Applied Intelligence (2022) 52:1406–1434
https://doi.org/10.1007/s10489-021-02420-4

The impact of time windows constraints on metaheuristics implementation: a study for the Discrete and Dynamic Berth Allocation Problem

Flávia Barbosa[1] · Priscila C. Berbert Rampazzo[2] · Anibal Tavares de Azevedo[2] · Akebo Yamakami[3]

Accepted: 6 March 2021 / Published online: 21 May 2021

Abstract

This paper describes the development of a mechanism to deal with time windows constraints. To the best of our knowledge, the time windows constraints are difficult to be fulfilled even for state-of-the-art methods. Therefore, the main contribution of this paper is to propose a new computational technique to deal with such constraints. Such technique was tested combined with two metaheuristics to solve the discrete and dynamic Berth Allocation Problem. The technique ensures obtaining feasible solutions in terms of vessels time windows constraints, which are treated as hard constraints. A data set generator was created, resulting in a diversity of problems in terms of time windows constraints. A detailed computational analysis was carried out to compare the performance of both metaheuristics considering the technique.

Keywords Berth allocation · Metaheuristics · Time windows constraints · Data generator

1 Introduction

According to the United Nations Conference on Trade and Development [29], maritime transport carries over 80% of global trade merchandise. However, the growth rate of seaborne shipping is the lowest it has been since 2009. The carrying capacity, on the other hand, has increased by 3.5% to 1.8 billion deadweight tons. Both movements have led to an increase in available capacity and a drop in freight rates. Freight rates should, however, decrease to attract volume at an increasing rate. It only is achieved if the ports reduce their operating costs without making significant investments.

The Berth Allocation Problem (BAP) is the first of a series of planning problems that need to be addressed by port operators. Any improvement in the quality of solutions for BAP will significantly impact the models of subsequent operations at the port (e.g. quay crane allocation and scheduling in quayside, equipment management operations). In this context, berths are an essential resource, and proper allocation of vessels to berths entails a reduction in handling costs. Therefore, the BAP comprises to determine where and when to allocate arriving vessels to a berth space over a planning horizon, taking into account constraints of time and space related to vessel length, arrival times, the number and position of quay cranes, charge stock location and windows of time, among others.

The main contribution of this paper is to propose a new computational technique to deal with time windows constraints. The majority of methods either take too much time to find a feasible solution or fail to find a feasible solution. Thereby, a genetic algorithm (GA) and a particle swarm algorithm (PSO) were implemented using the same technique and compare their performance.

This paper considers a discrete berth layout. In this case, the quay is viewed as a finite set of berths where each berth is described by fixed-length segments or points, as illustrated in Fig. 1.

Additionaly, the vessels are considered to arrive dynamically, turning the problem more difficult since the solution should respect a specific time window.

The original online version of this article was revised: the manuscript materials used was incorrect.

✉ Flávia Barbosa
flavia@fe.up.pt

1 Faculdade de Engenharia da Universidade do Porto, Porto, Portugal

2 Faculdade de Ciências Aplicadas, Universidade Estadual de Campinas, Limeira, Brazil

3 Faculdade de Engenharia Elétrica e Computação, Universidade Estadual de Campinas, Campinas, Brazil

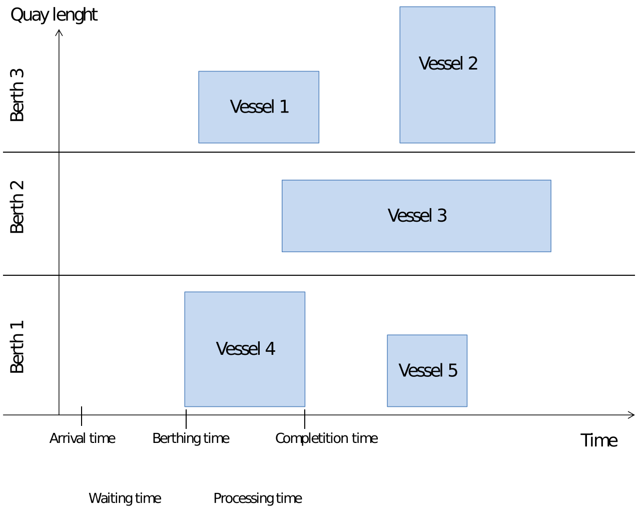

Fig. 1 BAP representation for discrete berths

Although researchers devoted a considerable effort to improve the discrete BAP models and methods, the instances available on literature do not emphasize the time windows constraints. Moreover, there are no standard benchmarks to verify performances focusing on instances with time windows constraints hard to attend. In this context, another contribution of this paper is the proposition of a data set generator employing a variety of statistical functions. It results in a diversity of problems, including instances with tighter time windows, which are harder to attend. The variety of instances allows to verify the robustness of the algorithm, especially if the mechanism is adequate to deal with the time window constraints. Hypothesis testing was used to compare the performance of both metaheuristics. It is not usual to find in literature papers that employ statistical tests to compare different methods, misleading the analysis.

This paper is organized as follows. Section 2 presents a literature review of the BAP, in which models are classified according to the characteristics of berthing space, vessel arrivals, service time, integration with other problems, methods of resolution, data used and comparisons regarding the classical combinatorial models. Section 3 describes the mathematical formulation considered for the BAP; Section 4 develops a Genetic algorithm with floating-point representation; Section 5 reports the results from the computational tests.

2 Literature review on the Berth Allocation Problem

The variety of characteristics and goals found in the real world led to a multitude of approaches for the BAP through literature. In the following, the most relevant and recent features on the BAP will be reviewed and classified.

First, the berthing space can be considered discrete or continuous. In the discrete BAP, the quay is a finite set of berths, and only one vessel can be assigned to each berth at any given moment in time. According to [6], the problem of allocating arriving vessels to discrete berth locations at container terminals is the most critical processes for any container terminal. Correcher et al. [9] considered a discrete layout to address the general case of terminals with irregular layouts involving adjacency, oppositional and blocking restrictions between berths. In the continuous Berth Allocation Problem, vessels can berth anywhere along the quay depending only on other vessels' position. Frojan et al. [14] extended the study of the continuous Berth Allocation Problem to the case of multiple quays. It adds to the problem the decision of assigning vessels to quays. The continuous problem is represented in a single space-time diagram; the problem with multiple quays requires a multiple space–time representation. Lin et al. [21] considered a continuous layout where each vessel has its preferred berthing position because the containers to be loaded might have already arrived at the yard.

Another essential characteristic of BAP concerns the vessel arrivals. The problem is classified as static (Static Berth Allocation Problem - SBAP) if all scheduled vessels are already in the port when scheduling begins. Buhrkal et al. [6]. Alternatively, the problem is classified as dynamic (Dynamic Berth Allocation Problem - DBAP) if not all vessels have arrived at the beginning of the planning horizon, even though the arrival times are known in advance. Jos et al. [18] proposed a model for the minimum cost berth allocation problem at a container terminal where the maritime vessels arrive dynamically. According to [24], the DBAP requires a reasonably fast optimization algorithm, and for larger instances, it requires considerable computational time and memory. The vessels arrivals can also be affected by the phenomenon of the tides. Tides are the rise and fall of sea levels caused by the combined effects of the gravitational forces exerted by the Moon and the Sun and the rotation of Earth. The port and vessel operators must have local knowledge of the tides, wind conditions, depths, and aids to procedure berthing. Due to different tide levels at different hours of the day, whether a vessel can be assigned to a berth is a lso dependent on the time when the mooring takes place. Modelling the tidal impacts requires retrofitting the berthing time requirements and departure time definitions under the tidal effects on incoming and outgoing navigations. Tides might influence vessels' ability to sail through the navigation channel when entering or leaving the port [11] measured the impacts of tides on the seaside operations of a container port with a mixed-integer linear programming model. Barros et al. [5] proposed a problem to allocate berth positions for vessels in tidal bulk port terminals. In tidal ports, draft conditions depend on high tide conditions. The available depth at low tide is not adequate for the movement of vessels. Both factors - dynamic arrivals and tide effects - imply time window constraints, which significantly increase the difficulty of solving BAP.

Servicing a vessel should never take a long time, as this represents an immobilization cost for the client and an opportunity cost for the port. The duration of a vessel (un)loading depends on the number of quay cranes allocated to the vessel. As the number of quay cranes increases, the duration of vessel berthing decreases. Therefore, a vessel's handling time may be different for different berths, regardless of the number of cranes to assign to each vessel [6]. Hence, the handling time can also be classified as static or dynamic. In the static case, the number of cranes that will serve each vessel is fixed. In the dynamic case, the number of cranes is variable and decided together with the berth position and service time. It results in the integrated Quay Crane Allocation Problem (QCAP) and Quay Crane Scheduling Problem (QCSP). Meisel and Bierwirth [22] provided a framework for aligning all decisions regarding the productivity rates for the cranes from the vessels' stowage plans, the berthing decisions and the assignment of cranes to vessels, and the crane scheduling. According to [25], in real-life applications, the handling time depends on the number of containers to be handled and the number of cranes deployed. Agra and Oliveira [1] integrated the BAP and the quay crane assignment and scheduling problem, where cranes are assigned to vessels, and their operations are scheduled. Moreover, the other two scheduling problems (besides the berth allocation) could be related to port operators: waterway scheduling, and quay crane assignment problem. Tavakkoli-Moghaddam et al. [27] integrated the three problems using a hybrid flow shop problem with unrelated parallel machines. A mixed-integer programming model was proposed and solved to evaluate the model performance for a small-size problem instance. Hsu and Wang [17] focused on allocating berth and quay cranes to vessels, with dynamic and continuous BAP. The first stage generates alternative vessel placement sequences with *first-come-first-served* heuristic (FCFS), particle swarm optimization, improved PSO (PSO2), and multiple PSO (MPSO). The second stage works with an event-based heuristic to place vessels, deal with overlaps of vessels, and assign quay cranes to develop a feasible solution. The MPSO, combined with the event-based heuristic, leads to a better result. The most recent study, [7] integrated berth and quay crane allocation, proposing a dynamical modelling framework based on discrete-event systems and an algorithm for solving the I-BCAP using the model predictive control (MPC) principle with a rolling event horizon. The objective functions considered are total handling and waiting costs. The proposed method outperforms the hybrid particle swarm optimization based [16] and the genetic algorithm-based method proposed in the recent literature. A Hybrid particle swarm optimization was proposed combining an improved PSO with an event-based heuristic for solving dynamic and discrete berth allocation problem and dynamic quay crane assignment problem simultaneously. The work reported that the performance of HPSO-based solution is comparable to the GA-based one. [20] studied the tactical berth allocation problem (TBAP). It consists in determining the berthing position, berthing time and allocation of quay cranes for vessels arriving at the port over a time horizon. They proposed a biased random key genetic algorithm (BRKGA). Correcher and Alvarez-Valdes [8] also proposed a biased random-key genetic algorithm (BRKGA) for solving the continuous berth allocation and quay crane assignment problem (BACAP). Our methodology will be useful when the BAP is solved in the first stage, and the solution is used as a starting point for any other problems mentioned above.

Several models have been developed for the BAP based on other problems in the literature. The BAP has a combinatorial nature. Therefore, most studies develop heuristics and metaheuristics to tackle the problem. In this context, it is well known that the data strongly influence heuristics and metaheuristics. However, most papers use randomly generated data to test their methods. A few approaches addressing problems found in real-world ports have been proposed in the literature, therefore using real port-based data. The proposition of a data set generator that results in a diversity of problems allows verifying the robustness of the algorithm, primarily if it deals with the time window constraints.

This paper considers the discrete BAP, with dynamic arrival of vessels and processing times fixed and dependent on the berth. The paper contributions for the literature are the following: (i) a two-stage search process with a penalization where optimal solutions with some degree of violation in time windows constraints are not allowed; (ii) two optimization methods incorporating such penalization mechanism; (iii) a BAP library with a wide range of instances in terms of difficulty to fulfil time windows constraints; (iv) and a proper framework to compare the methods based on statistical hypothesis testing.

3 A mathematical model for the Berth Allocation Problem

Consider the BAP formulated as a Heterogeneous Routing Problem with Time Windows (HVRPTW) [6]. Let V be the set of vertices, containing a vertex for each vessel as well as vertices o and d which mark the origin and destination nodes for any route in the graph. The set of arcs is a subset of $V \times V$. Let N be the set of vessels and M the set of berths. Each vessel $i \in N$ has an arrival time a_i, an expected departure time from the port b_i (which implies a time window $[a_i, b_i]$ for vessel i), processing times p_i^k that are dependent on the respective berth $k \in M$ locations and a relative importance v_i. For the origin (o) and destination (d) vertices, the time window depends on the berth k as berths can be available at different times $[s^k, e^k]$). Each binary decision variable l_{ij}^k, $k \in M$, $(i, j) \in A$, takes the value one if vessel j immediately succeeds vessel i at berth k and is zero otherwise. Each continuous variables x_i^k, $i \in V$, $k \in M$, gives the time that vessel i starts being serviced at berth k (if vessel i does not use berth k, $x_i^k = a_i$). The variables x_o^k and x_d^k define the start and end time of activities at berth $k \in M$ respectively. The problem is formulated as follow:

$$\min \sum_{i \in N} \sum_{k \in M} v_i \left((x_i^k - a_i) + p_i^k \sum_{j \in N \cup \{d\}} l_{ij}^k \right) \quad (1)$$

$$\text{s.t.} \sum_{k \in M} \sum_{j \in N \cup \{d\}} l_{ij}^k = 1 \;\; \forall i \in N \quad (2)$$

$$\sum_{j \in N \cup \{d\}} l_{oj}^k = 1 \;\; \forall k \in M \quad (3)$$

$$\sum_{j \in N \cup \{o\}} l_{id}^k = 1 \;\; \forall k \in M \quad (4)$$

$$\sum_{j \in N \cup \{d\}} l_{ij}^k = \sum_{j \in N \cup \{o\}} l_{ji}^k \;\; \forall k \in M, i \in N \quad (5)$$

$$x_i^k + p_i^k - x_j^k \leq (1 - l_{ij}^k) M_{ij}^k \;\; \forall k \in M, (i, j) \in A \quad (6)$$

$$a_i \leq x_i^k \;\; \forall k \in M, i \in N \quad (7)$$

$$x_i^k + p_i^k \sum_{j \in N \cup \{d\}} l_{ij}^k \leq b_i \;\; \forall k \in M, i \in N \quad (8)$$

$$s^k \leq x_o^k \;\; \forall k \in M \quad (9)$$

$$x_d^k \leq e^k \;\; \forall k \in M \quad (10)$$

$$l_{ij}^k \in (0, 1) \;\; \forall k \in M, (i, j) \in A \quad (11)$$

$$x_i^k \geq 0 \;\; \forall k \in M, i \in N \quad (12)$$

The objective function (1) minimizes the weighted sum of vessel service times (waiting time plus handling time). Constraint set (2) states that each vessel must be assigned to exactly one berth k. Constraints set (3) guarantees that for each berth k the degree of origin is one. Constraints set (4) guarantees that for each berth k the degree of destination nodes is one. Constraints set (5) ensures flow conservation for the remaining vertices. Constraints set (6) guarantees consistency for berthing time and mooring sequence on each berth, and $M_{ij}^k = \max\{b_i + h_i^k - a_j, 0\}$. Constraints sets (7) and (8) guarantee the time window requirements for each vessel. Constraints sets (9) and (10) guarantee the berth availability time windows. Although the literature reports the difficult to deal with time window constraints (7), (8), (9) and (10), there are no details about proper mechanisms that could help in attend them. Finally, constraint sets (11) and (12) define the domains of the decision variables.

4 Metaheuristics

According to [7], the particle swarm optimization and the genetic algorithm are the state-of-the-art methods proposed in the recent literature to solve the berth and quay crane allocation problems. It is noteworthy that, although this paper solves the integrated problem, it does not mention how occasional time windows could be treated. Therefore, both state-of-the-art methods (GA and PSO) are adapted to solve the dynamic and discrete BAP. Although this paper only considers the BAP, the model has the feature of considering time windows as hard constraints. Both algorithms were implemented with the same structures of individual codification; population initialization and idleness treatment; evaluation; and local search.

The Genetic Algorithm is a metaheuristic based on the evolutionary idea of natural selection and genetics. The pioneering work of J. H. Holland in the 1970s, with the publication of his book *Adaptation in Natural and Artificial Systems*, consolidated the contribution of Genetic Algorithms to operations research [15]. Easily adaptable, GA became a classical method for solving combinatorial problems. On the other hand, the PSO has fewer parameters to adjust. The PSO does not label its operations in the same way as GAs, but analogies exist depending on the implementation of the GA operation [13]. The effect of selection in a GA is supporting the fittest survival, a concept central to all evolutionary algorithms. PSO does not utilize selection once all particles continue as members of the population for the duration of the run. A particle does not explicitly exchange material with other particles, but they influence its trajectory. In PSO, each particle is stochastically accelerated toward its own previous best position, as well as toward the global best position or the local best position. These particles seem to be exploring a region that represents the geometric mean between two promising regions. Mutation allows a GA chromosome to reach any point in the problem space, particularly near the end of a run, because several mutations may be needed to reach a distant point. It may be that a PSO particle cannot reach any point in problem space in one iteration, although this might be possible at the beginning of the run.

4.1 Individual codification

Consider a floating-point coding [28] to represent the solutions, as shown in Fig. 2. A real number between (0;m) is assigned to each vessel. The integer part is the berth number to which the vessel is allocated, and the fractional part is used to sort the vessels allocated to each berth. Let $q_k = [q_k(1); q_k(2); ...; q_k(n)]$ represent an individual k.

4.2 Population initialization

All individuals were randomly generated except one that was generated based on the *first-come-first-served* heuristic (FCFS) [28] (the vessels were sorted by their arrival times in ascending order and assigned one at a time by choosing the combination of berth and vessel that will finish first).

4.3 Population evaluation and idleness treatment

Next to initialization, each individual must be evaluated. Usually, the fitness value incorporates all the aspects present in the objective function (1); moreover, our fitness value will also carry information about the infeasibility of solutions.

During the initialization of solutions, (7) and (9) will always be met. However, constraints (8) and (10) need special treatment, so that time windows are not violated [3].

The infeasible solutions were penalized in the fitness as follows. If a vessel i allocated in berth k violates the maximum departure time, i.e. $x_i^k + p_i^k > b_i$, then to the fitness is added the penalty

$$\sum_{i,k \in I} 1000 * (x_i^k + p_i^k - b_i) \tag{13}$$

Kavoosi et al. [19] is the most recent work that addressed the dynamical and discrete BAP by developing universal island-based metaheuristic algorithm (UIMA). The solutions with the integer-coded representation are mapped to the solutions with the real-coded representation, as proposed by [28]. The objective function considers costs of handling, waiting and late departure time. Each solution is checked by a *repairing operator* in case of infeasibility, but this treatment does not guarantee that time windows will not be violated. Optimal solutions with late departure are not allowed: the idleness treatment has the advantage of ensuring all vessels will be served within their time windows.

The search was not limited to the feasible region. The penalization divided the search process into two stages. First, a factibilization process was carried out. After all, individuals were feasible, that is, respecting their time windows, the search process continues with the optimization. The method presented in [19] can provide solutions infeasible in terms of time window constraints since the repairing operator allows late departure. This way, [19] considers time windows as soft constraints, and he does not indicate if the time windows were attended, or what was the degree of violation. Our method forces time windows to be hard constraints; that is, the optimal (or sub-optimal) solution is feasible to meet the time windows constraints.

In Table 1 and Fig. 3 the treatment of infeasible solutions for a small problem with 5 vessels and 2 berths is illustrated:

Vessel 2 violates the time window in 10 units, so the objective function is penalized by 1000 * 10; vessel 4 violates the time window in 5 units, so the objective function is penalized by 1000 * 5.

4.4 Local search

A local search procedure [28] was incorporated into the GA algorithm to improve the solution quality. This technique can be implemented in two different ways. The vessels can be swapped in the same berth, by comparing all the possible swapping pairs within the same berth and select the best improvement to exchange their fitness values; and between

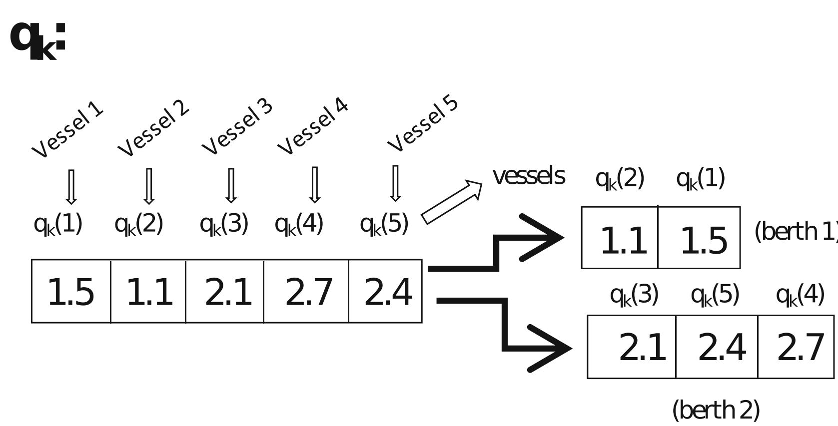

Fig. 2 Solution representation

Table 1 Illustrating the treatment of infeasible solutions

Vessel (i)	Processing time Berth 1 (p_i^1)	Processing time Berth 2 (p_i^2)	Arrival time (a_i)	Maximum departure time (b_i)	Start time of service (x_i^k)	Berth (k)	Processing time (p_i^k)	Waiting time time $(x_i^k - a_i)$	Real departure is violated $(x_i^k + p_i^k)$	How much the TW $(\lvert x_i^k + p_i^b - b_i \rvert)$	Objective function (1) and (13)
1	35	40	5	95	10	2	40	5	50	–	45
2	30	100	0	65	45	1	30	45	75	10	**10075**
3	25	20	5	55	5	1	25	0	30	–	25
4	35	30	5	85	60	2	30	55	90	5	**5085**
5	40	20	20	55	35	2	20	15	55	–	35

berths, by selecting randomly two vessels in two different berths and selecting the best improvement to exchange.

4.5 Genetic algorithm

The GA implementation begins by codifying the solution to generate a random population of individuals ($g = 0, P_g =$ initialization, and an idleness treatment was applied to attempt to reduce the idleness of each berth. Next, the structures are evaluated, and a fitness value is assigned to them. The fitness includes the objective function and penalization for infeasibilities. First, the algorithm aims to decrease the value of fitness seeking from one generation to another to reduce the amount of infeasibility in the solutions. Then, if all individuals are feasible, the fitness includes only of the objective function. Selection is then applied to the population to create an intermediate population. The binary tournament [2] was used to build a population of parents. Pairs of parents are selected to generate pairs of offspring through crossover and mutation.

Crossover and mutation are complementary operators. The crossover was implemented as the following: for each pair of parents, a random number rm is generated in the range [0.3, 0.7]. $rm\%$ from the first parent is copied to the new individual, and $(1 - rm)\%$ from the second parent is copied to the new individual, giving rise to the first offspring. The complementary portions of both parents form the second offspring. The Non-Uniform Mutation was used, which is more suitable for problems with constraints and floating-point coding, according to [23]. It is a dynamic operator designed to improve the search process. The new element resulting from the mutation is, with 50% probability each:

$$q_k'(i) = \begin{cases} q_k(i) + \Delta(g, ub - q_k(i)) & \text{or} \\ q_k(i) - \Delta(g, q_k(i) - lb) & \text{otherwise.} \end{cases} \tag{14}$$

where $[lb, ub]$ is $[0, m]$ for the BAP. Function $\Delta(g, y)$ in (15) [23] returns a value in the range $[0, y]$ such that the probability of $\Delta(g, y)$ being close to zero increases as g

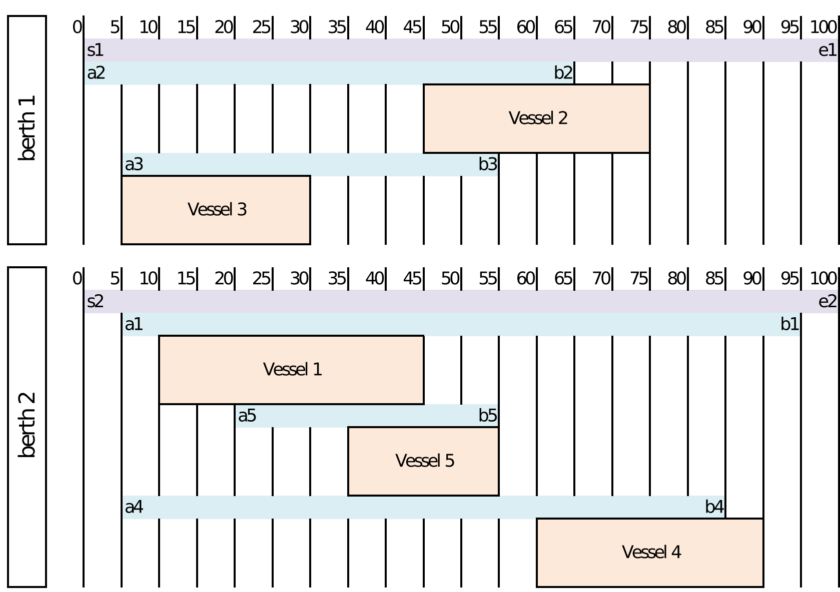

Fig. 3 Illustrating the treatment of infeasible solutions

increases ($g = 1, ..., G$). The operator explores first the search space extensively in the initial iterations, and locally in advanced iterations.

$$\Delta(g, y) = y.(1 - r^{(\frac{1-g}{G})^b}) \quad (15)$$

where r is a random number from [0,1], G is the maximal iteration number, and b is a system parameter determining the degree of dependency on iteration number (empirically, $b = 3$).

Consider as *one iteration* the process of going from the current population to the next population.

4.6 Particle swarm algorithm

The Particle Swarm Optimization simulates the movement of organisms in a bird flock [12]. It has been used across a wide range of applications because there are few parameters to adjust. For an optimization problem of n variables, a swarm has N_P particles [10]. Each particle has its trajectory, position x_i and velocity v_i, and moves in the search space by successively updating its trajectory. All particles have fitness values, and flown through the solution space by following the current optimum particles. The algorithm initializes a group of particles with random positions and then searches for optima by updating iterations. In every iteration, each particle is updated by the particle best *pbest*, denoted x_i^* , $i = 1, ..., N_P$, which is the best solution it has achieved so far. The global best *gbest*, denoted x^g, is also updated, which is the best value obtained so far by any particle in the population.

Because all particles in the swarm learn from *gbest* even if *gbest* is far from the global optimum, particles may easily be attracted to the *gbest* region and get trapped in a local optimum for multimodal problems. In case the *gbest* positions located on the local minimum, other particles in the swarm may also be trapped. If an early solution is suboptimal, the swarm can quickly stagnate around it without any pressure to continue exploration.

PSO can locate the region of the optimum faster than others. However, once in this region, it progresses slowly due to the fixed velocity stepsize. Linearly decreasing weight PSO effectively balances the global and local search abilities of the swarm by introducing a linearly decreasing inertia weight on the previous velocity of the particle into

$$\begin{aligned} v_i(t+1) = &\ W v_i(t) + c_1 r_1 [x_i^*(t) - x_i(t)] \\ &+ c_2 r_2 [x^g(t) - x_i(t)] \end{aligned} \quad (16)$$

where W is called the inertia weight, and the positive constants c_1 and c_2 are, respectively, cognitive and social parameters.

At iteration $t + 1$, the swarm can be updated by

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (17)$$

In Fig. 4, the flowchart summarizes the GA and the PSO for comparisons.

5 Computational tests

Through literature, papers use instances where vessels have time windows with the same size [6]. These are not suitable

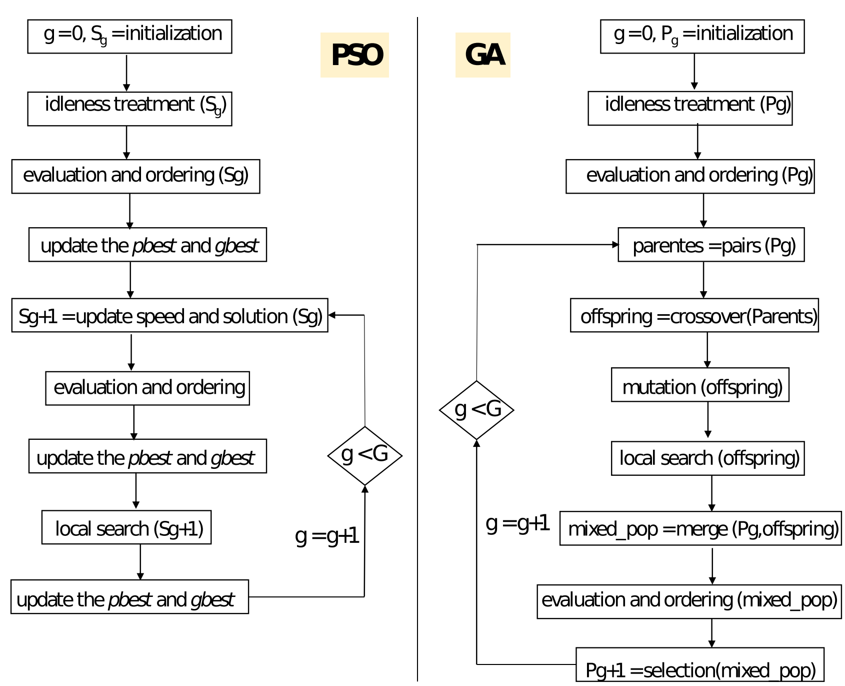

Fig. 4 Comparison of both algorithms structures

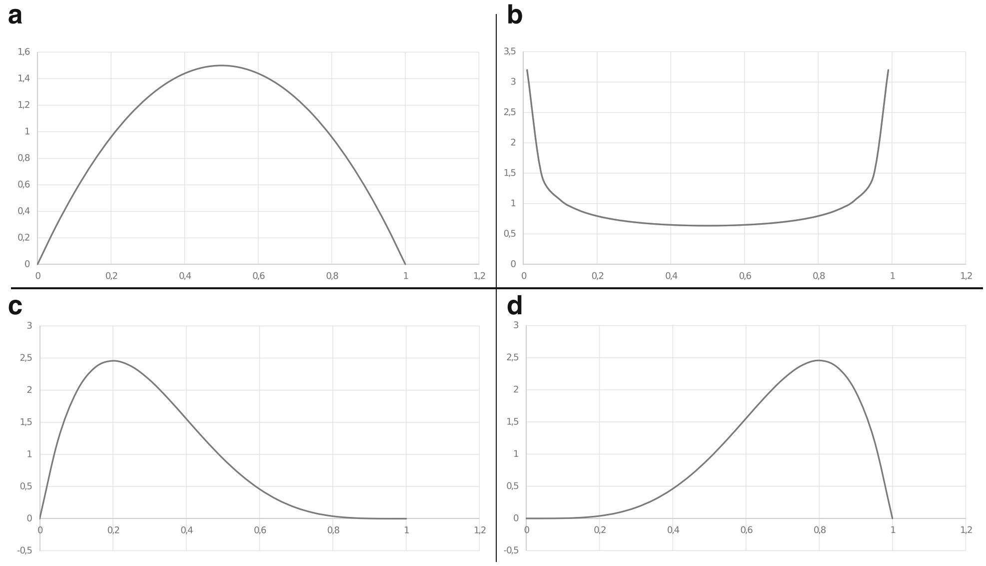

Fig. 5 Beta distribution shapes

for achieving the primary goal of this paper. To conduct computational testing using a wide variety of problem sizes (vessels and berths) and to avoid bias in algorithms that had been specially tuned to one set of parameters, a data generator was developed with Beta Distribution due to its extreme versatility [26]. It consists of a continuous probability distribution with a probability density function given by:

$$f(x;\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 \mu^{\alpha-1}(1-\mu)^{\beta-1}\partial\mu}, \quad 0 \le x \le 1, \alpha > 0, \beta > 0 \tag{18}$$

The inverse transform sampling technique was used for generating random numbers from the beta distribution, and a uniform pseudo-random generator was implemented to ensure the portability and reproducibility of the test data.

The parameters number of vessels (n) and number of berths (m) are user's choice. The berth time window is fixed for every berth and normalized: $s^k = 0$ and e^k is an input. For generating processing times, arrival times and departure times the user has to select one of four types of beta distributions considered in Fig. 5: $\alpha = 2$ and $\beta = 2$ (5a), $\alpha = 0.5$ and $\beta = 0.5$ (5b), $\alpha = 2$ and $\beta = 5$ (5c) and $\alpha = 5$ and $\beta = 2$ (5d).

For the generation of processing time values, first, a reference processing time p_ref_i is generated for each vessel in the interval $[1, \frac{m}{n}e^k]$.

If a berth does not possess the right equipment to (un)load a vessel, then the vessel cannot be moored at that berth. In some cases, there are draft or depth restrictions that forbid a vessel to be moored at a given berth. For these, consider that, with a given probability ($prob$) - defined as an input, the vessel will not be able to be processed in a berth. A random number $rand$ in (0, 1) is generated. If $rand \le prob$, then the processing time of vessel i at berth k is set to $2e^k$, i.e., vessel i cannot be moored ate berth k. Otherwise, p_i^k is generated using the beta distribution $f(x, 5, 2)$ in the interval $[\frac{1}{2}p\_ref_i, p\_ref_i]$.

The arrival time a_i values are generated in the interval $[0, e^k - \max_k p_i^k]$ and the departure time values are generated in the interval $[a_i + \max_k p_i^k, e^k]$.

For the relative importance v_i of each vessel i, the user must provide an interval between 0.5 and 1. The parameter is then sampled from the uniform distribution, which is the beta distribution $f(x, 1, 1)$. Table 2 summarizes the intervals defined for the parameters generation.

The difficulty of solving BAP varies according to the way each parameter was generated; the time windows can be very tight or loose. By generating the parameters in different ways, to obtain different classes of problems, it is possible to make a stronger evaluation of the methodology.

5.1 Evaluating the computational difficulty of each instance

The purpose of the next subsections is to tackle two aspects that are missing in literature: a robust comparison among performances of different methods in terms of difficulty to accomplish the berth window constraint. To

Table 2 Data summary

	min	max
p_ref	1	$\frac{m}{n}e^k$
p	$\frac{1}{2}p\_ref$	p_ref
a	0	$e^k - \max_k p_i^k$
b	$a_i + \max_k p_i^k$	e^k

achieve such robust comparison, we considered two aspects: (i) a properly variation of parameters as described in Figure 5; (ii)a statistical approach to compare the methods' performance for each set of parameters. It reinforces a proper evaluation of methods performances by avoiding usage of a limited set of instances. This common practice typically results in a misleading and biased analysis of the method's performance.

First, it is necessary to evaluate the influence of the data on the difficulty of solving the problem with CPLEX. Considered four shapes for the beta distribution: $f(x, 2, 2)$, $f(x, 0.5, 0.5)$, $f(x, 2, 5)$ and $f(x, 5, 2)$ (Fig. 5). Because the model has 3 parameters (p, a and b), there are 4^3 different combinations for generating the set of problem parameters. Considered $e^k = 100\ \forall k$ and $v_i = 1\ \forall i$. For each combination, 10 instances were generated and solved for all combinations of 20, 30 and 40 vessels with 5, 7 and 10 berths. It totalizes 5760 computational tests solved with CPLEX 12.6. The stop criterion was computational time, limited to 3600 seconds (one hour). In Appendix A the detailed results are presented to enable the reader accessing the performance of CPLEX for each instance.

The difficulty of the parameter combinations for the beta distribution was classified as follows. First, count how many of these 10 instances were complete. It means that either solution optimality or problem infeasibility was proved. Second, organize the average computational time taken by CPLEX in non-decreasing order. Third, count on how many of the 10 instances CPLEX was able to find a feasible solution. Fourth, for the instances for which a feasible solution was found, but optimality was not reached, organize in non-decreasing order of average gap. Finally, count on how many of the 10 instances CPLEX was not able to find a feasible solution within one hour or prove the problem was infeasible (unknown status). Table 5 in the Appendix presents the results ordained following such classification.

Analyzing the results, it is possible to see that CPLEX performs better the HVRPTW model as the *vessels*/*berths* ratio decreases. This happens due to the type of valid inequalities they introduced [6]. However, the number of variables must also be considered when measuring the difficulty. For example, the instances with 20 vessels and 5 berths and the instances with 40 vessels and 10 berths have the same ratio, 4, but the first one has $20*20*5 = 2000$ binary variables, while the last one has $40*40*10 = 16000$ binary variables. Besides, if the arrival times of all vessels are concentrated at the beginning of the planning period, it is hard for CPLEX to solve. It produces congestion, which is very difficult to manage [14]. For these cases, metaheuristics help to obtain good feasible solutions for the problem in a short computational time.

Table 3 Instances chosen to test the metaheuristics

Vessels	Berths	p	a	b	Class
20	5	$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 5)$	C1
30	5	$(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 2)$	C2
30	5	$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	C3
30	5	$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	C4
30	7	$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 5)$	C1
30	7	$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 2)$	C5
30	7	$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	C4
40	5	$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	C4
40	7	$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 5)$	C1
40	7	$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	C3
40	7	$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	C4
50	7	$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	C4
50	10	$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	C4
100	7	$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	C4
150	10	$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	C4

From this analysis, Table 3 lists the instances chosen to be tackled with proposed GA. Call a *class* the combination of parameters for the BAP. For example, class C1 has the processing time p generated using the distribution $f(x, 5, 2)$, the arrival times generated using the distribution $f(x, 2, 5)$ and the departure times generated using the distribution $f(x, 2, 5)$. Besides the most difficult instances point out by CPLEX, the GA performance was also tested on instances with a much larger number of vessels: 50, 100, and 150.

5.2 Testing the performance of both algorithms

The GA and the PSO were implemented in C language. The computational tests were executed on a personal computer, a Dell Inspiron 14Z with Intel Core I5-3337U 1.80GHz, RAM of 6GB, and a Solid State Drive of size 240 GB.

For the GA, the greater the probability of crossing, the more quickly new solutions appear in the population. A low probability of mutation prevents solutions from stagnant at a value; a very high value makes the search essentially random. Empirically and considering these observations mentioned, the probability of crossover was 0.9, and the probability of mutation was 0.1.

For the PSO, a larger W (Section 4.6) can prevent particles being trapped in the local optimum. The parameter-setting used was: $W = 0.9$, $c_1 = c_2 = 2$, as suggested by [28].

In the work of [28], the suggestion is for a population of 20 particles and 200 iterations. Therefore, for both algorithms the population size was set to 200 and the number of iterations to 500. The problem addressed here is more complicated for dealing with time windows constraints, which justifies a more extensive population's choice. The number of iterations was set empirically analyzing metaheuristics' behaviour over the generations (Fig. 6). In some instances, the best solutions were obtained in some iterations well below 500. The results are presented in Table 5 in Appendix B. Column "Iteration Number" shows the iteration in which each metaheuristic obtained the "Best Solution". The number of generations needed for a fair comparison was extrapolated.

Due to the stochastic nature of metaheuristics, each instance is run 30 times. Table 6 in the Appendix reports (i) the average computational time; (ii) the average objective function; (iii) the best solution; (iv) iteration that the best solution was obtained. Total, the computational tests were performed with 110 instances. The values with * mean that CPLEX has proved optimality.

Appendix B includes detailed results that enable the reader to access and compare the performance of each algorithm for each instance. Analyzing Table 6, it is possible to see that in 77.33% of the tests, GA outperforms PSO. In 71.33% of the tests, GA outperforms CPLEX. In 8.67% of the tests, GA and CPLEX tie. In 3.33% of the tests, PSO and CPLEX tie. Besides, the FCFS heuristic for generating an individual was infeasible in 52.67% of the tests, that is, it does not bring significant contribution by accelerating the convergence of metaheuristics. To make a more careful analysis of the results presented in this table, to verify the superiority of GA, hypothesis testing will be carried out.

An interesting point that to highlight, was the solution of the PSO to the instance 3 for 20 vessels, 5 berths, processing time generated with $f(x, 5, 2)$, arrival time generated with $f(x, 2, 5)$ and departure time generated with $f(x, 2, 5)$.

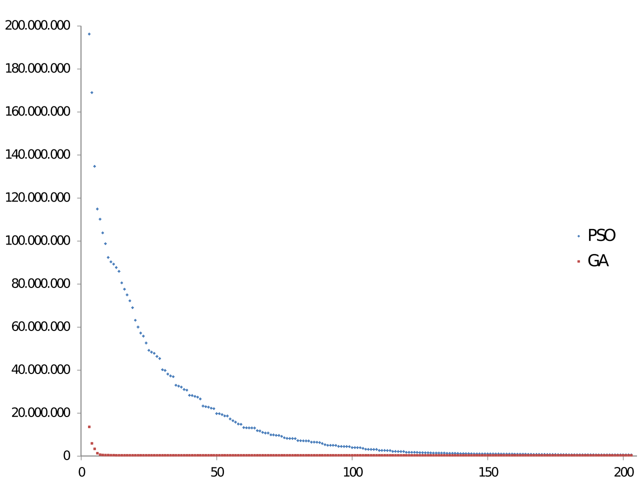

Fig. 6 Progression of the solution over the iterations

The FCFS heuristic solution is feasible, and all 30 PSO rounds converged to the FCFS heuristic solution, with average objective function and best solution with the same value, 357. GA has been able to achieve the solution that was proven by CPLEX to be optimal.

The penalty built into the objective function allowed us to observe that, in most instances, GA was able to reach the feasibility of the solutions faster than PSO. This behavior is exemplified with instance 1 for 30 vessels, 5 berths, processing time generated with $f(x, 5, 2)$, arrival time generated with $f(x, 2, 5)$ and departure time generated with $f(x, 2, 2)$. Figure 6 shows the progression of the solution over the iterations. In this case, the GA obtained a completely feasible population in iteration 8 with an average objective function of value 706.31. The PSO obtained a completely feasible population in the iteration 156 with an average objective function of value 966.145. It justifies the fact that the best solution (with objective function value 357 obtained by both algorithms) has been achieved by the GA in iteration 65 and by the PSO only in iteration 316. The feasibility mechanism used by the PSO and the same GA.

5.3 Hypothesis testing

Suppose two mean performance measures μ_1 and μ_2, based on objective function values, that are available from two different optimization methods. The purpose of hypothesis testing is to verify if $\mu_1 = \mu_2$ holds, while the confidence-interval approach evaluates the difference $\mu_1 - \mu_2$ and verify how close are both measures. The second one will be employed to make pairwise comparisons on results from CPLEX, GA, and PSO methods.

Now suppose there are K observations from both methods: $Z_1, Z_2, \cdots, Z_K$ and $W_1, W_2, \cdots, W_K$. If the K input data sets area relatively homogenous, it is reasonable to assume the K observed differences $d_j = Z_j - W_j$, $j = 1, 2, \cdots, K$ are identically distributed. By the *Central Limit Theorem* (CLT) the differences $d_j = Z_j - W_j$ are approximately normally distributed with mean μ_d and variance σ_d^2. The appropriate statistical test is then a t-test of the null hyphothesis of mean difference equal to zero:

$$H_0 : \mu_d = 0$$

versus the alternative of significative difference:

$$H_1 : \mu_d \neq 0.$$

The proper test is a paired t-test. First, compute the sample mean difference, μ_d, and the sample variance, S_d^2. Then, since $K \leq 30$, compute the t statistic with a confidence level $100(1-\alpha)\%$ and n replications: $t_{\alpha/2;n-1}$. The confidence interval limits are $[\theta_1, \theta_2]$ where:

$$\theta_1 = \overline{d} - \frac{t_{\alpha/2;n-1} S_d}{\sqrt{n}}$$

and

$$\theta_2 = \overline{d} + \frac{t_{\alpha/2;n-1} S_d}{\sqrt{n}}$$

Three cases could happen:

1. The confidence interval contains 0 and nothing can be concluded;
2. the confidence interval limits are greater than 0, meaning that $\overline{Z}$ is greater than $\overline{W}$;
3. the confidence interval limits are lower than 0, meaning that $\overline{W}$ is greater than $\overline{Z}$.

The comparisons will be made by employing the concept of a set of instances which aggregate the instances with same parameters values in the following features: number of vessels, number of berths and distribution functions parameters (p, a, and b) as defined in Table 3. For example, the instance set 20x5_C1 embraces the instances 1 to 10 that uses the following parameters: 20 vessels, 5 berths, and functions $f(x, 5, 2)$, $f(x, 2, 5)$, and $f(x, 2, 5)$. Table 4 presents the results for confidence intervals that compare the optimization methods in terms of objective function difference for each set of instances considering $\alpha = 5\%$[4]:

- Δ(GA,PSO) = obj(GA) - obj(PSO);
- Δ(GA,CPL) = obj(GA) - obj(CPLEX);
- Δ(PSO,CPL) = obj(PSO) - obj(CPLEX).

Table 4 shows that in eight sets of instances, the GA has a performance statistically equivalent to PSO. Although, in seven sets of instances, the GA produces objective function values that are best than PSO. It is important to observe that these seven sets of instances are the ones with the most significant number of berths and vessels. For example, for 40×7_C1 set of instances, the confidence interval is [−10.35; −3.01] showing a better performance of GA over PSO. This difference increases until reaching the largest one, which is for 150×10_C4 set of instances with a confidence interval of [−107.94; −74.59]. This interval shows that it is expected that using a GA will produce an objective function that is better than PSO in this corresponding set. Another interesting finding is that employing GA is statistically equivalent to using CPLEX even for the smallest instances. For example, for instances 30×5p_C3 and 30×5_C4, the confidence interval contains zero. It means that the null hypothesis that the difference between objective functions is zero cannot be rejected. Therefore the objective functions produced by both methods are not different in statistical terms. The only exception is for three instance sets: 20×5_C1, 30×5_C2, 50×10_C4.

Table 4 Confidence interval for $\alpha = 5\%$

Instance set	Δ(GA,PSO)	Δ(GA,CPL)	Δ(PSO,CPL)
20×5_C1	[−6.56; 1.77]	[4.27; 10.12]	[4.24; 14.94]
30×5_C2	[−3.84; 1.71]	[0.43; 7.69]	[0.31; 9.94]
30×5_C3	[−4.82; 2.50]	[−3.10; 8.88]	[−0.71; 8.81]
30×5_C4	[−4.81; 2.13]	[−0.70; 7.83]	[0.62; 8.73]
30×7_C1	[−2.50; 6.85]	NA	NA
30×7_C5	[−3.85; 1.43]	[−2.47; 8.77]	[−0.48; 9.92]
30×7_C4	[−4.84; 0.22]	[−0.99; 9.87]	[0.48; 13.03]
40×5_C4	[−7.83; 0.22]	[−3.38; 13.19]	[0.81; 16.59]
40×7_C1	[−10.35; −3.01]	NA	NA
40×7_C3	[−21.47; −4.78]	NA	NA
40×7_C4	[−6.92; −1.03]	[−0.09; 12.11]	[3.48; 16.49]
50×7_C4	[−11.29; −4.67]	[−38.77; 14.90]	[−29.50; 21.60]
50×10_C4	[−14.80; −8.65]	[3.56; 29.60]	[16.29; 40.33]
100×7_C4	[−72.67; −46.66]	NA	NA
150×10_C4	[−107.94; −74.59]	NA	NA

Since on these sets of instances, the confidence interval limits are all positive; it means CPLEX has a better performance than GA, because CPLEX has lower objective function values in some instances, and it makes a difference positive. In five sets of instances, it was not possible to make a comparison (the ones with a “NA” symbol) since CPLEX was not able to return even a feasible solution. A statistical comparison between PSO performance and CPLEX can be made with the values on column Δ(PSO,CPL). CPLEX is better than PSO; the confidence interval limits are all positive in seven sets of instances and worse in none. Five sets of instance, a comparison cannot be made (spaces with “NA” symbol), and only in three PSO has a performance equivalent to CPLEX.

6 Conclusion

Developing countries are gaining a more significant market share in world merchandise trade. It brings jobs and opportunities, but their ports lack the infrastructure for bigger vessels becoming a bottleneck in the global business operations. Optimizing the shipping lines on routes in Africa, Asia, and Latin America means ports in these regions will have to improve performance. They may need to spend heavily on training, personal skills development, and in building prominent and efficient terminals. Otherwise, they may face fewer port calls, less competitive markets, and higher shipping costs. The increasing demand for maritime transportation suggests for these ports to search for logistics to accommodate the incoming vessels minimizing the waiting and service times of vessels and the handling costs, giving raise for the Bert Allocation Problem.

Analyzing the results obtained with the metaheuristics, it is noteworthy that in computational times, GA and PSO were very similar. However, the GA was shown to be faster in the solution quality progression, which can be seen by observing that the best solution was obtained in the smaller iterations. Thus, the GA was more competitive than PSO.

Our paper tackled the Berth Allocation Problem. Such problem can be modelled in several ways, such as scheduling of jobs on parallel machines or vehicle routing. Thus, it is possible to extend the developed methodology to solve these problems, among others. The penalization mechanism proved to be efficient to work with time windows constraints, and in future works can be applied to other combinatorial problems.

Appendix A: Results from the computational tests using CPLEX

Table 5 Instances tests analysis

Processing time (p)	Arrival time (a)	Departure time (b)	Solved problems	Average comp. time (s)	Founded a solution	Average GAP	Unknown status
20 vessels and 5 berths							
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	0.216	0	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	0.356	8	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	0.36	9	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	0.362	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	0.424	8	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	0.431	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	0.436	8	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	0.438	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	0.438	8	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	0.438	9	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	0.453	9	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	0.463	8	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	0.464	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	0.472	9	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	0.486	8	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	0.498	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	0.514	9	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	0.531	8	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	0.559	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	0.569	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	0.583	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	10	0.607	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	0.637	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	0.684	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 2, 2)$	10	0.782	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	0.834	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	0.859	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	0.915	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 5, 2)$	10	0.926	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	0,994	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	1.037	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	1.17	10	–	0

Table 5 (continued)

Processing time (p)	Arrival time (a)	Departure time (b)	Solved problems	Average comp. time (s)	Founded a solution	Average GAP	Unknown status
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 2, 2)$	10	1.25	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	1.291	10	–	0
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	1.775	6	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 5, 2)$	10	1.776	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 2, 5)$	10	1.856	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	1.884	8	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	10	2,003	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 2, 2)$	10	5.286	9	–	0
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 2, 2)$	10	5.673	0	–	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	6.297	8	–	0
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 2, 2)$	10	7.699	9	–	0
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	10	8.223	8	–	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	10	13.075	8	–	0
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 5, 2)$	10	35.295	10	–	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	46.51	10	–	0
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 5, 2)$	10	74.244	1	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 2, 5)$	10	92,344	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	10	103.247	10	–	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	115.738	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 2, 2)$	9	63.17	10	0,00218	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	17.788	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 5, 2)$	10	27.343	9	–	0
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	10	29.794	0	–	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 2, 5)$	9	212.87	10	0,01158	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 5, 2)$	8	5.29	10	0,00262	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	8	466.28	10	0,00644	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 2, 2)$	7	227.98	10	0,00492	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	6	108.82	10	0,00492	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 5)$	4	940.67	10	0,00796	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 2)$	2	104.11	10	0,01039	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	2	107.58	10	0,00964	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	1	94.94	10	0,01064	0
20 vessels and 7 berths							
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	0.37	1	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	0.38	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	0.46	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	0.47	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	0.48	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	0.48	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	0.50	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	0.55	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	0.56	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	0.58	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	0.59	10	–	0
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	0.62	5	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	0.62	10	–	0

Table 5 (continued)

Processing time (p)	Arrival time (a)	Departure time (b)	Solved problems	Average comp. time (s)	Founded a solution	Average GAP	Unknown status
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	0.63	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	0.64	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	0.65	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	0.66	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 2, 5)$	10	0.66	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	0.69	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	0.70	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	0.71	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	0.76	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	10	0.78	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	10	0.79	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 2, 2)$	10	0.82	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 2, 2)$	10	0.86	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	0.88	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 5, 2)$	10	0.93	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	0.95	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	0.95	7	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 5, 2)$	10	0.96	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	1.03	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	1.05	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	1.05	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	1.08	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	1.19	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	1.19	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	1.22	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	1.29	10	–	0
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 2, 2)$	10	1.45	9	–	0
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	10	1.46	9	–	0
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 5, 2)$	10	2.60	10	–	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	4.83	9	–	0
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 2, 2)$	10	7.74	2	–	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 2, 2)$	10	10.14	7	–	0
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	10	15.44	4	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 2, 5)$	10	20.04	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	10	20.53	8	–	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	28.44	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	10	28.52	10	–	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	30.29	10	–	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	49.02	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 2, 2)$	10	69.77	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 5, 2)$	10	99.51	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 5, 2)$	9	6.49	9	–	1
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	9	6.82	10	0.010	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 2, 5)$	9	7.08	10	0.005	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 2, 2)$	9	8.24	10	0.008	0
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 5, 2)$	9	13.09	3	–	1
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	9	15.80	10	0.007	0

Table 5 (continued)

Processing time (p)	Arrival time (a)	Departure time (b)	Solved problems	Average comp. time (s)	Founded a solution	Average GAP	Unknown status
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 5)$	6	115.42	8	0.002	1
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	5	187.84	10	0.005	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 2)$	3	184.59	10	0.007	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	3	192.41	10	0.008	0
20 vessels and 10 berths							
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	0.43	2	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	0.43	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	0.53	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	0.57	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	0.58	10	–	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	0.59	8	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	0.59	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	0.63	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	0.63	8	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	0.63	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	0.64	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	0.65	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	0.66	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	0.662	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	0.67	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	0.67	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	0.68	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	0.70	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	0.71	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	0.72	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	0.74	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	0.77	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	0.77	10	–	0
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 2, 2)$	10	0.78	6	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	10	0.80	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	0.82	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 2, 2)$	10	0.82	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	0.85	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	10	0.86	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	0.86	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 2, 5)$	10	0.87	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 5, 2)$	10	0.88	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	0.91	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	0.92	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	0.93	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 2, 2)$	10	0.94	10	–	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 5)$	10	0.98	9	–	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	1.01	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	1.01	10	–	0
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	1.02	9	–	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	1.07	10	–	0

Table 5 (continued)

Processing time (p)	Arrival time (a)	Departure time (b)	Solved problems	Average comp. time (s)	Founded a solution	Average GAP	Unknown status
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 5, 2)$	10	1.09	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	1.10	10	–	0
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	10	1.14	9	–	0
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 2, 2)$	10	1.16	9	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	1.17	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 2, 2)$	10	1.29	9	–	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	1.39	10	–	0
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	10	1.42	5	–	0
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 5, 2)$	10	1.50	10	–	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 2, 5)$	10	1.54	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	10	1.63	9	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	10	1.85	10	–	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	10	1.88	10	–	0
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 5, 2)$	10	1.90	7	–	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 2, 2)$	10	1.90	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 2, 5)$	10	2.26	10	–	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	10	2.43	10	–	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 2)$	10	2.65	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 5, 2)$	10	3.21	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 2, 2)$	10	3.66	10	–	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	10	3.81	10	–	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	10	8.97	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 5, 2)$	10	12.95	9	–	0
30 vessels and 5 berths							
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	0.50	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	0.65	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	0.69	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	0.73	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	0.77	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	0.82	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	0.91	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	10	1.01	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	1.08	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	1.09	8	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	1.11	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	1.17	7	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	1.17	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 2, 5)$	10	1.18	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	1.20	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	1.30	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	1.31	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	10	1.39	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	1.44	7	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	1.46	9	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	1.50	7	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 2, 2)$	10	1.51	10	–	0

Table 5 (continued)

Processing time (p)	Arrival time (a)	Departure time (b)	Solved problems	Average comp. time (s)	Founded a solution	Average GAP	Unknown status
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 5, 2)$	10	1.60	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 2, 2)$	10	1.66	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	1.69	7	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	1.87	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	1.94	9	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	1.96	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 5, 2)$	10	2.31	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	2.80	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	3.03	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	4.42	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	11.17	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	11.83	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	17.28	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	20.95	10	–	0
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	41.19	0	–	0
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	101.45	8	–	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	273.08	3	–	0
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	8	15.51	8	–	2
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 2, 2)$	8	43.17	8	–	2
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 5, 2)$	8	60.34	8	–	2
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 2, 5)$	8	299.29	10	0.00255	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	7	38.83	10	0.00215	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 2, 5)$	7	52.20	8	0.00321	2
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 2, 2)$	7	71.40	10	0.00176	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	7	81.28	10	0.00572	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	7	93.07	10	0.00217	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 2, 5)$	7	144.94	10	0.00487	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	7	189.28	10	0.01303	0
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 2, 2)$	7	223.07	0	–	3
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 2, 2)$	7	316.11	10	0.01144	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 2, 2)$	7	337.66	9	0.00427	1
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 5, 2)$	7	612.29	10	0.00945	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 2, 2)$	6	520.93	10	0.00413	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 5, 2)$	6	65.77	10	0.00283	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	5	56.47	7	0.00182	2
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 5, 2)$	5	70.26	10	0.00370	0
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	5	193.82	0	–	5
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 5, 2)$	3	339.20	2	0.00164	5
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	0	–	10	0.01621	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 2)$	0	–	10	0.01645	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	0	–	10	0.01917	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 5)$	0	–	8	0.01371	2
30 vessels and 7 berths							
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	0.89	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	1.03	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	1.19	10	–	0

Table 5 (continued)

Processing time (p)	Arrival time (a)	Departure time (b)	Solved problems	Average comp. time (s)	Founded a solution	Average GAP	Unknown status
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	1.22	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	1.28	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	1.42	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	1.42	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	1.62	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	1.66	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	1.8	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	1.82	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	10	1.83	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 2, 5)$	10	1.93	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	2.08	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	2.12	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 2, 2)$	10	2.16	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 5, 2)$	10	2.64	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	2.93	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	10	2.95	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 2, 2)$	10	3.16	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	3.26	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 5, 2)$	10	3.69	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	4.48	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	4.75	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	4.77	9	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	5.44	9	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	5.64	9	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	5.85	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	6.61	9	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	6.65	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	6.78	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	10.88	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	13.38	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	14.45	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	17.7	10	–	0
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	19.7	0	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	21.83	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 5, 2)$	10	50.81	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	89.4	9	–	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 2, 2)$	9	14.66	10	0.0013	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	9	15.13	10	0.0017	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 2, 5)$	9	29.82	10	0.0047	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 2, 2)$	9	50.5	10	0.0044	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	9	55.77	10	0.0048	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	9	62.29	10	0.0053	0
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 2, 5)$	9	125.14	10	0.0009	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	9	255.44	10	0.0059	0
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	8	47.52	10	0.0013	0
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 5, 2)$	8	65.57	10	0.0018	0
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 2, 2)$	8	70.28	10	0.0014	0

Table 5 (continued)

Processing time (p)	Arrival time (a)	Departure time (b)	Solved problems	Average comp. time (s)	Founded a solution	Average GAP	Unknown status
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 2, 5)$	8	173.54	10	0.0039	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 5, 2)$	8	232.9	10	0.0041	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 2, 2)$	8	280.04	10	0.0034	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 2, 5)$	7	240.3	10	0.0034	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 2, 2)$	6	184.24	10	0.0027	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 5, 2)$	6	231.09	10	0.0032	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	6	388.25	10	0.0027	0
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 2, 2)$	3	697.45	0	–	7
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	1	138.23	0	–	9
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	0	3600.00	10	0.0151	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 2)$	0	3600.00	10	0.0163	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	0	3600.00	10	0.017	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 5)$	0	3600.00	8	0.0166	2
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 5, 2)$	0	3600.00	0	–	10
30 vessels and 10 berths							
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	0.874	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	1.041	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	1.056	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	1.094	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	1.18	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	1.22	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	1.28	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	1.36	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	1.43	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	1.46	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	1.49	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	1.54	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	1.64	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	1.68	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	1.73	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	1.82	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	1.85	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	1.91	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	2.00	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	2.10	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	10	2.11	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 2, 5)$	10	2.20	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	10	2.28	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	2.33	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 2, 2)$	10	2.37	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	2.50	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 2, 2)$	10	2.59	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 5, 2)$	10	3.07	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 5, 2)$	10	3.40	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	3.88	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	3.98	10	–	0

Table 5 (continued)

Processing time (p)	Arrival time (a)	Departure time (b)	Solved problems	Average comp. time (s)	Founded a solution	Average GAP	Unknown status
$f(x,2,2)$	$f(x,2,2)$	$f(x,2,2)$	10	4.78	10	–	0
$f(x,0.5,0.5)$	$f(x,2,2)$	$f(x,0.5,0.5)$	10	5,43	10	–	0
$f(x,2,2)$	$f(x,2,2)$	$f(x,5,2)$	10	5.71	10	–	0
$f(x,0.5,0.5)$	$f(x,2,2)$	$f(x,2,2)$	10	6.81	10	–	0
$f(x,0.5,0.5)$	$f(x,2,5)$	$f(x,2,5)$	10	8.60	10	–	0
$f(x,2,2)$	$f(x,5,2)$	$f(x,2,5)$	10	9.64	9	–	0
$f(x,0.5,0.5)$	$f(x,2,5)$	$f(x,0.5,0.5)$	10	10.47	10	–	0
$f(x,0.5,0.5)$	$f(x,2,2)$	$f(x,5,2)$	10	11.59	10	–	0
$f(x,0.5,0.5)$	$f(x,2,5)$	$f(x,2,2)$	10	16.24	10	–	0
$f(x,0.5,0.5)$	$f(x,2,5)$	$f(x,5,2)$	10	17.16	10	–	0
$f(x,2,2)$	$f(x,2,5)$	$f(x,2,2)$	10	21.17	10	–	0
$f(x,2,2)$	$f(x,2,5)$	$f(x,2,5)$	10	22.43	10	–	0
$f(x,2,2)$	$f(x,2,5)$	$f(x,5,2)$	10	25.40	10	–	0
$f(x,0.5,0.5)$	$f(x,5,2)$	$f(x,2,5)$	10	29.81	8	–	0
$f(x,2,2)$	$f(x,2,5)$	$f(x,0.5,0.5)$	10	32.06	10	–	0
$f(x,5,2)$	$f(x,5,2)$	$f(x,2,5)$	10	60.97	2	–	0
$f(x,0.5,0.5)$	$f(x,5,2)$	$f(x,0.5,0.5)$	10	82.55	9	–	0
$f(x,0.5,0.5)$	$f(x,5,2)$	$f(x,2,2)$	10	118.41	9	–	0
$f(x,0.5,0.5)$	$f(x,5,2)$	$f(x,5,2)$	10	144.46	10	–	0
$f(x,5,2)$	$f(x,2,2)$	$f(x,0.5,0.5)$	9	4.78	10	0.00021	0
$f(x,5,2)$	$f(x,2,2)$	$f(x,2,2)$	9	5.02	10	0.00027	0
$f(x,5,2)$	$f(x,2,2)$	$f(x,5,2)$	9	7.75	10	0.00032	0
$f(x,2,2)$	$f(x,5,2)$	$f(x,0.5,0.5)$	9	17.18	9	–	1
$f(x,2,2)$	$f(x,5,2)$	$f(x,2,2)$	9	19.11	9	–	1
$f(x,2,2)$	$f(x,5,2)$	$f(x,5,2)$	9	28.16	10	0.00479	0
$f(x,5,2)$	$f(x,2,2)$	$f(x,2,5)$	9	326.45	10	–	0
$f(x,5,2)$	$f(x,5,2)$	$f(x,2,2)$	8	341.71	3	0.00125	1
$f(x,5,2)$	$f(x,5,2)$	$f(x,0.5,0.5)$	6	45.61	4	0.00124	2
$f(x,5,2)$	$f(x,5,2)$	$f(x,5,2)$	4	102.66	4	0.00035	4
$f(x,5,2)$	$f(x,2,5)$	$f(x,2,5)$	2	461.27	10	0.00504	0
$f(x,5,2)$	$f(x,2,5)$	$f(x,0.5,0.5)$	2	1358.89	10	0.00520	0
$f(x,5,2)$	$f(x,2,5)$	$f(x,2,2)$	2	1781.70	10	0.00537	0
$f(x,5,2)$	$f(x,2,5)$	$f(x,5,2)$	1	53.28	10	0.00454	0
40 vessels and 5 berths							
$f(x,0.5,0.5)$	$f(x,0.5,0.5)$	$f(x,2,5)$	10	1.22	6	–	0
$f(x,2,2)$	$f(x,0.5,0.5)$	$f(x,2,5)$	10	1.42	8	–	0
$f(x,2,5)$	$f(x,0.5,0.5)$	$f(x,2,5)$	10	1.62	9	–	0
$f(x,0.5,0.5)$	$f(x,0.5,0.5)$	$f(x,0.5,0.5)$	10	1.71	8	–	0
$f(x,2,5)$	$f(x,0.5,0.5)$	$f(x,2,2)$	10	1.82	9	–	0
$f(x,0.5,0.5)$	$f(x,0.5,0.5)$	$f(x,2,2)$	10	1.84	7	–	0
$f(x,2,2)$	$f(x,0.5,0.5)$	$f(x,2,2)$	10	1.93	8	–	0
$f(x,2,2)$	$f(x,0.5,0.5)$	$f(x,0.5,0.5)$	10	1.98	8	–	0
$f(x,2,5)$	$f(x,2,2)$	$f(x,2,5)$	10	2.07	10	–	0
$f(x,2,5)$	$f(x,0.5,0.5)$	$f(x,0.5,0.5)$	10	2,07	9	–	0
$f(x,0.5,0.5)$	$f(x,0.5,0.5)$	$f(x,5,2)$	10	2,37	9	–	0
$f(x,2,2)$	$f(x,0.5,0.5)$	$f(x,5,2)$	10	2.37	8	–	0

Table 5 (continued)

Processing time (p)	Arrival time (a)	Departure time (b)	Solved problems	Average comp. time (s)	Founded a solution	Average GAP	Unknown status
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	2.52	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	2.62	9	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	2.77	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	3.79	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 2, 5)$	10	4.15	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	10	4.38	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	4.40	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	10	5.27	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 5, 2)$	10	5.37	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	5.55	4	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	14.54	4	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 5, 2)$	10	23.88	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	90.71	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	102.28	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	147.36	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	9	1.19	4	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 2, 2)$	9	4.54	10	0.00316	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 2, 2)$	9	8.05	9	–	1
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	9	8.50	10	0.00123	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 2, 5)$	9	9.10	10	0.00095	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 5, 2)$	9	12.79	10	0.00116	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 5, 2)$	9	16.45	10	0.00015	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	9	33.32	5	–	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 2, 5)$	9	45.79	5	–	1
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 2, 2)$	8	10.64	10	0.00079	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 2, 5)$	7	100.27	9	0.00266	1
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 2, 5)$	7	118.13	10	0.00399	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	7	155.98	10	0.00360	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	7	291.73	10	0.00356	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 2, 2)$	7	487.60	10	0.00478	0
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 2, 5)$	6	25.92	0	–	4
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	6	29.88	10	0.00645	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 2, 2)$	6	58.72	10	0.00341	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 5, 2)$	6	61.48	10	0.00346	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	6	77.51	7	0.00297	3
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	6	395.36	10	0.00770	0
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 2, 5)$	6	624.31	4	0.00086	3
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 2, 5)$	6	774.97	10	0.00768	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 5, 2)$	5	92.48	6	0.00054	4
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 2, 2)$	5	142.15	7	0.00240	3
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 2, 2)$	5	676.54	10	0.00869	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 5, 2)$	4	65.02	10	0.00480	0
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 2, 2)$	3	405.92	6	0.00119	4
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	3	621.01	7	0.00155	3
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 5, 2)$	3	878.61	8	0.00129	2
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 2, 2)$	2	1881.9	1	–	7
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	2	3600	0	–	8

Table 5 (continued)

Processing time (p)	Arrival time (a)	Departure time (b)	Solved problems	Average comp. time (s)	Founded a solution	Average GAP	Unknown status
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	0	–	10	0.46239	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 2)$	0	–	9	0.01840	1
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	0	–	6	0.02247	4
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 5)$	0	–	5	0.01630	5
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 5, 2)$	1	3600	1	0.00601	8
40 vessels and 7 berths							
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	1.21	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	1.5	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	1.69	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	1.75	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	1.89	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	1.93	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	2.01	9	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	2.15	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	2.25	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	2.30	9	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	2.45	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	2.51	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	2.8	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	2.80	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	2.84	9	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	3.48	9	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 2, 2)$	10	3.68	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	10	3.71	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 2, 5)$	10	4.70	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	10	5.26	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	6.33	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 5, 2)$	10	6.95	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 5, 2)$	10	8.77	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	10.14	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	14.55	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	68.46	8	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	9	1.95	10	0.00027	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 2, 5)$	9	4.032	10	0.00110	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	9	5.38	10	0.00064	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 2, 2)$	9	5.47	10	0.00138	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	9	6.94	6	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 5, 2)$	9	8.49	10	0.00104	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	9	9.84	8	0.00064	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 2, 2)$	9	10.13	10	0.00136	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 2, 2)$	9	12.50	10	0.00117	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	8	9.84	8	0.00033	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	7	45.29	10	0.00312	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 2, 2)$	7	55.47	10	0.00715	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 2, 2)$	7	62.74	10	0.00295	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	7	82.88	10	0.00321	0

Table 5 (continued)

Processing time (p)	Arrival time (a)	Departure time (b)	Solved problems	Average comp. time (s)	Founded a solution	Average GAP	Unknown status
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 5, 2)$	7	91.13	10	0.00302	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 2, 5)$	7	174.50	10	0.00340	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	7	218.93	10	0.00121	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 2, 2)$	7	411.98	10	0.00091	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 2, 5)$	7	465.57	10	0.00163	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	6	36.71	10	0.00233	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 2, 5)$	6	69.19	10	0.00386	0
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 2, 5)$	6	281.03	7	0.00064	1
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 5, 2)$	6	368.92	10	0.00110	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 2, 5)$	5	141.71	8	0.00126	2
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	5	168.91	9	0.00073	1
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	5	253.83	9	0.00105	1
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 2, 2)$	5	438.95	9	0.00086	1
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 2, 5)$	5	576.04	3	0.00143	3
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 5, 2)$	4	281.092	9	0.00074	1
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 5, 2)$	4	526.65	10	0.00084	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 2, 2)$	4	594.87	8	0.00098	2
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	1	2201.84	2	0.00224	8
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 5, 2)$	1	3600	2	0.00125	7
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 2)$	0	–	10	0.01318	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	0	–	10	0.01439	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	0	–	9	61.45	1
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 5)$	0	–	6	0.00942	4
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 2, 2)$	0	–	2	0.00011	8
40 vessels and 10 berths							
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	1.70	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	2;10	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	2.14	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	2.22	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	2.38	10	–	0
$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	2.52	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	2.67	10	–	0
$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	2.87	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	3.04	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 2, 5)$	10	3.11	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	3.12	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	3.83	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	10	4.29	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 2, 5)$	10	4.58	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 2, 2)$	10	4.58	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	10	4.67	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	4.69	10	–	0
$f(x, 2, 5)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	4.80	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	10	5.12	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 2, 2)$	10	5.80	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	10	5.81	9	–	0

Table 5 (continued)

Processing time (p)	Arrival time (a)	Departure time (b)	Solved problems	Average comp. time (s)	Founded a solution	Average GAP	Unknown status
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	5.98	10	–	0
$f(x, 2, 5)$	$f(x, 2, 5)$	$f(x, 5, 2)$	10	6.15	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	10	6.49	9	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	6.99	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 2, 5)$	10	7.21	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	7.58	10	–	0
$f(x, 2, 5)$	$f(x, 5, 2)$	$f(x, 5, 2)$	10	8.101	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	10	8.78	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	9.03	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	10.19	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	10	11.69	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 2, 2)$	10	12.62	10	–	0
$f(x, 0.5, 0.5)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	14.21	10	–	0
$f(x, 2, 2)$	$f(x, 2, 2)$	$f(x, 5, 2)$	10	14.68	10	–	0
$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	9	5.10	9	–	0
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 2, 5)$	8	10.72	1	0.00025	1
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 2, 5)$	8	45.07	9	0.00043	1
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	8	74.82	9	0.00040	1
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 2, 2)$	8	89.75	9	0.00051	1
$f(x, 2, 2)$	$f(x, 5, 2)$	$f(x, 5, 2)$	8	119.49	10	0.00074	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 2, 2)$	7	177.52	10	0.00181	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 2, 2)$	7	193.92	9	0.00064	1
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	7	237.53	10	0.00142	0
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 5, 2)$	7	242.22	10	0.00203	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 2, 2)$	7	298.10	10	0.00155	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 5, 2)$	7	311.17	9	0.00072	1
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	7	332.99	10	0.00163	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 2, 5)$	6	13.82	10	0.00122	0
$f(x, 2, 2)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	6	14.87	10	0.00148	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 0.5, 0.5)$	6	36.68	10	0.00286	0
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	6	126.74	8	0.00025	2
$f(x, 0.5, 0.5)$	$f(x, 2, 5)$	$f(x, 2, 5)$	6	144.263	9	0.00154	1
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 2, 5)$	5	17.84	9	0.00279	1
$f(x, 0.5, 0.5)$	$f(x, 5, 2)$	$f(x, 2, 5)$	5	38.14	8	0.00047	2
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 2, 2)$	5	58.79	10	0.00207	0
$f(x, 5, 2)$	$f(x, 2, 2)$	$f(x, 5, 2)$	5	72.80	10	0.00221	0
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 0.5, 0.5)$	3	2426.73	1	–	7
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 2, 2)$	2	1801.29	1	0.00061	7
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	0	–	10	0.00674	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 2)$	0	–	10	0.00690	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 0.5, 0.5)$	0	–	10	0.00836	0
$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 5)$	0	–	9	0.00556	1
$f(x, 5, 2)$	$f(x, 5, 2)$	$f(x, 5, 2)$	0	–	1	0,00030	9

Appendix B: Results from the computational tests using the metaheuristics

Table 6 Comparison results

	GA				PSO					
	Comp. time	Average objective	Best solution	Iteration number	Comp. time	Average objective	Best solution	Iteration number	CPLEX	FCFS
20×5 - $p: f(x,5,2), a: f(x,2,5), b: f(x,2,5)$ - C1										
1	2.44096	442.5	428	80	2.43371	455.45	429	390	428*	InFea
2	2.44048	376.867	373	27	2.43811	378.2	373	312	373	InFea
3	2.47887	356.7	354	11	2.44245	357	357	0	354*	357
4	2.43888	434.7	425	72	2.43617	437.333	428	494	425	InFea
5	2.4408	324.467	321	45	2.49607	338.5	329	300	321*	InFea
6	2.432	426.033	418	52	2.41964	426.367	423	448	419	InFea
7	2.44154	365.867	361	46	2.4342	365.3	364	223	361	InFea
8	2.37677	373.067	371	59	2.42829	371	371	206	370*	InFea
9	2.39217	394.933	388	40	2.38351	394.133	389	462	385	InFea
10	2.40779	422.867	416	63	2.43171	418.667	411	491	410	InFea
30×5 - $p: f(x,5,2), a: f(x,2,5), b: f(x,2,2)$ - C2										
1	5.44496	392.433	387	65	5.88105	393.167	387	316	390	427
2	5.62155	363.867	357	72	5.91886	367.167	363	341	354	Infea
3	5.36694	341.6	339	76	5.89994	342.667	341	172	331	InFea
4	5.5594	392.567	385	108	5.93147	392.567	382	291	388	442
5	5.51518	429.7	418	63	5.92249	426.733	421	303	424	InFea
6	5.57889	464.8	451	81	6.00652	476.3	458	350	459	InFea
7	5.47882	397.133	388	108	5.83806	398.533	388	494	394	494
8	5.46717	423.1	414	150	5.90615	418.467	414	386	416	InFea
9	5.46193	471.8	460	163	5.90029	471.7	469	454	481	InFea
10	5.51181	415.633	408	164	5.89064	416	408	403	415	InFea
30×5 - $p: f(x,5,2), a: f(x,2,5), b: f(x,0.5,0.5)$ - C3										
1	5.55014	409.667	395	254	6.06957	410.167	399	434	405	InFea
2	5.67975	368.733	357	82	6.07927	368.133	358	127	361	InFea
3	5.35032	343.033	338	158	5.93905	342.533	341	453	331	InFea
4	5.61758	393.733	385	61	6.05866	396.867	392	234	388	InFea
5	5.61267	443.833	421	113	6.04785	438.4	425	449	445	InFea
6	5.55913	474.933	458	154	6.06745	487.333	468	270	483	InFea
7	5.49641	399.133	388	98	5.90988	403.9	394	405	400	InFea
8	5.51259	446.7	428	135	6.07503	439.667	430	184	429	InFea
9	6.0647	492.267	472	330	6.08939	494.1	475	431	504	InFea
10	6.3454	429.867	414	131	6.03559	432.4	424	368	427	InFea
30×5 - $p: f(x,5,2), a: f(x,2,5), b: f(x,5,2)$ - C4										
1	5.36892	390.767	382	43	5.82077	389.833	385	404	391	427
2	5.58279	361.2	355	128	5.83069	365.133	363	439	356	422
3	5.29334	341	339	68	5.78876	343	343	94	331	365
4	5.49413	392.1	386	163	5.84025	391.867	382	418	393	442
5	5.44219	426.433	414	139	5.80001	425.267	422	492	422	490
6	5.48639	462.133	455	257	5.87109	471.8	455	332	460	562
7	5.43422	394.267	386	120	5.74966	399.467	394	172	388	494

Table 6 (continued)

	GA				PSO					
	Comp. time	Average objective	Best solution	Iteration number	Comp. time	Average objective	Best solution	Iteration number	CPLEX	FCFS
8	5.42002	427.533	414	65	5.80267	417.733	414	381	414	InFea
9	5.38488	468.5	461	216	5.77338	472.4	469	442	476	539
10	5.48115	414.467	406	98	5.81969	415.3	407	336	414	497
30×7 - $p: f(x,5,2), a: f(x,2,5), b: f(x,2,5)$ - C1										
1	4.42206	589.8	584	69	4.32267	591.633	588	313	594	InFea
2	4.33721	583.633	563	143	4.31193	582.7	567	325	–	InFea
3	4.31419	560.567	546	139	4.26428	558.4	548	295	554	InFea
4	4.29948	601.2	586	80	4.35129	600.8	590	275	594	InFea
5	4.28449	643.867	633	74	4.21272	647.9	634	451	632	InFea
6	4.41932	534.667	524	68	4.21935	535.067	531	473	519	InFea
7	4.30246	600.533	567	150	4.21311	590.067	564	451	567	InFea
8	4.36342	544.967	533	98	4.24737	546.1	537	490	538	InFea
9	4.35974	618.9	598	249	4.21599	614.033	598	420	615	InFea
10	4.3448	719.769	689	117	4.26803	709.462	687	205	–	InFea
30×7 - $p: f(x,5,2), a: f(x,2,5), b: f(x,2,2)$ - C5										
1	4.27634	586.033	583	68	4.17765	588.733	580	372	580	635
2	4.25298	581.833	563	173	4.16081	580.1	564	452	579	InFea
3	4.21417	547.1	541	84	4.14369	546.767	538	473	546	InFea
4	4.24585	587.767	577	206	4.17538	586.2	577	288	583	InFea
5	4.23992	630.5	615	70	4.1677	634.133	626	470	630	InFea
6	4.31371	523.5	517	119	4.17379	526.767	524	266	512	570
7	4.20656	572.433	560	197	4.17025	568.4	559	461	552	InFea
8	4.29258	534.7	526	93	4.15368	538.367	532	351	537	InFea
9	4.25596	601	594	78	4.16788	598.6	593	198	606	InFea
10	4.26808	669.667	657	188	4.21991	678.567	667	495	678	InFea
30×7 - $p: f(x,5,2), a: f(x,2,5), b: f(x,5,2)$ - C4										
1	4.16898	582.467	580	97	4.10468	583.067	580	429	579	635
2	4.18	581.567	563	182	4.1355	582.267	572	466	564	637
3	4.1415	545.8	541	93	4.08757	546.167	540	424	550	608
4	4.15392	577.9	575	207	4.11182	577.9	574	220	575	654
5	4.19005	629.533	623	69	4.10211	633.2	624	384	615	696
6	4.5087	520.833	509	236	4.09863	526.167	517	476	509	570
7	4.61945	565.933	555	108	4.10108	565.467	557	439	560	636
8	4.57433	533.3	527	103	4.08356	537.9	533	387	536	618
9	4.71918	592.2	589	90	4.10279	590.033	588	156	598	650
10	4.61019	656.9	653	122	4.13314	667.4	663	345	656	732
40×5 - $p: f(x,5,2), a: f(x,2,5), b: f(x,5,2)$ - C4										
1	11.0973	664	652	299	12.2981	678.567	662	409	678	842
2	10.8854	453.733	448	201	12.5186	449.7	439	483	450	542
3	10.6708	385.433	369	138	12.466	384.4	374	442	369	489
4	10.5934	501.1	493	336	12.4098	498.167	492	479	505	625
5	10.8279	487.8	480	95	12.6039	494	483	490	482	577

Table 6 (continued)

	GA				PSO					
	Comp. time	Average objective	Best solution	Iteration number	Comp. time	Average objective	Best solution	Iteration number	CPLEX	FCFS
6	10.6114	451.733	438	128	12.6231	453.867	439	489	428	576
7	11.2115	463.867	455	95	12.5223	470.2	460	424	472	525
8	10.864	347.033	331	156	12.5044	355.733	350	370	328	407
9	11.5177	420.833	413	282	12.605	420.867	415	405	413	497
10	10.8037	566.5	550	212	12.7989	574.567	566	452	568	698
40x7 - $p : f(x, 5, 2), a : f(x, 2, 5), b : f(x, 2, 5)$ - C1										
_227 1	7.52868	652.933	632	229	7.96548	667.967	648	488	–	InFea
2	7.53006	491.367	466	193	7.99012	495.033	470	500	463	559
3	7.61987	718.464	688	316	8.04117	732.897	694	476	–	InFea
4	7.60239	554.567	546	103	7.93714	554.4	550	492	567	InFea
5	7.44254	577	564	420	7.89156	575.4	563	402	560	InFea
6	7.546	586.7	576	143	7.98173	594.533	584	457	571	InFea
7	7.54718	519.6	508	136	7.88165	527.6	520	461	512	InFea
8	7.64337	680.267	660	166	8.03195	686.567	663	448	–	InFea
9	7.41872	479.6	472	157	7.95603	487.933	474	323	474	InFea
10	7.60144	622.667	604	139	7.9437	627.7	613	499	–	InFea
40×7 - $p : f(x, 5, 2), a : f(x, 2, 5), b : f(x, 0.5, 0.5)$ - C3										
1	7.56658	642.433	615	169	8.00369	672.1	639	409	651	InFea
2	7.59673	493.6	476	291	8.09169	506.767	470	491	469	InFea
3	7.59802	705.767	675	398	8.19223	739.067	694	495	–	InFea
4	7.65005	565.8	552	189	7.92617	567.567	554	444	563	InFea
5	7.474	572.533	559	274	7.89535	570.133	554	460	553	InFea
6	7.55181	593.333	574	223	8.06791	617.033	608	413	644	InFea
7	7.5683	523.333	508	144	7.87521	527.2	518	459	514	InFea
8	7.5857	666.3	637	313	8.13508	682.267	661	486	671	InFea
9	7.49133	490.5	476	147	8.14429	494.767	480	401	474	InFea
10	7.6489	642.833	624	232	8.02488	650.867	627	472	653	InFea
40×7 - $p : f(x, 5, 2), a : f(x, 2, 5), b : f(x, 5, 2)$ - C4										
1	7.40226	635.667	627	92	7.80689	633.767	604	416	614	729
2	7.42561	492.933	474	151	7.83343	501	492	424	474	559
3	7.40462	658.733	646	140	7.83034	663.433	652	321	663	821
4	7.57102	541.3	539	193	7.82683	541.967	535	482	545	590
5	7.36938	565.3	555	215	7.72465	563.4	553	436	558	636
6	7.33602	557.333	544	132	7.92143	565.3	560	487	550	646
7	7.29626	518.733	510	122	7.99787	522.3	517	278	512	589
8	7.46437	647.567	632	186	7.783	655.133	646	487	639	747
9	7.32683	475.367	468	192	7.79072	476.5	471	435	476	534
10	7.51488	606.167	598	174	7.91903	616.067	597	475	608	737
50×7 - $p : f(x, 5, 2), a : f(x, 2, 5), b : f(x, 5, 2)$ - C4										
1	12.0306	643.967	619	265	13.504	646.333	638	427	621	751
2	11.9234	536.4	533	122	13.5086	539.333	533	486	540	610
3	11.8435	563.2	552	229	13.5418	573.467	563	489	559	633
4	12.4204	632.667	617	137	13.6508	638.7	621	476	634	768

Table 6 (continued)

	GA				PSO					
	Comp. time	Average objective	Best solution	Iteration number	Comp. time	Average objective	Best solution	Iteration number	CPLEX	FCFS
5	11.9189	746.533	740	301	13.4251	751.367	742	418	758	85
6	12.2212	582.2	567	259	13.692	598.433	578	476	582	752
7	11.934	585.2	567	138	13.4849	595.167	578	387	575	664
8	11.9151	630.9	612	138	13.4107	633.967	623	487	639	730
9	12.2238	626.967	617	290	13.6373	636.967	625	490	639	795
10	12.0123	768.6	750	327	13.5286	782.733	768	500	889	932
50×10 - $p : f(x, 5, 2), a : f(x, 2, 5), b : f(x, 5, 2)$ - C4										
1	9.02244	746.167	735	164	8.91988	755.733	748	471	734	825
2	9.10821	789.033	763	282	8.93409	810.367	789	445	794	912
3	9.11461	705.567	687	243	8.97742	718	710	499	676	807
4	9.19993	754.6	740	284	8.99088	762.9	751	394	734	914
5	9.08836	881.667	868	302	8.91698	891.6	877	500	896	992
6	9.08731	737.867	715	165	8.91385	747	733	476	686	865
7	9.19825	921.767	913	275	8.95118	938.567	925	499	902	InFea
8	9.24116	799.5	786	395	8.97624	809.1	790	485	778	936
9	9.07784	799.033	791	174	8.89187	805.267	791	474	770	889
10	9.19301	698.633	687	168	8.942	712.6	694	411	698	821
100×7 - $p : f(x, 5, 2), a : f(x, 2, 5), b : f(x, 5, 2)$ - C4										
1	87.1763	755.767	738	385	92.0299	791.067	779	491	798	981
2	85.8305	763.5	737	456	91.485	800	784	490	742	971
3	85.3758	1065.33	1033	471	91.2732	1134.83	1101	463	Unknown	InFea
4	85.9964	989.867	975	441	91.5648	1042.6	1020	475	Unknown	InFea
5	85.5265	940.467	903	405	91.6787	1037.8	1008	451	1009	InFea
6	85.758	1005.77	963	497	91.467	1069.2	1041	490	1061	InFea
7	86.6334	668.467	655	417	91.7927	720.133	693	486	696	900
8	86.1883	1032.47	1007	498	92.0892	1108.3	1077	482	Unknown	InFea
9	87.0695	1078.8	1054	494	92.1898	1124.63	1092	496	Unknown	InFea
10	85.4683	1123.23	1096	407	91.3016	1191.8	1160	491	Unknown	InFea
150×10 - $p : f(x, 5, 2), a : f(x, 2, 5), b : f(x, 5, 2)$ - C4										
1	147.899	718.333	691	488	152.511	801.6	782	466	Unknown	907
2	145.457	705.167	681	481	152.932	778.533	765	499	Unknown	878
3	144.558	852.967	839	493	152.606	954.467	928	464	Unknown	InFea
4	147.315	801	777	459	153.099	888.567	869	477	Unknown	InFea
5	145.629	732.567	718	490	153	777.5	768	499	Unknown	860
6	146.693	762.7	751	484	153.725	845.267	831	498	Unknown	InFea
7	143.867	853.467	823	487	152.089	938.6	900	500	Unknown	InFea
8	143.816	754.367	730	471	152.334	872.567	838	473	Unknown	InFea
9	144.426	1218.2	1191	493	152.413	1352.23	1302	481	Unknown	InFea
10	144.492	785.267	757	486	152.756	887.367	851	494	Unknown	InFea

References

1. Agra A, Oliveira M (2018) Mip approaches for the integrated berth allocation and quay crane assignment and scheduling problem. Eur J Oper Res 264(1):138–148
2. Bäck T, Fogel DB, Michalewicz Z (2000) Evolutionary computation 1: basic algorithms and operators, vol 1. CRC press
3. Bäck T, Fogel DB, Michalewicz Z (2000) Evolutionary computation 2: advanced algorithms and operators, vol 1. CRC press
4. Banks J (2005) Discrete event system simulation. Prentice Hall
5. Barros VH, Costa TS, Oliveira AC, Lorena LA (2011) Model and heuristic for berth allocation in tidal bulk ports with stock level constraints. Computers & Industrial Engineering 60(4):606–613
6. Buhrkal K, Zuglian S, Ropke S, Larsen J, Lusby R (2011) Models for the discrete berth allocation problem: a computational comparison. Transportation Research Part E: Logistics and Transportation Review 47(4):461–473
7. Cahyono RT, Flonk EJ, Jayawardhana B (2019) Discrete-event systems modeling and the model predictive allocation algorithm for integrated berth and quay crane allocation. IEEE Trans Intell Transp Syst 21(3):1321–1331
8. Correcher JF, Alvarez-Valdes R (2017) A biased random-key genetic algorithm for the time-invariant berth allocation and quay crane assignment problem. Expert Syst Appl 89:112–128
9. Correcher JF, Van den Bossche T, Alvarez-Valdes R, Berghe GV (2019) The berth allocation problem in terminals with irregular layouts. Eur J Oper Res 272(3):1096–1108
10. Du KL, Swamy MNS (2016) Particle swarm optimization. Springer International Publishing, Cham
11. Du Y, Chen Q, Lam JSL, Xu Y, Cao JX (2015) Modeling the impacts of tides and the virtual arrival policy in berth allocation. Transp Sci 49(4):939–956
12. Eberhart, Shi Y (2001) Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 congress on evolutionary computation (IEEE Cat.No.01TH8546). vol 1, pp 81–86
13. Eberhart RC, Shi Y (1998) Comparison between genetic algorithms and particle swarm optimization. Springer Berlin Heidelberg
14. Frojan P, Correcher JF, Alvarez-Valdes R, Koulouris G, Tamarit JM (2015) The continuous berth allocation problem in a container terminal with multiple quays. Expert Syst Appl 42(21):7356–7366
15. Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning, 1 edn. Addison-Wesley, Reading
16. Hsu HP (2016) A HPSO for solving dynamic and discrete berth allocation problem and dynamic quay crane assignment problem simultaneously. Swarm and Evolutionary Computation 27:156–168
17. Hsu HP, Wang CN (2020) Resources planning for container terminal in a maritime supply chain using multiple particle swarms optimization (mpso). Mathematics 8(5):764
18. Jos BC, Harimanikandan M, Rajendran C, Ziegler H (2019) Minimum cost berth allocation problem in maritime logistics: new mixed integer programming models. Sadhana 44(6):149
19. Kavoosi M, Dulebenets MA, Abioye O, Pasha J, Theophilus O, Wang H, Kampmann R, Mikijeljevic M (2019) Berth scheduling at marine container terminals: a universal island-based metaheuristic approach. Maritime Business Review 5:1
20. Lalla-Ruiz E, González-Velarde JL, Melián-Batista B, Moreno-Vega JM (2014) Biased random key genetic algorithm for the tactical berth allocation problem. Appl Soft Comput 22:60–76
21. Lin SW, Ting CJ, Wu KC (2018) Simulated annealing with different vessel assignment strategies for the continuous berth allocation problem. Flex Serv Manuf J 30(4):740–763
22. Meisel F, Bierwirth C (2013) A framework for integrated berth allocation and crane operations planning in seaport container terminals. Transp Sci 47(2):131–147
23. Michalewicz Z (1996) Genetic algorithms + data structures = evolution programs, 3rd edn. Springer, Berlin
24. Nishi T, Okura T, Lalla-Ruiz E, Voß S (2020) A dynamic programming-based matheuristic for the dynamic berth allocation problem. Annals Oper Res 286(1):391–410
25. Raa B, Dullaert W, Schaeren RV (2011) An enriched model for the integrated berth allocation and quay crane assignment problem. Expert Syst Appl 38(11):14136–14147
26. Silva E, Oliveira JF, Wascher G (2014) 2DCPackGen: a problem generator for two-dimensional rectangular cutting and packing problems. Eur J Oper Res 237(3):846–856
27. Tavakkoli-Moghaddam R, Fatemi-Anaraki S, Abdolhamidi D, Vahedi-Nouri B (2019) Integrated waterway scheduling, berth allocation and quay crane assignment problem by using a hybrid flow shop concept. In: 2019 international conference on industrial engineering and systems management (IESM). IEEE, pp 1–5
28. Ting CJ, Wu KC, Chou H (2014) Particle swarm optimization algorithm for the berth allocation problem. Expert Systems with Applications 41(4, Part 1):1543–1550
29. UNCTAD (2016) United nations conference on trade and development - review of maritime transport