

A novel memetic algorithm with a deterministic parameter control for efficient berth scheduling at marine container terminals

Maxim A. Dulebenets

*Department of Civil and Environmental Engineering,
Florida A&M University-Florida State University, Tallahassee, Florida, USA*

Abstract

Purpose – The volumes of international containerized trade substantially increased over the past years. In the meantime, marine container terminal (MCT) operators are facing congestion issues at their terminals because of the increasing number of large-size vessels, the lack of innovative technologies and advanced handling equipment and the inability of proper scheduling of the available resources. This study aims to propose a novel memetic algorithm with a deterministic parameter control to facilitate the berth scheduling at MCTs and minimize the total vessel service cost.

Design/methodology/approach – A local search heuristic, which is based on the first-come-first-served policy, is applied at the chromosomes and population initialization stage within the developed memetic algorithm (MA). The deterministic parameter control strategy is implemented for a custom mutation operator, which alters the mutation rate values based on the piecewise function throughout the evolution of the algorithm. Performance of the proposed MA is compared with that of the alternative solution algorithms widely used in the berth scheduling literature, including a MA that does not apply the deterministic parameter control strategy, typical evolutionary algorithm, simulated annealing and variable neighborhood search.

Findings – Results demonstrate that the developed MA with a deterministic parameter control can obtain superior berth schedules in terms of the total vessel service cost within a reasonable computational time. Furthermore, greater cost savings are observed for the cases with high demand and low berthing capacity at the terminal. A comprehensive analysis of the convergence patterns indicates that introduction of the custom mutation operator with a deterministic control for the mutation rate value would provide more efficient exploration and exploitation of the search space.

Research limitations/implications – This study does not account for uncertainty in vessel arrivals. Furthermore, potential changes in the vessel handling times owing to terminal disruptions are not captured.

Practical implications – The developed solution algorithm can serve as an efficient planning tool for MCT operators and assist with efficient berth scheduling for both discrete and continuous berthing layout cases.

Originality/value – The majority of studies on berth scheduling rely on the stochastic search algorithms without considering the specific problem properties and applying the guided search heuristics. Unlike canonical evolutionary algorithms, the developed algorithm uses a local search heuristic for the chromosomes and population initialization and adjusts the mutation rate values based on a deterministic parameter control strategy for more efficient exploration and exploitation of the search space.

Keywords Marine container terminals, Berth scheduling, Deterministic parameter control, Memetic algorithms, Vessel service cost savings

Paper type Research paper



1. Introduction

According to the recent statistical data, provided by the United Nations Conference on Trade and Development (UNCTAD), volumes of the international maritime trade have been rapidly growing since 2009 (UNCTAD, 2015). A total of 9.8 billion tons of cargo were carried by vessels in 2014, which is 20.2 per cent more as compared to the total volume of cargo transported in 2009. In five years, the container volumes increased by 30.9 per cent and the dry cargo volumes increased by 11.8 per cent, while major bulk and liquid bulk cargoes increased by 33.0 and 6.5 per cent, respectively (UNCTAD, 2015). Such a significant increase in the international containerized trade volumes requires marine container terminal (MCT) operators to improve the terminal productivity and efficiently serve the growing demand (Bentolila *et al.*, 2016; Rahman *et al.*, 2016; Lu *et al.*, 2016).

However, the opposite tendencies have been recently observed at major MCTs. The largest European ports (i.e. Rotterdam, Hamburg, Antwerp) reported the inability of the port infrastructure to efficiently serve the increasing amount of megaships (Port Finance, 2014). Long congestion periods were observed at the Port of Los Angeles (USA) owing to labor strikes, which resulted in a decrease in the West Coast-laden traffic by 12 per cent in the first three months of 2015 (Journal of Commerce, 2015). The expected vessel waiting time of up to three days was reported in May 2016 at certain ports of Malaysia: Kuantan, Penang, Pasir Gudang, etc. (Ben Line Agencies, 2016). The American Shipper (2015) and World Shipping Council (2015) outline the following key factors causing the port congestion:

- liner shipping alliances;
- increasing size of vessels;
- lack of innovative technologies; and
- chassis availability.

The MCT operators have to develop new strategies and innovative approaches to enhance efficiency of their operations and alleviate negative externalities from the congestion issues (World Shipping Council, 2015). This paper focuses on the berth scheduling problem (BSP), which deals with the assignment of vessels to the available berths at the given MCT, and defining the sequence of vessels served at each berth. An efficient berth schedule is crucial for the MCT operations, as it directly affects the vessel turnaround time. The latter is considered as an important MCT performance indicator (Bierwirth and Meisel, 2015). A novel memetic algorithm (MA), which uses a local search heuristic for the chromosomes and population initialization and adjusts the mutation rate values based on a deterministic parameter control strategy, is proposed to assist the MCT operators with construction of efficient berth schedules. The rest of the paper is organized as follows. Section 2 presents a review of the BSP literature with focus on the developed solution algorithms, while Section 3 describes the problem studied in this paper. Section 4 presents a mixed-integer nonlinear mathematical model for the problem, while Section 5 provides a detailed description of the proposed solution algorithm. Section 6 discusses the computational experiments, conducted in this study, to evaluate the efficiency of the developed algorithm, while Section 7 summarizes the findings and outlines the future research extensions.

2. Literature review

The BSP can be reduced to the unrelated machine scheduling problem, where the arriving jobs should be assigned for processing on the available machines, which have different properties. The latter operations research problem is known to have *NP*-hard complexity (Pinedo, 2008; Bierwirth and Meisel, 2015). The *NP*-hard problems are unlikely to be solved

using the exact optimization algorithms in an acceptable computational time for the realistic-size problem instances. Many of the published to-date BSP studies applied the stochastic search algorithms (e.g. evolutionary algorithms – EAs, simulated annealing – SA, Tabu search – TS, particle swarm optimization – PSO, etc.), which do not guarantee the global optimality of the berth schedules, but they are able to obtain the “good-quality” berth schedules within a reasonable computational time. The literature review presented herein mainly focuses on the solution algorithms that were designed to solve the BSP. For a more detailed description of the BSP papers, this study refers to [Bierwirth and Meisel \(2010, 2015\)](#) and [Carlo *et al.* \(2013\)](#). An overview of the collected BSP papers is presented next.

[Nishimura *et al.* \(2001\)](#) proposed a nonlinear mathematical formulation for the dynamic BSP in the public berth system. The objective aimed to minimize the total vessel service time. An EA-based heuristic was developed to solve the problem. Computational experiments demonstrated that the presented solution approach provided solutions, close to the ones that were obtained by the Lagrangian relaxation-based algorithm for the large-size problem instances. [Kim and Moon \(2003\)](#) and [Dai *et al.* \(2008\)](#) developed SA-based algorithms to solve BSPs, aiming to minimize the total vessel service cost. [Cordeau *et al.* \(2005\)](#) presented a BSP formulation, minimizing the total weighted vessel service time. A TS algorithm was designed to solve the problem. [Zhou *et al.* \(2006\)](#) developed an EA for the BSP with variable vessel service priorities, minimizing the total waiting time of vessels. [Imai *et al.* \(2007a\)](#) formulated the BSP at the MCT with indented berths for faster service of megaships, minimizing the total vessel service time. An EA was developed to solve the problem. [Wang and Lim \(2007\)](#) presented a stochastic beam search (SBS) heuristic to solve the BSP. The objective minimized the total vessel service cost, associated with possible unallocation, and penalties owing to deviations from the desired berthing positions and late vessel departures. [Hansen *et al.* \(2008\)](#) and [Lee and Chen \(2009\)](#) used a variable neighborhood search (VNS) heuristic to solve BSPs, minimizing the total vessel service cost and maximizing the berth utility index, respectively.

[Imai *et al.* \(2008\)](#) formulated the BSP at the MCT, where vessels with excessive waiting times were diverted for service at an external terminal. The objective aimed to minimize the total vessel service time at the external terminal. The problem was solved using an EA. [Golias *et al.* \(2010a\)](#) developed an EA to solve BSP, aiming to minimize the total vessel service time, delayed departures, fuel consumption and emissions produced. [Golias *et al.* \(2010b\)](#) presented a lambda-optimal heuristic to solve the BSP, minimizing the total weighted service time of vessels. [Lee *et al.* \(2010\)](#) developed two versions of a greedy randomized adaptive search procedure to solve the BSP, aiming to minimize the total weighted turnaround time of vessels. [Lalla-Ruiz *et al.* \(2012\)](#) formulated the BSP, directed to minimize the total service time of vessels. The authors presented a heuristic based on TS and path relinking. [Imai *et al.* \(2013\)](#) evaluated various MCT layouts, including conventional, channel and indented layouts. The objective aimed to minimize the total vessel service time, and an EA was applied to solve the problem.

[Ting *et al.* \(2014\)](#) proposed a PSO algorithm for the BSP, minimizing the total turnaround time of vessels. [Frojan *et al.* \(2015\)](#) formulated a BSP at the MCT with multiple quays, minimizing the total vessel service cost. An EA with a local search heuristic was developed to solve the problem. [Hu \(2015a\)](#) studied a BSP, taking into consideration a periodic balancing utilization of quay cranes and aiming to minimize the total vessel service cost. A set of heuristics was developed to solve the problem. [Emde and Boysen \(2016\)](#) focused on the BSP, minimizing the total waiting time of vessels and the number of delayed containers. The authors presented an SA-based heuristic to solve the problem. [Mauri *et al.* \(2016\)](#) proposed an adaptive large neighborhood search heuristic for the BSP. The objective minimized the

total vessel service time. The heuristic changed the solutions using an adaptive probabilistic mechanism for destroy and repair operators. [Dulebenets \(2017a\)](#) proposed an adaptive EA for the BSP, where the mutation rate was altered based on feedback from the search. The objective of the proposed model aimed to minimize the total weighted vessel service cost. [Dulebenets et al. \(2017\)](#) developed a set of Hybrid EAs for the green BSP, where the carbon dioxide emission cost because of container handling was included in the objective function of the model.

Several papers captured uncertainty in vessel arrival and handling times. [Xu et al. \(2011\)](#) formulated the BSP with uncertain vessel arrival and handling times. The objective minimized the total late departures and maximized the length of buffer time. The problem was solved using the algorithm, which was based on SA and Branch-and-Bound Algorithm. [Zhen and Chang \(2012\)](#) formulated a bi-objective BSP, taking into account uncertainties of vessel arrivals and handling times. The first objective minimized the total operational cost, while the second one was directed to maximize the robustness of schedule. The authors presented a heuristic to solve the problem. [Golias et al. \(2014\)](#) proposed an EA for the BSP, capturing uncertainty in vessel arrival and handling times. The model was formulated as a bi-level problem, where the upper level minimized the average total service time of vessels, while the lower level minimized the total range of service times. [Legato et al. \(2014\)](#) studied an integrated tactical and operational BSP, considering uncertainty in vessel arrival and handling times and minimizing the total penalty cost because of deviation from the desired berthing position. Solutions at the tactical level were obtained using SBS, while SA was applied at the operational level.

A number of studies focused on multi-objective BSPs. [Moorthy and Teo \(2006\)](#) formulated a bi-objective BSP, minimizing the total vessel service delays and the connectivity cost. An SA-based algorithm was designed to solve the problem. [Imai et al. \(2007b\)](#) focused on a bi-objective BSP, aiming to minimize:

- the total vessel late departures; and
- the total vessel service time.

An EA was used to solve the problem. [Cheong et al. \(2008\)](#) developed an EA for a multi-objective BSP, aiming to minimize the makespan, waiting time of vessels and degree of deviation from a predetermined priority schedule. [Cheong and Tan \(2008\)](#) studied a similar problem, minimizing the total vessel service time and total vessel delayed departures. The authors developed the multi-objective multi-colony ant algorithm to solve the problem. [Golias et al. \(2009\)](#) presented a multi-objective BSP formulation, considering various vessel priority agreements and minimizing the total vessel service time for each vessel priority group. [Hu \(2015b\)](#) formulated a bi-objective BSP, minimizing:

- the total delayed workload; and
- the total night workload.

An EA was developed to solve the problem. [Han et al. \(2015\)](#) proposed a two-phase model for the BSP and the quay crane assignment problem. The objective of the first phase minimized the total vessel waiting time and the total additional time due to deviation from the desired berthing position. The second phase had two objectives:

- (1) minimize the range of maximum and minimum quay cranes; and
- (2) minimize the movements of quay cranes.

A PSO algorithm was designed to solve the problem.

The review of literature suggests that BSP receives an increasing attention from the community. However, the majority of studies rely on the stochastic search algorithms without considering any specific problem properties and applying any guided search heuristics (Bierwirth and Meisel, 2015). The contributions of this study to the state-of-the-art include the following:

- This study proposes a novel MA, which applies a local search heuristic for the chromosomes and population initialization and a deterministic parameter control strategy for more efficient exploration and exploitation of the search space.
- Performance of the developed MA is assessed against the exact optimization approaches for the small-size problem instances.
- The proposed solution algorithm is evaluated against the other state-of-the-art metaheuristic algorithms (including a typical EA, VNS and SA) for the realistic-size problem instances.
- A comparative analysis of the solution algorithms is performed not only for a discrete berthing layout case (which is commonly used in the BSP literature) but also for a continuous berthing layout case.

The developed solution algorithm is expected to assist the MCT operators with design of cost-effective berth schedules and reduce potential delays, associated with the vessel service.

3. Problem description

This study models the MCT with a discrete berthing layout (Figure 1). The terminal's wharf is partitioned in several berths, and one vessel can be moored at each berth at the time. The MCT operator is assumed to have the information regarding the vessel arrival times (i.e. dynamic vessel arrival case). The uncertainty in vessel arrival times is not modeled in this study. Once a vessel arrives at the MCT, it is towed to the assigned berth by push boats. If the assigned berth is not available (i.e. it is occupied by the other vessel), a vessel will be waiting for service in the dedicated area, located close to the MCT (Figure 1). It is assumed that the safety distance requirements between vessels, moored along the MCT's wharf, are satisfied. The vessels are served using a gang of quay cranes. The container handling productivity (measured in twenty-foot equivalent units [TEUs] per hour) is negotiated with a liner shipping company and is known before the vessel arrives at the MCT.

The container handling charges are imposed to the liner shipping company for each TEU handled. The handling time of a given vessel (measured in hours) is estimated based on the total number of TEUs handled and the handling productivity requested. The uncertainty in vessel handling time due to potential disruptions in the MCT operations is not considered. However, if a given vessel is diverted for the service from the originally scheduled berth (i.e. "preferred berth") to the other berth (to decrease the waiting time of that vessel), its handling time will increase proportionally to the distance between the actual and desired berthing positions (Bierwirth and Meisel, 2015). The increase in vessel handling time can be explained by the fact that the travel distance from the "preferred berth" to the assigned storage area in the marshaling yard is smaller compared to the travel distance from the other MCT berths.

An additional charge will be imposed to the MCT operator due to vessel waiting for service in the dedicated area. Furthermore, for each vessel, the liner shipping company requests a certain departure time, and violations owing to the difference between the actual and requested departure times will be penalized. If a vessel leaves the MCT before the requested departure time, the liner shipping company will pay the premium to the MCT

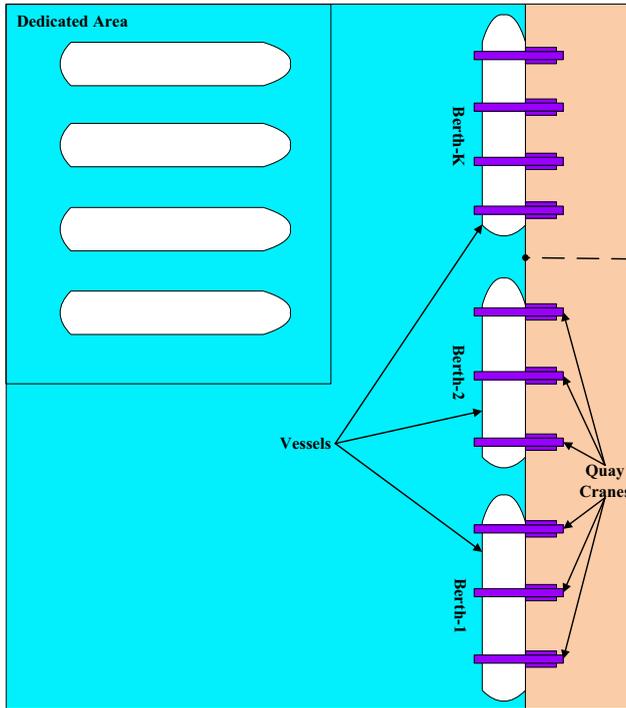


Figure 1.
Service of vessels at the MCT

operator (Dulebenets, 2015a). The objective of the MCT operator is to design an efficient berth schedule by minimizing the total vessel service cost.

4. Mathematical model

This section presents notations that will be used throughout the paper and a mixed-integer nonlinear mathematical model for the discrete dynamic BSP (DDBSP) studied herein.

Sets

$V = \{1, \dots, N\}$ = set of vessels to be served at the MCT; and

$B = \{1, \dots, K\}$ = set of berths at the MCT.

Decision variables

$x_{vb}, v \in V, b \in B$ = 1 if vessel v is served at berth b (= 0 otherwise);

$y_{\underline{v}\bar{v}}, \underline{v}, \bar{v} \in V, \underline{v} \neq \bar{v}$ = 1 if vessel \bar{v} is served at the same berth immediately after vessel \underline{v} (= 0 otherwise);

$f_v, v \in V$ = 1 if vessel v is served as the first vessel at the assigned berth (= 0 otherwise); and

$l_v, v \in V$ = 1 if vessel v is served as the last vessel at the assigned berth (= 0 otherwise).

Auxiliary variables

$ST_v, v \in V$ = start time of service for vessel v (hours);

$WT_v, v \in V$ = waiting time of vessel v (hours);

$ED_v, v \in V$ = hours of early departure for vessel v (hours); and
 $LD_v, v \in V$ = hours of late departure for vessel v (hours);

Parameters

$A_v, v \in V$ = arrival time of vessel v (hours);
 $NC_v, v \in V$ = number of containers assigned to vessel v (TEUs);
 $hp_{vb}, v \in V, b \in B$ = handling productivity for vessel v at berth b (TEUs per hour);
 $HT_{vb} = \frac{NC_v}{hp_{vb}}, v \in V, b \in B$ = handling time of vessel v at berth b (hours);
 $RD_v, v \in V$ = requested departure time of vessel v (hours);
 $c_v^H, v \in V$ = handling cost for vessel v (US\$ per TEU);
 $c_v^{WT}, v \in V$ = waiting cost for vessel v (US\$ per hour);
 $c_v^{ED}, v \in V$ = premium for early departure of vessel v (US\$ per hour);
 $c_v^{LD}, v \in V$ = penalty for late departure of vessel v (US\$ per hour); and
 M = large positive number.

DDBSP:

$$\min \left[\sum_{v \in V} (NC_v c_v^H) + \sum_{v \in V} (WT_v c_v^{WT}) + \sum_{v \in V} (LD_v c_v^{LD}) - \sum_{v \in V} (ED_v c_v^{ED}) \right] \quad (1)$$

Subject to:

$$\sum_{b \in B} x_{vb} = 1 \quad \forall v \in V \quad (2)$$

$$f_{\bar{v}} + \sum_{\underline{v} \in V: \underline{v} \neq \bar{v}} y_{\underline{v}\bar{v}} = 1 \quad \forall \bar{v} \in V \quad (3)$$

$$l_{\underline{v}} + \sum_{\bar{v} \in V: \bar{v} \neq \underline{v}} y_{\underline{v}\bar{v}} = 1 \quad \forall \underline{v} \in V \quad (4)$$

$$f_{\underline{v}} + f_{\bar{v}} \leq 3 - x_{\underline{v}b} - x_{\bar{v}b} \quad \forall \underline{v}, \bar{v} \in V, \underline{v} \neq \bar{v}, b \in B \quad (5)$$

$$l_{\underline{v}} + l_{\bar{v}} \leq 3 - x_{\underline{v}b} - x_{\bar{v}b} \quad \forall \underline{v}, \bar{v} \in V, \underline{v} \neq \bar{v}, b \in B \quad (6)$$

$$y_{\underline{v}\bar{v}} - 1 \leq x_{\underline{v}b} - x_{\bar{v}b} \leq 1 - y_{\underline{v}\bar{v}} \quad \forall \underline{v}, \bar{v} \in V, \underline{v} \neq \bar{v}, b \in B \quad (7)$$

$$ST_v \geq A_v \quad \forall v \in V \quad (8)$$

$$ST_{\bar{v}} \geq ST_{\underline{v}} + \sum_{b \in B} (HT_{\underline{v}b} x_{\underline{v}b}) - M(1 - y_{\underline{v}\bar{v}}) \quad \forall \underline{v}, \bar{v} \in V, \underline{v} \neq \bar{v} \quad (9)$$

$$WT_v \geq ST_v - A_v \forall v \in V \quad (10) \quad \text{Novel memetic algorithm}$$

$$LD_v \geq ST_v + \sum_{b \in B} (HT_{vb} x_{vb}) - RD_v \forall v \in V \quad (11)$$

$$ED_v = \max \left(0; RD_v - \left[ST_v + \sum_{b \in B} (HT_{vb} x_{vb}) \right] \right) \forall v \in V \quad (12)$$

$$x_{vb}, y_{v\bar{v}}, f_v, l_v \in \{0, 1\} \forall v, \underline{v}, \bar{v} \in V, \underline{v} \neq \bar{v}, b \in B \quad (13)$$

$$NC_v \in N \forall v \in V \quad (14)$$

$$ST_v, WT_v, ED_v, LD_v, A_v, hp_{vb}, HT_{vb}, RD_v, c_v^H, c_v^{WT}, c_v^{ED}, c_v^{LD}, M \in R^+ \forall v \in V, b \in B \quad (15)$$

In DDBSP, the objective function (1) aims to minimize the total service cost of vessels, which includes four components:

- (1) the total vessel handling cost;
- (2) the total vessel waiting cost;
- (3) the total penalty due to late vessel departures; and
- (4) the total premium due to early vessel departures (which is a negative cost component).

Constraint set (2) ensures that each vessel should be served once at one of the available MCT berths. Constraint set (3) indicates that each vessel is either served first or after another vessel at the given berth. Constraint set (4) ensures that each vessel is either served last or before another vessel at the given berth. Constraint set (5) indicates that only one vessel is served first at each berth of the MCT. Constraint set (6) ensures that only one vessel is served last at each berth of the MCT. Constraint set (7) indicates that a vessel can be served after another, if they are both assigned to the same berth of the MCT. Constraint set (8) ensures that service of a vessel starts only after its arrival. Constraint set (9) estimates start service time of each vessel. Constraint sets (10), (11) and (12) calculate waiting time, hours of late departure and hours of early departure for each vessel served at the MCT, respectively. Constraint sets (13)-(15) define ranges of variables and parameters in the DDBSP model.

Nonlinearity of the DDBSP mathematical model stems from constraint set (12). Note that replacing constraint set (12) with constraint set $ED_v \geq RD_v - \left[ST_v + \sum_{b \in B} (HT_{vb} x_{vb}) \right]$

$\forall v \in V$ [which is similar to inequality (11) that is used for estimating the late vessel departures] will make the solution to the problem unbounded (i.e. ED_v will be set to infinity “ ∞ ” for each vessel v , as the objective function of the DDBSP mathematical model minimizes the total service cost of vessels, and the total premium due to early vessel departures is treated as a negative cost component). Therefore, an alternative approach should be applied to linearize the DDBSP mathematical model (i.e. additional constraint sets will be required in the DDBSP mathematical model). Let $z_v^{ED} = 1$ if vessel v departs before

the requested departure time based on the suggested berth schedule (= 0 otherwise). Denote $ED_v^{aux}, v \in V$ as an auxiliary variable for estimating early vessel departures. Then, the original mixed-integer nonlinear DDBSP mathematical model can be reformulated as a mixed-integer linear mathematical model, which will be further referred to as DDBSP-L, as follows:

DDBSP-L:

$$\min \left[\sum_{v \in V} (NC_v c_v^H) + \sum_{v \in V} (WT_v c_v^{WT}) + \sum_{v \in V} (LD_v c_v^{LD}) - \sum_{v \in V} (ED_v c_v^{ED}) \right] \quad (16)$$

Subject to:

Constraint sets: (2)-(11), (13)-(15)

$$ED_v \geq RD_v - \left[ST_v + \sum_{b \in B} (HT_{vb} x_{vb}) \right] \forall v \in V \quad (17)$$

$$ED_v^{aux} \leq Mz_v^{ED} \forall v \in V \quad (18)$$

$$ED_v^{aux} \leq ED_v + M(1 - z_v^{ED}) \forall v \in V \quad (19)$$

The objective function (16) of the DDBSP-L mathematical model aims to minimize the total service cost of vessels. Constraint sets (17)-(19) estimate the hours of early departure for each vessel served at the MCT. Both DDBSP and DDBSP-L have *NP*-hard complexity. However, they can be solved to the global optimality using commercial optimization solvers for small-size problem instances. While the DDBSP mathematical model will require deployment of mixed-integer nonlinear programming solvers (e.g. BARON, AlphaECP, Couenne, DICOPT, Knitro), the DDBSP-L mathematical model can be solved using mixed-integer linear programming solvers (e.g. CPLEX, Gurobi, FICO-Xpress, MOSEK). A comparative analysis of the DDBSP and DDBSP-L mathematical models in terms of the computational time, required to solve the models, will be conducted in the numerical experiments section (details will be discussed in Section 6.3). Section 5 describes the solution algorithm, which will be used to solve the realistic-size problem instances of the DDBSP and DDBSP-L mathematical models.

5. Solution methodology

As discussed in Section 2, BSPs can be reduced to the machine scheduling problems, which are known to be *NP*-hard (Pinedo, 2008). Unlike many of published to-date BSP studies that mostly relied on the stochastic search algorithms, this paper proposes a MA with a deterministic parameter control (MA-DPC) to solve DDBSP (and its linearized version – DDBSP-L) for the realistic-size problem instances. Along with the stochastic search operators, MAs use local search heuristics, which typically improve the objective function value at termination and convergence pattern of the algorithm, as compared to canonical EAs and other stochastic search algorithms. Furthermore, application of the parameter control strategies allows better exploration and exploitation of the search space (Eiben and Smith, 2003). The main steps of the developed MA-DPC are outlined in Procedure 1.

Procedure 1. Memetic algorithm with a deterministic parameter control (MA-DPC)

MA-DPC(Input Data, PopSize, MutSteps, MutValues, StopCriterion)

in: InputData - values of the DDBSP parameters; PopSize - population size; MutSteps and MutValues - parameters of the piecewise function for the custom operator; StopCriterion - stopping criterion

out: BestChrom - the best vessel schedule

- 1: $|Pop| \leftarrow PopSize$; $|Fit| \leftarrow PopSize$; $|Parents| \leftarrow PopSize$;
 $|Offspring| \leftarrow PopSize$
- 2: Chrom \leftarrow **FCFS**(InputData) \triangleleft Chromosome initialization
- 3: $gen \leftarrow 0$
- 4: $Pop_{gen} \leftarrow$ **InitPop**(Chrom, PopSize) \triangleleft Population initialization
- 5: $Fit_{gen} \leftarrow$ **Evaluate**(Pop_{gen} , InputData) \triangleleft Initial population fitness evaluation
- 6: **while** StopCriterion \leftarrow FALSE **do**
- 7: $gen \leftarrow gen + 1$
- 8: $Parents_{gen} \leftarrow$ **SelectParents**(Pop_{gen}) \triangleleft Select parents
- 9: $Offspring_{gen} \leftarrow$ **MA-DPCoper**($Parents_{gen}$, MutSteps, MutValues) \triangleleft Produce offspring
- 10: $Fit_{gen} \leftarrow$ **Evaluate**($Offspring_{gen}$, InputData) \triangleleft Offspring fitness evaluation
- 11: $Pop_{gen+1} \leftarrow$ **Select**($Offspring_{gen}$, Fit_{gen}) \triangleleft Define population in the next generation
- 12: **end while**
- 13: **return** BestChrom

In Step 1, data structures for the algorithmic variables are initialized. In Steps 2-4, MA-DPC uses a local search heuristic for the chromosomes and population initialization. In Step 5, the fitness function values are estimated for the initial population chromosomes. Next, MA-DPC enters the main loop (Steps 6 through 12). In Step 8, function **SelectParents**(Pop_{gen}) chooses parents from the current population (i.e. variable $Parents_{gen}$) that will be used in Step 9 and produce the new offspring. In Step 9, function **MA-DPCoper**($Parents_{gen}$, $MutSteps$, $MutValues$) applies a custom operator with a deterministic parameter control to produce the new offspring (i.e. variable $Offspring_{gen}$). In Step 10, function **Evaluate**($Offspring_{gen}$, $InputData$) computes the values of fitness function (i.e. variable Fit_{gen}) for the offspring chromosomes. In Step 11, function **Select**($Offspring_{gen}$, Fit_{gen}) chooses the individuals based on their fitness that will become candidate parents in the following generation. Once the termination criterion is satisfied, MA-DPC exits the loop and returns the best chromosome (*BestChrom*), which represents the berth schedule with the least vessel service cost. A detailed description of the key algorithmic steps is presented next.

5.1 Chromosome representation

An integer chromosome representation is adopted in this paper to represent a solution (i.e. assignment of vessels to berths at the MCT). Each chromosome is composed of genes, which represent vessel identifiers (identifiers are denoted with integers). Note that such terms as vessel assignment, chromosome and individual will be used interchangeably throughout the paper. An example of a chromosome is presented in Figure 2, where 8 vessels request service at the MCT with two berths. Vessels “2”, “4”, “3” and “6” are served at berth “1” (in that order), while vessels “1”, “5”, “8”, and “7” are served at berth “2” (in that order). The length of chromosomes remains fixed throughout the MA-DPC evolution.

5.2 Chromosome/population initialization

Typically EAs rely on randomly initialized chromosomes; however, randomly initialized chromosomes and populations may contain a significant portion of low-quality and infeasible individuals, which may negatively affect the objective function value at termination and convergence pattern of the algorithm (Eiben and Smith, 2003; Sivanandam and Deepa, 2008). The algorithm, proposed in this study, applies a local search heuristic (which makes the algorithm memetic) to initialize the chromosomes. The heuristic is based on the first-come-first-served (FCFS) policy, widely used by the MCT operators in practice (Imai et al., 2008; Golias et al., 2009; Dulebenets, 2015a). Denote VS as the set of vessels sorted based on their arrival times; BA_b as the time when berth b becomes available for the first time in the planning horizon; BP_b as the berthing positions at berth b ; ST_v and FT_v as the start and finish service times of vessel v , respectively. The main steps of the FCFS heuristic are presented in Procedure 2.

Procedure 2. First-come-first-served (FCFS) policy

```

FCFS( $V, B, A_v, HT_{vb}$ )
in:  $V = \{1, \dots, N\}$  - set of vessels;  $B = \{1, \dots, K\}$  - set of berths;  $A_v$  -
arrival time of vessel  $v$ ;  $HT_{vb}$  - handling time of vessel  $v$  at berth  $b$ 
out:  $BP$  - vessel positions at each berth
1:  $|BA| \leftarrow K$ ;  $|BP| \leftarrow N \cdot K$ ;  $|VS| \leftarrow N$ ;  $|ST| \leftarrow N$ ;  $|FT| \leftarrow N$ 
                                      $\triangleleft$  Initialization
2:  $VS \leftarrow \mathbf{Sort}(V, A)$             $\triangleleft$  Sort vessels based on their arrival times
                                     in the ascending order
3:  $v \leftarrow 1$ 
4: for all  $v \in VS$  do
5:  $b \leftarrow \mathit{argmin}_b(BA_b)$         $\triangleleft$  Find the first available berth
6:  $BP_b \leftarrow BP_b \cup \{v\}$       $\triangleleft$  Assign a vessel to that berth
7:  $ST_v \leftarrow \max(A_v, BA_b)$     $\triangleleft$  Estimate vessel start service time
8:  $FT_v \leftarrow ST_v + HT_{vb}$       $\triangleleft$  Estimate vessel finish service time
9:  $BA_b \leftarrow FT_v$               $\triangleleft$  Update berth availability
10:  $v \leftarrow v + 1$ 
11: end for
12: return  $BP$ 

```

The initial population is constructed using the identical chromosomes. Various sizes of the initial population ($PopSize$) will be evaluated during the MA-DPC parameter tuning, and details will be discussed in Section 6. The population size is assumed to be constant and equal to the initial population size throughout the MA-DPC evolution.

5.3 Parent selection

The main objective of the parent selection procedure is to identify the chromosomes (i.e. “parents”) that will be selected to produce the offspring during the MA-DPC operations. This study adopts a deterministic parent selection scheme, which allows all individuals survived in the previous generation to become parents in the next generation. Such parent selection scheme is widely used in evolutionary programming (Eiben and Smith, 2003).

Figure 2.
Chromosome
representation

Berth	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
Vessel	2	4	3	6	0	0	0	0	1	5	8	7	0	0	0	0

5.4 MA-DPC operations

Crossover and mutation operators are commonly used for the EA/MA operations. However, for the proposed chromosome representation (Figure 2), typical crossover operators (e.g. one-point crossover, two-point crossover) may cause a complex infeasibility, as some offspring can inherit genes that represent the same vessels. Infeasible individuals may be repaired; however, the latter may incur a substantial computational time. To avoid a potential increase in time complexity of the algorithm, this study will use mutation only. Several types of mutation operators have been used in the past (Eiben and Smith, 2003): swap, invert, scramble, insert, etc. The swap mutation operator will be adopted in this study due to its efficiency for the chromosomes with integer representation (Golias et al., 2009; Dulebenets, 2015a). An example of the swap mutation is presented in Figure 3, where vessels “5” and “8” (originally scheduled at berth “2”) are switched with vessels “2” and “3” (originally scheduled at berth “1”), respectively.

The number of genes, swapped in each chromosome, is typically defined by the mutation rate (*MutRate*) in canonical EAs and MAs. However, this study uses MA-DPC with a deterministic parameter control for the mutation operator, where *MutRate* is no longer a pre-defined parameter. Specifically, the custom mutation operator, deployed within the MA-DPC algorithm, changes *MutRate* throughout the MA-DPC evolution based on values of the piecewise function in a given generation. Denote *I* as a set of segments in the piecewise function; *MutRate_{gen}* as a value of *MutRate* in generation *gen*, suggested by the piecewise function; *MutValues_i* as a *MutRate* value at segment *i* in the piecewise function; and *MutSteps_i* as a *gen* value at the beginning of segment *i* in the piecewise function. Then, the mutation rate *MutRate_{gen}* in generation *gen* that will be set by the MA-DPC mutation operator can be estimated using the following equation:

$$MutRate_{gen} = MutValues_i : MutSteps_i < gen < MutSteps_{i+1} \tag{20}$$

An example of the piecewise function for the custom mutation operator with *MutSteps* = [0;500; 1,000; 1,500; 2,000] and *MutValues* = [8; 6; 4; 2] is presented in Figure 4. For instance, the custom mutation operator will set *MutRate* = 6 in generation *gen* = 700 (Figure 4, segment *i* = 2 of the piecewise function). Setting high *MutRate* values at the beginning of the

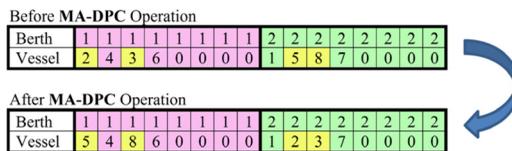


Figure 3. MA-DPC operation example

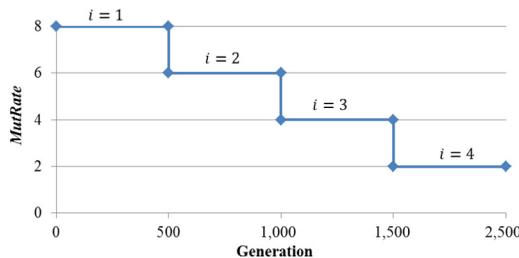


Figure 4. Piecewise function for the custom mutation operator

MA-DPC run allows better exploration of the search space, while decreasing *MutRate* values towards the MA-DPC convergence allows exploiting the promising domains (Eiben and Smith, 2003). Note that throughout the MA-DPC operations, the custom mutation operator is applied to each individual in the population. Along with the deterministic parameter control strategy, there exist the other alternatives for the parameter control (i.e. adaptive parameter control and self-adaptive parameter control – Eiben and Smith, 2003), which can be explored as a part of the future research.

5.5 Fitness function

The fitness function is typically associated with the objective function in EAs/MAs (Sivanandam and Deepa, 2008). In the developed MA-DPC, the fitness function is set equal to the objective function. No scaling mechanisms were applied to the fitness function.

5.6 Offspring selection

Offspring selection determines the individuals that will survive in the given generation and will become candidate parents in the next generation. This study will use a tournament selection mechanism, which has two parameters:

- (1) tournament size (*TourSize*) – number of individuals participating in each tournament; and
- (2) individuals selected (*IndSel*) – number of individuals selected based on their fitness at each tournament to become candidate parents in the next generation.

The tournament selection mechanism is based on multiple tournaments. During each tournament, a certain portion of individuals (*TourSize*) are randomly selected from the population. Then, a set of the best individuals (i.e. individuals with the highest fitness) are chosen from the tournament (a total of *IndSel* individuals) to become parents in the next generation. The tournament selection mechanism continuously performs tournaments until the desired population size is reached. The total number of tournaments (*NumTour*), required to obtain the desired population size, can be computed as follows: $NumTour = \frac{PopSize}{IndSel}$ (i.e. ratio of the desired population size to the total number of individuals chosen at one tournament). Various values of the offspring selection parameters will be evaluated during the MA-DPC parameter tuning and details will be discussed in Section 6. The elitist strategy is applied in the developed MA-DPC to ensure that the best individual lives more than one generation (Eiben and Smith, 2003).

5.7 Stopping criterion

The proposed MA-DPC will be terminated once the maximum number of generations (*LastGen*) is reached. The value of *LastGen* will be determined based on the preliminary MA-DPC runs (see Section 6 for details).

6. Computational experiments

This section presents a number of computational experiments that were conducted to assess efficiency of the developed MA-DPC in terms of computational time and solution quality. The proposed MA-DPC will be compared against the following alternative algorithms:

- MA that applies the FCFS heuristic for the chromosomes and population initialization, but does not apply the deterministic parameter control strategy for the mutation operator;

- typical EA that initializes the chromosomes and population randomly (i.e., without using any local search heuristics) and does not apply the deterministic parameter control strategy for the mutation operator;
- SA; and
- VNS.

Both SA and VNS algorithms can be considered as advanced state-of-the-art metaheuristic algorithms that have been widely used for solving BSP problems (as discussed in the literature review section). For a detailed description of the SA and VNS algorithms this study refers to [Emde and Boysen \(2016\)](#) and [Hansen et al. \(2008\)](#), respectively. The MA-DPC, MA, EA, SA and VNS algorithms were coded using MATLAB 2016a and executed on a CPU with Dell Intel(R) Core™ i7 Processor and 32 GB of RAM. The next sections of the paper elaborate on the input data used, parameter tuning, optimality gap analysis, algorithmic performance and comparative analysis of the developed algorithms for the continuous berthing layout case.

6.1 Input data

The numerical data, required for the computational experiments, were generated based on the available MCT literature ([Imai et al., 2007a, 2007b, 2008, 2013](#); [Goliás et al., 2009](#); [Dulebenets, 2012](#); [Zampelli et al., 2014](#); [Dulebenets, 2015a, 2015b](#); [Dulebenets, 2016a, 2016b, 2016c](#); [Dulebenets et al., 2016](#); [The Port Authority of New York and New Jersey, 2016](#); [Dulebenets and Ozguven, 2017](#)) and are presented in [Table I](#). A planning horizon of two weeks was modeled in this study. The berthing capacity (BC) at the MCT varied from two berths to four berths with an increment of one berth. A total of four vessel interarrival patterns (A1-4), varying from 2 h to 3 h, and following the exponential distribution ([Imai et al., 2007a, 2007b, 2008, 2013](#)) were considered. The number of containers to be handled for each vessel was assigned as $NC_v = U[500; 2,000] \forall v \in V$ (TEUs), where U – is a notation for uniformly distributed pseudorandom numbers.

The handling productivity ($hp_{vb}, v \in V, b \in B$) at the preferred berth was set to 125 TEUs per hour. Note that the preferred berth was identified based on the FCFS policy for each vessel. If a given vessel was diverted for service from the preferred berth to the other berth, the handling productivity was decreased proportionally to the distance between the actual and desired berthing positions ([Bierwirth and Meisel, 2015](#)). The requested departure time of a vessel was assigned based on the vessel arrival time and

Parameter	Value
Planning horizon	2 weeks
Number of berths at the MCT: [BC1; BC2; BC3]	[2; 3; 4]
Vessel interarrival patterns – exponential: [A1; A2; A3; A4] (hours)	[2.0; 2.3; 2.7; 3.0]
Containers assigned to each vessel: $NC_v, v \in V$ (TEUs)	$U[500; 2,000]$
Handling productivity at the preferred berth: $hp_{vb}, v \in V, b \in B$ (TEUs per hour)	125
Requested vessel departure: $RD_v, v \in V$ (hours)	$RD_v = A_v + HT_{vb} \cdot U[1.0; 1.2]$
Vessel handling cost: $c_v^H, v \in V$ (US\$ per TEU)	$U[400; 600]$
Vessel waiting cost: $c_v^{WT}, v \in V$ (US\$ per hour)	$U[1,500; 2,500]$
Early departure premium: $c_v^{ED}, v \in V$ (US\$ per hour)	$U[4,000; 6,000]$
Late departure penalty: $c_v^{LD}, v \in V$ (US\$ per hour)	$U[6,000; 8,000]$

Table I.
Numerical data

the handling time at the preferred berth. Based on the available literature (Zampelli *et al.*, 2014; The Port Authority of New York and New Jersey, 2016), the vessel handling cost was generated as $c_v^H = U[400; 600] \forall v \in V$ (US\$ per TEU), while the vessel waiting cost was initialized as $c_v^{WT} = U[1, 500; 2, 500] \forall v \in V$ (US\$ per hour). The early departure premium and late departure penalty were generated as $c_v^{ED} = U[4, 000; 6, 000] \forall v \in V$ (US\$ per hour) and $c_v^{LD} = U[6, 000; 8, 000] \forall v \in V$ (US\$ per hour), respectively (Zampelli *et al.*, 2014).

Based on the generated numerical data, a total of 30 problem instances were developed, which can be classified into the following two groups:

- (1) Small-size problem instances, which were generated by changing the berthing capacity at the MCT (i.e. BC1, BC2, BC3) and the number of vessels calling for service at the MCT (from 6 to 16 with an increment of 2 vessels), assuming frequent vessel arrivals (i.e., vessel interarrival pattern A1) – [3 berthing capacity scenarios] \times [6 vessel call types] = 18 problem instances (that will be referred to as S1-S18); and
- (2) Realistic-size problem instances, which were generated by changing the berthing capacity at the MCT (i.e. BC1, BC2, BC3) and the vessel interarrival patterns (i.e., A1, A2, A3, A4) – [3 berthing capacity scenarios] \times [4 vessel interarrival patterns] = 12 problem instances (that will be referred to as R1-R12).

Note that the small-size problem instances will be used for the optimality gap analysis of the developed algorithms (which will be described in Section 6.3 of the paper), while the realistic-size problem instances will be used for the comprehensive evaluation of the developed algorithms in terms of the objective function values at termination, computational time and convergence patterns (which will be described in Section 6.4 of the paper).

6.2 Parameter tuning

As mentioned in the beginning of Section 6, efficiency of the proposed MA-DPC will be evaluated based on a comparative analysis against the MA, EA, SA, and VNS algorithms. The developed algorithms have several parameters that have to be determined based on the parameter tuning analysis (Eiben and Smith, 2003). A total of three problem instances were randomly selected from the generated 12 realistic-size problem instances for the parameter tuning. For each instance, a total of 600 possible parameter combinations were tested for MA-DPC, considering the following candidate values of parameters: $PopSize = [20; 30; 40; 50; 60]$, $|I| = [2; 3; 4]$, $MutValues = [2; 4; 6; 8; 10]$, $MutSteps_{i+1} = MutSteps_i + \frac{LastGen}{|I|}$, $TourSize = [0.5 \cdot PopSize; 0.6 \cdot PopSize]$ and $IndSel = [0.1 \cdot PopSize; 0.2 \cdot PopSize]$. As for the MA and EA parameters, for each instance, a total of 100 possible parameter combinations were tested, considering the following candidate values of parameters: $PopSize = [20; 30; 40; 50; 60]$, $MutRate = [2; 4; 6; 8; 10]$, $TourSize = [0.5 \cdot PopSize; 0.6 \cdot PopSize]$, and $IndSel = [0.1 \cdot PopSize; 0.2 \cdot PopSize]$. A total of 5 replications were performed to estimate the average values of the objective function and computational time for each combination of parameters, i.e. a total of $600 \times 5 = 3,000$ MA-DPC runs, $100 \times 5 = 500$ MA runs and $100 \times 5 = 500$ EA runs for each problem instance. Throughout the computational experiments, it was noticed that the objective function values did not change significantly after generation $gen = 2,000$ for the MA-DPC, MA and EA algorithms. Hence, the termination criterion will be set to $LastGen = 2,000$ generations for those algorithms.

The SA algorithm has a total of three parameters:

- (1) initial temperature (T^0);
- (2) temperature interval (ΔT); and
- (3) maximum number of iterations ($LastIter$).

For each instance a total of 25 possible parameter combinations were tested for SA, considering the following candidate values of parameters: $T^0 = [500; 750; 1,000; 1,250; 1,500]$ and $\Delta T = [0.2; 0.3; 0.4; 0.5; 0.6]$. The VNS algorithm also has a total of three parameters:

- (1) number of neighborhood structures ($Nstr$);
- (2) exchange rate ($ExRate$), i.e. the number of vessels to be swapped in the current solution of a given neighborhood in order to generate a solution in another neighborhood; and
- (3) maximum number of iterations ($LastIter$).

For each instance, a total of 25 possible parameter combinations were tested for VNS, considering the following candidate values of parameters: $Nstr = [20; 30; 40; 50; 60]$ and $ExRate = [2; 4; 6; 8; 10]$. A total of five replications were performed to estimate the average values of the objective function and computational time for each combination of parameters, i.e. a total of $25 \times 5 = 125$ runs for both SA and VNS algorithms for each problem instance. Throughout the computational experiments, it was noticed that the objective function values did not change significantly after iteration $LastIter = 2,000$ for the SA and VNS algorithms. Hence, the termination criterion will be set to $LastIter = 2,000$ iterations for those algorithms. [Table II](#) shows results of the parameter tuning analysis, where the algorithmic parameter values were selected based on the tradeoff between the computational time and the objective function value at termination.

6.3 Optimality gap analysis

The first set of computational experiments focused on estimating the optimality gaps for the developed algorithms to determine how close the solutions, obtained by the algorithms, to the global optimal solutions. The small-size problem instances (S1-S18, described in Section 6.1) were used throughout the optimality gap analysis. The mixed-integer nonlinear DDBSP mathematical model was coded in General Algebraic Modeling System (GAMS) and solved using BARON for all the generated small-size problem instances. The relative optimality gap for BARON was assigned to be 0.1 per cent, while the allowable computational time was restricted to 2 h (or 7,200 s). Note that BARON is widely used in operations research for solving mixed-integer nonlinear mathematical models to the global optimality ([Sahinidis, 2017](#)). The MA-DPC, MA, EA, SA, and VNS algorithms were executed for the small-size problem instances as well. A total of five replications were performed for each problem instance to estimate the average objective function and computational time values for each algorithm. Results of the optimality gap analysis are summarized in [Table III](#), where the following data are presented:

- instance number;
- berthing capacity (BC);
- number of vessels calling for service at the MCT ($|V|$);
- objective function values (denoted as Z) for each solution approach; and
- computational time values (denoted as CPU) for each solution approach.

Table II.
Parameter tuning
analysis results

Algorithm	Parameter	Adopted value
MA-DPC	Population size: <i>PopSize</i>	50
MA-DPC	Number of segments in the piecewise function for the custom mutation operator: $ I $	4
MA-DPC	Mutation rate values at each segment in the piecewise function: <i>MutValues</i>	[6; 4; 2; 2]
MA-DPC	Generation values at the beginning of each segment in the piecewise function: <i>MutSteps</i>	[0; 500; 1,000; 1,500; 2,000]
MA-DPC	Tournament size: <i>TourSize</i>	30
MA-DPC	Number of individuals selected at each tournament: <i>IndSel</i>	5
MA-DPC	Maximum number of generations: <i>LastGen</i>	2,000
MA	Population size: <i>PopSize</i>	50
MA	Mutation rate: <i>MutRate</i>	2
MA	Tournament size: <i>TourSize</i>	30
MA	Number of individuals selected at each tournament: <i>IndSel</i>	5
MA	Maximum number of generations: <i>LastGen</i>	2,000
EA	Population size: <i>PopSize</i>	50
EA	Mutation rate: <i>MutRate</i>	2
EA	Tournament size: <i>TourSize</i>	30
EA	Number of individuals selected at each tournament: <i>IndSel</i>	5
EA	Maximum number of generations: <i>LastGen</i>	2,000
SA	Initial temperature: T^0	1,000
SA	Temperature interval: ΔT	0.5
SA	Maximum number of iterations: <i>LastIter</i>	2,000
VNS	Number of neighborhood structures: <i>Nstr</i>	50
VNS	Exchange rate: <i>ExRate</i>	2
VNS	Maximum number of iterations: <i>LastIter</i>	2,000

The optimality gap for algorithm a (G_a) was estimated based on the following relationship: $G_a = \frac{Z_a - Z_{BARON}}{Z_{BARON}}$. The obtained optimality gap values are presented in Figure 5.

It can be observed that due to complexity of the DDBSP mathematical model, BARON computational time was significantly affected with increasing problem size. For certain problem instances (i.e. instance S12 and S18), BARON was not able to provide the global optimal solution within the time limit imposed. The solutions, obtained by BARON for problem instances S12 and S18, were outperformed by the MA-DPC, MA, EA, SA and VNS algorithms in terms of the objective function values on average by 4.04, 3.51, 2.03, 1.82 and 2.31 per cent respectively. For the rest of small problem instances, the maximum optimality gap values comprised 0.46, 0.72, 1.94, 2.39 and 1.98 per cent for the MA-DPC, MA, EA, SA and VNS algorithms, respectively, which showcase a high accuracy of the developed algorithms. Furthermore, results indicate that the proposed MA-DPC, which applies a deterministic parameter control strategy, provided superior objective function values as compared to the MA, EA, SA and VNS algorithms. Specifically, the MA-DPC objective function value was lower, on average, by 0.19, 0.69, 0.81 and 0.68 per cent as compared to the objective function values, obtained by the MA, EA, SA and VNS algorithms, respectively. The average computational time comprised 48.4 s, 42.6 s, 41.5 s, 9.1 s and 31.7 s, respectively. Lower computational time was required by the SA and VNS algorithms due to

Inst.	BC	V	BARON		MA-DPC		MA		EA		SA		VNS	
			Z, 10 ⁶ US\$	CPU, s	Z, 10 ⁶ US\$	CPU, s	Z, 10 ⁶ US\$	CPU, s	Z, 10 ⁶ US\$	CPU, s	Z, 10 ⁶ US\$	CPU, s	Z, 10 ⁶ US\$	CPU, s
S1	[BC1]	6	3.815	5.22	3.815	38.97	3.815	34.43	3.815	33.71	3.815	7.29	3.815	25.51
S2	[BC1]	8	5.596	62.33	5.596	41.50	5.596	36.43	5.596	35.45	5.596	7.81	5.596	27.11
S3	[BC1]	10	7.518	148.80	7.518	43.70	7.518	38.47	7.518	37.28	7.518	8.19	7.518	28.62
S4	[BC1]	12	9.689	342.14	9.689	45.77	9.695	40.71	9.695	39.29	9.695	8.58	9.695	30.15
S5	[BC1]	14	11.468	595.01	11.468	47.80	11.502	42.35	11.521	41.05	11.554	9.04	11.554	31.56
S6	[BC1]	16	12.892	995.47	12.892	50.66	12.958	44.46	13.020	43.08	13.037	9.46	13.020	33.12
S7	[BC2]	6	3.689	15.29	3.689	42.47	3.689	37.70	3.689	36.56	3.689	7.81	3.689	27.89
S8	[BC2]	8	5.343	185.89	5.343	44.61	5.343	39.79	5.343	38.83	5.343	8.29	5.343	29.16
S9	[BC2]	10	7.031	499.86	7.031	47.29	7.031	41.63	7.031	41.08	7.031	8.91	7.031	31.00
S10	[BC2]	12	8.896	1205.18	8.896	49.80	8.918	43.73	9.043	43.11	9.053	9.34	9.046	32.63
S11	[BC2]	14	10.404	2331.02	10.424	51.97	10.461	45.84	10.593	44.94	10.640	10.01	10.593	34.12
S12	[BC2]	16	12.055	7201.49	11.585	55.88	11.643	48.25	11.846	46.79	11.857	10.40	11.796	35.50
S13	[BC3]	6	3.627	32.22	3.627	45.77	3.627	39.75	3.627	38.80	3.627	8.52	3.627	29.45
S14	[BC3]	8	5.220	342.44	5.220	48.00	5.220	41.98	5.220	41.39	5.220	8.95	5.220	31.17
S15	[BC3]	10	6.810	784.84	6.810	50.55	6.810	44.39	6.810	43.43	6.810	9.55	6.810	33.41
S16	[BC3]	12	8.578	1857.84	8.578	53.25	8.628	46.31	8.728	45.47	8.748	10.15	8.732	34.54
S17	[BC3]	14	9.953	3249.19	9.999	55.64	10.025	48.90	10.146	47.22	10.191	10.69	10.150	36.15
S18	[BC3]	16	11.504	7202.12	11.023	57.51	11.090	52.08	11.236	49.91	11.274	11.08	11.219	38.84
Average:			8.005	1503.13	7.956	48.40	7.976	42.62	8.027	41.52	8.039	9.11	8.025	31.66

Table III.
Comparison of the developed algorithms for the small-size instances

the fact that those algorithms are not population-based (and, therefore, require less evaluations of the generated solutions).

As discussed in Section 4, the mixed-integer nonlinear DDBSP mathematical model can be linearized to the DDBSP-L mathematical model and solved using mixed-integer linear programming solvers to the global optimality for the small-size problem instances. The scope of computational experiments includes evaluation of both DDBSP and DDBSP-L mathematical models in terms of the computational time, required to solve the models. To accomplish the latter objective, the mixed-integer linear DDBSP-L mathematical model was coded in GAMS and solved using CPLEX for all the generated small-size problem instances. CPLEX is widely used in operations research for solving large-scale mixed-integer linear programming models to the global optimality. The relative optimality gap for CPLEX was restricted to 0.1 per cent, while the allowable computational time was assigned to be 2 h (or 7,200 s). Note that the same settings were established for BARON. The comparative analysis results are illustrated in Figure 6, which presents the computational time, required by BARON and CPLEX to solve the small-size problem instances of the DDBSP and DDBSP-L mathematical models, respectively. Note that BARON and CPLEX were not compared in terms of solution quality, as they returned solutions with the same objective function values (i.e. the global optimal solutions) for the problem instances, which were solved within the time limit imposed. It can be observed that CPLEX was generally able to obtain the optimal solutions faster as compared to BARON. However, owing to the complexity of the DDBSP and DDBSP-L mathematical models, both solution approaches were not able to solve problem instance S18 to the global optimality within the time limit imposed.

6.4 Algorithmic performance

Upon completion of the optimality gap analysis for MA-DPC, MA, EA, SA, and VNS, the algorithms were executed for all 12 developed realistic-size problem instances (R1-R12,

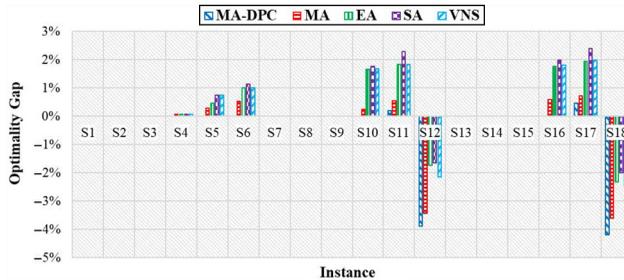


Figure 5.
Optimality gap values for the developed algorithms

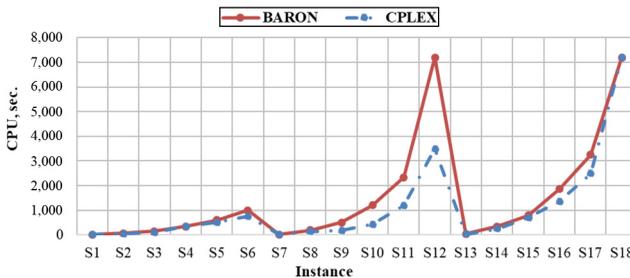


Figure 6.
Computational time of the exact optimization approaches: BARON vs CPLEX

described in Section 6.1). A total of five replications were performed for each problem instance to estimate the average objective function and computational time values for each algorithm. Results are presented in [Table IV](#), including the following information:

- instance number;
- berthing capacity (BC);
- vessel interarrival pattern (A);
- objective function values (denoted as Z) for each solution approach; and
- computational time values (denoted as CPU) for each solution approach.

Along with the objective function and computational time values, the coefficient of variation of the objective function values was recorded for each algorithm and each problem instance to measure variability in the objective function values at convergence over five replications. The estimated coefficient of variation values are presented in [Figure 7](#) for each algorithm and each instance.

We observe that MA-DPC reduced the total vessel service cost of berth schedules, on average, over all realistic-size problem instances and replications by 4.9, 6.9, 7.2, and 5.4 per cent as compared to the MA, EA, SA and VNS berth schedules, respectively. Furthermore, MA-DPC yielded greater cost savings for the cases with high demand and low MCT berthing capacity (i.e. the average cost savings over the MA, EA, SA and VNS berth schedules for problem instance R1 with BC1 and A1 comprised 12.0, 14.7, 13.5 and 8.0 per cent, respectively). The SA algorithm demonstrated the worst performance in terms of the objective function values at termination. The latter finding can be explained by the fact that SA is a single-solution-based algorithm (unlike the MA-DPC, MA and EA algorithms that are population-based), which limits the SA explorative capabilities. Moreover, the SA initial solution is generated randomly without using any local search heuristic. The VNS algorithm showed a good performance for the problem instances with low MCT berthing capacity and frequent vessel arrivals (i.e. problem instances R1, R5, and R9). The latter can be supported by the fact that VNS allows efficient exploration of various neighborhoods. However, for the rest of problem instances the VNS performance was worsening primarily due to the fact that it is a single-solution-based algorithm.

The coefficient of variation did not exceed 1.6 per cent over all the developed algorithms and considered problem instances, indicating stability of the algorithms. However, lower coefficient of variation values were generally recorded for the MA-DPC algorithm. The average computational time was 222.5 s, 222.7 s, 210.2 s, 98.9 s, and 160.8 s over all the generated realistic-size problem instances and performed replications for the MA-DPC, MA, EA, SA and VNS algorithms, respectively. The SA algorithm incurred the least computational time, as SA evaluates only two solutions at each iteration (i.e. the current solution and the neighbor solution), while VNS requires evaluation of solutions within different neighborhoods at each iteration of the algorithm. On the other hand, MA-DPC, MA and EA evaluate each solution in the given population at each generation, which incurs greater computational time as compared to the SA and VNS algorithms.

Throughout the computational experiments, the convergence patterns of the algorithms were recorded for each replication and each problem instance. Results are presented in [Figure 8](#) for the first replication of all realistic-size problem instances (note that the convergence patterns did not change significantly from one replication to the other for the MA-DPC, MA, EA, SA, and VNS algorithms). We observe that all the developed algorithms provide berth schedules, which are significantly superior to the initial berth schedule (produced based on the FCFS policy) in terms of the total vessel service cost, especially for

Table IV.
Comparison of the
developed algorithms
for the realistic-size
problem instances

Inst.	BC	A	MA-DPC		MA		E/A		SA		VNS	
			Z, 10 ⁶ US\$	CPU, s								
R1	[BC1]	[A1]	607.91	264.14	680.58	260.03	697.15	243.02	689.78	101.44	656.14	187.89
R2	[BC1]	[A2]	474.68	192.82	502.44	192.03	504.23	179.64	504.92	82.25	503.18	156.36
R3	[BC1]	[A3]	414.01	219.21	453.77	218.92	471.90	205.75	475.47	71.11	472.81	132.93
R4	[BC1]	[A4]	238.17	168.52	252.24	169.31	261.88	159.12	262.80	61.68	262.23	117.16
R5	[BC2]	[A1]	391.92	273.69	399.43	273.29	411.47	257.58	412.55	124.52	397.99	198.70
R6	[BC2]	[A2]	296.06	206.50	300.93	208.28	303.25	196.49	304.30	102.82	301.34	165.98
R7	[BC2]	[A3]	254.70	233.92	256.70	232.80	260.17	219.62	265.04	89.43	262.19	145.55
R8	[BC2]	[A4]	141.98	181.82	143.29	181.99	144.84	171.85	145.61	77.30	145.09	128.44
R9	[BC3]	[A1]	260.26	291.55	261.26	288.44	265.21	272.89	266.65	149.75	264.22	217.55
R10	[BC3]	[A2]	200.81	210.90	201.22	215.53	202.05	204.88	202.82	124.06	202.52	181.02
R11	[BC3]	[A3]	159.44	239.84	161.39	243.36	162.35	231.55	163.77	109.12	163.30	158.04
R12	[BC3]	[A4]	97.50	187.26	97.61	188.54	97.81	179.56	98.04	92.69	97.96	139.54
Average:			294.79	222.52	309.24	222.71	315.19	210.16	315.98	98.85	310.75	160.76

the instances with low MCT berthing capacity. However, a random initialization of the chromosomes and population negatively affects fitness of individuals within the initial population of the EA, SA and VNS algorithms (based on the convergence patterns, it can be noticed that the initial EA, SA and VNS solutions have greater total vessel service cost as compared to the solutions, constructed based on the FCFS policy).

Moreover, the introduction of the custom mutation operator with a deterministic parameter control allows MA-DPC more efficient exploration and exploitation of the search space. For example, in problem instance R1, both MA-DPC and MA identified solutions with the total vessel service cost of \approx US\$680m in generation \approx 60. Afterwards, MA was not able to explore any other domains of the search space due to low mutation rate, which resulted in a local (or “premature”) convergence. However, a high mutation rate in the beginning of the MA-DPC run and lower mutation rate toward the MA-DPC convergence allowed discovering berth schedules with significantly lower vessel service costs. The EA algorithm demonstrated worse performance than MA-DPC and MA, as it solely relies on the stochastic search operators and does not apply any parameter control strategies. As it was mentioned earlier, the VNS and SA algorithms were outperformed by the MA-DPC, MA and EA algorithms for the majority of realistic-size problem instances due to the fact that VNS and SA are single-solution-based algorithms that do not deploy any local search heuristics for initialization of the starting solutions.

Results from the extensive numerical experiments demonstrate that the proposed algorithm, which applies a local search heuristic for the chromosomes and population initialization and a deterministic parameter control strategy, can serve as an efficient planning tool for the MCT operators and assist with construction of cost-effective berth schedules. Furthermore, deployment of a local search heuristic and a custom mutation operator yields greater savings in terms of the total vessel service cost for the cases with high demand and low MCT berthing capacity without significantly affecting the computational time.

6.5 Comparative analysis for the continuous berthing layout case

The scope of computational experiments also includes a comparative analysis of the developed algorithms for the continuous berthing layout. In case of the continuous berthing layout, the wharf is not partitioned into separate berthing positions (as it is assumed in the discrete berthing layout case), and the total number of vessels that can be moored along the wharf is determined based on their overall length and the wharf’s length (i.e. the overall length of vessels, including safety distances between consecutive vessels, cannot exceed the wharf’s length). The wharf’s length was varied from 3,000 ft. to 6,000 ft. with an increment of 1,500 ft. The length of arriving vessels was set as $l_v = U[950; 1,350] \forall v \in V$ (ft.), where 950 ft. is a typical length of Panamax vessels, while 1,350 ft. is a typical length of Triple E vessels that are also classified as ultra-large container vessels (Port Technology, 2015). Additional 12 realistic-size problem instances (that will be

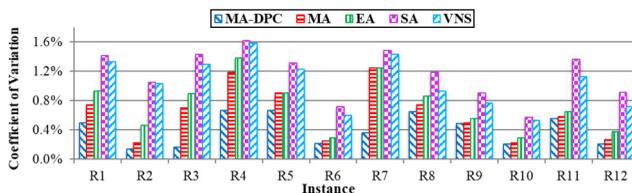


Figure 7. Objective function coefficient of variation for the developed algorithms

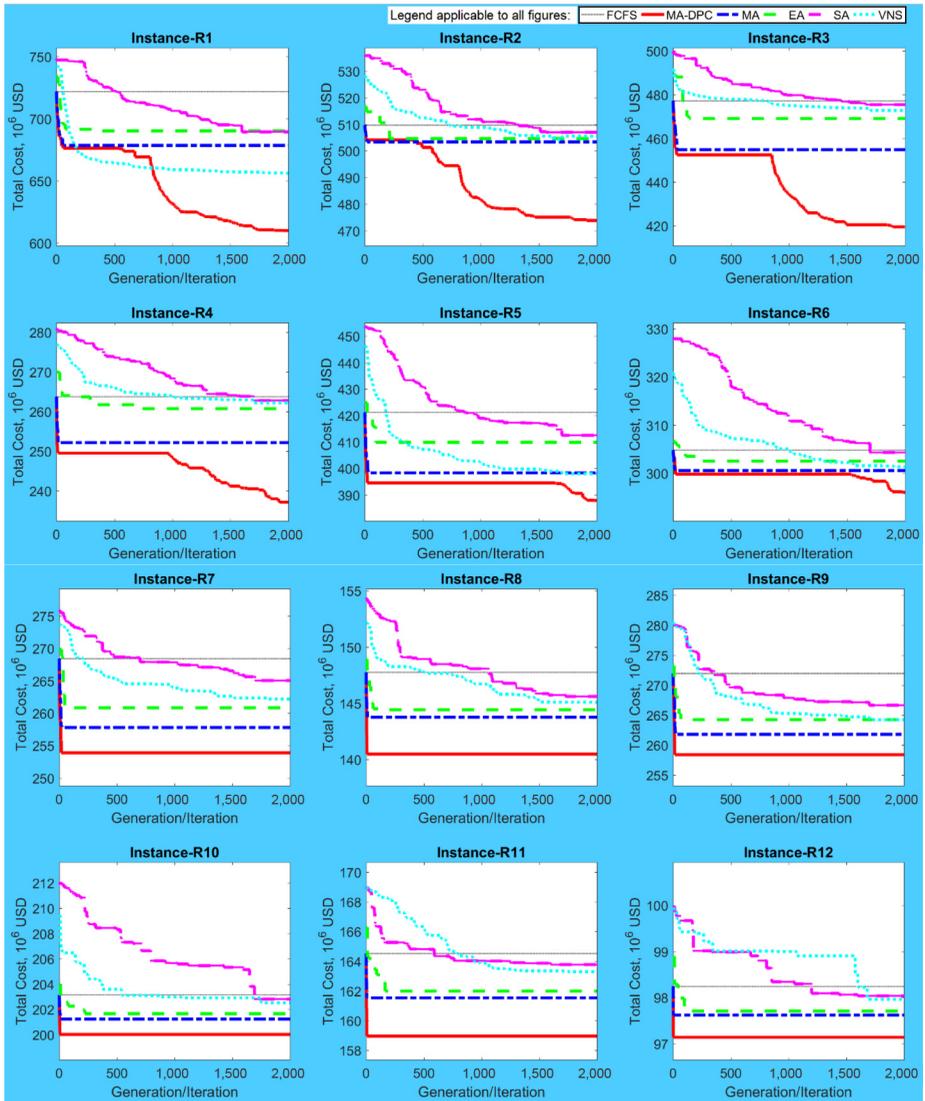


Figure 8.
Convergence patterns
of the considered
algorithms

referred to as R13-R24) were developed by changing the wharf's length at the MCT (i.e. 3,000 ft. – [WL1], 4,500 ft. – [WL2] and 6,000 ft. – [WL3]) and the vessel interarrival patterns (i.e. A1, A2, A3, A4). The developed algorithms were executed for additional 12 realistic-size problem instances (R13-R24). A total of five replications were performed for each problem instance to estimate the average objective function and computational time values for each algorithm. Results are presented in Table V, including the following information:

Inst.	WL	A	MA-DPC		MA		EA		SA		VNS	
			Z, 10 ⁶ US\$	CPU, s								
R13	[WL1]	[A1]	384.09	275.45	438.96	271.30	446.58	258.58	444.93	108.34	424.17	198.22
R14	[WL1]	[A2]	307.86	205.80	332.64	202.12	335.80	192.03	334.65	86.20	333.98	162.93
R15	[WL1]	[A3]	254.32	235.28	284.13	227.79	294.96	214.81	295.00	75.45	294.48	141.97
R16	[WL1]	[A4]	221.03	179.52	238.71	178.03	248.63	169.94	248.29	64.52	248.16	125.36
R17	[WL2]	[A1]	351.82	287.87	365.77	287.58	377.53	268.91	377.02	132.11	364.87	208.84
R18	[WL2]	[A2]	268.45	217.51	278.29	218.38	280.83	204.35	281.02	108.89	278.85	173.45
R19	[WL2]	[A3]	238.11	250.80	244.70	246.65	248.32	230.60	251.73	94.08	249.43	155.74
R20	[WL2]	[A4]	195.29	197.25	200.71	190.64	203.25	179.76	203.41	81.25	202.64	133.70
R21	[WL3]	[A1]	290.99	305.12	297.38	302.01	301.59	286.26	302.30	159.34	299.93	227.12
R22	[WL3]	[A2]	232.78	229.94	236.65	226.41	237.99	218.81	237.89	130.14	238.34	193.33
R23	[WL3]	[A3]	227.50	255.27	233.00	259.13	234.92	245.67	236.39	115.99	236.24	168.47
R24	[WL3]	[A4]	148.01	195.53	149.78	196.97	150.06	188.36	150.59	96.39	150.75	145.68
Average:			260.02	236.28	275.06	233.92	280.04	221.51	280.27	104.39	276.82	169.57

Table V. Comparison of the developed algorithms for the continuous berthing layout case

- instance number;
- wharf's length (WL);
- vessel interarrival pattern (A);
- objective function values (denoted as Z) for each solution approach; and
- computational time values (denoted as CPU) for each solution approach.

The MA-DPC cost savings over the alternative algorithms for the continuous berthing layout case are presented in Figure 9 for realistic-size problem instances R13-R24.

Similar to the results, obtained for the discrete berthing layout case, MA-DPC outperformed the other developed algorithms in terms of solution quality for the continuous berthing layout case. Specifically, the total vessel service cost of the MA-DPC berth schedules was lower, on average, by 5.8, 7.7, 7.8 and 6.5 per cent as compared to the MA, EA, SA and VNS berth schedules, respectively. Therefore, greater cost savings from the MA-DPC berth schedules were obtained for the continuous berthing layout case, as compared to the discrete berthing layout case. Moreover, substantial cost savings were recorded for the problem instances with lower capacity of the wharf and frequent vessel arrivals. The MA-DPC algorithm outperformed the MA, EA, SA and VNS algorithms in terms of the total vessel service cost, on average, by 10.5, 13.5, 13.2 and 11.8 per cent, respectively, for realistic-size problem instances R13-R16. The computational time of the developed algorithms was greater for the continuous berthing layout case, as compared the discrete berthing layout case. However, the increase in the average computational time did not exceed 6.3 per cent over the considered algorithms. Thus, the proposed MA-DPC algorithm will assist the MCT operators with efficient berth planning for both discrete and continuous berthing layout cases.

7. Conclusions and future research extensions

MCT operators have to seek new strategies that will allow to meet the growing demand for containerized trade and in the meantime coping with the existing congestion issues. This paper focused on improving the seaside operations of MCTs and proposed a novel MA with a deterministic parameter control to ensure efficiency of berth schedules by minimizing the total vessel service cost. Unlike published to-date berth scheduling studies that mostly rely on the stochastic search algorithms, the developed algorithm applied a local search heuristic for the chromosomes and population initialization and adjusted the mutation rate values based on a deterministic parameter control strategy. A set of computational experiments were performed to assess efficiency of the proposed MA with a deterministic parameter control based on a comprehensive comparative analysis against the alternative state-of-the-art solution algorithms widely used in the berth scheduling literature.

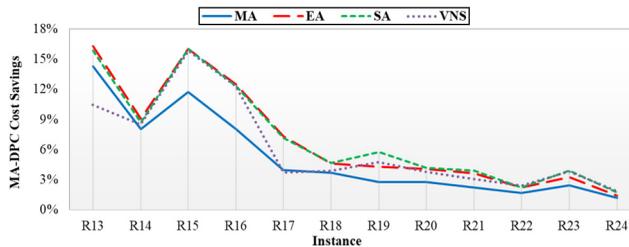


Figure 9. MA-DPC cost savings over the alternative algorithms

Results demonstrated that the developed MA with a deterministic parameter control was able to obtain superior berth schedules in terms of the total vessel service cost within a reasonable computational time. Furthermore, greater cost savings were observed for the cases with high demand and low berthing capacity at the terminal. A comprehensive analysis of the convergence patterns indicated that the introduction of the custom mutation operator with a deterministic control for the mutation rate value would provide more efficient exploration and exploitation of the search space. Moreover, a set of additional computational experiments indicated that cost savings from the application of the proposed solution algorithm would be even greater in the case of the continuous berthing layout.

Future research may focus on the following extensions:

- apply the proposed algorithm for the realistic operational data, collected from MCTs;
- develop local search heuristics for the MA-DPC operations;
- consider adaptive and self-adaptive operators within the proposed algorithm;
- evaluate alternative convergence criteria;
- model the vessel draft requirements;
- evaluate performance the proposed solution algorithm for the alternative berthing layouts (e.g., hybrid berthing layout, indented berthing layout, channel berthing layout); and
- perform a comprehensive comparative analysis of the developed MA-DPC algorithm against other algorithms, reported in the berth scheduling literature (i.e. Tabu search, particle swarm optimization, greedy randomized adaptive search procedure and others).

References

- American Shipper (2015), "Best practices and collaboration for curbing port congestion", White paper.
- Ben Line Agencies (2016), *Malaysia: Port Congestion Report*, available at: www.benlineagencies.com
- Bentolila, D., Ziedenveber, R., Hayuth, Y. and Notteboom, T. (2016), "Off-peak truck deliveries at container terminals: the 'good night' program in Israel", *Maritime Business Review*, Vol. 1 No. 1, pp. 2-20.
- Bierwirth, C. and Meisel, F. (2010), "A survey of berth allocation and quay crane scheduling problems in container terminals", *European Journal of Operational Research*, Vol. 202 No. 3, pp. 615-627.
- Bierwirth, C. and Meisel, F. (2015), "A follow-up survey of berth allocation and quay crane scheduling problems in container terminals", *European Journal of Operational Research*, Vol. 244 No. 3, pp. 1-15.
- Carlo, H., Vis, I. and Roddbergen, K. (2013), "Seaside operations in container terminals: literature overview, trends, and research directions", *Flexible Services and Manufacturing Journal*, Vol. 27 Nos 2/3, pp. 224-262.
- Cheong, C. and Tan, K. (2008), "A multi-objective multi-colony ant algorithm for solving the berth allocation problem", *Studies in Computational Intelligence*, Vol. 116, pp. 333-350.
- Cheong, C., Tan, K., Liu, D. and Lin, C. (2008), "Multi-objective and prioritized berth allocation in container ports", *Annals of Operations Research*, Vol. 180 No. 1, pp. 63-103.

- Cordeau, J., Laporte, G., Legato, P. and Moccia, L. (2005), "Models and Tabu search heuristics for the berth allocation problem", *Transportation Science*, Vol. 39 No. 4, pp. 526-538.
- Dai, J., Lin, W., Moorthy, R. and Teo, C. (2008), "Berth allocation planning optimization in container terminal", *Supply Chain Analysis*, Springer International Publishing, Berlin, pp. 69-104.
- Dulebenets, M.A. (2012), "Highway-rail grade crossing identification and prioritizing model development", M.Sc. Thesis, University of Memphis.
- Dulebenets, M.A. (2015a), "Models and solution algorithms for improving operations in marine transportation", Dissertation, University of Memphis.
- Dulebenets, M.A. (2015b), "Bunker consumption optimization in liner shipping: a metaheuristic approach", *International Journal on Recent and Innovation Trends in Computing and Communication*, Vol. 3 No. 6, pp. 3766-3776.
- Dulebenets, M.A. (2016a), "A new simulation model for a comprehensive evaluation of yard truck deployment strategies at marine container terminals", *Open Science Journal*, Vol. 1 No. 3, pp. 1-28.
- Dulebenets, M.A. (2016b), "The vessel scheduling problem in a liner shipping route with heterogeneous vessel fleet", *International Journal of Civil Engineering*, Vol. 16 No.1, pp. 19-32.
- Dulebenets, M.A. (2016c), "Advantages and disadvantages from enforcing emission restrictions within emission control areas", *Maritime Business Review*, Vol. 1 No. 2, pp. 107-132.
- Dulebenets, M.A. (2017a), "Application of evolutionary computation for berth scheduling at marine container terminals: parameter tuning versus parameter control", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 19 No. 1, pp. 25-37.
- Dulebenets, M.A. and Ozguven, E.E. (2017), "Vessel scheduling in liner shipping: modeling transport of perishable assets", *International Journal of Production Economics*, Vol. 184, pp. 141-156.
- Dulebenets, M.A., Ozguven, E.E., Moses, R. and Ulak, M.B. (2016), "Intermodal freight network design for transport of perishable products", *Open Journal of Optimization*, Vol. 05 No. 4, pp. 120-139.
- Dulebenets, M.A., Moses, R., Ozguven, E.E. and Vanli, A. (2017), "Minimizing carbon dioxide emissions due to container handling at marine container terminals via hybrid evolutionary algorithms", *IEEE Access*, Vol. 5, pp. 8131-8147.
- Eiben, A.E. and Smith, J.E. (2003), *Introduction to Evolutionary Computing*, Springer International Publishing, Berlin.
- Emde, S. and Boysen, N. (2016), "Berth allocation in container terminals that service feeder ships and deep-sea vessels", *Journal of the Operational Research Society*, Vol. 67 No. 4, pp. 551-563.
- Frojan, P., Correcher, J., Alvarez-Valdes, R., Koulouris, G. and Tamarit, J.M. (2015), "The continuous berth allocation problem in a container terminal with multiple quays", *Expert Systems with Applications*, Vol. 42 No. 21, pp. 7356-7366.
- Golias, M., Boile, M. and Theofanis, S. (2009), "Berth scheduling by customer service differentiation: a multi-objective approach", *Transportation Research Part E*, Vol. 45 No. 6, pp. 878-892.
- Golias, M., Boile, M. and Theofanis, S. (2010a), "A lambda-optimal based heuristic for the berth scheduling problem", *Transportation Research Part C*, Vol. 18 No. 5, pp. 794-806.
- Golias, M., Boile, M., Theofanis, S. and Efstathiou, C. (2010b), "The berth scheduling problem: maximizing berth productivity and minimizing fuel consumption and emissions production", *Transportation Research Record*, Vol. 2166, pp. 20-27.
- Golias, M., Portal, M.I., Konur, D., Kaisar, E. and Kolomvos, G. (2014), "Robust berth scheduling at marine container terminals via hierarchical optimization", *Computers & Operations Research*, Vol. 41, pp. 412-422.
- Han, X., Gong, X. and Jo, J. (2015), "A new continuous berth allocation and quay crane assignment model in container terminal", *Computers & Industrial Engineering*, Vol. 89, pp. 15-22.

- Hansen, P., Oguz, C. and Mladenovic, N. (2008), "Variable neighborhood search for minimum cost berth allocation", *European Journal of Operational Research*, Vol. 191 No. 3, pp. 636-649.
- Hu, Z. (2015a), "Heuristics for solving continuous berth allocation problem considering periodic balancing utilization of cranes", *Computers & Industrial Engineering*, Vol. 85, pp. 216-226.
- Hu, Z. (2015b), "Multi-objective genetic algorithm for berth allocation problem considering daytime preference", *Computers & Industrial Engineering*, Vol. 89, pp. 2-14.
- Imai, A., Nishimura, E. and Paradimitriou, S. (2008), "Berthing ships at a multi-user container terminal with a limited quay capacity", *Transportation Research Part E*, Vol. 44 No. 1, pp. 136-151.
- Imai, A., Nishimura, E. and Paradimitriou, S. (2013), "Marine container terminal configurations for efficient handling of mega-containerships", *Transportation Research Part E*, Vol. 49 No. 1, pp. 141-158.
- Imai, A., Sun, X., Nishimura, E. and Paradimitriou, S. (2007a), "Berth allocation at indented berths for mega-containerships", *European Journal of Operational Research*, Vol. 179 No. 2, pp. 579-593.
- Imai, A., Zhang, J., Nishimura, E. and Paradimitriou, S. (2007b), "The berth allocation problem with service time and delay time objectives", *Maritime Economics & Logistics*, Vol. 9 No. 4, pp. 269-290.
- Journal of Commerce (2015), *LA throughput Grows in May, Clouding Impact of Diversions*, available at: www.joc.com
- Kim, K. and Moon, K. (2003), "Berth scheduling by simulated annealing", *Transportation Research Part B*, Vol. 37 No. 6, pp. 541-560.
- Lalla-Ruiz, E., Melian-Batista, B. and Moreno-Vega, J. (2012), "Artificial intelligence hybrid heuristic based on Tabu search for the dynamic berth allocation problem", *Engineering Applications of Artificial Intelligence*, Vol. 25 No. 6, pp. 1132-1141.
- Lee, D., Chen, J. and Cao, J. (2010), "The continuous berth allocation problem: a greedy randomized adaptive search solution", *Transportation Research Part E*, Vol. 46 No. 6, pp. 1017-1029.
- Lee, Y. and Chen, C. (2009), "An optimization heuristic for the berth scheduling problem", *European Journal of Operational Research*, Vol. 196 No. 2, pp. 500-508.
- Legato, P., Mazza, R. and Gulli, D. (2014), "Integrating tactical and operational berth allocation decisions via simulation-optimization", *Computers & Industrial Engineering*, Vol. 78, pp. 84-94.
- Lu, C., Shang, K., Lin, C. and Lin, C. (2016), "Identifying crucial sustainability assessment criteria for container seaports", *Maritime Business Review*, Vol. 1 No. 2, pp. 90-106.
- Mauri, G., Ribeiro, G., Lorena, L. and Laporte, G. (2016), "An adaptive large neighborhood search for the discrete and continuous berth allocation problem", *Computers & Operations Research*, Vol. 70, pp. 140-154.
- Moorthy, R. and Teo, C. (2006), "Berth management in container terminal: the template design problem", *Or Spectrum*, Vol. 28 No. 4, pp. 495-518.
- Nishimura, E., Imai, A. and Paradimitriou, S. (2001), "Berth allocation planning in the public berth system by genetic algorithms", *European Journal of Operational Research*, Vol. 131 No. 2, pp. 282-292.
- Pinedo, M. (2008), *Scheduling: Theory, Algorithms, and Systems*, Springer International Publishing, Berlin.
- Port Finance (2014), *Port Congestion Looms Large Again*, available at: <http://portfinanceinternational.com>
- Port Technology (2015), *Containerships: Then and Now*, available at: www.porttechnology.org
- Rahman, N., Ismail, A. and Lun, V. (2016), "Preliminary study on new container stacking/storage system due to space limitations in container yard", *Maritime Business Review*, Vol. 1 No. 1, pp. 21-39.
- Sahinidis, N. (2017), *BARON user manual v. 17.8.9*, available at: www.minlp.com/downloads/docs/baron%20manual.pdf
- Sivanandam, S. and Deepa, S. (2008), *Introduction to Genetic Algorithms*, Springer International Publishing, Berlin.

-
- The Port Authority of New York and New Jersey (2016), *Marine Terminal Tariffs*, available at: www.panynj.gov/port/tariffs.html
- Ting, C., Wu, K. and Chou, H. (2014), "Particle swarm optimization algorithm for the berth allocation problem", *Expert Systems with Applications*, Vol. 41 No. 4, pp. 1543-1550.
- UNCTAD (2015), *Review of Maritime Transport 2015*, United Nations Conference on Trade and Development, New York, NY and Geneva.
- Wang, F. and Lim, A. (2007), "A stochastic beam search for the berth allocation problem", *Decision Support Systems*, Vol. 42 No. 4, pp. 2186-2196.
- World Shipping Council (2015), *Some Observations on Port Congestion, Vessel Size and Vessel Sharing Agreements*, available at: www.worldshipping.org
- Xu, Y., Chen, Q. and Quan, X. (2011), "Robust berth scheduling with uncertain vessel delay and handling time", *Annals of Operations Research*, Vol. 192 No. 1, pp. 123-140.
- Zampelli, S., Vergados, Y., Schaeren, R., Dullaert, W. and Raa, B. (2014), "The berth allocation and quay crane assignment problem using a CP approach", *Principles and Practice of Constraint Programming*, Springer Berlin Heidelberg, Berlin, pp. 880-896.
- Zhen, L. and Chang, D. (2012), "A bi-objective model for robust berth allocation scheduling", *Computers & Industrial Engineering*, Vol. 63 No. 1, pp. 262-273.
- Zhou, P., Wang, K., Kang, H. and Jia, J. (2006), "Study on berth allocation problem with variable service priority in multi-user container terminal", *Proceedings of the Sixteenth International Offshore and Polar Engineering Conference*, San Francisco, CA, pp. 684-688.

Corresponding author

Maxim A. Dulebenets can be contacted at: mdulebenets@eng.famu.fsu.edu