Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers & Industrial Engineering 50 (2006) 55–65
computers & industrial engineering
www.elsevier.com/locate/dsw

A multi-mode resource-constrained scheduling problem in the context of port operations

Luciano Lessa Lorenzoni [b,1], Hannu Ahonen [a,2], Arlindo Gomes de Alvarenga [a,*]

[a] Department of Computer Science, Federal University of Espirito Santo, Vitoria, Brazil
[b] Department of Computer Science, FAESA, Vitoria, Brazil

Received 30 October 2003; received in revised form 28 October 2005; accepted 17 November 2005
Available online 17 February 2006

Abstract

We discuss a problem of attending ships within agreed time limits at a port under the condition of the first come first served order. In addition, we indicate the use of the developed tool to support decisions at a strategic level with the objective of improving the attendance of the ships. The tool is based on a mathematical model of a multi-mode resource-constrained scheduling problem and on an extension of a differential evolution algorithm. We illustrate the implementation by computational tests with data generated on the basis of the characteristics of a real port environment.

Keywords: Port operations; Attending ships; Layday period; Multi-mode resource-constrained scheduling; Differential evolution algorithm; Local search

1. Introduction

Attending ships within the agreed time limits at a port depends essentially on the efficient use of the resources available for the port operations. Failing in this generally leads to high costs in the form of payments of penalties. In the case of the real port considered in this work, the ships have to be attended in the order of their arrivals, which implies that the attempts of optimizing the use of resources should be focused to the means of avoiding simultaneous or nearly simultaneous arrivals of ships competing with the same resources. This implies that the attention has to be paid to the mechanisms of defining the time limits for the attendance of the ships, usually expressed in terms of the so-called layday periods, which stipulate the earliest and latest dates the ships are supposed to arrive at the port. These periods, however, cannot be determined solely on the basis of the interests of the port, since they are defined on

* Corresponding author. Address: Departamento de Informatica, Universidade Federal do Espirito Santo, Centro Tecnologico, Av. Fernando Ferrari, Campus Universitario, Goiabeiras, CEP 29060-900 Vitoria, ES, Brazil. Tel.: +53 27 33352688.
E-mail addresses: luciano@faesa.br (L.L. Lorenzoni), hannu@inf.ufes.br (H. Ahonen), agomes@inf.ufes.br (A.G. Alvarenga).
[1] FAESA—Faculdades Integradas Espirito-Santenses—Campus I, Colegiado de Ciencia da Computacao, Rua Anselmo Serrat, 199, Ilha de Monte Belo, Cep 29040-410 Vitoria, ES, Brazil.
[2] Departamento de Informatica, Universidade Federal do Espirito Santo, Centro Tecnologico, Av. Fernando Ferrari, Campus Universitario, Goiabeiras, CEP 29060-900 Vitoria, ES, Brazil.

0360-8352/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2005.11.001

the basis of the negotiations between the shippers and the port operators. The objective of this work is to propose a computational tool for supporting this negotiation process.

The analysis of a proposal for layday periods is based on finding an efficient use of port resources for the attendance of the ships. The main difficulty here is that the arrival times of the ships are unknown. We propose two ways to find good solutions in spite of this difficulty. Firstly, given a proposal for the layday periods of a set of ships, we compute the 'ideal' arrival times for the ships minimizing the earliness and tardiness costs of the attendance. By this, we are able, at least, to eliminate totally infeasible proposals. Secondly, in order to have a more realistic view, we repeat the computations simulating the unknown real arrival dates of the ships by randomly generated date values. Thirdly, these computations can support certain planning tasks like the analysis of the impact of including new resources in the port environment or increasing the productivity rate in the port operations. In all these cases, we analyze the proposal by solving a scheduling problem, considered as a special case of a resource-constrained scheduling problem. In general, we have to solve a problem of this kind, when a set of tasks needs to be executed with resources available in some limited amounts. As defined in Blazewicz, Ecker, Pesch, Schmidt, and Weglarz (1996), scheduling means to determine the execution start times of the tasks attributing resources to the tasks until they all have been completed in the way that given restrictions have been obeyed and that an objective established in the problem context has been achieved in the best possible way.

Finding an efficient schedule for attending ships is a complex task, in which we have to consider the use of port resources including the berths, the canals of accessing the berths, equipments of loading and unloading, availability of the cargo to be loaded or stock area for unloading. In addition, the draft of a ship may restrict its entering to an access canal or a berth in certain time intervals due to the tidal conditions at the port. Similarly, some port resources may be temporarily unavailable, e.g. during a maintenance operation. Thus, it is useful to consider time windows for the availability of the resources.

The task of attending a ship, e.g. loading, usually requires a use of several resources like a combination of a canal of access, a berth and the same or another canal of access. There may be available several combinations of this kind. Each combination forms a processing mode for that task. The scheduling of this type turns out to be a special case of the multi-mode resource-constrained scheduling problem. For a review, see Brucker, Drexl, Mohring, Neumann, and Pesch (1999).

The way we consider resources in our approach differs from the one presented in Nishimura, Imai, and Papadimitriou (2001), in which the problem of allocating ships to berths is analyzed. Our approach makes it possible to include more than one step in the attendance of the ships at a port. For example, due to the different availabilities of port equipments, as observed in our case study, a ship may be unloaded in one berth and then loaded in another. A further difference is in assuming the first come first served regime of attendance, which in our case will be maintained, similarly to other authors referred in Nishimura et al. (2001).

It should be noted that the scheduling in our context concerns the use of resources available for port operations differing from the problem of ship scheduling (Christiansen, Fagerholt, & Ronen, 2004; Fagerholt, 2001).

Before entering the discussion on the solution procedure, we give a formalization of the scheduling problem associated to the problem of determining layday periods in Section 2. The solution procedure, based on the use of a differential evolution algorithm is presented in Section 3. The results of the computational tests are given in Section 4, while concluding remarks follow in Section 5.

2. Formalization of the problem

The problem of finding an efficient schedule based on a given proposal for layday periods can be formulated in the following manner.

Let R be a set of non-divisible and time-window-restricted resources consisting of berths and access canals. Each resource $r \in R$ is available at a given capacity c_r only within $l_r \geq 1$ disjoint time intervals $[a_{r,1},b_{r,1}],[a_{r,2},b_{r,2}],\ldots,[a_{r,l_r},b_{r,l_r}]$ such that $0 \leq a_{r,1} < b_{r,1} < a_{r,2} < b_{r,2} < \ldots < a_{r,l_r} < b_{r,l_r}$. Resources of this type have been characterized in the literature as partially renewable resources (Böttcher, Drexl, Kolisch, & Salewski, 1996).

The requests of operations coming from the shippers are represented as elements of the set of tasks indexed by set $T = \{1,2,\ldots,n\}$. A predefined precedence order between the tasks is denoted by $\prec$. The following data is associated to each task $j \in T$:

- a time interval $[\bar{l}_j,\bar{L}_j]$ proposed by the shipper as a layday period;
- a release time s_j of task j corresponding to the estimated arrival time of the ship associated to this task;
- coefficients ρ_j and γ_j that define the earliness and lateness penalties of task j with respect to its layday period, respectively;
- a set M_j of processing modes, that is, a set of all possible combinations of the resources sufficient for attending the request j. Each combination is given as an ordered v_m-tuple $R_m = \left(r_1^m, r_2^m, \ldots, r_{v_m}^m\right)$, where v_m is the number of resources in mode m;
- $p_{j,r,m}$ indicates the time request j occupies resource r when executed in mode m;
- $q_{j,r,m}$ indicates the quantity task j needs resource r when executed in mode m.

It should be noted that the earliness cost above does not correspond to a real operation cost of the port. It is included in the objective function in order to force the execution start times of the tasks to lie inside the corresponding layday periods.

Since the port resources are utilized successively, one after another, the total execution time of task j in mode m is greater than or equal to the sum of the resource utilization times $p_{j,r,m}$ over all resources r in R_m depending on the availability of resources due to their time windows. This differs from the conventional formulation of a multi-mode resource-constrained scheduling problem, in which resources in a given mode are used in parallel.

The sequential use of resources makes it necessary to compute the start times of the resource uses individually. For this, we define $\tau_{j,r,m}(t)$ as the start time of the use of resource r in mode m for task j, assuming that the execution of the task starts at time t.

The conditions for the start times of resource uses are related to the time windows of the resources. We assume that the modes m available for task j are feasible in the sense that, for any resource r in m and for any time instant t within the planning horizon, the resource will be available at some future time instant $\tau \geq t$, that is, there exists an interval $[a_{r,i},b_{r,i}]$ with $a_{r,i} \leq \tau$ and $\tau + p_{j,r,m} \leq b_{r,i}$. For simplicity, we define for each $j \in T$ and $m \in M_j$ an 'artificial' resource r_0^m with processing time $p_{j,r_0^m,m} = 0$.

Now, the start time $\tau_{j,r,m}(t)$ of the use of resource r in mode m for task j with execution start time t is successively defined in the following way:

$$\tau_{j,r_0^m,m}(t) = t; \tag{i}$$

$$\tau_{j,r_k^m,m}(t) = \min\left\{\tau | \tau_{j,r_{k-1}^m,m}(t) + p_{j,r_{k-1}^m,m} \leq \tau \,\&\, \exists\, i \text{ such that } a_{r_k^m,i} \leq \tau + p_{j,r_k^m,m} \leq b_{r_k^m,i}\right\} \quad \text{for } k = 1,\ldots,v_m. \tag{ii}$$

The two possible cases occurring in the computation of the minimum in Eq. (ii) are depicted in Figs. 1 and 2. In the first case, the use of resource r_k^m can start immediately after resource r_{k-1}^m is freed, while in the second case, task j has to wait until a time window of resource r_m^k becomes available.

The decision variables in the model consist of execution start times t_j of tasks $j \in T$ and the binary variables $y_{j,m}$, $m \in M_j$, $j \in T$, with $y_{j,m} = 1$ if mode m is selected for the execution of task j.

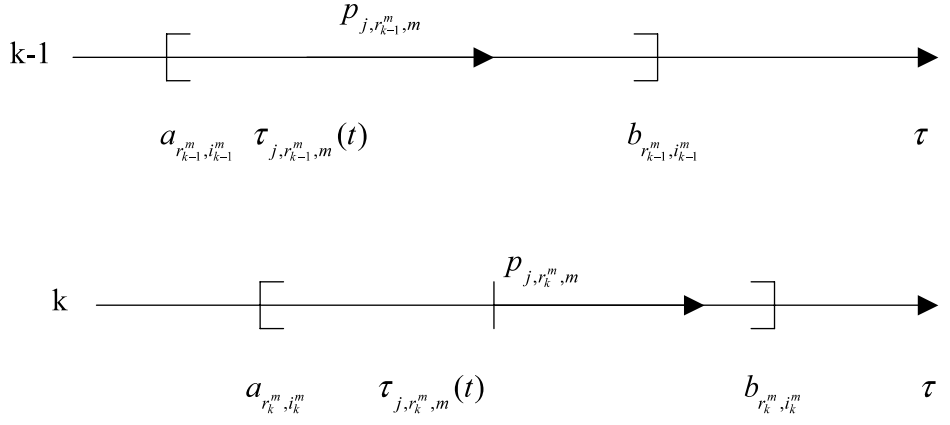

Fig. 1. Case 1—no interruption in the execution of task j.

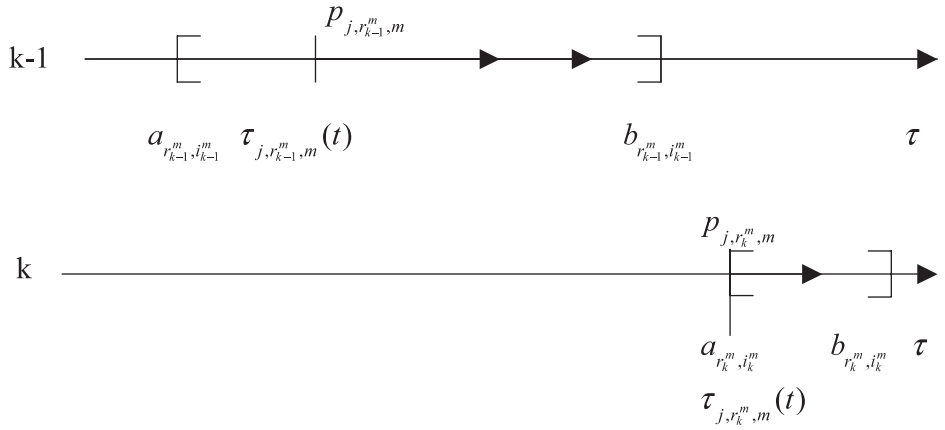

Fig. 2. Case 2—an interruption in the execution of task j.

As an abbreviation, we denote the set of tasks using resource $r \in R$, at a given time instant $t \in \aleph$, as $U(r,t) = \{j \in T | y_{j,m} = 1 \& \tau_{j,r,m}(t_j) \leq t < \tau_{j,r,m}(t_j) + p_{j,r,m}, m \in M_j\}$.

The objective function is defined with respect to the time intervals $[\bar{l}_j, \bar{L}_j]$ given for tasks $j \in T$. The objective is to minimize the earliness and tardiness costs of the tasks with respect to these intervals.

The formulation of the scheduling problem is now the following

$$\min \sum_{j \in T} \rho_j (\bar{l}_j - t_j)^+ + \gamma_j (t_j - \bar{L}_j)^+, \tag{1}$$

subject to

$$\sum_{j \in U(r,t)} \sum_{m \in M_j} y_{j,m} q_{j,r,m} \leq c_{r,t}, \quad \forall\, r \in R, t \in \aleph \tag{2}$$

$$\sum_{m \in M_j} y_{j,m} = 1, \quad \forall\, j \in T \tag{3}$$

$$s_j < t_j, \quad \forall\, j \in T \tag{4}$$

$$\sum_{m \in M_j} \left(\tau_{j,r^m_{v_m},m}(t_j) + p_{j,r^m_{v_m},m} \right) y_{j,m} \leq t_{j'}, \quad \forall\, j,\ j' \in T, \text{ with } j < j'. \tag{5}$$

$$y_{j,m} \in \{0,1\}, \quad \forall\, m \in M_j,\ j \in T \tag{6}$$

and

$$t_j \in \aleph, \quad \forall\, j \in T. \tag{7}$$

Inequalities 2 establish the restrictions in the use of resources. Equalities 3 indicate that exactly one of the execution modes available for task $j \in T$ must be selected. Inequalities 4 express the requirement that the execution of task j cannot start before its release time s_j. The precedence relations in 5 stipulate that execution of task j' cannot be started before all of its predecessors have been finished. The expression at the left hand side of the inequality picks the finish time of the use of the last resource in the selected execution mode of each predecessor j of task j'.

3. Solution procedure

It is worth of observing that the mathematical model proposed in Section 2 for solving the scheduling problem associated to the determination of layday periods is non-linear in various aspects, like by the introduction of the set $U(r,t)$ and by the definition of the precedence restrictions. Due to the complexity of the model, it is difficult to

conceive an exact algorithm which could efficiently find an optimal solution for it. Clearly, any straightforward enumerative search would lead to an intractable number of solution evaluations. Consequently, a natural way of solving the problem is to rely on some heuristic procedure. For this, we have selected an evolutionary algorithm, Differential Evolution Algorithm (Storn & Price, 1997), as a basis of our solution procedure. The algorithm is usually applied to the problems of global optimization with continuous decision variables, but with specific coding techniques it can be adapted to discrete combinatorial optimization problems, too. The differential evolution algorithm works with a population of solutions coded as vectors of real numbers trying to find better individuals with the help of a procedure that integrates the usual genetic operations of crossover and mutation. Here for each individual $\vec{x}_i$, three other individuals $\vec{x}_{r_1}$, $\vec{x}_{r_2}$ and $\vec{x}_{r_3}$ are selected at random, and the difference determined by the first two is used as a mutation operator applied to the third for generating another parent in addition to individual $\vec{x}_i$. More precisely, this parent, denoted by $\vec{v}_i$, is computed as

$$\vec{v}_i = \vec{x}_{r_3} + F(\vec{x}_{r_1} - \vec{x}_{r_2}), \tag{8}$$

where F is a predefined proportionality parameter. A simple crossover operation is defined between the two parents by the random choice of components from one or another individual. This choice is controlled by a parameter CR. Finally, the generated new individual, a candidate solution, is used to replace individual $\vec{x}_i$ if its quality is better than that of $\vec{x}_i$. A more detailed description of the Differential Evolution algorithm can be found in Corne, Dorigo, and Glover (1999).

In our case, the mapping between a solution and an individual, will be established between a schedule, expressed as start times and modes selected for the tasks, and vectors of real numbers. We apply here a modification of a commonly used random key representation. We call the resulting representation a sequential random key representation and define it in terms of a pair $(\vec{\alpha},\vec{\beta})$ of vectors. Each vector has real value components belonging to the interval [0,1], the number of which is equal to the number of tasks to be scheduled. The kth component α_k of vector $\vec{\alpha}$ is used to select the kth task for the schedule. The selection is made between the tasks in the set E_k of eligible tasks, defined as a set of tasks with no unscheduled predecessors. For this, the interval [0,1] is divided into n_k subintervals of equal lengths, where n_k is equal to the number of elements in set E_k. To each subinterval is associated an element of E_k according to a predefined order of enumeration. With this, a value of α_k is used as a pointer to one of these subintervals, i.e. to one of the eligible tasks. For example, if E_k consists of four tasks, the subintervals are [0,0.25), [0.25,0.5), [0.5,0.75), [0.75,1.0] and if $\alpha_k = 0.67$, the third task would be selected. This procedure is illustrated in Fig. 3.

Similarly, the kth component of vector $\vec{\beta}$ is used for selecting one of the modes in M_j after the selection of the kth task in the schedule. Again, the interval [0,1] is divided into subintervals, the number of which equals to the number of modes in M_j, and the component β_k is used to select one of the intervals and consequently one of the modes. As in the task selection, a predefined enumeration of modes within each set M_j is assumed.

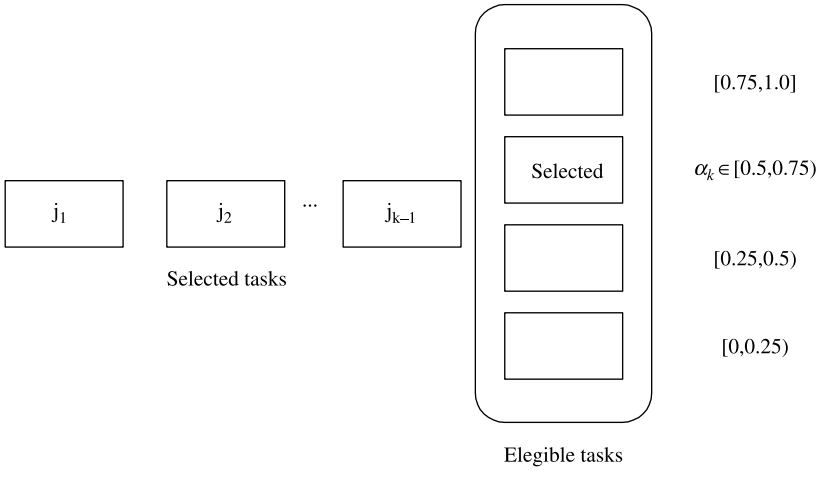

Fig. 3. Selection of the kth task.

```
Algorithm DE Scheduling Algorithm;
    Input;
    Generation of the Initial Population;
    Apply Local Search to the feasible individuals in the population;
    While (stopping criterion not satisfied) Do
        Generate new candidate solutions by DE Algorithm;
        Apply Local Search to the feasible candidate solutions;
        Select the individuals for the next iteration;
        If (the best solution was not improved during L last iterations)
            Then diversify the population;
    End-While;
End.
```

Fig. 4. The solution procedure.

The solution procedure for the scheduling problem summarized in Fig. 4 contains the differential evolution algorithm as its basic module in the role of a 'genetic machine' responsible for generating new individuals into the population. Other two components of the solution procedure consist of the modules of local search and diversification.

The input of the algorithm consists of parameters F, CR as well as the parameters indicating the population size, the number of trials until the diversification step is applied and the stopping criterion. The initial population is generated randomly.

The local search algorithm is applied to all feasible individuals in the population both before starting the iteration loop and then inside it. It is based on the operation of multi-mode left shift introduced by Sprecher, Hartmann, and Drexl (1997). In a multi-mode left shift, we try to complete a scheduled task earlier by selecting some other processing mode without violating the resource restrictions, without modifying the processing modes and without increasing the completion times of other tasks.

The diversification procedure tries to focus the search to less explored areas in the search space. For this, instead of selecting tasks and modes by the use of vectors $\vec{\alpha}$ and $\vec{\beta}$, an alternative random selection mechanism is used. The computation of the selection probabilities is based on task and mode priorities determined by two frequency matrices constructed during the iteration steps. Each element $TF_{j,k}$ of the first matrix stores the number of times task j was selected as the kth task in the schedules, while an element $MF_{j,m,k}$ of the second matrix indicates the number of times the kth task j was processed in mode m. The task priorities are computed as

$$u(j,k) = 1 - \frac{TF_{j,k}}{\sum_{j' \in E_k} TF_{j',k}}, \tag{9}$$

$$v(j,m,k) = 1 - \frac{MF_{j,m,k}}{\sum_{m' \in M_j} TF_{j,m',k}}. \tag{10}$$

As it can be observed, the less frequent choices will be given a higher priority. Now, the diversification procedure consists of applying two rules of random selection at a randomly selected kth step:

Rule 1. (Selection of a new task) Select task $j \in E_k$ with the probability $p(j,k)$, where $p(j,k) = \frac{u(j,k)}{\sum_{j' \in E_k} u(j',k)}$.

Rule 2. (Selection of a new mode) Select mode $m \in M_j$ with the probability $p(j,m,k)$, where $p(j,m,k) = \frac{v(j,m,k)}{\sum_{m' \in M_j} v(j,m',k)}$.

4. Computational tests

Firstly, for the purpose of validation, the implementation of the solution procedure was tested with a set of benchmark problems given in Kolisch & Sprecher (1997). The results obtained in these tests showed to be at the level of the best ones reported in the literature. A detailed discussion on the results is included in Lorenzoni et al. (Unpublished manuscript).

The computational tests were conducted in the context of a Brazilian port situated at Vitória—ES. This port, Port of Tubarão, can be characterized as an ore terminal, owned and operated by a single company. The contracts of affreightment consist of short, medium and long-term contracts. The last ones may have durations up to 5 years.

The objective of the tests was to analyze the performance of the implemented tool in different practical situations occurring in the context of negotiating layday periods of ships. Since no real data was available by commercial reasons, a randomly generated data based on the characteristics of the port environment was used. Due to the complexity of the problem, a comparison to the optimal solutions was not possible. The validation tests conducted with the known test problems make us conclude, however, that the quality of the results would be sufficiently good for practical purposes, too.

The port considered for the tests has three berths and one access canal shared by all of them. Table 1 shows some of their characteristics. The last column in the table indicates the maximum capacity a ship may have for being attended at a berth.

In addition, the tide prediction tables provided by the port operators were consulted for computing the time intervals with respect to the availability of the resources. No restrictions due to the maintenance operations were assumed.

We considered four types of cargo to be loaded, in this case types of ore, that were extra fine (X), fine (F), granulated (G) and pellets (P). The task of loading a ship requires an allocation of two types of resources, berth and access canal. The loading time depends both on the particular berth and type of cargo as shown in Table 2. The time of passing the access canal was considered to be equal to 2 hours. Although this time is constant, it has to be considered separately, since the access canal forms a resource required only at the beginning and the end of the task sequence of each mode.

It may be observed in Table 2 that all the berths can handle all the types of the cargo, but that the processing times are different. This implies that the berths have to be distinguished one from another.

Four classes of problems were defined, differing from each other in the number of task requests. We refer to these classes as S100, S200, S300 and S400, consisting of 100, 200, 300 and 400 requests, respectively. For each class, three sets of data was generated with repeated random choices of layday periods, cargo types and cargo quantities. We used the uniform probability distribution on the range of each generated value.

The layday period of each request was placed at a randomly selected date within a period of 1 year and its length was selected out of the interval of 8–15 days. The cargo type of a request was chosen to be one of the four types and the cargo quantity was picked out of the interval of 40,000–180,000 tons.

The estimated arrival times of the ships were selected as uniformly distributed random values in the intervals of the corresponding layday periods. The generation of the values was repeated five times for each of the three sets of data. Thus, the total number of problem instances used in the tests was equal to 60.

The resource vector corresponding to each processing mode is of the form (C1, ⟨berth⟩/⟨cargo⟩, C1), where C1 is the access canal, ⟨berth⟩ one of the berths 1S, 1N, and 02, and ⟨cargo⟩ one of the cargo types X, F, G, and P. Thus, there are 12 processing modes in total. The processing times for each mode were calculated on the basis of the time to pass the access canal and the loading times given in Table 2.

We constructed three test suites, the first two of which were used to evaluate the feasibility of a given set of layday period requests, while the third aims at discovering the consequences of an increase in the port's resources or in its rate of productivity.

In the first test suite, we analyze the attendance of the requests by computing the 'ideal' arrival times of the ships for a planned set of layday periods. For this, we assume that the attendance of the ships may start at any moment, that is, we relax inequality 3 in the mathematical model of the problem. We then define the 'ideal' arrival times of the ships as the start times of their attendance. The test of this kind could be used in practice as a quick test to eliminate totally infeasible proposals. The resulting arrival times can also be considered as proposals for the center points of new layday periods during the negotiation process.

Table 1
Technical characteristics of the resources

Resource	Length (m)	Draft (m)	Operation type	Tonnage
1S-pier one south	285	15.5	Embarkation	120,000
1N-pier one north	301	17	Embarkation	180,000
02-pier two	350	20	Embarkation	365,000
C1-access canal	–	25	–	–

Table 2
Loading times related to berths and cargo types

Berth	Loading time t/h			
	X	F	G	P
1S	3500	4800	4500	4000
1N	4000	6000	5500	5000
02	6500	7000	6500	6000

The second test suite provides us with a more detailed and realistic analysis, in which we executed the computations simulating the unknown real arrival dates of the ships by randomly generated date values. We used the uniform probability distribution on the interval defined by the corresponding layday period. The third suite can be considered as a part of strategic planning, in which we vary either the amounts of resources available at the port or the times of loading or unloading. In these test suites, we fixed the order of the tasks as defined by the generated arrival times of the ships, and executed only the selection of resource modes. Thus, we respected the required first come first served order. Although the objective of the solution procedure was to minimize an earliness–tardiness cost, we express the final results as percentages of unattended requests at the end of their layday periods related to the number of all requests.

After a calibration process of the parameters, we adopted the values $F = 0.96$, $CR = 0.85$ and the population sizes 100, 100, 140 and 160 for the classes S100, S200, S300 and S400, respectively.

The results of the first test suite showed that the generated layday periods were feasible in the sense that the attendance of all requests was possible, if assumed that the ships would arrive at the port in the first day of the period.

Fig. 5 illustrates the results of the second test showing the average percentages of the unattended requests at the end of the layday periods established for each class of problems. The averages were computed over the 15 problem instances created for each class. These values were 0.3, 2.6, 7.4 and 13.5% for the classes S100, S200, S300 and S400, respectively.

Although the optimal values for these cases are not known, it can be noted that these lie between 0 and the obtained values, since in the ideal case all requests would be attended.

Fig. 6 shows the average values of the deviations of the start times of the unattended requests from the finish times of their layday intervals. The obtained values for the cases S100, S200, S300 and S400 were 0.5, 1.1, 1.9 and 3.3 days, respectively.

The test with the simulated arrival dates shows clearly, that it is not sufficient to analyze the feasibility of a set of layday periods on the basis of the assumption that the ships would arrive at their 'ideal' time instant. Naturally, the more requests we have, the more critical will be the need of planning a good combination of layday periods.

The third test suite consists of two parts corresponding to the issues of strategic planning discussed above.

Firstly, we analyzed the alternative of constructing new berths simulating three scenarios, with high, medium and low allowed tonnages in Scenario 1, Scenario 2 and Scenario 3, respectively. A tonnage, a quantity of cargo expressed in tons, was considered high, if it exceeds 300,000 t, medium if it lies between 200,000 and 300,000 t and low if it remains under 200,000 t. The results obtained for class S100 showed that all requests could be attended in all of the three scenarios. The average percentages of unattended requests in Scenario 1 were 1.1, 3.8 and 7.8% for classes

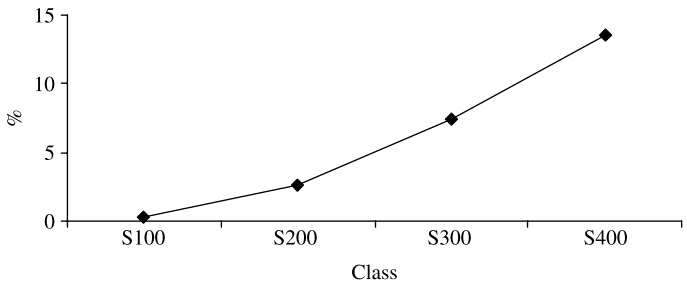

Fig. 5. Average percent of the unattended requests at the end of their layday periods.

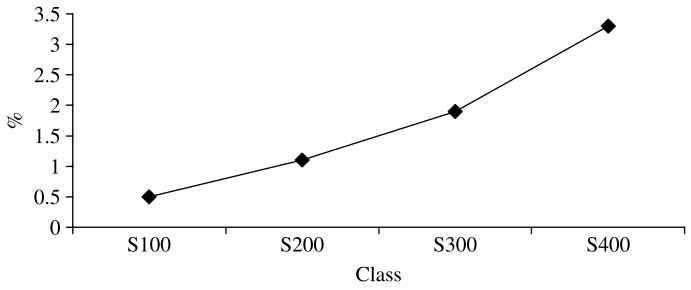

Fig. 6. Average deviations of the start times of unattended requests.

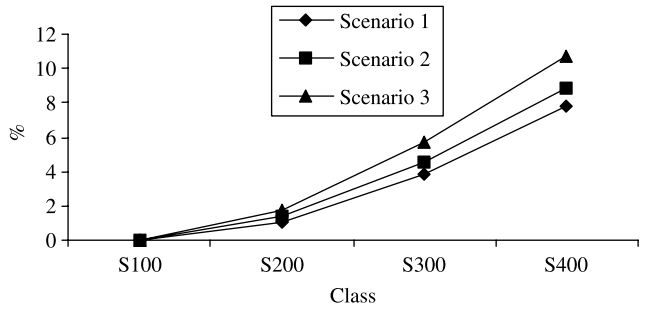

Fig. 7. Average percent of unattended requests at the end of their layday periods with the inclusion of new berths.

S200, S300 and S400, respectively. In Scenario 2, these values were 1.4, 4.5 and 8.8%, while in Scenario 3 the values were 1.7, 5.7 and 10.7%. Fig. 7 shows the results of the tests with the three scenarios.

Secondly, we considered a possibility to increase the efficiency of the port operations. Again, we conducted the tests with three scenarios, in each of which the productivity rate of one of the berths was increased by 30% and other two were kept unmodified. In Scenario 4, berth 02 was modified, in Scenario 5, berth 1S and in Scenario 6, berth 1N. Again, all requests were attended in the case of class S100 in all these scenarios. The average percentages of the unattended requests were 0.8, 4.7 and 7.8% for S200, S300 and S400, respectively, in Scenario 4. In Scenario 5 and Scenario 6, these values were 1.4, 4.9, 8.2 and 1.7, 5.9, 11.0%, respectively. Fig. 8 presents these results.

The results of the three last tests are summarized in Table 3. As expected, comparison of the results indicates that inclusion of a new berth or increase in the productivity rate of one of the berths will improve the attendance of the ships in a considerable manner. Especially, an inclusion of a high tonnage berth (Scenario 1) or an increase of the productivity rate at berth 02 (Scenario 4) would be the most efficient decision.

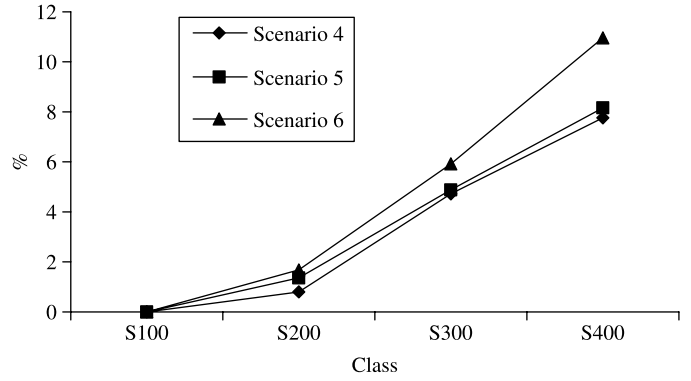

Fig. 8. Average percent of unattended requests at the end of their layday periods, with the increase in the productivity rate.

Table 3
Comparison of the results

	Orig.	New berths			Productivity		
		Scen. 1	Scen. 2	Scen. 3	Scen. 4	Scen. 5	Scen. 6
S100	0.3	0.0	0.0	0.0	0.0	0.0	0.0
S200	2.6	1.1	1.4	1.7	0.8	1.4	1.7
S300	7.4	3.8	4.5	5.7	4.7	4.9	5.9
S400	13.5	7.8	8.8	10.7	7.8	8.2	11.0

Table 4
Limits for the number of evaluated solutions and average computation times

Class	Limit	Time
S100	10,000	1 s
S200	15,000	7 min
S300	25,000	24 min
S400	32,000	50 min

The computations were executed with a Pentium III 1 GHz processor. We defined for each problem class a limit for the number of evaluated solutions. The objective was to adjust the number of evaluations so that the quality of the obtained solutions would be approximately equal in all problem classes. In the case of the first test suite, the execution time of the differential evolution algorithm was approximately 1 s. The execution times for the second test suite are shown in Table 4.

Besides, the natural increase in the computational effort with increasing problem size, it can be observed that the growth in the average computation times is in concordance with results shown in Fig. 5, since they greatly depend on the stoppin criterion of the algorithm. Thus, in the case of problem class S100, the algorithm almost always stopped before attaining the maximum number of evaluations, but in the case of class S400 the limit was achieved in approximately 13.5% of cases.

5. Conclusions

We discussed a problem of determining layday periods for ships under the condition that once the ships have arrived at the port, they have to be attended in the first come first served order. It is advantageous for the port to suggest shippers layday periods, which do not cause any excessive overloading in the execution of port operations. For this, we proposed a two-step procedure for verifying, whether a given set of layday periods would be feasible, that is, not causing a delay in the attendance of the ships at the end of their layday periods. The first step consists of determining quickly the ‘ideal’ arrival times for the ships by minimizing the earliness and lateness costs. The second step offers a more realistic analysis, in which we simulate the arrival times of the ships with randomly generated values and then verify, whether the resources can be selected without causing delays in the attendance of the ships.

A further application of our approach aims at supporting decisions at a strategic level giving answers to questions like whether constructing a new berth or increasing the productivity rate of an existing berth would bring improvements in the attendance of the ships.

The computational procedure used in our proposal is based on a formulation as a special case of a scheduling problem known as a multi-mode resource-constrained scheduling problem. Since the problem is, in general, not efficiently solvable with some exact approach, an algorithm based on differential evolution algorithm was selected to serve as a solution procedure. We implemented the algorithm for the computations based on the characteristics of a real port environment. Thus, we were able to illustrate the proposed approach with randomly generated sets of layday periods and execution times of port operations.

For illustration, the generation of the data was based on the use of the uniform random distribution. Thus, a future implementation of the proposed tool could be improved by the use of distributions based on a real experimental data. It may be expected that the real arrival times of the ships are mostly concentrated to the middle of the layday periods.

In a further application of our work, the port operator might propose, on the basis of the result of computation, modifications on the travel schedules of the ships, e.g. by informing that an early arrival of a ship may lead to a long waiting time before its attendance. More generally, a future work might integrate the determination of the layday periods with routing and scheduling of the ships (Brown, Lawphongpanich, & Thurman, 1994; Christiansen et al., 2004; Fagerholt, 2001; Kao & Lee, 1996). Although this may not be generally achievable due to the planning of these operations performed by different organizations, this might be somehow possible in the special case of the port referred in this work, since the company that operates the port also acts as a shipper. In this case, the proposed tool would be transformed to an internal planning tool of the company. Of course, it would not make sense to designate the tardiness cost in the objective function as a real penalty, but interpret it, similarly to the earliness cost, as a fictitious cost used in the computation of the task start times.

A future extension of our work would be to take in consideration other types of resources associated to special characteristics of the port operations. An example of this is the need of material handling resources in transshipment of containers as reviewed in Vis & de Koster (2003).

References

Blazewicz, J. J., Ecker, K. H., Pesch, E., Schmidt, G., & Weglarz, J. (1996). *Scheduling computer and manufacturing processes*. Berlin: Springer.

Böttcher, J., Drexl, A., Kolisch, R., & Salewski, F. (1996). Project scheduling under partially renewable resource constraints. *Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel,* no. 398.

Brown, G. G., Lawphongpanich, S., & Thurman, K. P. (1994). Optimizing ship berthing. *Naval Research Logistics*, *41*, 1–15.

Brucker, P., Drexl, A., Mohring, R., Neumann, K., & Pesch, E. (1999). Resource constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, *112*, 3–41.

Christiansen, M., Fagerholt, K., & Ronen, D. (2004). Ship routing and scheduling: Status and perspectives. *Transportation Science*, *38*(1), 1–18.

Corne, D., Dorigo, M., & Glover, F. (1999). *New ideas in optimization*. New York: McGraw-Hill.

Fagerholt, K. (2001). Ship scheduling with soft time windows: An optimisation based approach. *European Journal of Operational Research*, *131*, 559–571.

Kao, C., & Lee, H. T. (1996). Discrete time parallel-machine scheduling: A case of ship scheduling. *Engineering Optimization*, *26*, 287–294.

Kolisch, R., & Sprecher, A. (1997). PSPLIB—A project scheduling library. *European Journal of Operational Research*, *112*, 205–216.

Lorenzoni, L. L., Alvarenga, A. G., & Ahonen, H. T. (2002). *A differential evolution algorithm for multi-mode resource-constrained scheduling problems.* Department of Computer Science, Federal University of Espírito Santo, Vitória-ES, Brazil. Unpublished manuscript.

Nishimura, E., Imai, A., & Papadimitriou, S. (2001). Berth allocation planning in the public berth system by genetic algorithms. *European Journal of Operational Research*, *131*, 282–292.

Sprecher, A., Hartmann, S., & Drexl, A. (1997). An exact algorithm for project scheduling with multiples modes. *OR Spektrum*, *19*, 195–203.

Storn, R., & Price, K. (1997). Differential evolution—A fast and efficient heuristic for global optimization over continuous space. *Journal of Global Optimization*, *11*, 341–359.

Vis, I. F. A., & de Koster, R. (2003). Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research*, *147*, 1–16.