Computers & Industrial Engineering 97 (2016) 1–14

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A hierarchical approach to solve a production planning and scheduling problem in bulk cargo terminal

Gustavo Campos Menezes [a,b,*], Geraldo Robson Mateus [a], Martín Gómez Ravetti [c]

[a] Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Av. Antônio Carlos 6627, CEP 31270-010, Belo Horizonte, MG, Brazil
[b] Departamento de Eletroeletrônica e Computação, Centro Federal de Educação Tecnológica de Minas Gerais, Avenida Doutor Antônio Chagas Diniz 655, CEP 32210-160, Cidade Industrial, Contagem, MG, Brazil
[c] Departamento de Engenharia de Produção, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627, CEP 31270-901, Belo Horizonte, MG, Brazil

ARTICLE INFO

Article history:
Received 6 November 2015
Received in revised form 22 March 2016
Accepted 9 April 2016
Available online 19 April 2016

Keywords:
Production planning
Scheduling
Hierarchical approach

ABSTRACT

The integration of planning and scheduling decisions is key to obtain an efficient and reliable production operation in a modern manufacturing and service company. In this work we propose a mathematical model for this integration, the model is defined considering logistic operations at bulk port, however is generic enough to be adapted to several situations. The integration takes place in a hierarchical scheme where the problems exchange data and they are solved through a commercial solver and heuristics. When scheduling is not feasible, capacity information is forwarded to production planning to adjust or indicate the use of new tasks. The model and algorithms are validated considering data from a real case. Computational results show the efficiency of the approach, producing strong bounds for large instances.

1. Introduction

In production systems, production planning and scheduling problems are critical for profitability of companies, correct use of resources and to meet deadlines. These problems are applicable in a broad range of sectors, such as the casting industry (Camargo, Mattiolli, & Toledo, 2012; de Souza, Jr, Bretas, & Ravetti, 2015), the food industry (Rocco & Morabito, 2014), and cargo transportation in port terminals (Robenek, Umang, & Bierlaire, 2014). Even though planning and scheduling belong to two different decision levels (from strategic to operational), there is a strong relation between them and there is extensive literature on solution models and strategies (Drexl & Kimms, 1997; Meyr & Mann, 2013; Phanden, Jain, & Verma, 2013; Ramezanian, Saidi-Mehrabad, & Teimoury, 2013). Published strategies can be divided into hierarchical and integrated approaches.

In a broad sense the production planning decides when and how many products must be produced, and the decisions are usually associated to cost trade-offs. Instead, scheduling problems take into account shop-floor settings and determine how the production must be executed. Their objectives are usually time-related. The independent optimization of these problems can clearly lead to non-optimal solutions, thus the need to combine the decisions levels. Integrated methods consider both problems simultaneously; that brings better solutions in exchange of a greater computational complexity. Another approach is a heuristic procedure, where in a hierarchical fashion the production planning and scheduling problems and solutions exchange data.

The problem motivating this research can be defined as follows: lets consider a variety of products arriving at a logistics terminal (supply), they need to be transferred to meet the demand or to a local storage area. To make this transfer, products need a feasible route of equipment. On the one hand, the planning problem must take decisions regarding when to move the material and where to move it. On the other hand, the scheduling problem deals with making the planning feasible, that is, determining a route of equipment to be use at each time period. Different routes have different capacities. They may share equipment creating conflicts when used during the same time period.

The focus of our work and the main contributions of this article are related to the integrated solution methodology to deal with a complex problem with real size instances. In this manuscript, we propose the use of a hierarchical framework to solve a production planning and scheduling problem for the delivery of products. The methodology uses a combination of heuristics and mathematical formulations; the novelty of the method is on the combination of this algorithms to deal with the trade-off between medium and short term decisions. Moreover, based on the scheduling solution

* Corresponding author at: Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Av. Antônio Carlos 6627, CEP 31270-010, Belo Horizonte, MG, Brazil. Tel.: +55 31 3409 5860; fax: +55 31 3409 5858.

E-mail addresses: gcm@dcc.ufmg.br (G.C. Menezes), mateus@dcc.ufmg.br (G.R. Mateus), martin.ravetti@dep.ufmg.br (M.G. Ravetti).

http://dx.doi.org/10.1016/j.cie.2016.04.007
0360-8352/

additional constraints are generated to strengthen the production planning. To link this problem with two levels of information we use the capacity of the routes.

In addition, to this contribution, the hierarchical framework, and a mathematical model are built to address a real storage and transportation problem, that occurs in a Brazilian bulk terminal. This problem is also common in iron ore port terminals and has not been fully investigated in the literature. The experiments are validated considering data from a real case and the computational results show the effectiveness of algorithms and model.

The remainder of this article is structured as follows: Section 2 presents the literature review. Section 3 defines the problem on which a mathematical programming model is based. Section 4 presents some sets and variables used in the mathematical formulation (the complete model is available in the appendix). Section 5 discusses the solution strategy applied and the main algorithms developed. Section 6 is dedicated to computational results and the manuscript ends with conclusions and future research directions.

2. Literature review

The interaction (integrated or hierarchical form) between Planning and Scheduling is not a new concept, and various efforts have been made toward this goal, such as in Ozdamar and Yazgac (1999), Meyr (2000), Wu and Ierapetritou (2007), Gaglioppa, Miller, and Benjaafar (2008), Mateus, Ravetti, Souza, and Valeriano (2010), Kis and Kovcs (2012), and, more recently, Meyr and Mann (2013) and Wolosewicz, Dauzre-Prs, and Aggoune (2015). You, Grossmann, and Wassick (2011) and Calfa, Agarwal, Grossmann, and Wassick (2013) also address integrated problems. In You et al. (2011), it is investigated an integrated production problem, whose goal is to determine at each period which products to manufacture, as well as to establish an optimal capacity modification plan, such that future demand is satisfied. Calfa et al. (2013) investigate the integration of Planning and Scheduling of a Network of Batch Plants. The problem is to define the amount of products to be produced in each time period, the allocation of products to batch units and the detailed timing of operations and sequencing of products.

The solution strategies adopted by the works were: Bilevel Decomposition and Lagrangean Decomposition in You et al. (2011), and Bilevel and Temporal Lagrangean (Calfa et al., 2013). These approaches have succeeded in solving large-scale industrial problems. Although the problems considered in those two works are different from the one analyzed in this article, the solution approach is similar. They deal with real and complex industrial problems and explore decomposition and communication mechanisms between the subproblems. The strategies proposed (Bilevel and Lagrangian) can also be seen as hierarchical, since the problems are decomposed and solved separately.

As previously discussed, the central problem study in this article involves the flow of products between supply nodes, storage areas and demand nodes. In this sense, the primary contributions from the literature are related to the product flow in bulk cargo terminals (iron ore, coal, grains). The references highlighted below are related to mathematical models and exact and heuristic algorithms for problems in this sector.

Bilgen and Ozkarahan (2007), study the problem of blending and allocating ships for grain transportation. The authors develop a mixed-integer linear programming model with constraints involving blending, loading, transportation, and storage of products. Conradie, Morison, and Joubert (2008) address the optimization of the flow of products (in this case coal) between mines and factory. Kim, Koo, and Park (2009) study the allocation of products in the stockyard. This problem is solved using a mixed-integer programming model. Barros, Costa, Oliveira, and Lorena (2011) develop an integer linear programming model for the problem of allocating berths in conjunction with the storage conditions of the stockyard. Solutions are obtained using optimization packages and Simulated Annealing. Boland, Gulezynski, and Savelsbergh (2012) address the problem of managing coal stockpiles in Australia. In the study, it is necessary to choose which equipment will be used for transporting goods to be piled in the stockyard (preferably near the berth where the ship will be loaded), and how to synchronize the whole process. Singh et al. (2012) present a mixed-integer programming model for the problem of planning the capacity expansion of the coal production chain in Australia. The model seeks alternatives to expand capacity to fulfill the demand while minimizing infrastructure costs and demurrage. Finally, Robenek et al. (2014) proposes an integrated model for the integrated berth allocation and yard assignment problem in bulk ports, with solutions obtained by a branch and price algorithm.

Although these research works address various important aspects of the challenges found in bulk cargo terminals, we did not find articles investigating the integration of product flow and scheduling routes. Such problematic is very usual and must be solved in several bulk terminals.

3. Problem description

The port terminal under study possesses several types of equipment for loading iron ore onto ships: car dumpers, conveyor belts, ore stackers, reclaimers and ship loaders. Iron ore is the main commercialized product, and it is the only product considered in this work. There are primarily three types of iron ore being handled: lump, sinter and pellet. Several other products can be derived from these raw materials and differ in their chemical and physical characteristics.

To better understand the planning and scheduling problem consider the following scenario. There is a set of supply nodes or reception subsystem, where products are available for transportation, storage nodes or stockyards and demand nodes or delivery subsystem (points of shipping products). Specialized equipment with predefined capacities is used to transport the products within the network. An equipment route between nodes has a given capacity and handle one product at a time. Fig. 1 provides a schematic representation of the problem.

The number of routes is limited and they may share equipment. Thus, if two different products are assigned to routes sharing equipment, these routes must be active at non-overlapping intervals. Fig. 2 shows a case where two routes (routes 1 and 2) share the same equipment.

The stockyard subsystem consists of large areas for storage. Each storage area is further subdivided into smaller subareas called storage blocks. The dimensions of each storage block can vary and

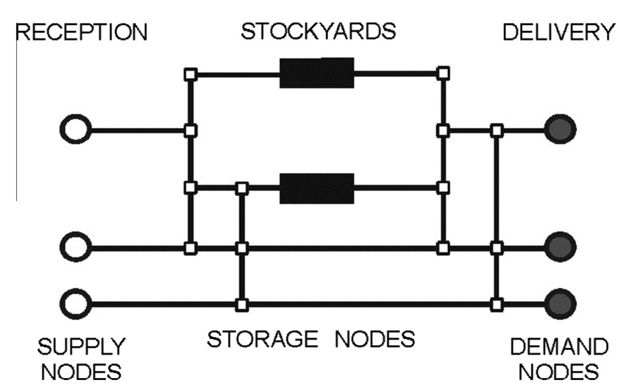

Fig. 1. Reception, stockyards and delivery systems.

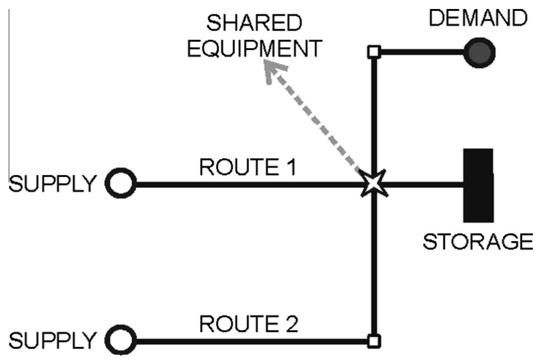

Fig. 2. Routes with shared equipments.

depends on the type of product, among other factors. An area between bordering storage blocks must be kept free to avoid blending or contamination of a stack by neighboring products. In the remainder of this article, the terms storage block and subarea are used interchangeably.

The problem analyzed in this study can be defined as follows: assume that there is product offer available at supply and storage nodes, and demands to be met during a time horizon. The problem consists in defining the amount and destination of each product from a supply node to a storage node or demand node or from a storage node to a demand node, and simultaneously establishing a set of feasible routes (where there is no conflict regarding equipment allocation) to guarantee that the products are transported on schedule. In addition, it is necessary to minimize the costs associated with the exchange of products in subareas and to meet the demand, and select the lowest cost route for cargo transporting. From now on, this problem will be referred as Product Flow Planning and Scheduling Problem or PFPSP.

4. PFPSP formulation

The PFPSP mathematical model was initially proposed in Menezes and Mateus (2013). All production is planned for a given time horizon, divided into T periods. Product supply nodes are related to the arrival of products to meet demand nodes. Routes are classified into three types: routes x that transport products from the Reception to the Stockyard, routes y from the Reception to delivery, and routes z from the Stockyard to the delivery.

When there is not enough product p to meet the demand, different products p' of quality close to p could be used instead to guarantee that the total demand of the ship is met. In this case, the use of these products implies a loss of income measured by the parameter $\lambda_{pp'}$ (Table 2). This alternative is used only if there is not enough product p available at the reception subsystem or in the stockyard.

The main challenges are the simultaneous allocation of products into storage areas and the selection of a feasible set of routes. The main sets and variables used in the hierarchical approach are described below. The complete formulation, all sets, parameters and variables are described in the appendix.

Table 1 defines the main sets used for PFPSP modeling.

Table 2 shows the main decision variables used in PFPSP modeling. The next section describes the hierarchical approach to solve the PFPSP.

Table 1
Main sets for the PFPSP model.

Set	Description
T	Set of periods
P	Set of products
S	Set of storage sub-areas
R^x	Set of routes (reception/stockyard)
R^y	Set of routes (reception/delivery)
R^z	Set of routes (stockyard/delivery)

Table 2
Main variables and parameters.

Variable	Description
f^s_{pt}	Has unit value when subarea s is allocated for product p in period t
x^r_{pt}	Time taken in period t to transport product p from the reception to the stockyard using route $r \in R^x$
$y^r_{pp't}$, $z^r_{pp't}$	Time taken to transport product p' to meet the demand of product p in period t using route r from sets R^y and R^z. The replacement of the product p by p' is only used if there is not enough product p available at the reception subsystem or in the stockyard
e^s_{pt}	Amount of product p stored at subarea s in period t
IR_{pt}	Represents the amount of product p in the Reception subsystem that was not delivered at the end of period t
IP_{npt}	Represents the amount of product p that was not delivered at mooring berth n at the end of period t
$\lambda_{pp'}$	Cost associated with the loss of income by replacing product p by product p' to meet the demand of product p. When $p = p'$, $\lambda_{pp'} = 0$

5. Solution approach

Solving the PFPSP is a great challenge. The number of variables and the amount of combinations generated by equipment and tasks in the scheduling phase make the model unlikely to solve real-world instances through optimization packages. To work around this problem, we adopt a hierarchical approach where production planning and scheduling are solved separately. Fig. 3 presents the solution strategy to solve the PFPSP.

In this approach, production planning and scheduling are solved period by period. In the production phase, a relaxed version of the PFPSP is solved through a commercial solver. In this version, scheduling and integrality constraints are relaxed, (constraints (A.14)–(A.31), (A.38), (A.40)–(A.42)), readers are referred to the appendix. In the remainder of this article, this relaxed problem will be called relaxed PFPSP. The production variables ($x^r_{pt}, y^r_{pp't}$ and $z^r_{pp't}$) are sent to the scheduling phase to select and schedule the routes for a given period t. The scheduling phase defines the start time and the end time for each task, considering the sharing of equipment among the selected routes.

If a feasible schedule is not found, two decisions can be made: return to the planning phase and reselect the activities or simply transfer activities to the following period (backlog). We return to the PFPSP formulation, if during the scheduling phase we are able to find information to impose capacity constraints in the relaxed PFPSP, to improve task selection (these algorithms will be discussed in Section 5.3). In case of finding a feasible schedule, we just move towards the next period.

Although integrated methods may offer better solutions at a higher computational cost, in comparison with fully integrated procedures, the hierarchical approach has some advantages: real planning situations present a dynamic environment with uncertainties regarding supply, demand and equipment failure. Therefore, day-to-day operations often require efficient algorithms. In addition, the proposed method solves the problem period by period. Thus, rolling horizon strategies, such as those proposed in Clark (2005) and Li and Ierapetritou (2010) can also be applied. In the following sections details of the methodology are explained and analyzed.

5.1. Production phase

At this phase, the method solves a capacitated Lot sizing. The problem allocates products into sub-areas, with the aim of selecting

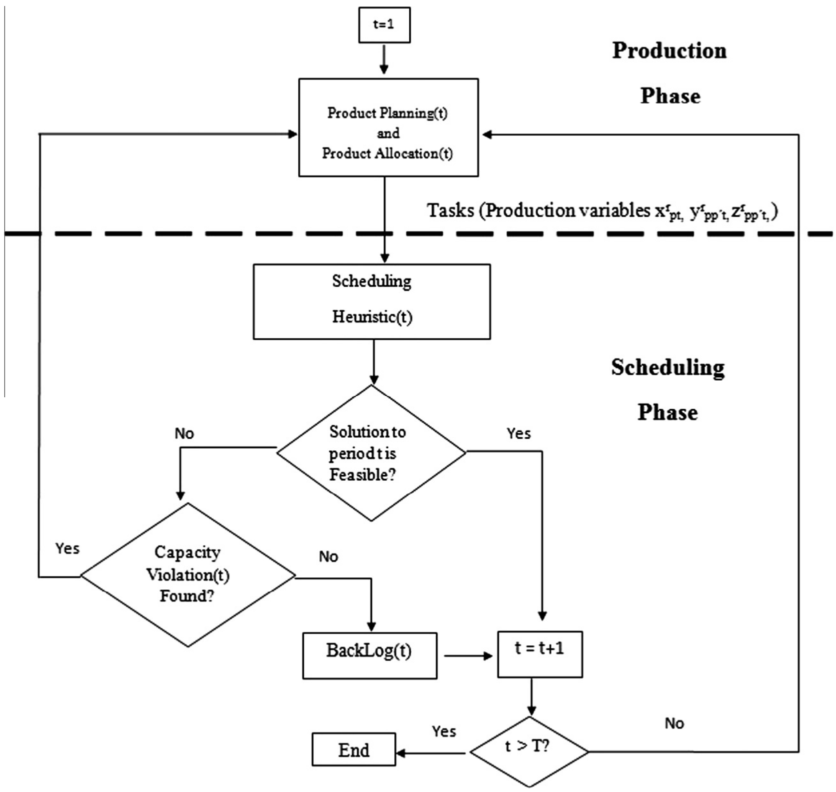

Fig. 3. Solution strategy.

the best sub-area for each product. While the Lot sizing is easily solved by a commercial solver, the allocation of products considers several integer variables, making the problem harder to be solved. Therefore we use a relax-and-fix strategy to efficiently solve the allocation problem. Such heuristics are commonly used in various problems: capacitated lotsizing (Mohammadi, Fatemi Ghomi, Karimi, & Torabi, 2008); production planning and scheduling (Ferreira, Morabito, & Rangel, 2010) and multi-level lotsizing problems (Toledo, Silva Arantes, Hossomi, França, & Akartunalı, 2015).

5.1.1. Relax-and-fix heuristic

The relax-and-fix heuristic works by fixing decision variables in a sequence until reaching a feasible solution. In our case, we first deal with the number of products allocated in each sub-area, variable f^s_{pt}, selecting the variable with more allocated products. The relaxed PFPSP is solved again, and the process is repeated until reaching a feasible set of variables f^s_{pt}. In a similar fashion, the algorithm works fixing fractional variables. In an iterative procedure, the variable of the sub-area with the largest fractional value is set to one. By the end of the production phase, an integer solution is found (considering the allocation problem and lot sizing) for each period of the time horizon. After that, the scheduling problem is solved in the second phase.

5.2. Scheduling phase

The PFPSP production variables ($x^r_{pt}, y^r_{pp't}$ and $z^r_{pp't}$) define the product type, the quantity and the route used to transport the products. Therefore, during the planning phase, values for these variables are defined and they will be the set of tasks for the scheduling phase. The scheduling problem consists of establishing the start and end times for these tasks, considering incompatibility restrictions. Preemption is not allowed and the objective is to minimize the makespan. Hereafter, this scheduling problem is called the scheduling problem with incompatibility jobs (SPIJ).

An example is used to clarify the definition of the SPIJ and to illustrate the tasks and incompatibility restrictions. Assume a solution for the relaxed PFPSP, as presented in Table 3.

In the first row of the Table 3, the variable x^1_{23} indicates that the product 2 should be carried by Route 1 in period 3, and the total time to transport this product on Route 1 will be 4 h. The SPIJ can be modeled using a conflict graph $G = (V, E)$ for each period t, where each vertex corresponds to a varible $x^r_{pt}, y^r_{pp't}$ or $z^r_{pp't}$, and an edge is created for each pair of tasks whose routes r and r' are conflicting (Fig. 4). Therefore, considering Table 3, the variable x^1_{23} corresponds to task or vertex A in the conflict graph (Fig. 5). The remaining rows of Table 3 have similar operations.

Fig. 4 illustrates the conflict between routes. The routes 1 and 2 share equipment and therefore cannot operate simultaneously. The same is true for routes numbered 6 and 9 and three other routes sharing equipment among themselves (routes 6, 8, and 10). In the conflict graph G (Fig. 5), the vertices are the tasks and the edges are created from the conflicts presented in Fig. 4. As previously discussed, the SPIJ consists of establishing the start and end times for each task, while guaranteeing that pairs of tasks sharing equipment are not simultaneously executed and minimizing the total execution time.

Table 3
Solution example of the relaxed PFPSP.

Tasks/vertices	Variables	Values	Routes
A	x^1_{23}	4	R_1
B	x^2_{53}	6	R_2
C	y^8_{553}	3	R_8
D	y^9_{333}	5	R_9
E	z^{10}_{663}	6	R_{10}
F	z^6_{443}	3	R_6

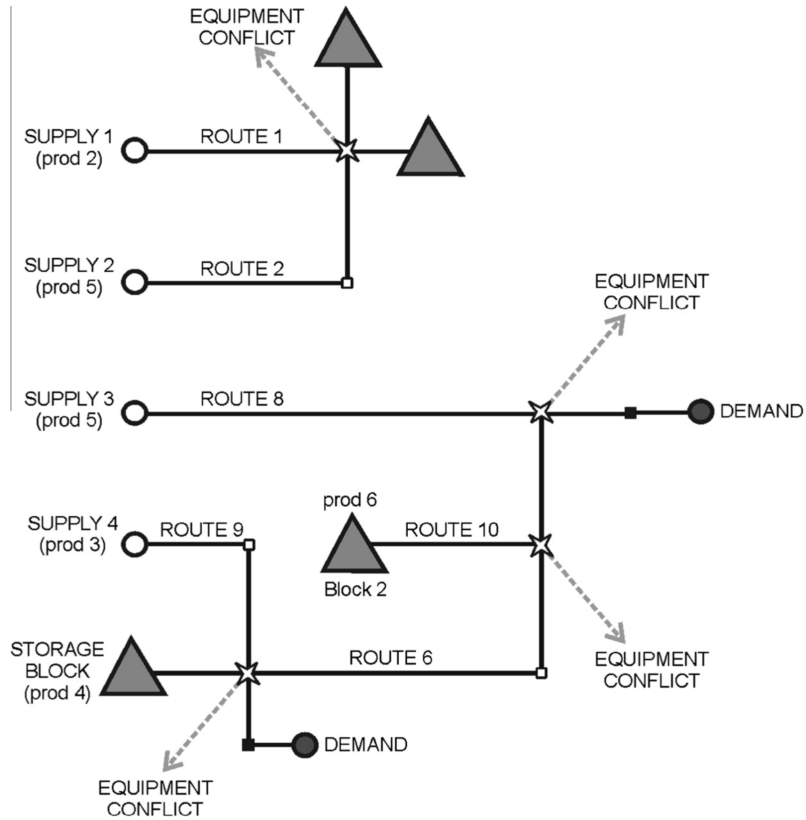

Fig. 4. Routes with conflicts.

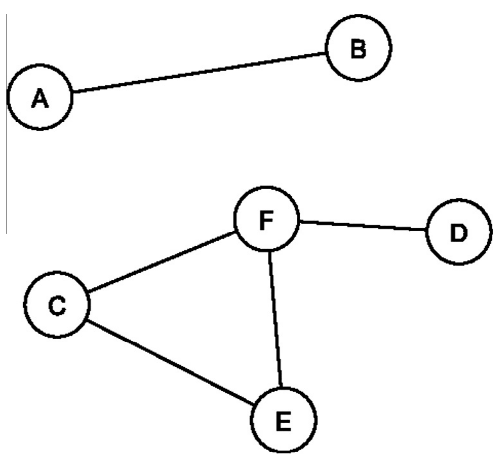

Fig. 5. Conflict graph extracted from the Fig. 4.

The SPIJ is not a new problem, many solutions have been proposed in the literature, such as scheduling with incompatible jobs (Bodlaender, Jansen, & Woeginger, 1994), scheduling jobs using an agreement graph (Bendraouche & Boudhar, 2012), and multicoloring and job-scheduling (Blchliger & Zufferey, 2013). Previous works as Bodlaender et al. (1994) and Gandhi, Halldrsson, Kortsarz, and Shachnai (2005), have shown that the SPIJ and several variations are NP-Hard.

5.2.1. SPIJ heuristic

To efficiently find good solutions, a greedy randomized search procedure (GRASP) was implemented. GRASP is an iterative algorithm first proposed to solve a set covering problem (Feo & Resende, 1989) (still without the name GRASP), and subsequently as already defined as GRASP (Feo & Resende, 1995), that basically consists of two phases: greedy construction and local search.

At each iteration of the construction phase, the algorithm considers the jobs extracted from planning phase and not yet scheduled, as the list of candidate elements. A greedy solution for the SPIJ is constructed as follows: Select randomly a job i from list of candidate elements. Next, define the lowest start time for the job, keeping already scheduled jobs that conflict with i without overlapping. Once all jobs are scheduled, a non-overlapping solution is provided. The local search consists in exchanging the order of jobs found in the greedy construction phase.

Details concerning GRASP adaptation for the SPIJ problem, as well the local search can be found in Menezes, Mateus, and Ravetti (2015). GRASP is here considered by its simplicity and ability to produce good results as those obtained at Binato, Hery, Loewenstern, and Resende (2000), Rocha, Ravetti, Mateus, and Pardalos (2008) and Rajkumar, Asokan, Anilkumar, and Page (2011). However, other heuristics can be evaluated, such as: Interated greedy (Pan & Ruiz, 2014; Ruiz & Sttzle, 2007), Iterated local search (Xu, L, & Cheng, 2014), Genetic algorithm (Omara & Arafa, 2010) among many others.

5.3. Communication between planning and scheduling

Allow backlogging in PFPSP is very expensive. At the port terminal in question, delays unloading a ship at the terminal (demurrage) or even unloading it before schedule (dispatch) result in a fine: for the terminal operator in the event of demurrage or for the shipowner in the event of dispatch. Preliminary results considering a hierarchical approach without the scheduling feedback, show poor PFPSP feasible solutions, due to a high backlog cost. As a capacity or conflict problem is resolved by pushing task for the following periods. The capacity information feedback is vital to ensure that the lot-sizing problem is able to carefully select the right set of tasks for each period planned. This ensures data consistency in both decision levels. In this section we will investigate two ways to accomplish this communication: The first based on a route capacity reduction heuristic, and one more effective approach based on maximal cliques.

5.3.1. Route Capacity Reduction Heuristic (RCH)

The PFPSP has constraints related to the time each route can remain active within a period. The objective is to insert a set of restrictions (similar to constraints A.11, A.12 and A.13 of the PFPSP in the appendix) to limit the duration of some tasks (only when the scheduling is not feasible). By reducing this time, planning will be forced to select other tasks to continue to meet the supply and demand. This constraint is created in the following way: Let n be total number of active tasks whose completion time are longer than the duration of the period t, $Conflict(j)$ represents the number of tasks that cannot be simultaneously executed with task j and $Length(j)$ represents the duration of task j.

$$Task_j <= Length(j) - ((Conflict(j)/n) * Length(j)), \qquad \forall j \tag{1}$$

Constraints (1) limits the duration of a task considering the number of conflicts. If task i has more conflicts than task j, its maximum duration in the next iteration will be shorter. The RCH heuristic is used for a fixed number of iterations in each period where the scheduling is not feasible. If the number of iterations is reached and scheduling continues unfeasible, tasks violating the capacity of the period are transferred to the next (backlog). In the computational experiments, this number of iterations was set at 10. Several instances were analyzed by varying the number of iterations. Through the results obtained, 10 iterations provides a balance between the quality of solutions and performance.

5.3.2. Maximal clique approach

The clique problem refers to the problem of finding complete subgraphs in a graph. This problem and its variants are widely studied in computer science and optimization as arises in many real-world problems. In this work, the maximal clique problem is exploited to facilitate the generation of a feasible solution for the PFPSP. In our work, it is used a well-known strategy called clique cuts, readers are referred to Dijkhuizen and Faigle (1993), Ji and Mitchell (2007), Mndez-Daz and Zabala (2008), Boland, Bley, Fricke, Froyland, and Sotirov (2011), among others.

A maximal clique is a clique that is not included in a higher clique. A constraint related to a maximal clique can be inserted into the production planning phase in order to strengthen the formulation and facilitate the search for a feasible scheduling. Consider the conflict graph (Fig. 5). In this graph, there are three maximal cliques: one maximal clique with the vertices A and B, another with F and D and, finally, one clique with C, F and E. Note that the vertices represent the production variables $x^r_{pt}, y^r_{pp't}$ and $z^r_{pp't}$ and the value of these variables is associated with the time that a route r is used for transporting a product p. Thus, the vertices in a clique represent tasks that cannot be performed simultaneously.

The sum of the task processing times in a maximal clique cannot exceed the period length (the start and end times of each task in a clique cannot overlap). After solving the scheduling phase and if the schedule is not feasible, a maximal clique cut (whose sum of the duration of its vertices is longer than the period duration), can be inserted into PFPSP formulation (production planning phase) to guide the production planning in selecting new routes. Some solution methods for maximal cliques are those proposed by Bron and Kerbosch (1973), Stix (2004) and Tomita, Tanaka, and Takahashi (2006). They are mostly recursive algorithms. Due the complexity, some pruning strategies are evaluated in the enumeration tree to improve the algorithm performance. In this article, a variation of the Bron-Kerbosch algorithm is implemented to generate the cliques. Information about the algorithm and details on implementation are available in the Appendix.

5.4. Hierarchical approach

As mentioned above, this approach separates the planning and scheduling problems. The following pseudo-code illustrates the primary procedures adopted to obtain an upper bound (feasible solution) for the PFPSP.

```
1:  procedure HIERARCHICAL APPROACH(N)   ▷
2:      t = 1                                    ▷ period counter
3:      repeat
4:          RelaxedSolutionPFPSP(t)      ▷ Solve relaxed PFPSP
5:          if CheckAllocationProduct(t) = true then      ▷ If all the variables f^s_pt are integer for period t
6:              SPIJHeuristic(t)     ▷ GRASP Heuristic
7:              if SPIJFeasible(t) = true then      ▷ all tasks are scheduled and respect the time period t
8:                  t = t + 1      ▷ If SPIJ is feasible to t, starts search for feasibility to period t + 1
9:              else
10:                 if CapacityViolation(t) = False then      ▷ if capacity violation is found
11:                     Backlogging(t)      ▷ Transfers supply and demand for the next period
12:                     t = t + 1
13:                 else
14:                     AddCapacityViolation(t)      ▷ add constraints associated with capacity
15:                 end if
16:             end if
17:         else
18:             Relax-And-FixHeuristic(t)      ▷ Relax-and-Fix to ensure the integrality of the variables f^s_pt
19:         end if
20:     until t > TotPeriod
21:     return Solution
22: end procedure
```

On line 4 of the pseudo-code, the relaxed PFPSP planning problem is solved. Hence, the relaxed PFPSP is an easily solved linear programming problem (even for large instances). The feasibility of the PFPSP is guaranteed in every period, i.e., the feasibility of the period (scheduling and product allocation) t is guaranteed before initiating the search for the feasibility of period $t+1$.

Once the relaxed PFPSP is solved, on line (4), the algorithm checks if a feasible allocation of products (all variables f^s_{pt} are integer) for the period t can be obtained. If positive, the heuristic to establish the start and end times for the tasks (scheduling) is initiated on line (6). Otherwise, there are fractional product allocation variables, a relax-and-fix heuristic is executed to achieve a feasible allocation of products (line 18), and the relaxed PFPSP is solved again (line 4). Even though the integrality of the variables f^s_{pt} and the task scheduling is solved one period at a time, the relaxed PFPSP is always solved from the first unfeasible period to the last one. These steps are repeated until a feasible solution is obtained for the allocation of products.

After the execution of the heuristic for the scheduling problem (line 6), it is necessary to verify that all tasks were scheduled respecting the duration of the period. If positive, this period is solved. Then, the solution of the next period is initiated, i.e., the current period (t) is incremented by 1 (line 8).

In situations where the scheduling heuristic has not found a feasible solution, it is made an attempt to find constraints (line 10) to assist the activation of new routes and thereby finding a feasible scheduling. If no capacity violation constraint is found, there is nothing left to be performed for this period. Supplies and demands that cannot be met within the limit of duration for this period are passed to the next period (backlogging) (line 11). Once all periods are explored (line 20), the hierarchical solution approach ends and a solution is obtained (line 21). In the following section, the computational results performed to validate the hierarchical approach are described.

6. Computational experiments

The experiments are performed based on an real product flow problem in an iron ore port terminal in Brazil, recognized as one of the largest worldwide. The basic parameters are the number of periods, the products and the routes. In general, they work with seven periods of one day or fourteen periods of twelve hours. Table 4 highlights the main parameters used to create the instances. The parameters $\alpha_{pt}, \beta_{npt}, \gamma^s_{p,p',t}, \lambda_{pp'}$ and σ^r are part of the PFPSP formulation and are described in the appendix.

In the iron ore port terminal, the demand nodes are those where the ship moors to receive the products. For the experiments, three demand nodes are considered: two moored ships in berth 1 and one ship in berth 2. Likewise, the supply nodes represent the points where the wagons unload the products. These points are related to the car dumpers. For the experiments, five car dumpers were considered. In this system, various products and quantities can arrive at different periods.

The priority in a port terminal is to meet the demand, so the penalty of not meeting it β_{npt}, is usually higher than the nonfulfillment of the supply (α_{pt} parameter). Even among the berths, the penalty is differentiated (β_{npt} value in Table 4). In our particular case, berth 2 meets larger ships so the priority is always to meet the ships of this berth. In the experiments, the cost of exchange products in the stockyard $\gamma^s_{pp't}$ is the same for any pair.

The monetary value of the iron ore unit is determined based on the percentage of iron in the product. Therefore, products with a higher percentage of iron are considered of better quality. Thus, to generate the $\lambda_{pp'}$ costs must be considered that its value increases proportionally to the difference of percentage of iron. Finally, the parameter σ^r is calculated based on the route length in meters. Regarding the supply/demand to be met in each period, the values are based on real data extracted from reports obtained from the company that manages the port terminal.

Table 4
Data used to generate the instances.

Parameter	Description
Stockyard	The product storage area is divided into four stockyards. Each stockyard has 10 subareas, each one capable of storing 100.000 tons of ore. In total, the four stockyards can store 4.000.000 tons of ore
Delivery	Two berths: two ships can be loaded simultaneously at berth 1
Equipment	Five car dumpers, four ore reclaimers, three stackers/reclaimers (equipment that performs both tasks), and eight stackers. At the terminal in question, there is approximately 50 km of conveyor belts. Each route uses one or various belt segments. Thus, the port terminal was considered to have in total 50 equipment, including conveyor belts
α_{pt}	2 (two monetary units)
β_{npt}	10 (ten monetary units for the berth number one), 50 (fifty monetary units for the berth number two)
$\gamma^s_{p,p',t}$	10 (ten monetary units)
$\lambda_{pp'}$	Based on the following formula: 0.01 (monetary unit) $*\lvert p-p'\rvert$, where $\lvert p-p'\rvert$ represents the quality deviation between the product p and p'
σ^r	Based on the following formula: 0.01 (monetary unit) $*$ length of route r

6.1. Instances

Three set of instances are considered in the experiments. For the first set (Tables 5 and 6), the quantity of products supplied is equal to the requested demand. For the second set (Tables 7 and 8), the quantity of products supplied is greater than the demand. For the third set (Tables 9 and 10), the quantity of products available in the supply is less than the demand. For each of the three sets, instances are created considering the following features: initial empty stock and initial stock with 30% of capacity, products supplied equal to those demanded and instances where the supplied and demanded products are different (when the product type is switched to meet the demand) and the cost of the exchange is calculated based on the parameter $\lambda_{pp'}$ from Table 4.

For the experiments described next, the equipment capacities varied between 8000 and 12000 t/h, and 110 routes in the terminal are considered. These values were based on current data of the port terminal. For all experiments in this section, the duration of each period is 12 h. The experiments are conducted using a computer with a 6-core Intel(R) Core(TM) i7 980 processor and 24 GB physical memory, running version 12.5 of the CPLEX solver. For all instances, a computational time limit of 3 h is set. In the following tables, the character $(-)$ represents instances for which the solver could not obtain a solution for the PFPSP due to insufficient memory.

The first, second and third columns of the Tables 5 and 6, contain the number of the instance, the type and the name. For example, instance 8P5$Prod$ corresponds to a planning horizon of eight periods and five different products being handled. Column Z_{LB} provides the lower limit for the PFPSP obtained by its linear relaxation. Columns Z_{UB} and Z_{BB} provide the upper and lower bounds (best bounds) obtained with the branch-and-cut algorithm of the CPLEX solver. The GAP column provides the gap given by $GAP = 100(Z_{UB} - Z_{BB})/Z_{UB}$. Regarding the hierarchical procedure, columns Z_{UB_1} and Z_{UB_2} provide the upper bound obtained for the hierarchical solution considering the use of RCH heuristics and maximal cliques respectively. The columns GAP_1 and GAP_2 provide the gap: $GAP_1 = 100(Z_{UB_1} - Z_{LB})/Z_{UB_1}$ and $GAP_2 = 100(Z_{UB_2} - Z_{LB})/Z_{UB_2}$. Finally, t_{LB}, t_{UB}, t_{UB1} and t_{UB2} are the elapsed computational time to obtain the values of Z_{LB}, Z_{UB}, Z_{UB_1} and Z_{UB_2} respectively, expressed in seconds.

The results shown in Table 5 indicate that solving the PFPSP in optimization packages is not feasible. From 18 (eighteen) instances, the solver was able to produce solutions only for half (nine) of them. In the rest, because insufficient memory, it was not possible to obtain even an upper bound.

With the heuristic based on a hierarchical approach (Table 6), it is possible to obtain solutions for all instances, all supplies and demands were met, and all tasks are scheduled respecting the duration of each period. Regarding the strategies adopted to ensure better exchange of information between the two decision-making levels (RCH heuristic and cuts based on the maximal clique problem), for smaller instances (instances 1–12), the two approaches showed similar results.

However, when considering hard instances, the use of maximal cliques allowed a considerable gain both as the upper bound as in performance (instances 13–18). In particular, for instances 13, 14,

Table 5
110 routes – capacity ranging from 8000 to 12000 t/h – supply equal demand (solver).

Number	Type	Instance	CPLEX					
			Z_{LB}	t_{LB}	Z_{UB}	t_{UB}	Z_{BB}	GAP (%)
1		4P5Prod	138.16	5	142.14	205	142.126	0.01
2		4P10Prod	140.4	34	143.25	785	143.25	0.00
3	Empty stock	8P5Prod	276.32	10	281.62	683	281.62	0.0
4	products	8P10Prod	279.7	1065	–	–	–	–
5	equal	10P5Prod	345.4	12	351.96	1159	351.931	0.01
6		10P10Prod	348.67	2089	–	–	–	–
7		4P5Prod	838.16	5	842.14	168	842.14	0.00
8		4P10Prod	2688.16	5674	–	–	–	–
9	Empty stock	8P5Prod	1476.32	13	1481.67	1202	1481.56	0.01
10	different	8P10Prod	6226.32	9871	–	–	–	
11	products	10P5Prod	1945.4	33	1952.96	1977	1952.78	0.01
12		10P10Prod	7345.4	2381	–	–	–	
13		4P5Prod	197.35	5.00	200.74	289.00	200.74	0.00
14		4P10Prod	204.22	1236.00	206.64	11003.00	206.64	0.00
15	Stock in 30%	8P5Prod	492.56	11.00	–	–	–	–
16	different	8P10Prod	425.94	983.00	–	–	–	–
17	products	10P5Prod	612.46	11.00	–	–	–	–
18		10P10Prod	510.84	1682.00	–	–	–	–

Table 6
110 routes – capacity ranging from 8000 to 12000 t/h – supply equal demand (hierarchical).

Number	Type	Instance	Hierarchical (RCH)			Hierarchical (CliqueCuts)		
			Z_{UB_1}	T_{UB_1}	GAP_1 (%)	Z_{UB_2}	t_{UB_2}	GAP_2 (%)
1		4P5Prod	142.24	4	2.87	142.24	7	2.87
2		4P10Prod	145.40	6	3.44	144	9	2.50
3	Empty stock	8P5Prod	323.12	13	14.48	323.12	25	14.48
4	products	8P10Prod	325.34	22	14.03	325.34	29	14.03
5	equal	10P5Prod	412.20	16	16.21	412.2	38	16.21
6		10P10Prod	426.78	31	18.30	413.56	45	15.69
7		4P5Prod	842.24	3	0.48	842.24	8	0.48
8		4P10Prod	2692.37	8	0.16	2692.37	11	0.16
9	Empty stock	8P5Prod	1490.63	14	0.96	1490.63	36	0.96
10	different	8P10Prod	6272.29	32	0.73	6272.29	41	0.73
11	products	10P5Prod	1981.33	24	1.81	1981.33	49	1.81
12		10P10Prod	7442.33	46	1.30	7412.47	81	0.90
13		4P5Prod	252.16	135	21.74	222.91	24	11.47
14		4P10Prod	542.94	94	62.39	212.90	32	4.08
15	Stock in 30%	8P5Prod	545.86	81	9.77	512.85	86	3.96
16	different	8P10Prod	1098.56	1032	61.23	543.70	237	21.66
17	products	10P5Prod	698.56	176	12.33	647.83	128	5.46
18		10P10Prod	799.82	2345	36.13	629.16	145	18.81

16 and 18, the use of cliques showed extremely more effective. The GAP are much smaller. In Tables 7 and 8 the results are analyzed assuming that the amount of products supplied is greater than the demand.

In the experiments described in Table 7, it was necessary to manipulate more products and routes (as excess supply must remain at the stockyard); therefore, the optimization package was not able to find a feasible solution for any case. Even the calculation of the lower limit (based on the linear relaxation) is not always possible, as the solver aborted due to insufficient memory in three opportunities (30, 32 and 36).

Similar to the results found in Table 6, with the hierarchical approach (Table 8), it is possible to obtain solutions for all instances, and all tasks are scheduled respecting the duration of each period. The strategy of producing maximal cliques to strengthen the planning phase also showed better performance. Especially for the last twelve instances (numbers 25–36).

Finally, the experiments from Table 9 describe instances where the number of products available in the supply is smaller than the demanded quantities. In such cases, in the absence of stock, the penalty of not meeting the demand is generated (associated with variables IP_{npt}). Similar to the results of Table 7, the CPLEX is not able to produce the upper bound for any case.

In the experiments described in Table 10, the hierarchical approach generated results for all experiments, but the GAP found for instances with no initial stock (instances 37–42) is high, once the penalties are been considered in the upper bound values. As discussed earlier, in the absence of stock, the penalty of not meeting the demand is generated (associated with variables IP_{npt}). For the cases with initial stock, all of the demand is met. This is possible because the initial stock is able to attend the lack of products in the supply. Additionally, for these cases, the upper bound is stronger, and the gaps found were smaller. For five instancies (numbered 43, 44, 45, 50 and 51), all the gaps are smaller than 3% (hierarchical approach with the use of cliques). Similarly to the above results, the use of maximal cliques showed to be more effective than the use of RCH heuristic.

Table 7
110 routes – capacity ranging from 8000 to 12000 t/h – supply greater than demand (solver).

Number	Type	Instance	CPLEX					
			Z_{LB}	t_{LB}	Z_{UB}	t_{UB}	Z_{BB}	GAP (%)
19		4P5Prod	148.3	4	–	–	–	–
20		4P10Prod	150.4	34	–	–	–	–
21	Empty stock	8P5Prod	297.495	9	–	–	–	–
22	products	8P10Prod	305.46	879	–	–	–	–
23	equal	10P5Prod	372.825	12	–	–	–	–
24		10P10Prod	380.9	1683	–	–	–	–
25		4P5Prod	651.66	5	–	–	–	–
26		4P10Prod	1876.34	854	–	–	–	–
27	Empty stock	8P5Prod	710.145	10	–	–	–	–
28	different	8P10Prod	2541.04	9422	–	–	–	–
29	products	10P5Prod	860.483	12	–	–	–	–
30		10P10Prod	–	–	–	–	–	–
31		4P5Prod	186.56	13	–	–	–	–
32		4P10Prod	–	–	–	–	–	–
33	Stock in 30%	8P5Prod	419.51	10	–	–	–	–
34	different	8P10Prod	381.74	1195	–	–	–	–
35	products	10P5Prod	499.76	15	–	–	–	–
36		10P10Prod	–	–	–	–	–	–

Table 8
110 routes – capacity ranging from 8000 to 12000 t/h – supply greater than demand (hierarchical).

Number	Type	Instance	Hierarchical (RCH)			Hierarchical (CliqueCuts)		
			Z_{UB_1}	T_{UB_1}	GAP_1 (%)	Z_{UB_2}	t_{UB_2}	GAP_2 (%)
19		4P5Prod	153.27	7	3.24	153.27	21	3.24
20		4P10Prod	175.38	11	14.24	175.38	23	14.24
21	Empty stock	8P5Prod	325.45	58	8.59	323.10	28	7.92
22	products	8P10Prod	343.49	77	11.07	343.49	47	11.07
23	equal	10P5Prod	417.38	147	10.68	414.01	80	9.95
24		10P10Prod	417.89	234	8.85	403.39	112	5.58
25		4P5Prod	671.05	20	2.89	657.25	65	0.85
26		4P10Prod	2074.91	21	9.57	2056.60	41	8.76
27	Empty stock	8P5Prod	795.25	78	10.70	766.91	116	7.40
28	different	8P10Prod	2961.18	379	14.19	2666.33	123	4.70
29	products	10P5Prod	950.01	83	9.42	887.65	178	3.06
30		10P10Prod	3515.61	437	–	3139.15	233	–
31		4P5Prod	347.56	229	46.32	215.06	92	13.25
32		4P10Prod	672.98	249	–	241.04	31	–
33	Stock in 30%	8P5Prod	499.20	131	15.96	486.70	111	13.81
34	different	8P10Prod	1687.93	1568	77.38	568.64	401	32.87
35	products	10P5Prod	596.36	201	16.20	529.80	270	5.67
36		10P10Prod	2262.84	2679	–	746.17	395	–

6.2. Overview about the experiments

Three solution approaches have been used to solve instances: the use of a compact model with optimization package, the hierarchical strategy with RCH heuristic (hierarchical-RCH) and the use of maximal cliques (hierarchical-CliqueCuts). The optimization package managed to produce better results in time and GAP only for smaller instances.

When considering instances with characteristics closer to the terminal real conditions, the solver is unable to produce feasible solutions. However, the hierarchical approaches have produced satisfactory results. For simpler instances the hierarchical-RCH heuristic is more efficient, however as the instances get bigger and complex the hierarchical-CliqueCuts shows a stronger performance in CPU time and gap. Therefore, considering that the major objective of the terminal operator is to provide low-cost solutions and optimize their resources, the hierarchical-CliqueCuts heuristic showed to be more efficient and effective.

7. Conclusions and final remarks

In this article, a hierarchical approach to solve a production planning and scheduling problem is proposed. The mathematical model developed, can be used to represent various problems related to the transportation of products and stock conditions, particularly problems that involve the flow of products in bulk cargo (iron ore, coal and grains) terminals. Although each case has its own particularities, our methodology is general enough to be easily adapted. The framework is validate by considering a real case from a Brazilian port terminal.

Regarding the hierarchical approach, the method is more efficient in producing a feasible solution than the solver. Furthermore, it is possible to solve medium and large instances, that with optimization packages is computationally unfeasible. The exchange of information between the two decision levels, guarantee good feasible solutions. The use of constraints based on maximal cliques, strengthened the PFPSP relaxed formulation and considerably improved the upper bound.

Table 9
110 routes – capacity ranging from 8000 to 12000 t/h – supply lower than demand (solver).

Number	Type	Instance	CPLEX					
			Z_{LB}	t_{LB}	Z_{UB}	t_{UB}	Z_{BB}	GAP (%)
37		4P5Prod	4.00E+06	5	–	–	–	–
38		4P10Prod	4.50E+06	36	–	–	–	–
39	Empty stock	8P5Prod	9.00E+06	10	–	–	–	–
40	products	8P10Prod	1.05E+07	973	–	–	–	–
41	equal	10P5Prod	1.20E+07	16	–	–	–	–
42		10P10Prod	1.50E+07	1585	–	–	–	–
43		4P5Prod	181.25	5	–	–	–	–
44		4P10Prod	158.63	35	–	–	–	–
45	Stock in 30%	8P5Prod	346.66	13	–	–	–	–
46	products	8P10Prod	321.51	1139	–	–	–	–
47	equal	10P5Prod	441.06	33	–	–	–	–
48		10P10Prod	416.03	2096	–	–	–	–
49		4P5Prod	213.29	5	–	–	–	–
50		4P10Prod	214.52	2875	–	–	–	–
51	Stock in 30%	8P5Prod	606.76	9	–	–	–	–
52	different	8P10Prod	479.76	1314	–	–	–	–
53	products	10P5Prod	928.87	16	–	–	–	–
54		10P10Prod	725.03	2300	–	–	–	–

Table 10
110 routes – capacity ranging from 8000 to 12000 t/h – supply lower than demand (hierarchical).

Number	Type	Instance	Hierarchical (RCH)			Hierarchical (CliqueCuts)		
			Z_{UB_1}	T_{UB_1}	GAP_1 (%)	Z_{UB_2}	t_{UB_2}	GAP_2 (%)
37		4P5Prod	5.00E+06	14	20.00	5.00E+06	4	20.00
38		4P10Prod	5.00E+06	16	10.00	5.00E+06	6	10.00
39	Empty stock	8P5Prod	1.80E+07	23	50.00	1.80E+07	13	50.00
40	products	8P10Prod	1.80E+07	67	41.66	1.80E+07	40	41.66
41	equal	10P5Prod	2.75E+07	39	56.36	2.75E+07	19	56.36
42		10P10Prod	2.75E+07	64	45.45	2.75E+07	58	45.45
43		4P5Prod	184.99	17	2.02	183.42	41	1.18
44		4P10Prod	163.30	19	2.86	163.34	48	2.28
45	Stock in 30%	8P5Prod	352.63	22	1.69	351.99	56	1.51
46	products	8P10Prod	369.36	46	12.95	357.76	67	10.13
47	equal	10P5Prod	457.32	38	3.55	456.39	93	3.36
48		10P10Prod	499.07	99	16.64	448.33	110	7.20
49		4P5Prod	243.07	54	12.25	232.72	32	8.35
50		4P10Prod	256.45	73	16.35	215.45	33	0.43
51	Stock in 30%	8P5Prod	613	124	1.02	610.34	36	0.59
52	different	8P10Prod	597.67	456	19.73	585.66	36	18.08
53	products	10P5Prod	994.27	234	6.58	965.10	141	3.75
54		10P10Prod	980.70	823	26.07	967.53	313	25.06

Is important to point out that the algorithms here presented, do not deal with idle capacity. This case happens when the SPIJ solution produces makespan values lower that the duration of a period. Future works are planned to overcome these limitations, together new methods to efficiently solve the SPIJ, and integrate the solutions.

Acknowledgements

This research is supported by the following institutions: VALE, Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Appendix A. PFPSP formulation

The mathematical model described next integrates the planning and scheduling decisions. All production is planned for a given time horizon, divided into T periods. Product supply is related to the arrival of a train, and demand is related to the arrival of a ship. Routes are classified into three types: routes x that transport products from the Reception to the Stockyard, routes y from the Reception to the Ships, and routes z from the Stockyard to the Ships. The main challenges are the allocation of products in the stockyard and the allocation and scheduling of routes that fulfill demands and supplies. The sets, parameters, variables, and equations of the model are described next.

Table A.1 defines the sets used for PFPSP modeling. Table A.2 gives the input parameters of the model, which define the capacity limits for periods, storage subareas, equipment, routes, and costs associated with the objective function.

Table A.3 shows all decision variables used in PFPSP modeling. These variables are associated with the allocation of storage subareas and the allocation and scheduling of routes and for returning which demands were not met in each period.

Table A.1
Set definition for the PFPSP model.

Set	Description
T	Set of periods
P	Set of products
S	Set of storage sub-areas
R^x	Set of routes (reception/stockyard)
R^x_s	Subset of routes x that reach subarea s
R^y	Set of routes (reception/piers)
R^z	Set of routes (stockyard/pier)
R^z_s	Subset of routes z from subarea s
M	Set of equipment
R^x_m	Subset of routes x that use equipment m
R^y_m	Subset of routes y that use equipment m
R^z_m	Subset of routes z that use equipment m
$R = R^x \cup R^y \cup R^z$	Set of all available routes
N	Set of available mooring berths
E	Pairs of routes that share at least one piece of equipment to transport products

Table A.2
Input parameters.

Parameter	Description
a_{pt}	Supply of product p at the beginning of period t
d_{npt}	Demand of product p at a ship moored at berth n at the beginning of period t
K	High-value constant
l^s_{pt}	Storage capacity of subarea s for product p in period t
b^m	Capacity of equipment m (in ton/hour)
j^m_t	Available time (in hours) for the use of equipment m in period t
$NumMax_t$	Total time in period t (in hours)
c^{rx}	Capacity (in tons/hour) of route $r \in R^x$
c^{ry}	Capacity (in tons/hour) of route $r \in R^y$
c^{rz}	Capacity (in tons/hour) of route $r \in R^z$
β_{npt}	Penalty for not meeting the demand of a ship moored at berth n with product p in period t
α_{pt}	Penalty for not meeting the supply at the reception of product p in period t
$\gamma^s_{pp't}$	Preparation cost associated with replacing product p by product p' in subarea s at period t
$\lambda_{pp'}$	Cost associated with the loss of income by replacing product p by product p' to meet the demand of product p. When $p = p', \lambda_{pp'} = 0$
σ^r	Maintenance cost of using route $r \in R$

- **Objective function**

The objective function seeks to minimize the penalty of not meeting the supply of products at the Reception subsystem, the penalty of not meeting the demand of ships, the cost of product allocation in the stockyard and the cost of using routes to transport products.

$$\begin{aligned}\min & \sum_{p\in P}\sum_{t\in T}\alpha_{pt}IR_{pt} \\ & +\sum_{n\in N}\sum_{p\in P}\sum_{t\in T}\beta_{npt}IP_{npt} \\ & +\sum_{s\in S}\sum_{p\in P}\sum_{p'\in(P\cup 0)}\sum_{t\in T}\gamma^s_{pp't}Sf^s_{pp't} \\ & +\sum_{p\in P}\sum_{p'\in P}\sum_{t\in T}\lambda_{pp'}\left(\sum_{r\in R^y}c^{ry}y^r_{pp't}+\sum_{r\in R^z}c^{rz}z^r_{pp't}\right) \\ & +\sum_{p\in P}\sum_{t\in T}\sum_{r\in R^x}\sigma^r(c^{rx}x^r_{pt}) \\ & +\sum_{p\in P}\sum_{p'\in P}\sum_{t\in T}\sum_{r\in R^y}\sigma^r(c^{ry}y^r_{pp't}) \\ & +\sum_{p\in P}\sum_{p'\in P}\sum_{t\in T}\sum_{r\in R^z}\sigma^r(c^{rz}z^r_{pp't})\end{aligned} \tag{A.1}$$

The first term represents the penalty of not meeting the supply of products at the Reception subsystem. IR_{pt} represents the supply of product p at the Reception subsystem that was not met by the end of period t, and α_{pt} is its unit cost (penalty). The next term represents the penalty of not meeting the demand at the Pier subsystem. IP_{npt} represents the demand for product p at the Pier subsystem that was not met by the end of period t, and β_{npt} is its unit cost (penalty). The next term represents the cost of exchanging products in a subarea. For some product pairs, e.g., manganese and iron ore, it is necessary to clean the subarea before replacing one product with another to prevent contamination. The cleaning cost is represented by $\gamma^s_{pp't}$, and the variable $Sf^s_{pp't}$ indicates whether product p has been replaced with product p' in subarea s at period t. When there is not enough product p to meet the demand, different products p' of quality close to p could be used instead to guarantee that the total demand of the ship is met. In this case, the use of these products implies a loss of income measured by the parameter $\lambda_{pp'}$. This alternative is used only if there is not enough product p available at the reception subsystem or in the stockyard. When $p = p', \lambda_{pp'} = 0$. Finally, σ^r represents the cost of using route r to transport a product. This cost is associated with the amount of products transported by a route and its distance (measured in meters from the point of origin to the final destination), whether the route is of the x type (distance from a given car dumper to a given subarea), y type (distance from the car dumper to a berth/ship) or z type (distance from a subarea in the stockyard to a berth).

Table A.3
Variable definition.

Variable	Description
f^s_{pt}	Has unit value when subarea s is allocated for product p in period t
$Sf^s_{pp't}$	Has a value of 1 when product p has been replaced with product p' at period t. This replacement can occur only when the amount of product p in subarea s has been exhausted in the preceding period $t-1$
x^r_{pt}	Time taken in period t to transport product p from the reception to the stockyard using route $r \in R^x$
$y^r_{pp't}$, $z^r_{pp't}$	Time taken to transport product p' to meet the demand of product p in period t using route r from sets R^y and R^z. When p' is equal to p, the product delivered is the same as was requested
e^s_{pt}	Amount of product p stored at subarea s in period t
IR_{pt}	Represents the amount of product p in the Reception subsystem that was not delivered at the end of period t
IP_{npt}	Represents the amount of product p that was not delivered at mooring berth n at the end of period t
$t^r_{pp't}$	Start time for the processing operation using route $r \in R$ in period t. For each variable $x^r_{pt}, y^r_{pp't}$ and $z^r_{pp't}$, there is one start time ($t^r_{pp't}$). When $t^r_{pp't}$ is associated with x^r_{pt}, p and p' are the same product
$u^r_{pp't}$	Binary variable. It has a value of 1 if the product p' used to meet the demand of product p uses the route r from set R in period t. For all $r \in R^x, p = p'$
$\theta^{rr'}_{pp'\hat{p}\hat{p}'t}$	Binary variable. It has a value of 1 if the product p or p' (used to meet p) precedes the product $\hat{p}$ or $\hat{p}'$ (used to meet $\hat{p}$) in the conflicting routes $r, r' \in E$ in period t

- **Supply and demand constraints**

Constraints associated with meeting the product supply at the Reception subsystem and demand at the Piers are presented below.

$$\sum_{r\in R^x}c^{rx}x^r_{pt}+\sum_{r\in R^y}c^{ry}\left(\sum_{p'\in P}y^r_{p'pt}\right)-IR_{p(t-1)}+IR_{pt}=a_{pt}, \quad \forall p\in P,\ \forall t\in T. \tag{A.2}$$

Constraints (A.2) formulate the meeting of the supply at the Reception subsystem. As previously stated, meeting the supply at this subsystem consists of unloading the trains, and therefore, these constraints guarantee that unmet supplies are updated. The unmet supplies IR_{p0} at period zero are an input data of the problem.

$$\sum_{r\in R^z} c^{rz}\left(\sum_{p'\in P} z^r_{pp't}\right) + \sum_{r\in R^y} c^{ry}\left(\sum_{p'\in P} y^r_{pp't}\right) - IP_{np(t-1)} + IP_{npt} = d_{npt},$$
$$\forall n\in N, \forall p\in P,\ \forall t\in T. \tag{A.3}$$

Meeting the demand at the Pier subsystem, i.e., loading cargo onto the ships, is imposed by constraints (A.3). The unmet demands (IP_{np0}) at period zero are an input data of the problem.

- **Stock control at subareas**

Constraints (A.4) guarantee that stocks are kept up-to-date at each subarea. Initially, the stocked amounts for product p in subarea s at period one (e^s_{p1}) are input data for the problem. Constraints (A.5) define the storage capacity of each subarea. Routes belonging to set R^x can be divided into subsets R^x_s, which contain all routes x that reach subarea s. The same process is performed for routes belonging to R^z.

$$e^s_{p(t+1)} = e^s_{pt} + \sum_{r\in R^x_s} c^{rx} x^r_{pt} - \sum_{r\in R^z_s} c^{rz}\left(\sum_{p'\in P} z^r_{p'pt}\right),$$
$$\forall s\in S,\ \forall p\in P,\ \forall t\in T. \tag{A.4}$$

$$e^s_{pt} \leqslant l^s_{pt}, \forall s\in S, \forall p\in P, \qquad \forall t\in T. \tag{A.5}$$

- **Equipment capacity constraints**

An piece of equipment may be used in more than one route. Constraints (A.6) ensure that no equipment will have its capacity exceeded. The time available for using equipment m in period t is represented by j^m_t, and the equipment capacity is represented by b^m.

$$\sum_{p\in P}\left(\sum_{r\in R^x_m} c^{rx} x^r_{pt} + \sum_{r\in R^z_m} c^{rz}\left(\sum_{p'\in P} z^r_{pp't}\right) + \sum_{r\in R^y_m} c^{ry}\left(\sum_{p'\in P} y^r_{pp't}\right)\right)$$
$$\leqslant j^m_t b^m, \qquad \forall m\in M,\ \forall t\in T. \tag{A.6}$$

These constraints allow two routes that share the same equipment to be used simultaneously as long as they do not exceed its capacity. However, these constraints do not prevent routes that transport different products and share equipment from being used simultaneously. To enforce this condition, scheduling constraints are added.

- **Constraints for stockyard allocation**

Constraints (A.7) control the stockyard allocation. Once a subarea is allocated for a product, it cannot be used for any other product in the same period. When a subarea is empty, the product 0 is allocated to it, i.e., $f^s_{0t} = 1$.

$$\sum_{p\in(P\cup 0)} f^s_{pt} = 1, \qquad \forall s\in S,\ \forall t\in T. \tag{A.7}$$

Constraints (A.8) control the replacement of products in a subarea. If $Sf^s_{pp't} = 1$, then product p has been replaced with p' at period t. This replacement may occur when a subarea becomes empty at the end of a given period and another product is stored in the subarea during the subsequent period. As previously stated, this constraint is associated with a subarea's maintenance/cleaning cost and is represented as a term in the model's objective function.

$$Sf^s_{pp't} \geqslant f^s_{p,(t-1)} + f^s_{p't} - 1, \qquad \forall s\in S,\ \forall t\in T,\ \forall p\in P,\ \forall p' \in (P\cup 0),\ p\neq p'. \tag{A.8}$$

The requirement that there can be only one product p stockpiled at a subarea s in period t if the stockyard allocation decision variable is valued 1 is enforced by constraints (A.9) and (A.10), where K is a high-value constant.

$$Kf^s_{pt} - e^s_{pt} \geqslant 0, \qquad \forall s\in S,\ \forall p\in P,\ \forall t\in T. \tag{A.9}$$

$$Kf^s_{pt} - \sum_{r\in R^x_s} c^{rx} x^r_{pt} \geqslant 0, \qquad \forall s\in S,\ \forall p\in P,\ \forall t\in T \tag{A.10}$$

- **Route allocation constraints**

If a route r at period t is used ($u^r_{pp't} = 1$) to carry a product p or p', constraints A.11, A.12 and A.13 guarantee that its availability and capacity (measured in hours) are met.

$$x^r_{pt} \leqslant NumMax_t\ u^r_{pp't}, \qquad \forall p,p'\in P,\ p=p',\ \forall t\in T,\ \forall r\in R^x \tag{A.11}$$

$$y^r_{pp't} \leqslant NumMax_t\ u^r_{pp't}, \qquad \forall p,p'\in P,\ \forall t\in T,\ \forall r\in R^y \tag{A.12}$$

$$z^r_{pp't} \leqslant NumMax_t\ u^r_{pp't}, \qquad \forall p,p'\in P,\ \forall t\in T,\ \forall r\in R^z \tag{A.13}$$

- **Product scheduling constraints**

Eqs. (A.14) and (A.15) define disjunctive constraints for each pair of conflicting routes ($r,r'\in E$) and $(r,r')\in R^x$. They also establish the order of products p and $\hat{p}$ sharing equipment. If $\theta^{rr'}_{pp'\hat{p}\hat{p}t} = 1$, (A.15) is redundant, and (A.14) ensures that product p or (p') precedes $\hat{p}$ or ($\hat{p}'$) and that the start time of $\hat{p}$ is greater than the start time of p; if $\theta^{rr'}_{pp'\hat{p}\hat{p}t} = 0, \hat{p}$ precedes p. The same is true for all the other pairs of conflicting routes $(r,r'\in E)$ such that $r\in R^x$ and $r'\in R^y$ constraints (A.16, A.17); $r\in R^x$ and $r'\in R^z$ constraints (A.18), (A.19); $r,r'\in R^y$ constraints (A.20), (A.21); $r,r'\in R^z$ constraints (A.22), (A.23), and $r\in R^z$ and $r'\in R^y$ constraints (A.24), (A.25).

$$K(1-u^r_{pp't}) + K(1-u^{r'}_{\hat{p}\hat{p}'t}) + K(1-\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^{r'}_{\hat{p}\hat{p}'t} \geqslant t^r_{pp't} + x^r_{pt} \tag{A.14}$$

$$K(1-u^r_{pp't}) + K(1-u^{r'}_{\hat{p}\hat{p}'t}) + K(\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^r_{pp't} \geqslant t^{r'}_{\hat{p}\hat{p}'t} + x^{r'}_{\hat{p}t} \tag{A.15}$$

$$\forall (r,r')\in E,\ (r,r')\in R^x,\ \forall p=p'\in P,\ \forall \hat{p}=\hat{p}'\in P,\ \forall t\in T$$

$$K(1-u^r_{pp't}) + K(1-u^{r'}_{\hat{p}\hat{p}'t}) + K(1-\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^{r'}_{\hat{p}\hat{p}'t} \geqslant t^r_{pp't} + x^r_{pt} \tag{A.16}$$

$$K(1-u^r_{pp't}) + K(1-u^{r'}_{\hat{p}\hat{p}'t}) + K(\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^r_{pp't} \geqslant t^{r'}_{\hat{p}\hat{p}'t} + y^{r'}_{\hat{p}\hat{p}'t} \tag{A.17}$$

$$\forall (r,r')\in E,\ r\in R^x,\ r'\in R^y,\ \forall p=p'\in P,\ \forall \hat{p},\hat{p}'\in P,\ \forall t\in T$$

$$K(1-u^r_{pp't}) + K(1-u^{r'}_{\hat{p}\hat{p}'t}) + K(1-\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^{r'}_{\hat{p}\hat{p}'t} \geqslant t^r_{pp't} + x^r_{pt} \tag{A.18}$$

$$K(1-u^r_{pp't}) + K(1-u^{r'}_{\hat{p}\hat{p}'t}) + K(\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^r_{pp't} \geqslant t^{r'}_{\hat{p}\hat{p}'t} + z^{r'}_{\hat{p}\hat{p}'t} \tag{A.19}$$

$$\forall (r,r')\in E,\ r\in R^x,\ r'\in R^z,\ \forall p=p'\in P,\ \forall \hat{p},\hat{p}'\in P,\ \forall t\in T$$

$$K(1-u^r_{pp't}) + K(1-u^{r'}_{\hat{p}\hat{p}'t}) + K(1-\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^{r'}_{\hat{p}\hat{p}'t} \geqslant t^r_{pp't} + y^r_{pp't} \tag{A.20}$$

$$K(1-u^r_{pp't}) + K(1-u^{r'}_{\hat{p}\hat{p}'t}) + K(\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^r_{pp't} \geqslant t^{r'}_{\hat{p}\hat{p}'t} + y^{r'}_{\hat{p}\hat{p}'t} \tag{A.21}$$

$$\forall (r,r')\in E,\ (r,r')\in R^y,\ \forall p,p',\hat{p},\hat{p}'\in P,\ \forall t\in T$$

$$K(1-u^r_{pp't}) + K(1-u^{r'}_{\hat{p}\hat{p}'t}) + K(1-\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^{r'}_{\hat{p}\hat{p}'t} \geqslant t^r_{pp't} + z^r_{pp't} \tag{A.22}$$

$$K(1-u^r_{pp't}) + K(1-u^{r'}_{\hat{p}\hat{p}'t}) + K(\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^r_{pp't} \geqslant t^{r'}_{\hat{p}\hat{p}'t} + z^{r'}_{\hat{p}\hat{p}'t} \tag{A.23}$$

$$\forall (r,r')\in E,\ (r,r')\in R^z,\ \forall p,p',\hat{p},\hat{p}'\in P,\ \forall t\in T$$

$$K(1-u^r_{pp't})+K(1-u^{r'}_{\hat{p}\hat{p}'t})+K(1-\theta^{rr'}_{pp'\hat{p}\hat{p}'t})+t^{r'}_{\hat{p}\hat{p}'t} \geqslant t^r_{pp't}+z^r_{pp't} \tag{A.24}$$

$$K(1-u^r_{pp't})+K(1-u^{r'}_{\hat{p}\hat{p}'t})+K(\theta^{rr'}_{pp'\hat{p}\hat{p}'t})+t^r_{pp't} \geqslant t^{r'}_{\hat{p}\hat{p}'t}+y^{r'}_{\hat{p}\hat{p}'t} \tag{A.25}$$

$$\forall (r,r') \in E,\ r \in R^z,\ r' \in R^y,\ \forall p,p',\hat{p},\hat{p}' \in P,\ \forall t \in T$$

Constraints (A.26)–(A.29), (A.31) ensure that a product cannot be introduced before or after the period in which it is established in the production plan.

$$t^r_{pp't} + x^r_{pt} \leqslant \sum_{i=1}^{t} NumMax_i \tag{A.26}$$

$$t^r_{pp't} \geqslant \sum_{i=1}^{t-1} NumMax_i \tag{A.27}$$

$$\forall r \in R^x,\ \forall p = p' \in P,\ \forall t \in T$$

$$t^r_{pp't} + y^r_{pp't} \leqslant \sum_{i=1}^{t} NumMax_i \tag{A.28}$$

$$t^r_{pp't} \geqslant \sum_{i=1}^{t-1} NumMax_i \tag{A.29}$$

$$\forall r \in R^y,\ \forall p,p' \in P,\ \forall t \in T$$

$$t^r_{pp't} + z^r_{pp't} \leqslant \sum_{i=1}^{t} NumMax_i \tag{A.30}$$

$$t^r_{pp't} \geqslant \sum_{i=1}^{t-1} NumMax_i \tag{A.31}$$

$$\forall r \in R^z,\ \forall p,p' \in P,\ \forall t \in T$$

- **Nonnegativity and integrality**

The following constraints define the limits for planning, storage, supply, demand, stockyard allocation, and product scheduling variables.

$$x^r_{pt} \geqslant 0, \qquad \forall r \in R^x,\ \forall p \in P,\ \forall t \in T. \tag{A.32}$$

$$y^r_{pp't} \geqslant 0, \qquad \forall r \in R^y,\ \forall p,p' \in P,\ \forall t \in T. \tag{A.33}$$

$$z^r_{pp't} \geqslant 0, \qquad \forall r \in R^z,\ \forall p,p' \in P,\ \forall t \in T. \tag{A.34}$$

$$e^s_{pt} \geqslant 0, \qquad \forall s \in S,\ \forall p \in P,\ \forall t \in T. \tag{A.35}$$

$$IR_{pt} \geqslant 0, \qquad \forall p \in P,\ \forall t \in T. \tag{A.36}$$

$$IP_{npt} \geqslant 0, \qquad \forall n \in N,\ \forall p \in P,\ \forall t \in T. \tag{A.37}$$

$$f^s_{pt} \in \{0,1\}, \qquad \forall s \in S,\ \forall p \in P,\ \forall t \in T. \tag{A.38}$$

$$0 \leqslant Sf^s_{pp't} \leq 1, \qquad \forall s \in S,\ \forall p \in P,\ \forall p' \in (P \cup 0),\ p \neq p',\ \forall t \in T. \tag{A.39}$$

$$t^r_{pp't} \geqslant 0, \qquad \forall r \in R,\ \forall p,p' \in P,\ \forall t \in T. \tag{A.40}$$

$$u^r_{pp't} \in \{0,1\}, \qquad \forall r \in R,\ \forall p,p' \in P,\ \forall t \in T. \tag{A.41}$$

$$\theta^{rr'}_{pp'\hat{p}\hat{p}'t} \in \{0,1\}, \qquad \forall r,r' \in E,\ \forall p,p',\hat{p},\hat{p}' \in P,\ \forall t \in T. \tag{A.42}$$

Appendix B. Maximal clique algorithm

In this article, the goal is to detect maximal cliques that is larger than a minimum weight. The weight of a clique is calculated as the sum of the weights of its vertices. In this work, the weight is the task duration times. Therefore, in one clique, if the sum of the weights of its vertices is greater than the duration of the period t, this clique must be inserted in PFPSP formulation.

Consider the conflict graph (Fig. 5), with information regarding the duration of each task available in the Table 3, and that the length (period capacity) is 8 h. Two maximal clique constraints can be inserted into PFPSP:

$$x^1_{23} + x^2_{53} <= 8 \tag{B.1}$$

$$y^8_{553} + z^{10}_{663} + z^6_{443} <= 8 \tag{B.2}$$

The constraint (B.1) is the maximal clique associated with the vertices A and B (Fig. 5), constraint (B.2) is related to the vertices C, F and E forming another maximal clique. Note that the sum of the duration tasks on both constraints violates the period length (values extracted from the Table 3). In the constraint (B.1) the sum is 10, and the constraint (B.2) the sum of the tasks execution times is 12 h. As the maximal clique refers to F and D vertices (Fig. 5) do not violate the length of period, this constraint is not inserted in PFPSP formulation.

The following algorithm is an extension of Bron-Kerbosch algorithm with pivoting and weight estimate (Bron & Kerbosch, 1973). Consider the following sets: C, the set of vertices already defined as part of the clique; P, set of candidates to join the click and S, vertices already been analyzed and which do not lead to increased set of candidate P. The *MinWeight* value is the length of period t.

```
procedure BK3(C,P,S)      ▷ Bron-Kerbosch algorithm with pivoting and weight estimate
    if (P and S are empty) then
        if (w(C) > MinWeight) then      ▷ If weight of the clique C exceeding the duration of period t.
            AddMaximalCliqueConstraint(C)      ▷ inserts the clique C in PFPSP formulation.
        end if
    end if
    if (w(C) + h(C) > MinWeight) then      ▷ if weight of the clique C plus the weight estimated of the other vertices which may be part of the clique C, exceeds the duration of the period t.
        Random pivot (u) between P ∪ S
        for i do 1 P \ N(u)      ▷ neighborhood of the vertex u.
            BK3(C ∪ {i}, P ∩ N(i), S ∩ N(i))
            P := P \ {i}
            S := S \ {i}
        end for
    end if
end procedure
```

The main advantage of this modification in BK algorithm, is related to weight estimate. Since the intent is only maximal cliques that meet the minimum weight condition, some pruning can be made, if these branches will not produce a maximal clique that violates the weight estimated.

References

Barros, V. H., Costa, T. S., Oliveira, A. C. M., & Lorena, L. A. N. (2011). Model and heuristic for berth allocation in tidal bulk ports with stock level constraints. *Computers & Industrial Engineering, 60*, 606–613.

Bendraouche, M., & Boudhar, M. (2012). Scheduling jobs on identical machines with agreement graph. *Computers & Operations Research, 39*, 382–390.

Bilgen, B., & Ozkarahan, I. (2007). A mixed-integer linear programming model for bulk grain blending and shipping. *International Journal of Production Economics, 107*, 555–571.

Binato, S., Hery, W., Loewenstern, D. M., & Resende, M. G. C. (2000). A grasp for job shop scheduling. In *Essays and surveys on metaheuristics* (pp. 59–79). Kluwer Academic Publishers.

Blchliger, I., & Zufferey, N. (2013). Multi-coloring and job-scheduling with assignment and incompatibility costs. *Annals of Operations Research, 211*, 83–101.

Bodlaender, H. L., Jansen, K., & Woeginger, G. J. (1994). Scheduling with incompatible jobs. *Discrete Applied Mathematics, 55*, 219–232.

Boland, N., Bley, A., Fricke, C., Froyland, G., & Sotirov, R. (2011). Clique-based facets for the precedence constrained knapsack problem. *Mathematical Programming, 133*, 481–511.

Boland, N., Gulezynski, D., & Savelsbergh, M. (2012). A stockyard planning problem. *EURO Journal on Transportation and Logistics, 1*, 197–236.

Bron, C., & Kerbosch, J. (1973). Algorithm 457: Finding all cliques of an undirected graph. *Communication of the ACM, 16*, 575–577.

Rocco, C. D., & Morabito, R. (2014). Scheduling of production and logistics operations of steam production systems in food industries. *Journal of the Operational Research Society, 65*, 1896–1904.

Calfa, B. A., Agarwal, A., Grossmann, I. E., & Wassick, J. M. (2013). Hybrid bilevel-lagrangean decomposition scheme for the integration of planning and scheduling of a network of batch plants. *Industrial & Engineering Chemistry Research, 52*, 2152–2167.

Camargo, V. C. B., Mattiolli, L., & Toledo, F. M. B. (2012). A knapsack problem as a tool to solve the production planning problem in small foundries. *Computers & Operations Research, 39*, 86–92.

Clark, A. R. (2005). Rolling horizon heuristics for production planning and set-up scheduling with backlogs and error-prone demand forecasts. *Production Planning & Control, 16*, 81–97.

Conradie, D., Morison, L. E., & Joubert, J. W. (2008). Scheduling at coal handling facilities using simulated annealing. *Mathematical Methods of Operations Research, 68*, 277–293.

Dijkhuizen, G., & Faigle, U. (1993). A cutting-plane approach to the edge-weighted maximal clique problem. *European Journal of Operational Research, 69*, 121–130.

Drexl, A., & Kimms, A. (1997). Lot sizing and scheduling: Survey and extensions. *European Journal of Operational Research, 99*, 221–235.

Feo, T. A., & Resende, M. G. (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters, 8*, 67–71.

Feo, T. A., & Resende, M. G. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization, 6*, 109–133.

Ferreira, D., Morabito, R., & Rangel, S. (2010). Relax and fix heuristics to solve one-stage one-machine lot-scheduling models for small-scale soft drink plants. *Computers & Operations Research, 37*, 684–691.

Gaglioppa, F., Miller, L. A., & Benjaafar, S. (2008). Multitask and multistage production planning and scheduling for process industries. *Operations Research, 56*, 1010–1025.

Gandhi, R., Halldrsson, M., Kortsarz, G., & Shachnai, H. (2005). Improved bounds for sum multicoloring and scheduling dependent jobs with minsum criteria. In G. Persiano & R. Solis-Oba (Eds.), *Approximation and online algorithms*. *Lecture notes in computer science* (Vol. 3351, pp. 68–82). Berlin Heidelberg: Springer.

Ji, X., & Mitchell, J. E. (2007). Branch-and-price-and-cut on the clique partitioning problem with minimum clique size requirement. *Discrete Optimization, 4*, 87–102. Mixed Integer Programming IMA Special Workshop on Mixed-Integer Programming.

Kim, B., Koo, B., & Park, B. S. (2009). A raw material storage yard allocation problem for a large-scale steelworks. *The International Journal of Advanced Manufacturing Technology, 41*, 880–884.

Kis, T., & Kovcs, A. (2012). A cutting plane approach for integrated planning and scheduling. *Computers & Operations Research, 39*, 320–327.

Li, Z., & Ierapetritou, M. G. (2010). Rolling horizon based planning and scheduling integration with production capacity consideration. *Chemical Engineering Science, 65*, 5887–5900.

Mateus, G. R., Ravetti, M. G., Souza, M. C., & Valeriano, T. M. (2010). Capacitated lot sizing and sequence dependent setup scheduling: An iterative approach for integration. *Journal of Scheduling, 13*, 245–259.

Menezes, G. C., Mateus, G. R. (2013). Integrated planning and scheduling in bulk cargo terminal. Technical report, Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Brasil.

Menezes, G. C., Mateus, G. R., & Ravetti, M. G. (2015). Scheduling with incompatible jobs: Model and algorithms. In *Proceedings of the 7th multidisciplinary international conference on scheduling: Theory and applications (MISTA), 25–28 August, Prague, Czech Republic* (pp. 776–781).

Meyr, H. (2000). Simultaneous lotsizing and scheduling by combining local search with dual reoptimization. *European Journal of Operational Research, 120*, 311–326.

Meyr, H., & Mann, M. (2013). A decomposition approach for the general lotsizing and scheduling problem for parallel production lines. *European Journal of Operational Research, 229*, 718–731.

Mndez-Daz, I., & Zabala, P. (2008). A cutting plane algorithm for graph coloring. *Discrete Applied Mathematics, 156*, 159–179. Computational Methods for Graph Coloring and it's Generalizations.

Mohammadi, M., Fatemi Ghomi, S. M. T., Karimi, B., & Torabi, S. A. (2008). Rolling-horizon and fix-and-relax heuristics for the multi-product multi-level capacitated lotsizing problem with sequence-dependent setups. *Journal of Intelligent Manufacturing, 21*, 501–510.

Omara, F. A., & Arafa, M. M. (2010). Genetic algorithms for task scheduling problem. *Journal of Parallel and Distributed Computing, 70*, 13–22.

Ozdamar, L., & Yazgac, T. (1999). A hierarchical planning approach for a production-distribution system. *International Journal of Production Research, 37*, 3759–3772.

Pan, Q. K., & Ruiz, R. (2014). An effective iterated greedy algorithm for the mixed no-idle permutation flowshop scheduling problem. *Omega, 44*, 41–50.

Phanden, R. K., Jain, A., & Verma, R. (2013). An approach for integration of process planning and scheduling. *International Journal of Computer Integrated Manufacturing, 26*, 284–302.

Rajkumar, M., Asokan, P., Anilkumar, N., & Page, T. (2011). A grasp algorithm for flexible job-shop scheduling problem with limited resource constraints. *International Journal of Production Research, 49*, 2409–2423.

Ramezanian, R., Saidi-Mehrabad, M., & Teimoury, E. (2013). A mathematical model for integrating lot-sizing and scheduling problem in capacitated flow shop environments. *The International Journal of Advanced Manufacturing Technology, 66*, 347–361.

Robenek, T., Umang, N., & Bierlaire, M. (2014). A branch-and-price algorithm to solve the integrated berth allocation and yard assignment problem in bulk ports. *European Journal of Operational Research, 235*, 399–411.

Rocha, P. L., Ravetti, M. G., Mateus, G. R., & Pardalos, P. M. (2008). Exact algorithms for a scheduling problem with unrelated parallel machines and sequence and machine-dependent setup times. *Computers & Operations Research, 35*, 1250–1264.

Ruiz, R., & Sttzle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research, 177*, 2033–2049.

Singh, G., Sier, D., Ernst, A. T., Gavriliouk, O., Oyston, R., Giles, T., et al. (2012). A mixed integer programming model for long term capacity expansion planning: A case study from the hunter valley coal chain. *European Journal of Operational Research, 220*, 210–224.

de Souza, M. C., Jr, A. C. G., Bretas, A. M., & Ravetti, M. G. (2015). Models for scheduling charges in continuous casting: Application to a Brazilian steel plant. *Optimization Letters*, 1–17.

Stix, V. (2004). Finding all maximal cliques in dynamic graphs. *Computational Optimization and Applications, 27*, 173–186.

Toledo, C. F. M., Silva Arantes, M., Hossomi, M. Y. B., França, P. M., & Akartunalı, K. (2015). A relax-and-fix with fix-and-optimize heuristic applied to multi-level lot-sizing problems. *Journal of Heuristics, 21*, 687–717.

Tomita, E., Tanaka, A., & Takahashi, H. (2006). The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science, 363*, 28–42. Computing and Combinatorics10th Annual International Conference on Computing and Combinatorics (COCOON 2004).

Wolosewicz, C., Dauzre-Prs, S., & Aggoune, R. (2015). A Lagrangian heuristic for an integrated lot-sizing and fixed scheduling problem. *European Journal of Operational Research, 244*, 3–12.

Wu, D., & Ierapetritou, M. (2007). Hierarchical approach for production planning and scheduling under uncertainty. *Chemical Engineering and Processing: Process Intensification, 46*, 1129–1140.

Xu, H., L, Z., & Cheng, T. (2014). Iterated local search for single-machine scheduling with sequence-dependent setup times to minimize total weighted tardiness. *Journal of Scheduling, 17*, 271–287.

You, F., Grossmann, I. E., & Wassick, J. M. (2011). Multisite capacity, production, and distribution planning with reactor modifications: Milp model, bilevel decomposition algorithm versus lagrangean decomposition scheme. *Industrial & Engineering Chemistry Research, 50*, 4831–4849.