A Novel Priority-Slot Based Continuous-Time Formulation for Crude-Oil Scheduling Problems

Sylvain Mouret,[†] Ignacio E. Grossmann,[*,†] and Pierre Pestiaux[‡]

Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA, and Centre de Recherche de Gonfréville, Total France, 76700 Le Havre, France

E-mail: grossmann@cmu.edu

October 2008

Abstract

The optimal scheduling of crude-oil operations in refineries has been studied by various groups during the past decade leading to different mixed integer linear programming (MILP) or mixed integer nonlinear programming (MINLP) formulations. This paper presents a new continuous-time formulation, called single-operation sequencing (SOS) model, which can be used to solve the crude-oil operations problem introduced by Lee et al. [1]. It is different from previous formulations as it requires to postulate the number of priority-slots in which operations take place instead of specifying the number of time intervals or event points to be used in the schedule. This MINLP model is also based on the representation of a crude-oil schedule by a single sequence of transfer operations. It allows breaking symmetries involved in the problem, thus tremendously reducing the computational expenses (all instances can be solved within 2 minutes). A simple two step MILP - NLP procedure has been used to solve the non-convex MINLP model leading to an optimality gap lower than 5% in all cases.

[†]Carnegie Mellon University

[‡]Total France

Introduction

Scheduling problems are among the most challenging optimization problems, both in terms of modeling and solution algorithm. Mostly mixed integer linear programming (MILP), constraint programming (CP) and genetic algorithm (GA) techniques have been used to tackle these problems. CP has proved to be very efficient for solving scheduling problems but it is rarely used to solve problems arising in the chemical engineering field. One of the reason is that CP is very efficient at sequencing tasks or jobs clearly defined a priori (e.g. job-shop problems in discrete manufacturing). However, the scheduling of chemical processes usually involves both defining and sequencing the tasks that should be performed. Defining tasks means choosing a batch size or a unit operating mode for example. As a consequence, LP based techniques have been preferred with formulations essentially based on time grids as it easily allows modeling reservoir or unit capacity at the ends of each time interval.

Uniform time discretization (usually referred as discrete-time) formulations have first been successfully used to solve batch processes based on an STN (state-task-network, Kondili et al. [2]) or RTN (resource-task-network, Pantelides [3]) representation of the process. This formulation has two main advantages: it can easily be applied to many different problems and it has a very tight continuous relaxation. However, when a large number of time intervals is needed (for example in the a case of a large scheduling horizon), the model size may become intractable, even for efficient commercial solvers.

Nonuniform time discretization (or global event) formulations have been introduced based on the RTN representation (Zhang and Sargent [4], Schilling and Pantelides [5]) or STN representation (Mockus and Reklaitis [6]). The main difference with the discrete-time approach is that the duration of the time intervals is not fixed and has to be determined by the solver. This approach is also easy to implement and may be used for larger scheduling horizon, as it leads to more compact models. However, its continuous relaxation is in general less tight making the model hard to solve, even with few time intervals.

A unit specific (or operation specific) event formulation has been introduced by Ierapetritou and Floudas [7]. In this approach, each unit (or operation) has its own set of time invervals which is not shated with other units (or operations). It has led to better computational performances compared to the previous continuous-time formulations as fewer events need to be postulated.

A detailed comparison between these three approaches can be found in Floudas and Lin [8]. Even though a great amount of work has been done in order to decrease the model size while tightening its continuous relaxation, these scheduling formulation have a common disadvantage. As mentioned by Kallrath [9] they often display many degeneracies or symmetries leading to poor computational performances when postulating a large number of time intervals or events.

The aim of this paper is to propose a novel continuous-time model for the crude-oil scheduling problem introduced by Lee et al. [1] and to efficiently solve it by using symmetry-breaking techniques. The single-operation sequencing (SOS) model introduced is based on the representation of a schedule as a sequence of operations. A sequencing rule is used to remove symmetric solutions and reduce the size of the search space.

Commercial software such as GRTMPS (Haverly Systems), PIMS (Aspen Tech) and RPMS (Honeywell Hi-Spec Solutions), mostly based on successive linear programming techniques, have been used to address refinery planning since the early 80's and have led to increased operational benefits. The optimal scheduling of crude-oil operations have been studied since the 90's and has been shown to lead to multimillion dollar benefits by Kelly and Mann [10] as it is the first stage of the oil refining process. It involves crude-oil unloading from crude marine vessels (at berths or jetties) or from a pipeline to storage tanks, transfers from storage tanks to charging tanks and atmospheric distillations of crude-oil mixtures from charging tanks. The crude is then processed in order to produce basic products which are then blended into gasoline, diesel, and other final products. Assuming that the schedule of crude supply and production demands are determined by the long-term refinery planning, this paper studies the short-term scheduling problem maximizing gross margins of crude-oil mixtures.

Shah [11] proposed to use mathematical programming techniques to find crude-oil schedules exploiting opportunities to increase economic benefits. Lee et al. [1] defined a precise crude-oil scheduling problem involving crude unloading at berths, developped a discrete-time MINLP model, and solved a linear relaxation of the model. Later, Wenkay and Hui [12] improved the model and proposed an iterative approach to solve the MINLP model, taking into account nonlinear blending constraints. Pinto et al. [13], Moro and Pinto [14], and Reddy et al. [15] used a global event formulation to model refinery systems involving crude-oil unloading from pipeline or jetties. The scheduling horizon is divided into fixed length sub-intervals, which are then divided in several variable length time-slots. In parallel, Jia et al. [16] developed an operation specific event model and applied it to the problems introduced by Lee et al. [1], using a linear approximation of storage costs. A comparison of computational performances between both continuous-time and discrete-time models is given, showing tremendous decrease in CPU time. Also, locally optimal solutions have been obtained using a MINLP algorithm. Karuppiah et al. [17] later addressed the global optimization of this model using a Lagrangean decomposition of the refinery system. While rigorous, this method is computationally expensive.

This paper is organized as follows. First, the problem definition is given and a continuous-time single-operation sequencing (SOS) formulation is proposed. Next, a simple solution method to solve the SOS model is presented and a new approach for breaking symmetries using a sequencing rule is introduced. Finally, computational results are given to show the effectiveness of the proposed model and solution method.

Problem Statement

This paper aims at solving a refinery crude-oil operations problem. A crude-oil operations system is composed of four types of resources: crude marine vessels, storage tanks, charging tanks and crude distillation units (CDUs). Three types of operations, all transfers between resources, are allowed: crude-oil unloading from marine vessels to storage tanks, transfer between tanks, and transfers of charging tanks to CDUs.

A crude-oil scheduling problem is defined by: (a) a time horizon, (b) arrival time of marine vessels, (c) capacity limits of tanks, (d) transfer flowrate limitations, (e) initial composition of vessels and tanks, (f) crude property specifications for distillations, (g) and demands for each crude blend.

The logistics constraints of the problem are defined as follows.

(i) only one berth is available at the docking station for vessel unloadings,

(ii) simultaneous inlet and outlet transfers on tanks are forbidden,

(iii) a tank may charge only one CDU at a time,

(iv) a CDU can be charged by only one tank at a time,

(v) and CDUs must be operated continuously throughout the scheduling horizon.

The goal is to determine how many times each operation will be executed, when it will be performed (start time and duration), and the volume of crude to be transferred in order to maximize the gross margins of distilled mixed oil. The gross margin of each crude can be estimated from the sale income of the final products minus its purchase value and related refining operational costs. Depending on the market value of the different crudes and final products, the model tries to process the most profitable crudes and to store the other crudes. Unloading and storage costs are ignored.

As CDU switches between different crude blends are costly, it is considered that the number of distillation batches is bounded. The bounds on the number of distillation operations can be set by the operator or, for the lower bound, can be obtained by solving a model minimizing the number of distillations. Typically, the number and type of marine vessels is determined at the planning level as well as demands in each crude blend.

Single-Operation Sequencing Model

Basic idea

The continuous-time formulation presented in this section is based on the representation of a schedule as a sequence of operations. To obtain the optimal schedule, a sequence of priority-slots is used. A priority-slot is a position i in the sequence of operations, which has a higher scheduling priority than other priority-slots j such that $j > i$. Each priority-slot is to be assigned to exactly one specific operation (e.g. unloading, tank-to-tank transfer, distillation). The number of priority-slots corresponds to the length of the sequence of operations, which is the total number of operations that will be executed during the scheduling horizon. The number of priority-slots has to be postulated a priori by the user.

One of the most common logistics constraints appearing in scheduling problems is the non-overlapping constraint between two operations v and w, noted $v][w$. The logistics constraints (i), (ii), (iii), and (iv) from the previous section can all be expressed as non-overlapping constraints. Indeed, for the problem 1 (see Figure 1), (i) is equivalent to 1][2, (ii) to 1][3, 1][4, 2][5, 2][6, 3][7, 4][8, 5][7, 6][8, and (iv) to 7][8.

Assuming that two non-overlapping operations v and w are assigned to priority-slots i and j such that $i < j$ (i has a higher scheduling priority than j), we define S_{iv} and S_{jw} their respective start times and D_{iv} and D_{jw} their respective durations. As the operation v has the highest priority, operation w must start after the end of operation v.

$$S_{iv} + D_{iv} \leq S_{jw}$$

Based on a given sequence of operations, any schedule defined using this precedence rule is such that any pair of non-overlapping operations do not overlap. Any given schedule can be obtained by the model by ordering the sequence of operations with respect to their start time.

Case-study

Figure 1 depicts the refinery configuration for the first example presented in Lee et al. [1]. The scheduling horizon is composed of 8 days and two marine vessels are scheduled to arrive at the beginning of day 1 ($t = 0$) and day 5 ($t = 4$) and contain 1 million bbl of crude-oil A and B, respectively. There is one CDU which has to process 1 million bbl of each crude-oil mixture, X and Y. The property that is being tracked is the sulfur concentration, 0.01 for crude A and 0.06 for crude B. The sulfur concentration of crude mix X and Y should be in the ranges $[0.015, 0.025]$ and $[0.045, 0.055]$, respectively. The two storage tanks initially contain 250,000 bbl of crude A and 750,000 bbl of crude B. The two charging tanks initially contain 500,000 bbl of crude C and D, with sulfur concentrations of 0.02 and 0.05, respectively. Crudes C and D are in fact blends of crudes A and B that match the sulfur concentration specifications for crude mix X and Y. The gross margins of crudes A, B, C, and D are 9 $/bbl, 4 $/bbl, 8 $/bbl, 5 $/bbl, respectively. Gross margins for crudes C and D have been calculated from gross margins of crudes A and B. The data for problem 1 is given in Table 1. Flowrate limitations are expressed in Mbbl/day.

Figure 2 depicts the Gantt chart of a sub-optimal solution for problem 1 with a profit of $6,925,000, that might be obtained by using heuristics. It can be represented as the sequence of operations **8313746852**. Each task is represented by a horizontal bar labeled with its scheduling priority (or position in the sequence), and each row corresponds to a specific operation. Tank inventories are also displayed, showing that tank capacity limits are satisfied. Figure 3 shows in contrast the Gantt chart of the optimal solution with a profit of $7,975,000, which represents a 13.2% increase in profit. It can be represented as the sequence of operations **7683513762**. Clearly, finding such a solution is non-trivial.

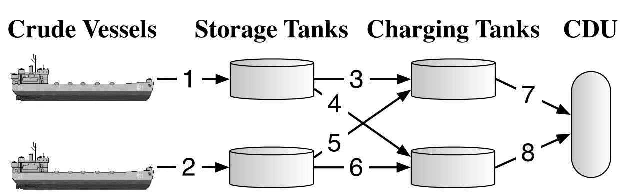

Figure 1: Crude-oil operations system for problem 1

Table 1: Problem 1 data

Scheduling horizon			8 days
Vessels	Arrival time	Composition	Amount of crude (Mbbl)
Vessel 1 Vessel 2	0 4	100% A 100% B	1,000 1,000
Storage tanks	Capacity (Mbbl)	Initial composition	Initial amount of crude (Mbbl)
Tank 1 Tank 2	[0, 1,000] [0, 1,000]	100% A 100% B	250 750
Charging tanks	Capacity (Mbbl)	Initial composition	Initial amount of crude (Mbbl)
Tank 1 (mix X) Tank 2 (mix Y)	[0, 1,000] [0, 1,000]	100% C 100% D	500 500
Crudes	Property 1 (sulfur concentration)		Gross margin ($/bbl)
Crude A Crude B Crude C Crude D	0.01 0.06 0.02 0.05		9 4 8 5
Crude mixtures	Property 1 (sulfur concentration)		Demand (Mbbl)
Crude mix X Crude mix Y	[0.015, 0.025] [0.045, 0.055]		[1,000, 1,000] [1,000, 1,000]
Unloading flowrate	[0, 500]	Transfer flowrate	[0, 500]
Distillation flowrate	[50, 500]	Number of distillations	3

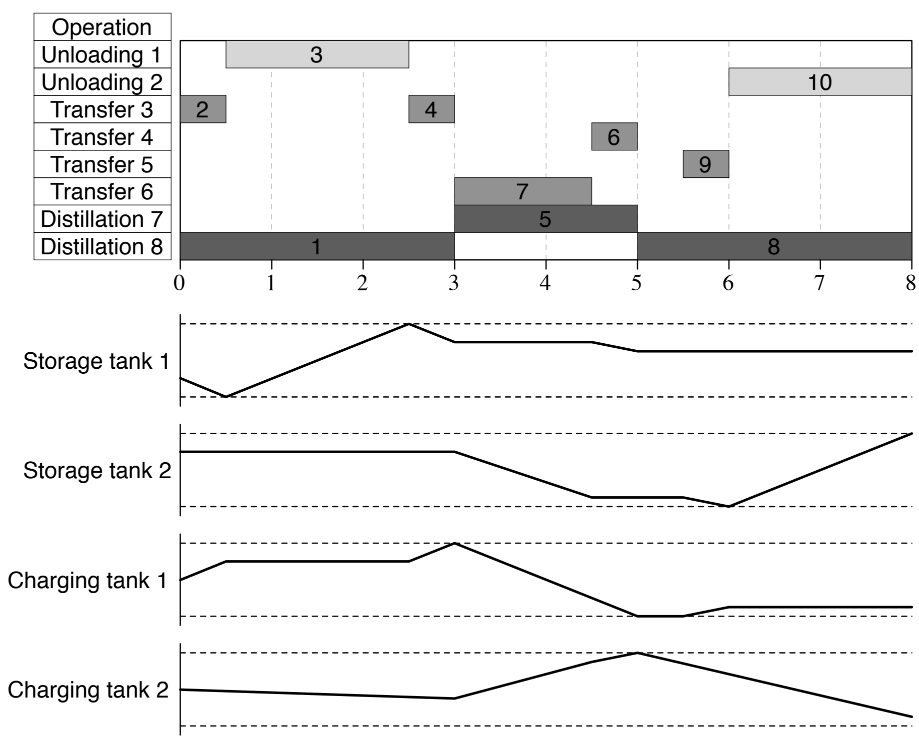

Figure 2: Sub-optimal schedule for problem 1 (profit: $6,925,000)

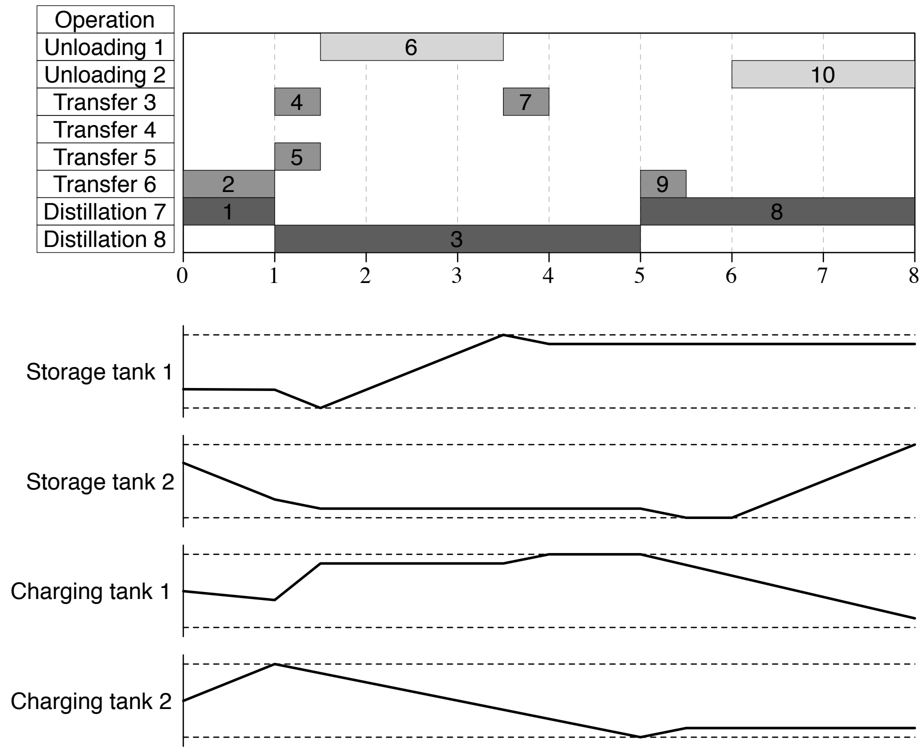

Figure 3: Optimal schedule for problem 1 (profit: $7,975,000)

Sets

The following sets will be used in the model.

- $T = \{1,\ldots,n\}$ is the set of priority-slots
- W is the set of all operations
- $W_U \subset W$ is the set of unloading operations
- $W_T \subset W$ is the set of tank-to-tank transfer operations
- $W_D \subset W$ is the set of distillation operations
- R is the set of resources (i.e. reservoirs)
- $R_V \subset R$ is the set of vessels
- $R_S \subset R$ is the set of storage tanks

- $R_C \subset R$ is the set of charging tanks
- $R_D \subset R$ is the set of distillation units
- $I_r \subset W$ is the set of inlet transfer operations on resource r
- $O_r \subset W$ is the set of outlet transfer operations on resource r
- C is the set of products (i.e. crudes)
- K is the set of product properties (e.g. crude sulfur concentration)

Variables

The variables used in the model can be decomposed into assignment variables, time variables, operation variables and resource variables.

- **Assignment variables** Z_{iv}.

  $Z_{iv} = 1$ if operation v is assigned to priority-slot i, $Z_{iv} = 0$ otherwise.

- **Time variables** S_{iv} and D_{iv}.

  S_{iv} is the start time of operation v if it is assigned to priority-slot i, $S_{iv} = 0$ otherwise.

  D_{iv} is the duration of operation v if it is assigned to priority-slot i, $D_{iv} = 0$ otherwise.

- **Operation variables** V_{iv} and V_{ivc}.

  V_{iv} is the total volume of crude transferred during operation v if it is assigned to priority-slot i, $V_{iv} = 0$ otherwise.

  V_{ivc} is the volume of crude c transferred during operation v if it is assigned to priority-slot i, $V_{ivc} = 0$ otherwise.

- **Resource variables** L_{ir} and L_{irc}.

  L_{ir} is the *intermediate* total level of crude in tank $r \in R_S \cup R_C$ before the operation assigned to priority-slot i.

L_{irc} is the *intermediate* level of crude c in tank $r \in R_S \cup R_C$ before the operation assigned to priority-slot i.

It should be noted that the crude composition of blends in tanks is tracked instead of their properties. The distillation specifications are later enforced by calculating a posteriori the properties of the blend in terms of its composition. For instance, in problem 1, a blend composed of 50% of crude A and 50% of crude B has a sulfur concentration of 0.035 which does not meet the specification for crude mix X nor for crude mix Y.

Assignment constraints

Assignment constraints restrict the operations that can be assigned to each priority-slot. In the SOS model, exactly one operation has to be assigned to each priority-slot.

$$\sum_{v \in W} Z_{iv} = 1 \qquad i \in T \tag{1}$$

Variable constraints

Variable constraints are given by their definitions. Start time, duration and global volume variables are defined with big-M constraints.

$$S_{iv} + D_{iv} \leq H \cdot Z_{iv} \qquad i \in T, v \in W \tag{2}$$

$$V_{iv} \leq \overline{V_v} \cdot Z_{iv} \qquad i \in T, v \in W \tag{3}$$

Crude volume variables are positive variables whose sum equals the corresponding global volume variable.

$$\sum_{c \in C} V_{ivc} = V_{iv} \qquad i \in T, v \in W \tag{4}$$

Total and crude level variables are defined by adding to the initial level in the tank all inlet and outlet transfer volumes of operations of higher priority than the considered priority-slot.

$$L_{ir} = L_{0r} + \sum_{j \in T, j<i} \sum_{v \in I_r} V_{iv} - \sum_{j \in T, j<i} \sum_{v \in O_r} V_{iv} \qquad i \in T, r \in R \qquad (5)$$

$$L_{irc} = L_{0rc} + \sum_{j \in T, j<i} \sum_{v \in I_r} V_{ivc} - \sum_{j \in T, j<i} \sum_{v \in O_r} V_{ivc} \qquad i \in T, r \in R, c \in C \qquad (6)$$

Sequencing constraints

Sequencing constraints restrict the set of possible sequences of operations. Cardinality and unloading sequence constraints are specific cases of sequencing constraints. More complex sequencing constraints will also be discussed later.

Cardinality constraint. Each crude-oil marine vessel has to unload its content exactly once.

$$\sum_{i \in T} \sum_{v \in O_r} Z_{iv} = 1 \qquad r \in R_V \qquad (7)$$

The total number of distillations is bounded by $\underline{N_D}$ and $\overline{N_D}$ in order to reduce the cost of CDU switches.

$$\underline{N_D} \leq \sum_{i \in T} \sum_{v \in W_D} Z_{iv} \leq \overline{N_D} \qquad (8)$$

Unloading sequence constraint Marine vessels have to unload in order of arrival to the refinery. Considering two vessels $r_1, r_2 \in R_V$, $r_1 < r_2$ signifies that r_1 unloads before r_2.

$$\sum_{j \in T, j<i} \sum_{v \in O_{r_2}} Z_{jv} + \sum_{j \in T, j \geq i} \sum_{v \in O_{r_1}} Z_{jv} \leq 1 \qquad i \in T, i \neq 1, r_1, r_2 \in R_V, r_1 < r_2 \qquad (9)$$

Scheduling constraints

Scheduling constraints restrict the values taken by time variables according to logistics rules.

Non-overlapping constraint. A non-overlapping constraint between two sets of operations $W_1 \subset W$ and $W_2 \subset W$ states that any pair of operations $(v_1, v_2) \in W_1 \times W_2$ must not be executed simultaneously.

Unloading operations must not overlap.

$$\sum_{v \in W_U} (S_{iv} + D_{iv}) \leq \sum_{v \in W_U} S_{jv} + H \cdot (1 - \sum_{v \in W_U} Z_{jv}) \qquad i, j \in T, i < j \tag{10}$$

Inlet and outlet transfer operations on a tank must not overlap.

$$\sum_{v \in I_r} (S_{iv} + D_{iv}) \leq \sum_{v \in O_r} S_{jv} + H \cdot (1 - \sum_{v \in O_r} Z_{jv}) \qquad i, j \in T, i < j, r \in R_S \cup R_C \tag{11}$$

$$\sum_{v \in O_r} (S_{iv} + D_{iv}) \leq \sum_{v \in I_r} S_{jv} + H \cdot (1 - \sum_{v \in I_r} Z_{jv}) \qquad i, j \in T, i < j, r \in R_S \cup R_C \tag{12}$$

Although we do not consider crude settling in storage tanks after vessel unloadings, it could be included in the model, with a modified version of constraint (11) taking into account transition times. We note TR_v the transition time after unloading operation $v \in W_U$ and TR the maximum transition time, $TR = \max_{v \in W_U} TR_v$.

$$\sum_{v \in I_r} (S_{iv} + D_{iv} + TR_v \cdot Z_{iv}) \leq \sum_{v \in O_r} S_{jv} + (H + TR) \cdot (1 - \sum_{v \in O_r} Z_{jv}) \qquad i, j \in T, i < j, r \in R_S \tag{13}$$

Constraint (13) is valid in the 4 possibles cases:

$$(\exists v_1 \in I_r, Z_{iv_1} = 1) \wedge \left(\exists v_2 \in O_r, Z_{jv_2} = 1\right) \Rightarrow S_{iv_1} + D_{iv_1} + TR_{v_1} \leq S_{jv_2}$$

$$(\exists v_1 \in I_r, Z_{iv_1} = 1) \wedge \left(\forall v_2 \in O_r, Z_{jv_2} = 0\right) \Rightarrow S_{iv_1} + D_{iv_1} \leq H + TR - TR_{v_1}$$

$$(\forall v_1 \in I_r, Z_{iv_1} = 0) \wedge \left(\exists v_2 \in O_r, Z_{jv_2} = 1\right) \Rightarrow 0 \leq S_{jv_2}$$

$$(\forall v_1 \in I_r, Z_{iv_1} = 0) \wedge \left(\forall v_2 \in O_r, Z_{jv_2} = 0\right) \Rightarrow 0 \leq H + TR$$

A tank may charge only one CDU at a time.

$$\sum_{v \in O_r} (S_{iv} + D_{iv}) \leq \sum_{v \in O_r} S_{jv} + H \cdot (1 - \sum_{v \in O_r} Z_{jv}) \qquad i, j \in T, i < j, r \in R_C \tag{14}$$

A CDU may be charged by only one tank at a time.

$$\sum_{v \in I_r} (S_{iv} + D_{iv}) \leq \sum_{v \in I_r} S_{jv} + H \cdot (1 - \sum_{v \in I_r} Z_{jv}) \qquad i, j \in T, i < j, r \in R_D \tag{15}$$

In order to avoid schedules in which a transfer is being performed twice at a time, thus possibly violating the flowrate limitations, constraint (16) is included in the model.

$$S_{iv} + D_{iv} \leq S_{jv} + H \cdot (1 - Z_{jv}) \qquad i, j \in T, i < j, v \in W \tag{16}$$

Continuous distillation constraint. It is required that CDUs operate without interruption. As CDUs perform only one operation at a time, the continuous operation constraint is defined by equating the sum of the duration of distillations to the time horizon.

$$\sum_{i \in T} \sum_{v \in I_r} D_{iv} = H \qquad r \in R_D \tag{17}$$

Resource availability constraint. Unloading of crude-oil vessels may start only after arrival to the refinery. Let S_r be the arrival time of vessel r.

$$S_{iv} \geq S_r \cdot Z_{iv} \qquad i \in T, r \in R_V, v \in O_r \tag{18}$$

Operation constraints

Operation constraints restrict the values taken by operation and time variables according to operational rules.

Flowrate constraint. The flowrate of transfer operation v is bounded by $\underline{FR_v}$ and $\overline{FR_v}$.

$$\underline{FR_v} \cdot D_{iv} \leq V_{iv} \leq \overline{FR_v} \cdot D_{iv} \qquad i \in T, v \in W \qquad (19)$$

Minimum volume constraint. As vessels need to be entirely discharged, the volume transferred during unloading of vessel r must be greater than its initial content V_r.

$$V_{iv} \geq V_r \cdot Z_{iv} \qquad i \in T, v \in W \qquad (20)$$

Property constraint. The property k of the blended products transferred during operation v is bounded by $\underline{x_{vk}}$ and $\overline{x_{vk}}$. The property k of the blend is calculated from the property x_{ck} of crude c assuming that the mixing rule is linear.

$$\underline{x_{vk}} \cdot V_{iv} \leq \sum_{c \in C} x_{ck} V_{ivc} \leq \overline{x_{vk}} \cdot V_{iv} \qquad i \in T, v \in W, k \in K \qquad (21)$$

Composition constraint. It has been shown[18] that processes including both mixing and splitting of streams cannot be expressed as a linear model. Mixing occurs when two streams are used to fill a tank and is expressed linearly in constraint (6). Splitting occurs when partially discharging a tank, resulting in two parts: the remaining content of the tank and the transferred products. This constraint is nonlinear. The composition of the products transferred during a transfer operation must be identical to the composition of the origin reservoir.

$$\frac{L_{irc}}{L_{ir}} = \frac{V_{ivc}}{V_{iv}} \qquad i \in T, r \in R, v \in O_r, c \in C \qquad (22)$$

Constraint (22) is reformulated as an equation involving bilinear terms.

$$V_{ivc} \cdot L_{ir} = L_{irc} \cdot V_{iv} \qquad i \in T, r \in R, v \in O_r, c \in C \qquad (23)$$

Note that constraint (23) is correct even when operation v is not assigned to priority-slot i, as $V_{iv} = V_{ivc} = 0$.

Resource constraints

Resource constraints restrict the use of resources throughout the scheduling horizon.

Reservoir capacity constraint. The level of materials in the tank r must remain between minimum and maximum capacity limits $\underline{L_r}$ and $\overline{L_r}$, respectively. Let L_{0r} be the initial total level and L_{0rc} be the initial level of crude c in the reservoir r. As simultaneous charging and discharging of tanks is forbidden, the following constraints are sufficient.

$$\underline{L_r} \leq L_{ir} \leq \overline{L_r} \qquad i \in T, r \in R_S \cup R_C \tag{24}$$

$$0 \leq L_{irc} \leq \overline{L_r} \qquad i \in T, r \in R_S \cup R_C, c \in C \tag{25}$$

$$\underline{L_r} \leq L_{0r} + \sum_{i \in T} \sum_{v \in I_r} V_{iv} - \sum_{i \in T} \sum_{v \in O_r} V_{iv} \leq \overline{L_r} \qquad r \in R_S \cup R_C \tag{26}$$

$$0 \leq L_{0rc} + \sum_{i \in T} \sum_{v \in I_r} V_{ivc} - \sum_{i \in T} \sum_{v \in O_r} V_{ivc} \leq \overline{L_r} \qquad r \in R_S \cup R_C, c \in C \tag{27}$$

Demand constraint. Demand constraints define lower and upper limits, $\underline{D_r}$ and $\overline{D_r}$, on the products transferred out of each charging tank r during the scheduling horizon.

$$\underline{D_r} \leq \sum_{i \in T} \sum_{v \in O_r} V_{iv} \leq \overline{D_r} \qquad r \in R_C \tag{28}$$

Objective function

The objective is to maximize the gross margins of the distilled crude blends. Let G_c be the individual gross margin of the crude c.

$$\max \sum_{i \in T} \sum_{r \in R_D} \sum_{v \in I_r} \sum_{c \in C} G_c \cdot V_{ivc} \tag{29}$$

Solution Method

The non-convex MINLP model given in the previous section can be solved using any generic MINLP solver such as DICOPT (outer-approximation method) or BARON (global solver using a spatial branch-and-bound search). These methods can be prohibitively expensive when used to solve this model. Therefore, a simple two-stage procedure has been implemented (see Figure 4), leading to locally optimal solutions with an estimation of the optimality gap. In the first stage, a linear MILP relaxation of the model, defined by removing nonlinear constraints, is solved. The solution returned by the MILP solver may violate the bilinear composition constraint (23). In this case, the binary variables Z_{iv} are fixed, which means that the sequence of operations is fixed, and the resulting nonlinear programming (NLP) model is solved using the solution of the MILP as a starting point. This NLP model contains the same constraints as in the MILP model plus the nonlinear constraints. The solution obtained at this stage might not be the optimum of the full model, but the optimality gap can be estimated from the lower bound given by the MILP solution and the upper bound given by the NLP solution. It should be noted that the NLP is non-convex and thus may lead to different locally optimal solutions depending on the starting point. It could also be locally infeasible although it did not occur in our experiments. For such a case, one could add integer cuts to the MILP model and restart the procedure until a solution is found. However, there is no proof that the solution obtained with this algorithm is globally optimal. Also, it cannot be ensured that a solution will be found, even if the MINLP has feasible solutions.

Symmetry-Breaking Constraints

As mentioned by Kallrath [9], degeneracies and symmetries often cause scheduling problems to be difficult to solve. The author suggests that nonlinearities involved in refinery scheduling problems may reduce these effects. In this section, another approach is proposed in order to break the symmetries which can be detected in the SOS model. It is based on the concept of static symmetry-breaking constraints as presented by Margot [19].

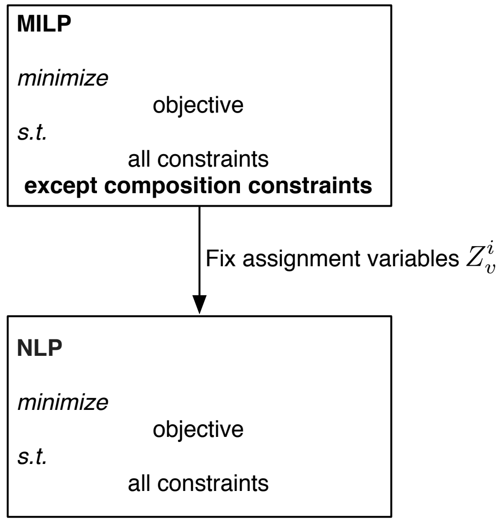

Figure 4: Two-stage decomposition strategy

Symmetric sequences of operations

It is possible to exhibit different sequences of operations leading to the same schedule. For the optimal solution of problem 1 (Figure 3), there are 96 symmetric sequences of operations, 4 of which are displayed in Figure 5. These sequences are obtained from the optimal one by moving operations in the sequence in such a way that the same non-overlapping constraints are active. As there are many different optimal discrete solutions, the Branch & Bound algorithm will explore many redundant nodes of the search tree. Therefore, an efficient symmetry-breaking tool is needed to avoid searching irrelevant solutions.

A sequencing rule based on a regular language

A sequencing rule is defined in order to select the sequences of operations to be explored. This rule is expressed as a regular language which can be recognized by a deterministic finite automaton (DFA). A regular language is a set of words (sequences of letters) defined from an alphabet (i.e. the set of operations) and the empty word ε by the operations concatenation '$\cdot$' (symbol usually omitted), union '+', and Kleene star '*'. Given two languages L_1 and L_2, these operations are defined by the following formulas. The reader may refer to Hopcroft and Ullman [20] for a complete

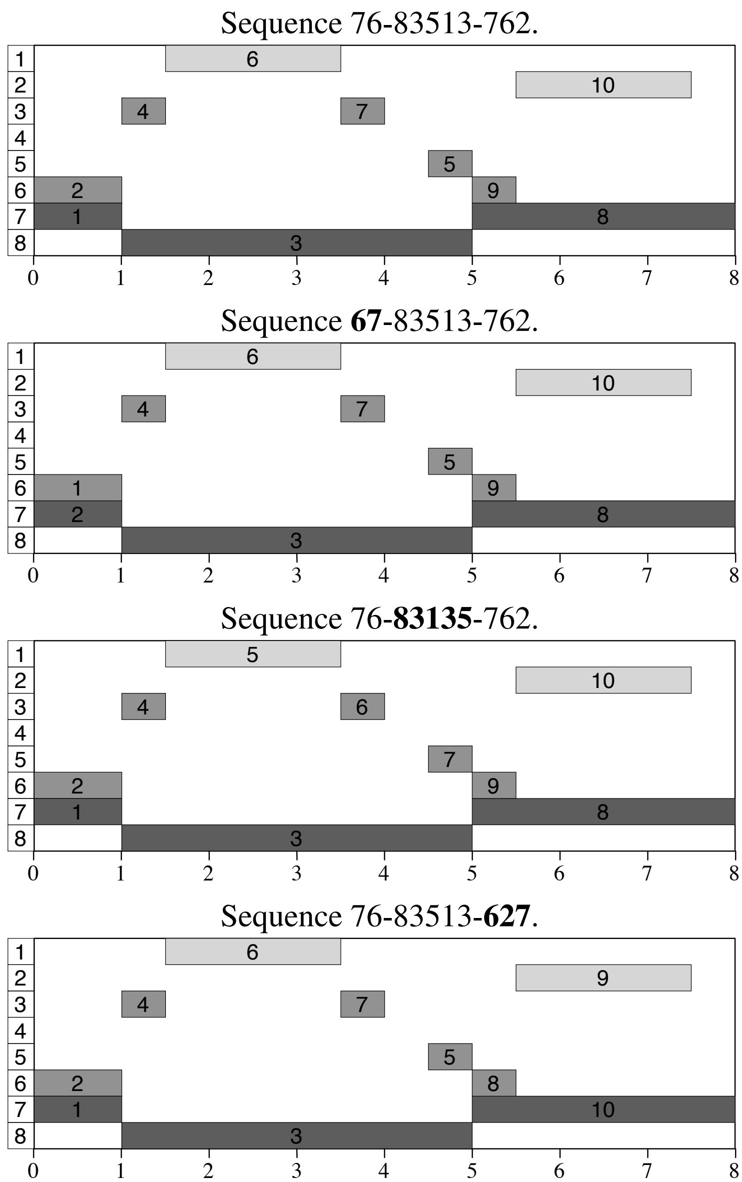

Figure 5: Symmetric sequences of operations for problem 1

definition of regular languages.

$$L_1 \cdot L_2 = \{w = w_1 \cdot w_2 \text{ s.t. } w_1 \in L_1 \text{ and } w_2 \in L_2\}$$
$$L_1 + L_2 = \{w \text{ s.t. } w \in L_1 \text{ or } w \in L_2\}$$
$$L_1{}^* = \{\varepsilon\} \cup L_1 \cup L_1 \cdot L_1 \cup L_1 \cdot L_1 \cdot L_1 \cup \dots$$

Rule derivation for problem 1

The rule presented in this section has been designed in order to remove as many symmetric sequences of operations as possible, regardless of its complexity. In problem 1, two distillation states exist as either distillation 7 or distillation 8 is being executed at any time. Thus, the sequence of operations can be decomposed into subsequences, each corresponding to one distillation state, as shown in Figure 5.

Let L_7 (resp. L_8) be the regular language describing the possible sequences of operations during distillation state 7 (resp. distillation state 8). Note that only transfer operations 1, 2, 4 and 6 are allowed to be executed during distillation state 7 due to the non-overlapping constraints.

If no unloading operation is performed, operations 4 and 6 need to be executed at most once. Thus, in that case, we choose to define the regular language L_7 so that a subsequence corresponding to the distillation state 7 starts with distillation 7 and may follow by at most one occurrence of transfer operations 4 and 6, in this order.

$$L_7 = \{7, 74, 76, 746\} = 7(\varepsilon + 4)(\varepsilon + 6)$$

If unloading operation 1 is allowed to be executed during distillation state 7, then it can be executed at most once. Also, it might be necessary to perform transfer operation 4 before and after this unloading. Thus, in that case, we define the regular language L_7 as follows.

$$L_7 = 7(\varepsilon + 4)(\varepsilon + 6)(\varepsilon + 1 + 14)$$

If both unloading operations 1 and 2 are allowed to be executed during distillation state 7, then it might also be necessary to perform transfer operation 6 before and after this unloading. Also, unloading 1 has to be sequenced before operation 2 due to the precedence constraint between unloading operations (crude-oil vessels have to unload in order of arrival). Thus, in this general case, we choose to define the regular language L_7 as follows.

$$L_7 = 7(\varepsilon+4)(\varepsilon+6)(\varepsilon+1+14)(\varepsilon+2+26)$$

The set of all sequences of operations belonging to the regular language L_7 is displayed in Table 2. It can be represented by the DFA depicted in Figure 6, noted DFA_7. This DFA reads a sequence starting with operation 7, and then reads the following operation in the sequence by moving though the corresponding labeled arc. A sequence is accepted if it can be entirely be read by DFA_7.

Table 2: List of sequences belonging to regular language L_7

Length	Sequences belonging to L_7
1	7
2	71, 72, 74, 76
3	712, 714, 726, 741, 742, 746, 761, 762
4	7126, 7142, 7412, 7414, 7426, 7461, 7462, 7612, 7614, 7626
5	71426, 74126, 74142, 74612, 74614, 74626, 76126, 76142
6	741426, 746126, 746142, 761426
7	7461426

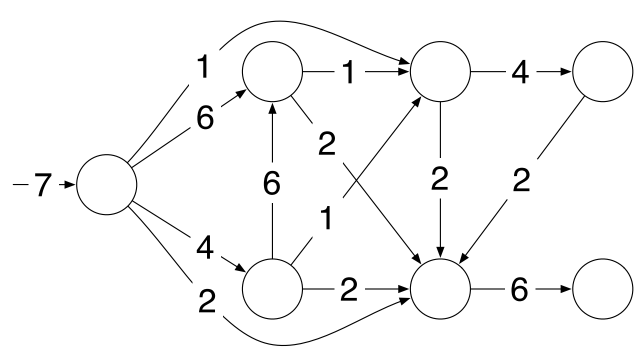

Figure 6: Automaton DFA_7 recognizing regular language L_7

Similarly, the regular language L_8, represented by DFA_8, is defined as follows.

$$L_8 = 8(\varepsilon + 3)(\varepsilon + 5)(\varepsilon + 1 + 13)(\varepsilon + 2 + 25)$$

Finally, the regular language L describing an efficient sequencing rule for the problem 1 can be defined using language L_7 and L_8. Figure 7 shows a scheme of the DFA recognizing the regular language L.

$$L = (\varepsilon + L_7)(L_8 \cdot L_7)^*(\varepsilon + L_8)$$

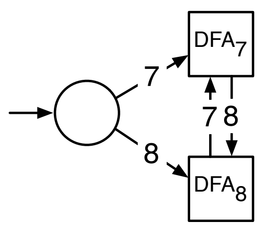

Figure 7: Automaton recognizing the regular language L

It should be noted that this symmetry-breaking rule captures all possible schedules and removes many redundant sequences of operations. However, there are some symmetric sequences that remain such as 78132 and 71832. Indeed, in these two sequences belonging to the language L, exchanging operations 1 and 8 does not change the active non-overlapping constraints.

Regular constraint

Once the regular language L and its corresponding DFA has been defined, it is possible to include the sequencing rule using the linear system of equation proposed by Côté et al.[21]. The equations represent a network flow through a directed layered graph initially introduced by Pesant[22] for Constraint Programming.

Let $M = (Q, \Sigma, \delta, q_1, F)$, where Q is the set of states, Σ is the alphabet, δ is the transition function, q_1 is the initial state, and F is the set of final states, be a DFA recognizing the regular language L. The idea is to unfold the automaton states on $n+1$ layers (where n is the length of the sequence), the first layer corresponding to the initial state, the last layer corresponding to the possible final states, and the transition between layers corresponding to the automaton transitions defined by δ. Then, a sequence is recognized by the DFA M if and only if there is a flow of unit 1 from the initial state in the first layer to a final state in the last layer such that the transition taken between any two adjacent layers i and $i+1$ corresponds to the ith letter of the sequence.

The regular language membership constraint makes use of flow binary variables S_{ivq}, where $q \in Q$. $S_{ivq} = 1$ if the automaton is in state q when operation v is assigned to priority-slot i, $S_{ivq} = 0$ otherwise. The network flow problem is defined with the following constraints.

$$\sum_q S_{ivq} = Z_{iv} \qquad i \in T, v \in W \qquad \text{linking constraint} \tag{30}$$

$$\sum_v S_{1vq_1} = 1 \qquad \text{initial unit flow} \tag{31}$$

$$\sum_{v,q',q=\delta(q',v)} S_{(i-1)vq'} - \sum_v S_{ivq} = 0 \qquad i \in T, i \neq 1, q \in Q \qquad \text{flow conservation} \tag{32}$$

$$\sum_{v,q,\delta(q,v)\in F} S_{nvq} = 1 \qquad \text{final unit flow} \tag{33}$$

Constraint (30) links the Z_{iv} variable to the S_{ivq} variables ($q \in Q$) ensuring that the assignment of operation v to priority-slot i is equivalent to a unit flow traversing the corresponding arcs in the network. Constraint (31) sets the flow leaving the network source node $(1,q_1)$ to 1. Constraint (32) ensures that the flow entering network node (i,q) is equal to the flow leaving it. Constraint (33) sets the flow leaving the network nodes (n,q) through transitions $\delta(q,v) \in F$ to 1.

Although $O(|T| \cdot |W| \cdot |Q|)$ new variables and $O(|T| \cdot |W| + |T| \cdot |Q|)$ new constraints are added to the model, the search space is substantially reduced as only sequences belonging to the regular language L are explored. It should be noted that it is not necessary to declare the variable S_{ivq} as binary. Indeed, if all variables Z_{iv} are fixed, then the sequence of operations is fixed to $v_1 \ldots v_i \ldots v_n$. As the automaton M is deterministic, there is a unique sequence of states $q_1 \ldots q_i \ldots q_n q_{n+1}$ (where

$q_{n+1} \in F$) corresponding to the states visited upon processing the sequence $v_1...v_i...v_n$. Therefore, the network flow problem stated above has a unique solution defined by $S_{ivq} = 1$ if $v = v_i$ and $q = q_i$, $S_{ivq} = 0$ otherwise.

Computational Results

All experiments have been performed on an Intel Core 2 Duo 2.16GHz processor with GAMS as modeling language, Xpress 17.1 as MILP solver and CONOPT 3 as NLP solver. The CPU limit for solving the MILP has been set to 1 hour. The DFAs for problems 2, 3, and 4 have been defined in a similar way as for problem 1, adapted to each refinery configuration.

Scheduling results

Figures 8 and 13 depict the refinery configuration for problems 2, 3, and 4. The data involved in these problems are given in Tables 3, 4, and 5. Figures 9 and 12 depict the optimal solutions for problems 2 and 3. As there is no incentive to keep inventory in the charging tanks, their final level is in general close to zero. However, the crude-oil that arrived late to the refinery is mostly kept in the storage tanks. This leads to higher transfer activity at the beginning than at the end of the scheduling horizon.

The main uncertain parameter in refinery crude-oil scheduling problems is the arrival time of tankers. The expected date of arrival of these marine vessels is known long in advance, but is also subject to many changes before it actually arrives at the refinery. Figure 10 shows the optimal schedule for problem 2 when vessels are scheduled to arrive one day later, leading to a 3.4% profit decrease (from \$10,117,00 to \$9,775,000). It should be noted that the sequence of distillations is different in this case and that there is a higher transfer activity in the very beginning of the scheduling horizon. However, this schedule assumes that the late arrival of vessels is known at time $t = 0$.

Assuming that the exact arrival dates are known slightly later ($t = \varepsilon \ll 1$), the initial decisions are fixed, so that priority-slots 1, 2, 3 and 4 (which start at time $t = 0$) are fixed to operations 12, 14, 4, and 6 respectively. In this case, the optimal solution is depicted in Figure 11. This solution is very similar to the original optimal solution (Figure 9) as most transfers are simply delayed, although the profit is reduced to \$9,609,000. Only transfer operation 10 from $t = 8$ to $t = 8.5$ is no longer used. The fact that the exact arrival of vessels is known only at $t = \varepsilon$ leads to a 1.7% profit decrease (from \$9,775,000 to \$9,609,000) compared to the case where it is known at $t = 0$. This result shows that the original schedule determined at $t = 0$ is not very affected by the late arrival of vessels.

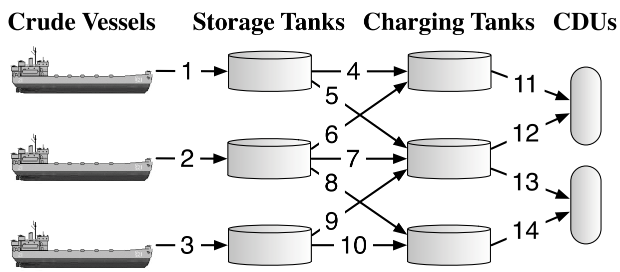

Figure 8: Crude-oil operations system for problems 2 and 3

Effect of symmetry-breaking constraints

In this section, we study the effect of the symmetry-breaking rule on solving the MILP linear relaxation of the MILP-NLP decomposition method established earlier is studied. Two models will be compared: the basic model that includes all linear constraints of the MILP except the symmetry-breaking constraints, and the extended model that includes all linear constraints as well as the symmetry-breaking constraints.

Consider a single instance of problem 1 (Figure 1, Table 1) for which 12 priority-slots are postulated. As seen in Table 6, although the extended model contains many more new variables and relatively fewer additional constraints than the basic model, the number of nodes explored is greatly reduced (from more than 2 million to 63), which leads to very low CPU time (from more than 3600s to 2s). This is due to the removal of many symmetric solutions from the search space by using the symmetry-breaking rule. It should be noted that both models have the same LP

Table 3: Problem 2 data

Scheduling horizon			10 days
Vessels	Arrival time	Composition	Amount of crude (Mbbl)
Vessel 1	0	100% A	1,000
Vessel 2	3	100% B	1,000
Vessel 3	6	100% C	1,000
Storage tanks	Capacity (Mbbl)	Initial composition	Initial amount of crude (Mbbl)
Tank 1	[0, 1,000]	100% A	200
Tank 2	[0, 1,000]	100% B	500
Tank 3	[0, 1,000]	100% C	700
Charging tanks	Capacity (Mbbl)	Initial composition	Initial amount of crude (Mbbl)
Tank 1 (mix X)	[0, 1,000]	100% D	300
Tank 2 (mix Y)	[0, 1,000]	100% E	500
Tank 3 (mix Z)	[0, 1,000]	100% F	300
Crudes	Property 1	Property 2	Gross margin ($/bbl)
Crude A	0.01	0.04	1
Crude B	0.03	0.02	3
Crude C	0.05	0.01	5
Crude D	0.0167	0.333	1.67
Crude E	0.03	0.23	3
Crude F	0.0433	0.133	4.33
Crude mixtures	Property 1	Property 2	Demand (Mbbl)
Crude mix X	[0.01, 0.02]	[0.03, 0.038]	[1,000, 1,000]
Crude mix Y	[0.025, 0.035]	[0.018, 0.027]	[1,000, 1,000]
Crude mix Z	[0.04, 0.048]	[0.01, 0.018]	[1,000, 1,000]
Unloading flowrate	[0, 500]	Transfer flowrate	[0, 500]
Distillation flowrate	[50, 500]	Number of distillations	5

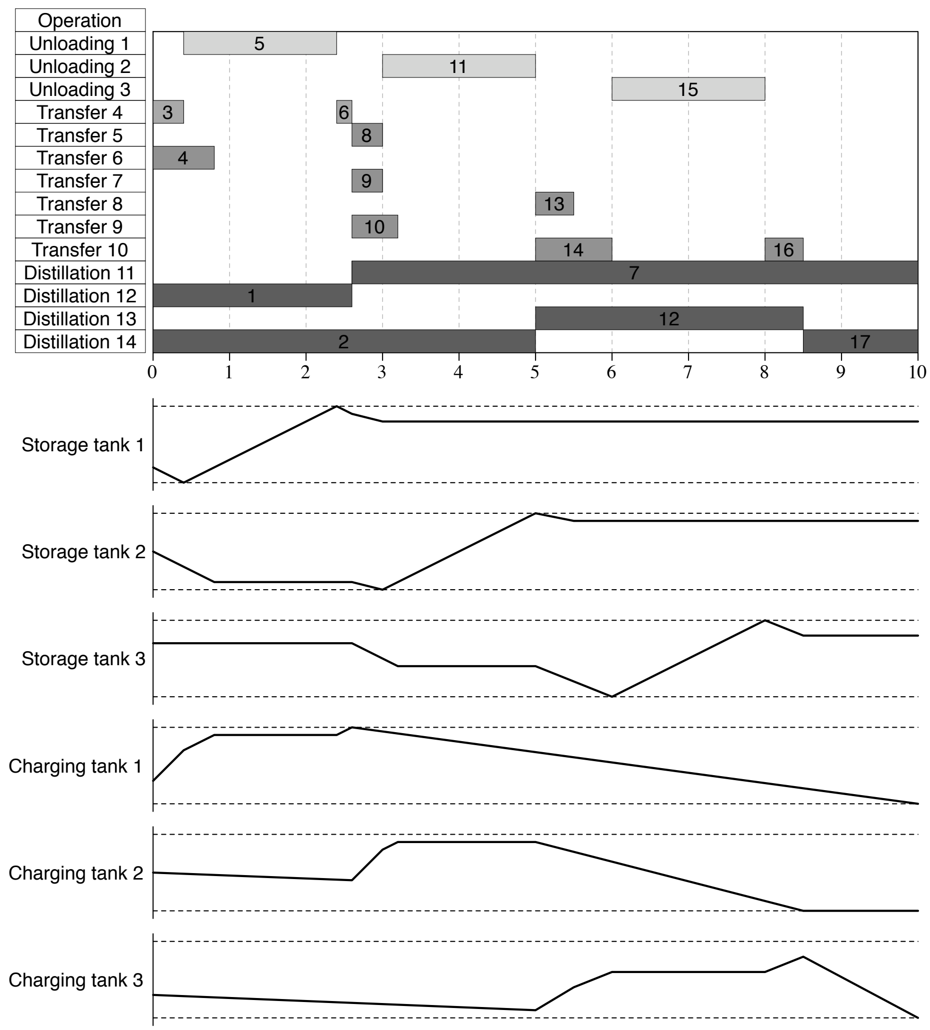

Figure 9: Optimal schedule for problem 2 (profit: $10,117,000)

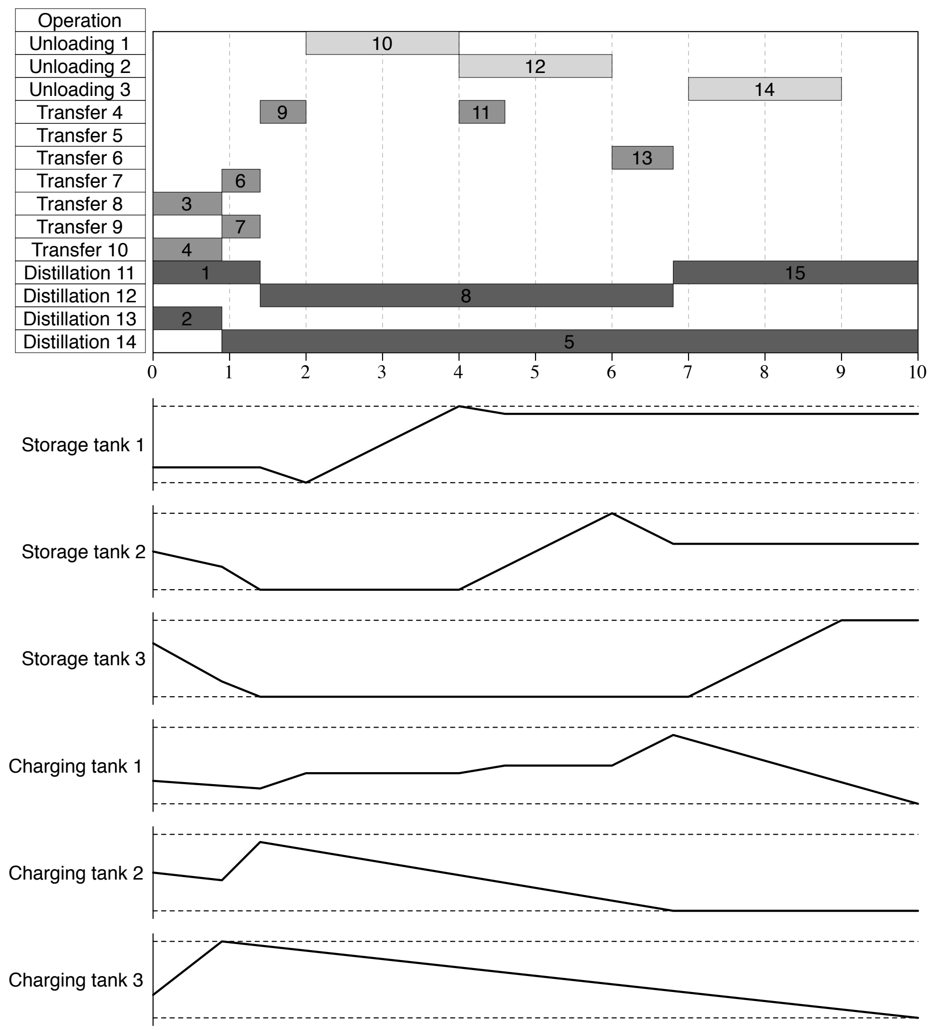

Figure 10: Optimal schedule for problem 2 with late vessel arrivals (profit: $9,775,000)

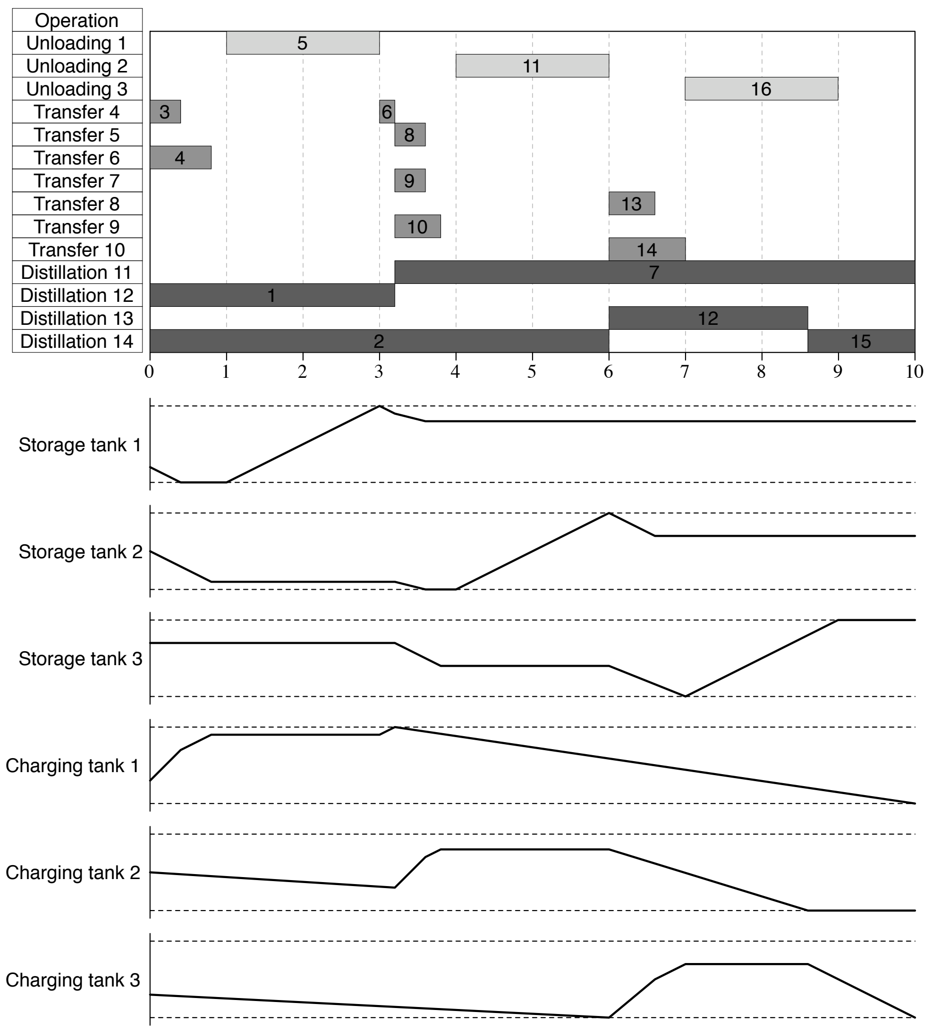

Figure 11: Optimal schedule for problem 2 with late vessel arrivals and fixed initial decisions (profit: $9,609,000)

Table 4: Problem 3 data

Scheduling horizon			12 days
Vessels	Arrival time	Composition	Amount of crude (Mbbl)
Vessel 1	0	100% A	500
Vessel 2	4	100% B	500
Vessel 3	8	100% C	500
Storage tanks	Capacity (Mbbl)	Initial composition	Initial amount of crude (Mbbl)
Tank 1	[0, 1,000]	100% D	200
Tank 2	[0, 1,000]	100% E	200
Tank 3	[0, 1,000]	100% F	200
Charging tanks	Capacity (Mbbl)	Initial composition	Initial amount of crude (Mbbl)
Tank 1 (mix X)	[0, 1,000]	100% G	300
Tank 2 (mix Y)	[0, 1,000]	100% E	500
Tank 3 (mix Z)	[0, 1,000]	100% F	300
Crudes	Property 1		Gross margin ($/bbl)
Crude A	0.01		1
Crude B	0.085		6
Crude C	0.06		8.5
Crude D	0.02		2
Crude E	0.05		5
Crude F	0.08		8
Crude G	0.03		3
Crude mixtures	Property 1		Demand (Mbbl)
Crude mix X	[0.025, 0.035]		[500, 500]
Crude mix Y	[0.045, 0.065]		[500, 500]
Crude mix Z	[0.075, 0.085]		[500, 500]
Unloading flowrate	[0, 500]	Transfer flowrate	[0, 500]
Distillation flowrate	[50, 500]	Number of distillations	5

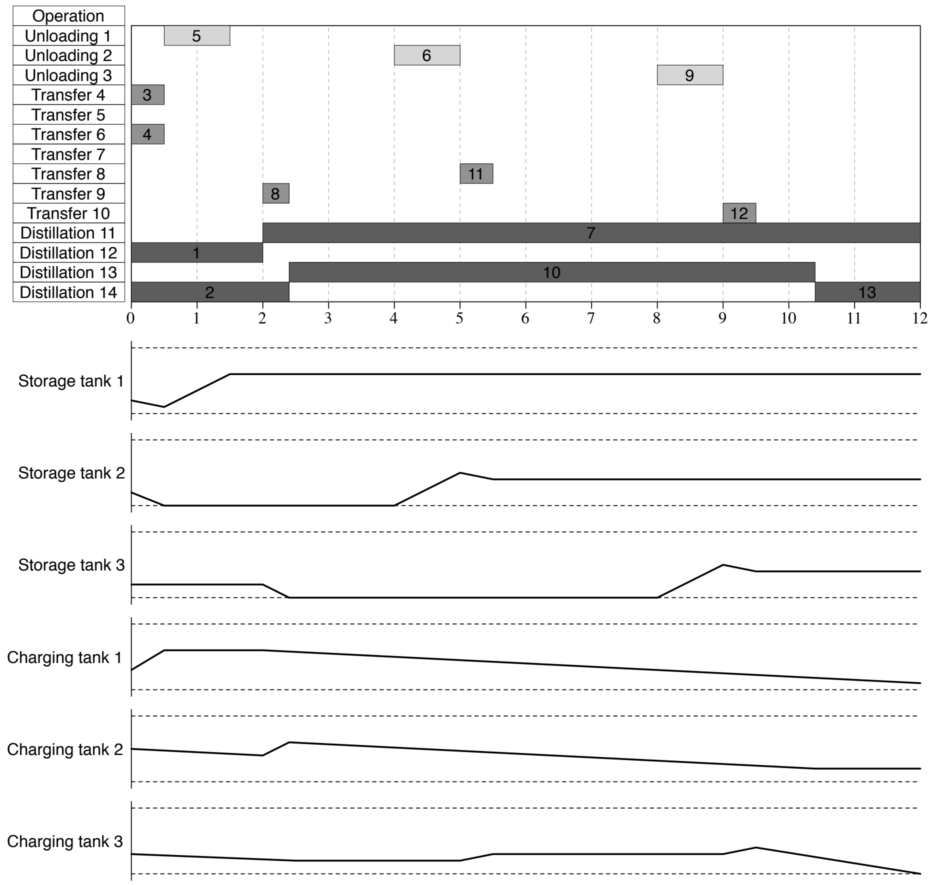

Figure 12: Optimal schedule for problem 3 (profit: $8,540,000)

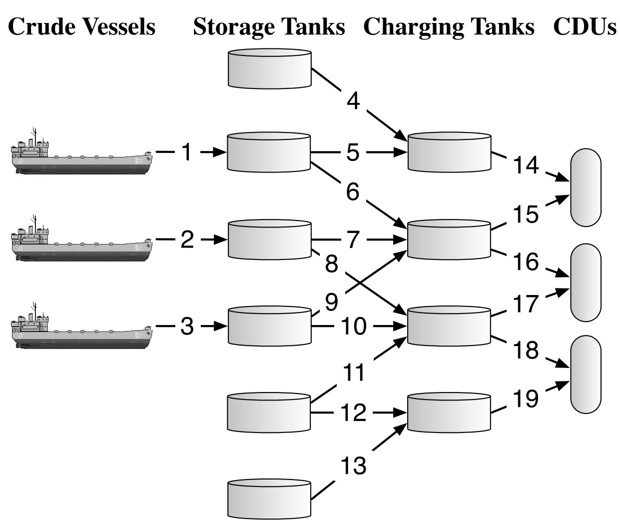

Figure 13: Crude-oil operations system for problem 4

Table 5: Problem 4 data

Scheduling horizon			15 days
Vessels	Arrival time	Composition	Amount of crude (Mbbl)
Vessel 1	0	100% A	600
Vessel 2	5	100% B	600
Vessel 3	10	100% C	600
Storage tanks	Capacity (Mbbl)	Initial composition	Initial amount of crude (Mbbl)
Tank 1	[100, 900]	100% D	600
Tank 2	[100, 1,100]	100% A	100
Tank 3	[100, 1,100]	100% B	500
Tank 4	[100, 1,100]	100% C	400
Tank 5	[100, 900]	100% E	300
Tank 6	[100, 900]	100% E	600
Charging tanks	Capacity (Mbbl)	Initial composition	Initial amount of crude (Mbbl)
Tank 1 (mix X)	[0, 600]	100% F	50
Tank 2 (mix Y)	[0, 600]	100% G	300
Tank 3 (mix Z)	[0, 600]	100% H	300
Tank 4 (mix W)	[0, 600]	100% E	300
Crudes	Property 1		Gross margin ($/bbl)
Crude A	0.03		3
Crude B	0.05		5
Crude C	0.065		6.5
Crude D	0.031		3.1
Crude E	0.075		7.5
Crude F	0.0317		3.17
Crude G	0.0483		4.83
Crude H	0.0633		6.33
Crude mixtures	Property 1		Demand (Mbbl)
Crude mix X	[0.03, 0.035]		[600, 600]
Crude mix Y	[0.043, 0.05]		[600, 600]
Crude mix Z	[0.06, 0.065]		[600, 600]
Crude mix W	[0.071, 0.08]		[600, 600]
Unloading flowrate	[0, 500]	Transfer flowrate	[0, 500]
Distillation flowrate	[20, 500]	Number of distillations	7

relaxation. Also, the optimal solution is found early during the resolution of the basic model, but a large amount of time is used to prove its optimality.

Figure 14 shows how both models perform on problem 1 when varying the number of priority-slots from 6 to 13. Clearly, the computational expense needed to solve the basic model grows exponentially with the number of priority-slots. However, both the number of nodes and the CPU time remain stable when solving the extended model.

Table 6: Size and performance of the Basic and Extended models on problem 1 (12 priority-slots)

	Variables	Binary variables	Constraints	Non-zeroes	LP	Nodes	CPU
Basic	1,189	96	2,383	15,555	80	+1,990,700	+3,600s
Extended	2,629	96	2,646	18,923	80	63	2s

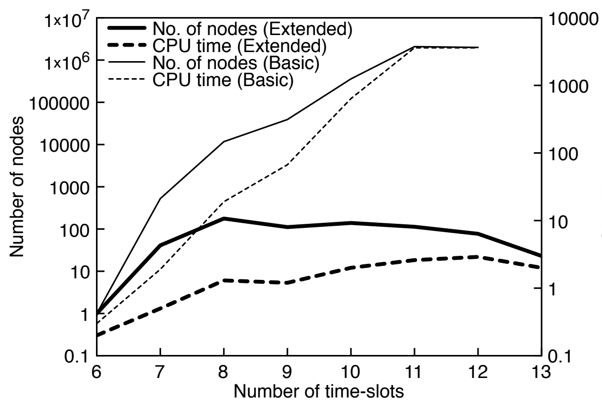

Figure 14: Performance of the Basic and Extended models on problem 1 (6 to 12 slots)

Performance of the MILP-NLP decomposition strategy

Table 7 shows the performance of different solution methods on problem 1. The Extended model is solved by the two-step MILP-NLP procedure and several MINLP solvers available in GAMS. MINLP solvers are given as a starting point the solution of the LP obtained by removing the nonlinear constraints of the model and relaxing the integrality constraints on binary variables. Other problems do not appear in Table 7 as they are intractable for these MINLP solvers. Results show that the two-step procedure is able to get the optimal solution and keep the computational expense low compared to other solvers.

Table 7: Performance of difference solution method on problem 1

Solution method	Problem	Slots	Solution	CPU time
MILP-NLP decomposition	1	13	79.75	1s
DICOPT	1	13	79.75	20s
SBB	1	13	79.35	56s
AlphaECP	1	13	79.28	242s
BARON	1	13	79.75	419s

Effect of the number of priority-slots

It is critical to postulate a relevant number of priority-slots to be used in the model. Indeed, a small number of priority-slots may lead to infeasibility, while a large number of priority-slots may result in an intractable model. The strategy used to determine the number of priority-slots is dependent on the problem.

In the case of the refinery problems introduced by Lee et al. [1], it can be shown that the number of relevant operations is bounded as long as the number of distillations is bounded. For example, if problem 1 is constrained to be solved with a maximum of 3 distillation operations, the maximum number of operations is 13 as in the sequence 7461483525746. Any sequence of 14 operations will be rejected by the sequencing rule in combination with the distillation cardinality constraints. Therefore, it is unnecessary to postulate a number of priority-slots greater than 13.

Figure 15 shows the evolution of the number of nodes explored, the CPU time, the MILP and NLP solutions objective value ($100,000 unit), and the optimality gap with respect to the number of priority-slots when solving problems 1 to 4 and using the extended formulation. The grey area represents the gap between MILP and NLP solutions. For all problems, the computational expense is small when the number of priority-slots is small, as the size of the problem and the feasible space are small. Also, the computational expense is small when the number of priority-slots is large (close to its maximum), as the solver must assign one operation to each priority-slot while satisfying both the cardinality constraints and the symmetry-breaking rule, thus making the feasible space small. In between, the number of nodes explored and the CPU time reach a maximum although not at the same number of priority-slots. It can also be observed that the objective value of the optimal solutions of the MILP relaxation increases with the number of priority-slots as more flexibility is given to the solver to find feasible solution. The optimality gap tends to decrease as well.

Table 8 gives some details on the solution of instances with the maximum number of priority-slots. The optimality gap is rather small (less than 4%) even though the solution strategy does not necessarily converge to a global optimum. This is explained by the fact that the composition constraints are always satisfied, even if dropped, under specific conditions. For each transfer v assigned to priority-slot i, the composition constraint (23) is satisfied if either of the following conditions hold true.

- the origin tank contains only one crude before the transfer
- the origin tank is fully discharged during the transfer

These conditions are always met in the optimal solutions of problems 1, 2, and 4. They do not always hold true in the optimal solution of problem 3, which explains the positive gap obtained.

Table 8: Performance of the model on problem 1 to 4 (maximum number of priority-slots)

Problem	No. of slots	LP	MILP			NLP		Gap
		Solution	Solution	Nodes	CPU	Solution	CPU	
1	13	80	79.75	17	2s	79.75	0s	0%
2	21	103	101.17	23	15s	101.17	1s	0%
3	21	100	87.4	45	32s	84.5	2s	3.3%
4	26	132.59	132.55	21	79s	132.55	3s	0%

Remark

When no symmetry-breaking sequencing rule is used, the global optimal solution of the SOS model with maximum number of priority-slots leads to the best possible schedule. Indeed, the optimal sequence of operations can be extended with unused operations, for which the volume transferred is set to 0. Thus, it can be obtained even if the user postulated more priority-slots than needed. However, in general, if a sequencing rule is used, this property does not necessarily hold true.

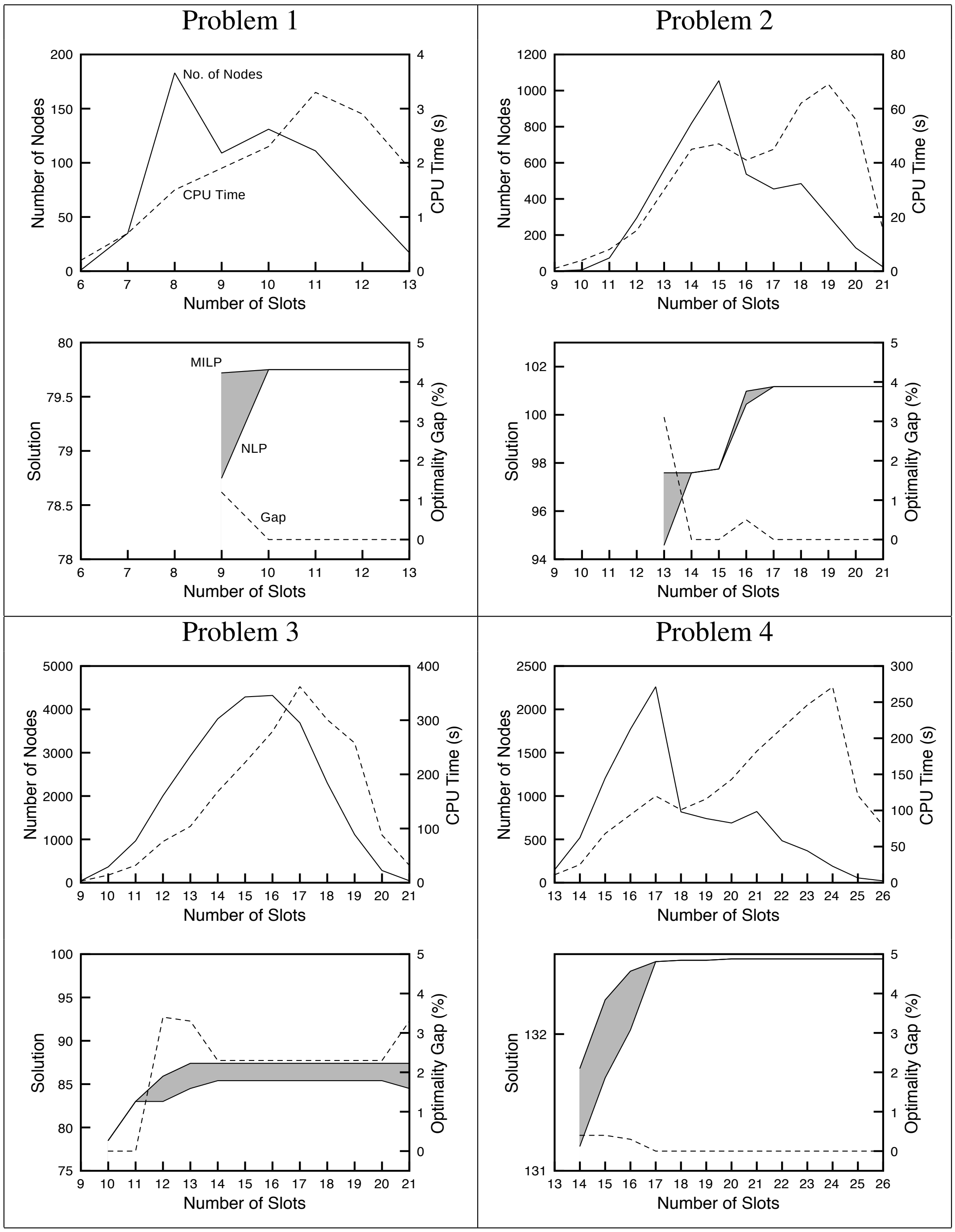

Figure 15: Performance of the model on problem 1 to 4 (varying number of priority-slots)

Conclusion

A new continuous-time scheduling formulation has been proposed in order to address crude-oil scheduling problems. It requires to postulate the number of tasks to be performed instead of the number of time intervals or event points as in previous formulations. Besides, it allows representing a solution schedule as a single sequence of operations that can be restricted with respect to a symmetry-breaking sequencing rule. A simple two step procedure consisting of solving an MILP and an NLP has been used to solve the MINLP model leading to solutions with optimality gaps lower than 5% with CPU times under 2 minutes.

The focus of future work is to use inference techniques such as contraint propagation in CP to handle the sequencing rule as it can be expensive in terms of variables and constraints, and thus in terms of node relaxation time. Also, techniques tightening the linear relaxation of such MINLP problems are under development.

References

1. Lee, H.; Pinto, J. M.; Grossmann, I. E.; Park, S. Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management. *Industrial and Engineering Chemistry Research* **1996**, *35*, 1630–1641.

2. Kondili, E.; Pantelides, C. C.; Sargent, R. W. H. A general algorithm for short-term scheduling of batch operations. I: MILP formulation. *Computers and Chemical Engineering* **1993**, *17*, 211–227.

3. Pantelides, C. C. Unified frameworks for optimal process planning and scheduling. *Proceedings of the Second International Conference on Foundations of Computer-Aided Process Operations*, 1994; pp 253–274.

4. Zhang, X.; Sargent, R. W. H. The optimal operation of mixed production facilities: A gen-

eral formulation and some approaches for the solution. *Computers and Chemical Engineering* **1996**, *20*, 897–904.

5. Schilling, G.; Pantelides, C. C. A simple continuous-time process scheduling formulation and a novel solution algorithm. *Computers and chemical engineering* **1996**, *20*, S1221–S1226.
6. Mockus, L.; Reklaitis, G. V. Mathematical programming formulation for scheduling of batch operations based on nonuniform time discretization. *Computers and Chemical Engineering* **1997**, *21*, 1147–1156.
7. Ierapetritou, M. G.; Floudas, C. A. Effective Continuous-Time Formulation for Short-Term Scheduling. 1. Multipurpose Batch Processes. *Industrial and Engineering Chemistry Research* **1998**, *37*, 4341–4359.
8. Floudas, C. A.; Lin, X. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Computers and Chemical Engineering* **2004**, *28*, 2109–2129.
9. Kallrath, J. Planning and scheduling in the process industry. *OR Spectrum* **2002**, *24*, 219–250.
10. Kelly, J. D.; Mann, J. L. Crude oil blend scheduling optimization: an application with multi-million dollar benefits. *Hydrocarbon Processing* **2003**, *82*, 47–53.
11. Shah, N. Mathematical programming techniques for crude oil scheduling. *Computers and Chemical Engineering* **1996**, *20*, 1227–1232.
12. Wenkay, L.; Hui, C. Scheduling Crude Oil Unloading, Storage, and Processing. *Industrial and Engineering Chemistry Research* **2002**, *41*, 6723–6734.
13. Pinto, J. M.; Joly, M.; Moro, L. F. L. Planning and scheduling models for refinery operations. *Computers and Chemical Engineering* **2000**, *24*, 2259–2276.
14. Moro, L. F. L.; Pinto, J. M. Mixed-Integer Programming Approach for Short-Term Crude Oil Scheduling. *Industrial and Engineering Chemistry Research* **2004**, *43*, 85–94.

15. Reddy, P. C. P.; Karimi, I. A.; Srinivasan, R. A new continuous-time formulation for scheduling crude oil operations. *Chemical Engineering Science* **2004**, *59*, 1325–1341.

16. Jia, Z.; Ierapetritou, M. G.; Kelly, J. D. Refinery Short-Term Scheduling Using Continuous Time Formulation: Crude-Oil Operations. *Industrial and Engineering Chemistry Research* **2003**, *42*, 3085–3097.

17. Karuppiah, R.; Furman, K. C.; Grossmann, I. E. Global Optimization for Scheduling Refinery Crude Oil Operations. *Computers and Chemical Engineering* **2008**, *32*, 2745–2766.

18. Quesada, I.; Grossmann, I. E. Global Optimization of Bilinear Process Networks with Multicomponent Flows. *Computers and Chemical Engineering* **1995**, *19*, 1219–1242.

19. Margot, F. *Symmetry in Integer Linear Programming*, Tepper Working Paper 2008 E-37, 2008.

20. Hopcroft, J. E.; Ullman, J. D. *Introduction to Automata Theory, Languages and Computation*; Addison Wesley, 1979.

21. Côté, M. C.; Gendron, B.; Rousseau, L. M. Modeling the Regular Constraint with Integer Programming. *CPAIOR 2007*, 2007; pp 29–43.

22. Pesant, G. A Regular Language Membership Constraint for Finite Sequences of Variables. *Principles and Practice of Constraint Programming - CP 2004*, 2004; pp 482–495.