, (13), 9, 5, (14), and 12* will be assigned to *QC2.*

It should be noticed that the dummy holds introduce a problem of having an additional transfer time of one minute (4 min in total) in the original solution, this allows for some feasible solutions that are not possible otherwise. One has only to wait one or two minutes to reach better solutions. This can be regarded both as problem and as an additional flexibility in the design of the solution.

6 Conclusion

In this paper, a methodology based on GA is developed for solving the two quay-crane scheduling problem for the port of Narvik. This work is applied to the two quay cranes of Narvik Container Terminal. A GA has been developed through GeneHunter tool interfaced with MS Excel. Advantages of this methodology is the fact that near optimal solutions are obtained while the software is user-friendly as the user interacts with a known program (MS Excel) and the input can be changed easily (vessel's capacity; holds' capacity, cranes' throughput, and transfer time). GeneHunter can be then rerun and new solutions will be available. Furthermore, with some modeling changes, the number of quay cranes can be extended to three cranes but for a higher number of cranes, interference check will become difficult to carry on this way and developing one's own code (VBA or C++) will be more appropriate. As a result, this software could easily be used by operators in Narvik Container terminal. The limitations of this work, which constitute further research alleys, are neither the stability of the boat nor the handling priorities of the holds are taken into consideration here. Specific knowledge about the holds' volume distribution should be exploited to obtain minimal *makespan*..

References

1. Cheng, J.L., You, F.H., Yang, Y.: Research on quay crane scheduling problem by hybrid genetic algorithm. In: Proceedings of the IEEE International Conference on Automation and Logistics, ICAL 2008, pp. 2131–2135 (2008)
2. GeneHunter v 2.4 Getting Started Manual, `http://www.wardsystem.com`

3. Holland, J.H.: Adaptation in natural and artificial systems. The University of Michigan Press (1975)
4. Kim, K.H., Park, Y.M.: A crane scheduling methods for port container terminal. European Journal of Operational Research 156, 752–768 (2004)
5. Lee, D.H., Wang, H.Q., Miao, L.: Quay crane scheduling with non-interference constraints in port container terminals. Transportation Research, Part E 44, 124–135 (2008)
6. Meersmans, P.J.M., Dekker, R.: Operation research supports container handling, Econometric Institute Report EI, pp. 2001–2022 (2001)
7. Park, Y.M., Kim, K.H.: OR Spectrum 25, 1–23 (2003)
8. Sun, J.Q., Li, P., Han, M.: The crane scheduling problem and the hybrid intelligent optimization algorithm GASA. In: Proceedings of the 26th Chinese Control Conference, CCC 2007, pp. 92–96 (2007)
9. Wang, K.: Applied Computational Intelligence in Intelligent Manufacturing Systems, Advanced Knowledge International Pty Ltd., Australia (2005)