A Multi-objective Programming Method for Vehicle Dispatching in Container Terminal

1Yang Xiaoming, 2Zhao Ning, 3Mi Chao, 2Shu Fan and 3Liu Haiwei
1Logistics Research Center,
2Logistics Engineering College,
3Container Supply Chain Technology Engineering Research Center, Shanghai Maritime University, 201306, Shanghai, China

Abstract: It is a fundamental decision making process in container terminals to allocate container transporting works among vehicles. Several categories of methods such as mathematical programming, queuing theory, network models, or Markov decision making as well as heuristics are employed in the research. In this study a multi-objective programming method is proposed to resolve the problem in container terminal. The objective is the minimization of the total working time of the vehicles as well as the associated total cost of the travel. Then a genetic algorithm is developed to resolve the problem. Numerical tests are carried out and the results show the effectiveness and feasibility of the algorithm.

Key words: Multi-objective programming, container, quay side transport, genetic algorithm

INTRODUCTION

The vehicle dispatching is one of the most important decision making process in the container terminal. The efficiency of the travel to some extent determines the total working time of loading and unloading containers from the vessels. The decision-making problem for vehicle dispatching in container terminal has been widely recognized in the previous literature. Different algorithms were applied to enhance the operational efficiency of a container yard.

Zhao et al. (2002) presented a dynamic vehicle routing programming model, in which the routing between two objects is changeable and the blocking of travel is taken into account. The shortest path problem is resolved by a greedy algorithm. Mitrovic-Minic et al.(2000) focus on the simultaneously vehicles routing programming. The decision making process is divided into several sub-problems. The whole solution is combined with the sub-solutions. Bish (2003) proposes a heuristic method to solve the problem of dispatching vehicles to containers, determining storage location for each discharging container while scheduling loading and discharging operations on QCs. Kim and Eae (2004) propose an extended approach for AGV dispatching with a network-based MP model. The objective is the minimization of the total idle time of a quay crane resulting from late arrivals of AGVs and the total travel time of the AGV. It is assumed that storage locations of containers and schedules for (un)loading operations are given. The extension considers a pooled dispatching strategy taking multiple quay cranes and dual-cycle operations of AGVs into account. Lv and Zhang (2004) describe an integer programming model to solve the unloading sequence decision making problem. Two heuristic algorithms, look ahead and look backward search algorithms are presented to solve the problem. Bish et al. (2005) focus on the quayside process of discharging and uploading containers to and from a single vessel. They discuss the problem of dispatching single load vehicles to the containers for the minimization of the total time for discharging and uploading all containers. It is assumed that a fleet of vehicles is already assigned to the vessel. Several variations of a greedy algorithm are proposed that can be easily implemented in particular for large problem instances. Briskorn et al. (2006) focus on the assignment of transportation jobs to AGVs within a terminal control system operating in real time. A rather common problem formulation based on due times for the jobs is described and a greedy priority rule based heuristic with an exact algorithm was developed to solve this problem. Then an alternative formulation of the assignment problem which does not include due times is presented. This formulation is based on a rough analogy to inventory management and is solved using an exact algorithm. The idea behind this alternative formulation is to avoid estimates of driving times, completion times, due times and tardiness because such estimates are often highly unreliable in practice and do not allow for accurate planning. Zheng and Tao (2006) propose a combinatorial optimization method to minimize

Corresponding Author: Liu Haiwei, Logistics Engineering College, Shanghai Maritime University, 201306, Shanghai, China
the idle travel and the vessels serving time. And a simulation model is built to evaluate the method. The result is the average of the vessels’ serving time reduce 7%. Nguyen and Kim (2007) discuss the dispatching of ALVs. Information about locations and times of future delivery tasks is utilized in an MIP model aiming at an optimal assignment of delivery tasks to ALVs. The NP-hard problem is formulated as a scheduling problem with precedence and buffer constraints similar to the multiple traveling salesmen problem with precedence constraints and time windows. In Chen et al. (2007), the dispatching problem is formulated as a hybrid flow shop scheduling problem with precedence and blocking constraints. A Tabu-search based algorithm is developed to solve this problem. The results show that a good initial solution is important. Angeloudis and Bell (2010) proposed a dispatching model and algorithm structured on a cost/benefit concept, according to unreliable information from terminal operations. An approach as measurement to terminal uncertainties is presented as well and it is proved that the model and algorithm works better in lower uncertainty level. Lee et al. (2010) makes similar assumptions as Nguyen and Kim, aside from the one that handling times of cranes are taken into consideration. A mixed integer programming model is proposed to treat the vehicle dispatching for job sequences of multiple cranes and a heuristic to solve the problem. Unfortunately, it could be referred from experiments that, in case the number of cranes and works are large, the computation time will be very long. Moussani et al. (2011) focus on the Lifting AGV scheduling problem at container terminal of Normandy in Le Havre port. Lifting AGVs are vehicles which could lift a container itself while stacking and retrieving containers in storage yards. An integer programming model is proposed, based on assumptions including congestion-free traffic and fixed vehicle speed and travel distance. A generic algorithm which is compared with simulating annealing algorithm is developed to solve the problem. Le et al. (2012) formulated the vehicle dispatching problem as a mixed integer programming model. It is proposed an algorithm combining DCA and branch-and-Bound method to solve the model. Skinner et al. (2013) presents a GA-based optimization approach to solve the straddle carrier scheduling problem for container handling in the Patrick AutoStrad Terminal. The deterministic mathematical model is extended from a former one and the proposed approach has been fully implemented in terminal operation.

Nowadays, following the trend of jumbo ships, container terminals are changing gradually so that they could serve larger ships. CMA-COM has brought into uses several ships of 16000 TEUs and Maersk has received even larger container ships, with capacity up to 18000 TEUs. One of the major problems that a container terminal operator is confronted with is how to quickly make a decision for the vehicle dispatching for job sequences of multiple cranes. In summary, many research works have been carried out in the field. But few of them focus on the work to help the operator to quickly make a decision to minimize the total working time and total travel costs. In this study, a multi-objective programming method is proposed. The objective is to minimize the total working time and the related total travel costs has also been carefully taken into account. Then a genetic algorithm is designed to resolve the problem.

The rest of study is structured as follows. Section 2 briefly explains the problem addressed in this study. In section 3 and section 4 the multi-objective programming and the genetic algorithm is discussed which is related to the vehicle dispatching. Then the detailed computational results are given in Section 5. Conclusions are discussed in Section 6.

**PROBLEM DESCRIPTION**

While unloading and loading containers to or from a vessel, several vehicles will be dispatched to carry out the travel jobs. In traditional, the vehicles are bound to the quay cranes. Which means that a vehicle can only serves for one quay crane. When it finished an unloading job in the block it must travel back to the bounding quay crane for another unloading job. The idle travel always happens in this dispatching method. The bounding model of quay side transports shows in Fig. 1.

A dynamic dispatching model means that the vehicles can serve any quay crane. When a vehicle finished its unloading job in the blocks, it can choose a loading job near it. And it also can choose a unloading job in the quay side when it finished its loading job. A dynamic dispatching model will help to reduce the idle travels. The dynamic dispatching model of quay side transports shows in Fig. 2.

![Fig. 1: Bounding model of quay side transports](image-url)
Fig. 2: Dynamic dispatching model

Table 1: Distances between each quay cranes (m)

<table>
<thead>
<tr>
<th>QC No.</th>
<th>QC1</th>
<th>QC2</th>
<th>QC3</th>
<th>QC4</th>
<th>QC5</th>
<th>QC6</th>
</tr>
</thead>
<tbody>
<tr>
<td>QC1</td>
<td>0</td>
<td>137</td>
<td>274</td>
<td>411</td>
<td>548</td>
<td>685</td>
</tr>
<tr>
<td>QC2</td>
<td>137</td>
<td>137</td>
<td>0</td>
<td>274</td>
<td>411</td>
<td>548</td>
</tr>
<tr>
<td>QC3</td>
<td>274</td>
<td>137</td>
<td>0</td>
<td>137</td>
<td>274</td>
<td>411</td>
</tr>
<tr>
<td>QC4</td>
<td>411</td>
<td>274</td>
<td>137</td>
<td>0</td>
<td>137</td>
<td>274</td>
</tr>
<tr>
<td>QC5</td>
<td>548</td>
<td>411</td>
<td>137</td>
<td>137</td>
<td>0</td>
<td>137</td>
</tr>
<tr>
<td>QC6</td>
<td>685</td>
<td>548</td>
<td>411</td>
<td>274</td>
<td>137</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2: Distances between each blocks (m)

<table>
<thead>
<tr>
<th>B No.</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>137</td>
<td>137</td>
<td>411</td>
<td>685</td>
<td>959</td>
<td>1233</td>
<td>1507</td>
<td>1781</td>
</tr>
<tr>
<td>B</td>
<td>137</td>
<td>137</td>
<td>137</td>
<td>411</td>
<td>685</td>
<td>959</td>
<td>1233</td>
<td>1507</td>
</tr>
<tr>
<td>C</td>
<td>411</td>
<td>137</td>
<td>137</td>
<td>137</td>
<td>411</td>
<td>685</td>
<td>959</td>
<td>1233</td>
</tr>
<tr>
<td>D</td>
<td>685</td>
<td>411</td>
<td>137</td>
<td>137</td>
<td>137</td>
<td>411</td>
<td>685</td>
<td>959</td>
</tr>
<tr>
<td>E</td>
<td>959</td>
<td>685</td>
<td>411</td>
<td>137</td>
<td>137</td>
<td>137</td>
<td>411</td>
<td>685</td>
</tr>
<tr>
<td>F</td>
<td>1233</td>
<td>959</td>
<td>685</td>
<td>411</td>
<td>137</td>
<td>137</td>
<td>137</td>
<td>685</td>
</tr>
<tr>
<td>G</td>
<td>1507</td>
<td>1233</td>
<td>959</td>
<td>685</td>
<td>411</td>
<td>137</td>
<td>137</td>
<td>137</td>
</tr>
<tr>
<td>H</td>
<td>1781</td>
<td>1507</td>
<td>1233</td>
<td>959</td>
<td>685</td>
<td>411</td>
<td>137</td>
<td>137</td>
</tr>
</tbody>
</table>

Table 3: Distance between blocks and quay cranes (m)

<table>
<thead>
<tr>
<th>First day</th>
<th>QC1</th>
<th>QC2</th>
<th>QC3</th>
<th>QC4</th>
<th>QC5</th>
<th>QC6</th>
</tr>
</thead>
<tbody>
<tr>
<td>A01</td>
<td>110.0</td>
<td>493.3</td>
<td>876.6</td>
<td>1259.9</td>
<td>1643.2</td>
<td>2026.5</td>
</tr>
<tr>
<td>B01</td>
<td>382.5</td>
<td>219.3</td>
<td>602.6</td>
<td>965.9</td>
<td>1349.2</td>
<td>1752.5</td>
</tr>
<tr>
<td>C01</td>
<td>656.5</td>
<td>273.2</td>
<td>328.6</td>
<td>711.9</td>
<td>1095.2</td>
<td>1478.5</td>
</tr>
<tr>
<td>D01</td>
<td>930.5</td>
<td>547.2</td>
<td>163.9</td>
<td>437.9</td>
<td>821.2</td>
<td>1204.5</td>
</tr>
<tr>
<td>E01</td>
<td>1204.5</td>
<td>821.2</td>
<td>437.9</td>
<td>163.9</td>
<td>547.2</td>
<td>930.5</td>
</tr>
<tr>
<td>F01</td>
<td>1478.5</td>
<td>1095.2</td>
<td>711.9</td>
<td>328.6</td>
<td>273.2</td>
<td>656.5</td>
</tr>
</tbody>
</table>

In this study, the problem is to dispatch the loading and unloading jobs among the vehicles and determine the task sequence. There are 6 quay cranes and 8 blocks in the container terminal. The distances between each other are given as below Table 1-3.

When a vehicle finished its job in a block and start a new job in the same block, it need to driving from the head of the block to the end of the block, so the distance between the same block is the length of the block.

**SOLUTION METHOD**

**Assumptions:** The actual loading and unloading operation in container terminal is complicated. In order to simplify the model some assumption is made:

- The number of quay cranes and yard cranes and vehicles are given
- The processing time of quay crane and yard crane is constant
- The speed of vehicles is constant
- The vehicle can start its next task immediately
- The blocking of the traffic is not taken into account

**Notations:** The relative symbol definitions contain the set of quay cranes, the set of yard cranes, the starting time of each task and so no which are listed as follows:

- \( \{m_1, m_2, m_3, \ldots, m_p\} \) is the set of quay cranes, \( p \) is the total number of quay cranes
- \( \{n_1, n_2, n_3, \ldots, n_q\} \) is the set of quay cranes, \( q \) is the total number of quay cranes
- \( \{v_1, v_2, v_3, \ldots, v_k\} \) is the set of vehicles, \( k \) is the total number of vehicles
- \( \{c_1, c_2, c_3, \ldots, c_j\} \) is the set of loading and unloading task, \( h \) is the total number of tasks
- \( T_{\gamma_i} \) is the start time for \( v_j \) to start \( c_i \)
- \( r_{\gamma_i} \) is the end time for \( v_j \) to finish \( c_i \)
- \( \max(T_{\gamma_i}) \) is the total working time
- \( T_{\gamma_i}^{\text{V}} \) is the idle travel time during \( v_j \) serving for \( c_i \)
- \( T_{\gamma_i}^{\text{L}} \) is the loaded travel time during \( v_j \) serving for \( c_i \)
- \( T_{\gamma_i}^{\text{R}} \) is the total idle travel time of \( v_j \)
- \( T_{\gamma_i}^{\text{L}} \) is the total loaded travel time of \( v_j \)
- \( \max(r_{\gamma_i}, s_{\gamma_i}) \) is the total task number of \( v_j \)
- \( a_{\gamma_i} \) is permitted total working time
- \( a_{\gamma_i} \) is the decision variable, when \( c_i \) is dispatched to \( v_j \), \( a_{\gamma_i} \) is 1, otherwise 0

**Objective functions:** In order to minimize the total working time of all vehicles, assume \( T_{\gamma_i} \) is the total working time of \( v_j \):

\[
T_{\gamma_i} = T_{\gamma_i}^{\text{V}} + T_{\gamma_i}^{\text{L}} + T_{\gamma_i}^{\text{R}}
\]

\[
T_{\gamma_i}^{\text{V}} = \sum_{c_i} (r_{\gamma_i} - s_{\gamma_i})
\]

\[
T_{\gamma_i}^{\text{L}} = \sum_{c_i} T_{\gamma_i}^{\text{L}}
\]

The first objective function is to minimize the total working time:

\[
\min(\max(T_{\gamma_j}), j=1, 2, 3, \ldots k)
\]
In order to minimize the total travel costs all vehicles, assume \( p_{1} \), \( p_{2} \) to be the unit cost of loaded travel time and idle travel time. The second objective function is to minimize the total travel costs:

\[
\text{Min}(y) = p_{1} \sum_{i} \sum_{j} (t_{\text{end}_{ij}} - t_{\text{start}_{ij}}) + p_{2} \sum_{i} \sum_{j} t_{\text{idle}_{ij}}
\]  

(5)

Which can be simplified to:

\[
\text{Min}(y) = p_{1} \sum_{i} \sum_{j} t_{\text{end}_{ij}} + p_{2} \sum_{i} \sum_{j} t_{\text{idle}_{ij}}
\]  

(6)

**Constraints:** The constraints are listed below:

\[
t_{\text{end}_{ij}} = t_{\text{start}_{ij}}
\]  

(7)

Equation 7 indicates the start time of each task must before the end time:

\[
t_{\text{idle}_{ij}} = t_{\text{end}_{ij}} - t_{\text{start}_{ij}}
\]  

(8)

Equation 8 indicates that the idle travel is the first task end time and the second task start time:

\[
t_{\text{start}_{ij}} \geq t_{\text{end}_{ij}}
\]  

(9)

Equation 9 is for that the first task end time must after the second task start time:

\[
\forall j, \sum_{i} a_{ij} = 1
\]  

(10)

Equation 10 is for that a loading or unloading task must be dispatched to one vehicle:

\[
\max(t_{\text{end}_{ij}}) \leq t_{\text{on}}
\]  

(11)

Equation 11 indicates that the final task end time must before the permit finish time.

**PROPOSED SCHEME**

In order to resolve the multi-objective programming, here a genetic algorithm is designed. Genetic algorithm is a common heuristic algorithm which is effective and feasible to get a good solution among the solution space.

**Notations:** The notations used in the algorithm are listed as follows:

- \( \{v_{1}, v_{2}, \ldots, v_{i}, \ldots, v_{n}\} \) is the set of the vehicle.
- \( v_{i} \) means that task \( e_{i} \) is dispatched to \( v_{i} \).
- \( N = 50 \) is the total number of the chromosomes in each generation.
- GenNUM = 150 is the generation number.
- \( \{f_{1}, f_{2}, f_{3}, \ldots, f_{n}\} \) is the set of the fitness.
- \( P_{c} = 1.1 \) is the genetic operators.
- \( P_{m} = 0.7 \) is the crossover operators.
- \( P_{m} = 0.05 \) is the mutation operators.

**Chromosome representation:** In the first population, 50 chromosomes are created. One of the chromosome is:

\[
\{56813823641453681225121111022122145691101114149471424215933311884514721112151221261513771411271378812731531569428711147153151051592111444\}
\]

In the set, the first number 5 means the first task is dispatched to vehicle 5 and the third number 8 means the third task is dispatched to vehicle 8 and so on.

**Parent selection:** The fitness function is:

\[
c - \text{Min}(y) = c - p_{1} \sum_{i} \sum_{j} (t_{\text{end}_{ij}} - t_{\text{start}_{ij}}) - p_{2} \sum_{i} \sum_{j} t_{\text{idle}_{ij}}
\]  

(12)

In which \( C = 1200 \). The parent select function is:

\[
f_{j} / \sum_{j} f_{j} \geq P_{c} / n
\]  

(13)

\[
\frac{t_{\text{on}}}{\sum_{i} t_{\text{on}_{i}}} < \frac{1}{N}
\]  

(14)

The parent select function indicates that if Eq 13 and 14 is true the solution will be chosen to be the parent chromosomes. In order to ensure that the best individual always survive to the next generation, the parent chromosomes will be selected to the next generation.

**Crossover:** In the crossover section, the parent chromosomes will be chosen to cross with the other chromosomes. After the crossing operations, the new chromosomes will be selected to the next generation. The crossing operation is to exchange some genes in the chromosomes with the same genes in the parent chromosomes randomly. The exchanged genes will be selected in the probability \( P_{c} \), which is the crossover operator.
**Table 5: Task list**

<table>
<thead>
<tr>
<th>Task No.</th>
<th>Start points</th>
<th>End points</th>
<th>Task type</th>
<th>Travel distance (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>QC01</td>
<td>A11</td>
<td>unload</td>
<td>414.2</td>
</tr>
<tr>
<td>2</td>
<td>QC01</td>
<td>B08</td>
<td>unload</td>
<td>551.7</td>
</tr>
<tr>
<td>3</td>
<td>QC01</td>
<td>E11</td>
<td>unload</td>
<td>1688.7</td>
</tr>
<tr>
<td>4</td>
<td>QC01</td>
<td>F13</td>
<td>unload</td>
<td>1845.1</td>
</tr>
<tr>
<td>5</td>
<td>QC01</td>
<td>G10</td>
<td>unload</td>
<td>1508.7</td>
</tr>
<tr>
<td>6</td>
<td>QC01</td>
<td>D07</td>
<td>unload</td>
<td>2015.1</td>
</tr>
<tr>
<td>7</td>
<td>QC01</td>
<td>A12</td>
<td>unload</td>
<td>1102.7</td>
</tr>
<tr>
<td>8</td>
<td>QC01</td>
<td>B01</td>
<td>unload</td>
<td>437.2</td>
</tr>
<tr>
<td>9</td>
<td>QC01</td>
<td>D02</td>
<td>unload</td>
<td>382.5</td>
</tr>
<tr>
<td>10</td>
<td>QC01</td>
<td>H13</td>
<td>unload</td>
<td>593.5</td>
</tr>
<tr>
<td>11</td>
<td>QC01</td>
<td>A11</td>
<td>unload</td>
<td>2388.1</td>
</tr>
<tr>
<td>12</td>
<td>QC01</td>
<td>D02</td>
<td>unload</td>
<td>583.0</td>
</tr>
<tr>
<td>13</td>
<td>QC01</td>
<td>A12</td>
<td>unload</td>
<td>744.9</td>
</tr>
<tr>
<td>14</td>
<td>A06</td>
<td>QC01</td>
<td>load</td>
<td>1407.7</td>
</tr>
<tr>
<td>15</td>
<td>B03</td>
<td>QC01</td>
<td>load</td>
<td>1643.3</td>
</tr>
<tr>
<td>16</td>
<td>D09</td>
<td>QC01</td>
<td>load</td>
<td>667.5</td>
</tr>
<tr>
<td>17</td>
<td>F18</td>
<td>QC01</td>
<td>load</td>
<td>673.3</td>
</tr>
<tr>
<td>18</td>
<td>G03</td>
<td>QC02</td>
<td>load</td>
<td>276.7</td>
</tr>
<tr>
<td>19</td>
<td>A11</td>
<td>QC02</td>
<td>load</td>
<td>1947.4</td>
</tr>
<tr>
<td>20</td>
<td>C03</td>
<td>QC02</td>
<td>load</td>
<td>1152.6</td>
</tr>
<tr>
<td>21</td>
<td>E04</td>
<td>QC03</td>
<td>load</td>
<td>1029.0</td>
</tr>
<tr>
<td>22</td>
<td>A11</td>
<td>QC03</td>
<td>load</td>
<td>2339.7</td>
</tr>
<tr>
<td>23</td>
<td>F18</td>
<td>QC03</td>
<td>load</td>
<td>1165.9</td>
</tr>
<tr>
<td>24</td>
<td>E11</td>
<td>#2</td>
<td>load</td>
<td>1234.7</td>
</tr>
</tbody>
</table>

- The loading and unloading information is studied by using the actual data. And moreover, the experimental tests are carried out to evaluate the effectiveness and reliability of the algorithm. The input data are listed as follow:

  - The distance between each working points has been given in the problem description section.
  - Fifteen vehicles are given and the speed is assumed to be 36 km h⁻¹, the initial locations are the start points of their first task.
  - The processing time of quay cranes is assumed to be 2 and 2.5 min move⁻¹ for yard cranes.
  - The unit cost of loaded travel is 240 and 180 h⁻¹ for the idle travel.
  - There are 100 tasks comprise 30 loading tasks and 70 unloading tasks. The task list is show in Table 5.

In the section 10 experimental tests are carried out. One of the results is show following:

  - The optimal solution is got in the last generation which has a good fitness, low travel cost and less total working time.
  - The fitness of the solution is 267.14, the total cost of travel is 932 and the total working time is 21.35 min.
  - The average fitness of the 150 generations is show in Fig. 4. And the lowest cost of the 150 generations is show in Fig. 5.
The optimal solution is shown in Table 6. In this study, 10 numerical experiments are performed. The results are shown in Table 7.

ACKNOWLEDGMENTS

This study is sponsored by National Natural Science Foundation (71201059), Shanghai Municipal Education Commission Project (14ZZ140), Local University Capacity Promotion Special Programs (13510501800) of Science and Technology Commission of Shanghai Municipality and Shanghai Maritime University Foundation Project (20120104). We also thank anonymous referees and the editor-in-chief for their ideas and suggestions.

REFERENCES


