

Integration of Refinery Planning and Crude-Oil Scheduling using Lagrangian Decomposition

Sylvain Mouret^a, Ignacio E. Grossmann^{a,*}, Pierre Pestiaux^b

^aDepartment of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213, USA

^bTotal Refining & Marketing, Research Division, 76700 Harfleur, France

Abstract

The aim of this paper is to introduce a methodology to solve a large-scale mixed integer nonlinear program (MINLP) integrating the two main optimization problems appearing in the oil refining industry: refinery planning and crude-oil operations scheduling. The proposed approach consists of using Lagrangian decomposition to efficiently integrate both problems. The main advantage of this technique is to solve each problem separately. A new hybrid dual problem is introduced to update the Lagrange multipliers. It uses the classical concepts of cutting planes, subgradients, and trust-regions. The proposed approach is compared to a basic sequential approach and to standard MINLP solvers. The results obtained on a case-study and a larger refinery problem show that the Lagrangian decomposition algorithm is more robust than the other approaches and produces better solutions in reasonable times.

Keywords: refinery planning, crude-oil scheduling, mixed-integer nonlinear programming, Lagrangian decomposition

1. Introduction

The oil refining industry is a prolific field for the application of mathematical programming techniques (see Bodington and Baker, 1990). Refinery operators have to make decisions on the logistics operation taking into account a large number of crude-oils, finished products such as liquified petroleum gas, gasoline, diesel fuel, and a wide variety of high flexibility production units involving many different chemical processes. Furthermore, the economic impact of optimizing operations can be very significant (see Kelly and Mann, 2003).

The refinery planning problem often involves the pooling problem (see Foulds et al., 1992; Floudas and Visweswaran, 1993; Quesada and Grossmann, 1995; Audet et al., 2004; Misener and Floudas, 2009), which has been addressed since the early 80s and usually consists of optimizing feedstocks, unit settings, as well as final product blending and shipping. Some examples of nonlinear refinery planning problems including pooling constraints and nonlinear process models can be found in Pinto and Moro (2000), Li et al. (2005), and Alhajri et al. (2008). Although commercial solvers such as GRTMPS (Haverly Systems), PIMS (Aspen Tech), and RPMS (Honeywell Hi-Spec Solutions) implement successive linear programming algorithms to solve this problem (see Zhang et al., 1985), any standard NLP solvers can also be used although they may not guarantee global optimality of the solution.

A major issue with refinery planning is that most models are single-period models where the refinery is assumed to operate in the same state over the whole planning period (typically 1 month). Therefore, the planning solution is used as a tactical goal for refinery operators rather than as an operational tool. In particular, CDU (Crude Distillation Unit) feedstock decisions returned by the refinery planning problem are usually not applicable in the field due to crude logistics constraints. These are described in the crude-oil operations scheduling problem, which includes unloading from crude-oil tankers, preparation of

*Corresponding author

crude-blends, and CDU feed charging. This problem has been addressed since the late 90s by many different research groups (see Lee et al., 1996; Jia et al., 2003; Moro and Pinto, 2004; Mouret et al., 2009).

Although, integration of planning and scheduling has recently been addressed in the context of multiproduct continuous and batch production plants (see Erdirik-Dogan and Grossmann, 2008; Maravelias and Sung, 2009), very little work has been done towards the integration of planning and crude-oil scheduling problems in the context of refineries. This is due to the fact that in this case, the planning model is not an aggregate scheduling model. Therefore, the decomposition methods developed for batch and continuous plants are not directly applicable to refineries. In particular, planning and scheduling correspond to two different problems solely linked through CDU feedstocks. Therefore, instead of using a hierarchical decomposition, a spatial Lagrangian decomposition is preferred. The reader may refer to Fisher (1985) and Guignard (2003) for extensive reviews on Lagrangian relaxation and decomposition techniques. These approaches have been applied to many industrial problems such as production planning and scheduling integration (see Li and Ierapetritou, 2009) or multiperiod refinery planning (see Neiro and Pinto, 2006). Thus, it seems natural to apply Lagrangian decomposition to solve the integrated refinery planning and crude-oil scheduling problem.

The content of this paper is organized as follows. The planning and scheduling problems are stated in Section 2 as well as the full-space integrated problem. In Section 3, a Lagrangian decomposition scheme based on the dualization of CDU feedstock linking constraints is presented. A new hybrid method is introduced to solve the Lagrangian relaxation in Section 4. A heuristic algorithm is developed to obtain good feasible solutions for the integrated full-space problem in Section 5. A numerical application of the proposed approach is presented in Section 7 and Section 9 concludes the paper.

2. Problem Statement

2.1. Refinery Planning Problem

The refinery planning problem can be regarded as a flowsheet optimization problem with multiple periods during which the refinery system is assumed to operate in steady-state. Due to extensive stream mixing, the model for each period is based on a pooling problem that is extended in order to include process models for each refining unit. The different periods in the model are connected through many material inventories. In this work, we consider a single-period planning model based on a pooling problem inspired from the literature (see for instance Adhya et al. (1999)). A basic refinery planning system is represented in Figure 1. A set of crudes $i \in I$ are to be mixed in different types of crude-oil blends $j \in J$ (e.g. low-sulfur and high-sulfur blends), each associated to a specific CDU operating mode. For each mode and each crude, several distillation cuts are obtained with different yields. These crude cuts are then blended into intermediate pools which are used to prepare several final products. Therefore, the refinery planning system is composed of the following elements:

- One input stream for each selected crude $i \in I$ and each type of crude blend $j \in J$
- One CDU with fixed yields for each distillation cut
- Set of distillation cuts $k \in K$
- One pool for each type of crude blend $j \in J$ and each cut $k \in K$
- One intermediate stream between each pool $(j, k) \in J \times K$ and each final product $l \in L$
- Set of final products $l \in L$
- One sales stream for each final product $l \in L$

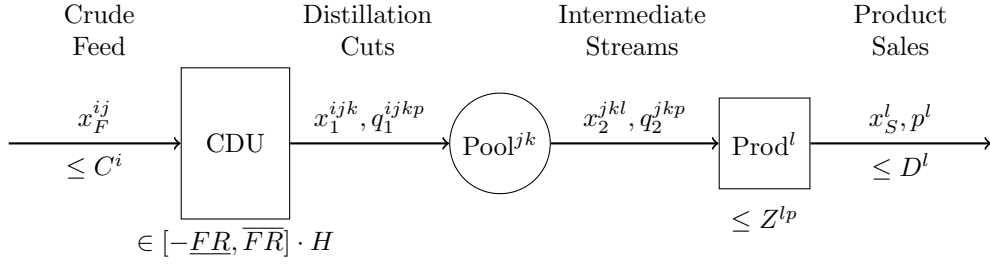


Figure 1: Basic refinery planning system.

- Set of qualities $p \in P$

The yield of crude $i \in I$ in distillation cut $k \in K$ when processed in crude blend $j \in J$ is assumed to be fixed and is denoted by α^{ijk} . In terms of stream qualities, it is assumed that distillation cuts have fixed qualities while pool qualities are calculated by bilinear quality balance constraints. A pure flow-based model is used to formulate the pooling problem as shown below. CDU flowrate limitations are considered independent of the operating mode and are enforced globally for all crudes processed during the period.

$$\begin{aligned}
\max \quad & \sum_{l \in L} p^l x_S^l && \text{(sales revenue maximization)} \\
\text{s.t.} \quad & 0 \leq \sum_{j \in J} x_F^{ij} \leq C^i && i \in I \quad \text{(crude availability)} \\
& \underline{FR} \cdot H \leq \sum_{i \in I} \sum_{j \in J} x_F^{ij} \leq \overline{FR} \cdot H && \text{(CDU flowrate limitations)} \\
& x_1^{ijk} = \alpha^{ijk} \cdot x_F^{ij} && (i, j, k) \in I \times J \times K \quad \text{(CDU yield calculation)} \\
& \sum_{i \in I} x_1^{ijk} = \sum_{l \in L} x_2^{jkl} && (j, k) \in J \times K \quad \text{(pool mass balance)} \\
& \sum_{i \in I} q_1^{jkp} x_1^{ijk} = q_2^{jkp} \sum_{l \in L} x_2^{jkl} && (j, k, p) \in J \times K \times P \quad \text{(pool quality balance, nonlinear)} \\
& \sum_{j \in J} \sum_{k \in K} x_2^{jkl} = x_S^l && l \in L \quad \text{(product mass balance)} \\
& x_S^l \leq D^l && l \in L \quad \text{(maximum product demand)} \\
& \sum_{j \in J} \sum_{k \in K} q_2^{jkp} x_2^{jkl} \leq Z^{lp} x_S^l && (l, p) \in L \times P \quad \text{(product quality requirement, nonlinear)} \\
& x_F^{ij}, x_1^{ijk}, x_2^{jkl}, x_S^l \geq 0, q_2^{jkp} \in \mathbb{R}
\end{aligned}$$

The nomenclature used is as follows:

- Variables:
 - x_F^{ij} a variable representing the amount of crude i selected for CDU distillation in blend j
 - x_1^{ijk} a variable representing the amount of cut k extracted from crude i in blend j
 - x_2^{jkl} a variable representing the flow of material between pool (j, k) and product l
 - q_2^{jkp} a variable representing the quality p of pool (j, k)
 - x_S^l is a variable representing the amount of final product l sold

- Parameters:

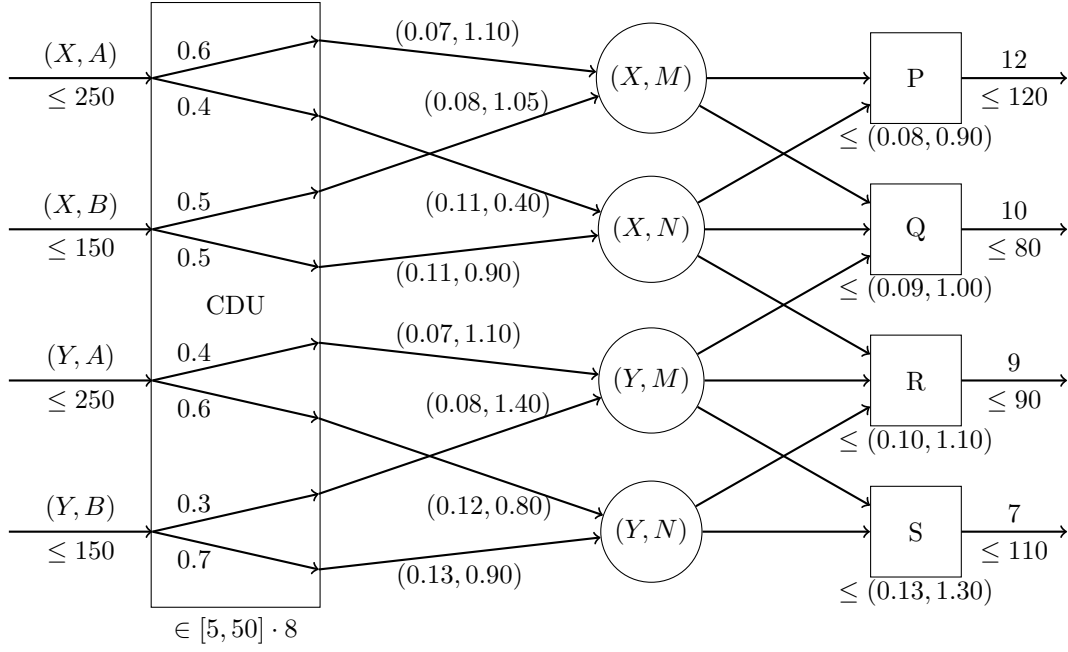


Figure 2: Refinery planning case-study.

- q_1^{ijkp} is the (fixed) quality p of cut k extracted from crude i in blend j
- p is the market value of final products
- C^i is the amount of available crude i
- α^{ijk} is the yield of cut k extracted from crude i in blend j
- H is the planning horizon
- $[FR, \overline{FR}]$ is the bounds on CDU flowrate
- D^l is the maximum demand in product l
- Z^{lp} is the maximum specification for quality p of product l

Figure 2 displays the pooling structure of a case-study with corresponding data for crudes (A, B), blends (X, Y), distillation cuts (M, N), and final products (P, Q, R, S). Two different qualities are considered in this case.

In the remainder of the paper, we consider the following NLP, which is a simplified version of the planning model (P_P).

$$(P_P) \begin{cases} \max & V_P^T x_S \\ \text{s.t.} & f_P(x_F, x_I, x_S) \leq 0 \\ & g_P(x_I) \leq 0 \\ & x_F \in \mathbb{R}^{|F|}, x_I \in \mathbb{R}^{|I|}, x_S \in \mathbb{R}^{|S|} \end{cases}$$

The nomenclature used is as follows:

- V_P is the market value of final products
- x_F is a set of continuous variables representing CDU feedstock quantities over the single planning period
- x_S is a set of continuous variables representing final products sales
- x_I is a set of intermediate continuous variables (e.g. pool quantity and quality variables)

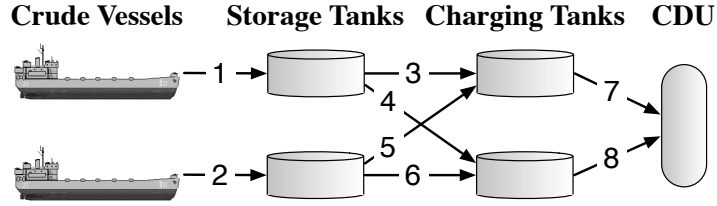


Figure 3: Crude-oil scheduling system 1 (Lee et al., 1996).

- $f_P(x_F, x_I, x_S) \leq 0$ is the set of *linear* constraints (e.g. material balance constraints)
- $g_P(x_I) \leq 0$ is the set of *nonlinear* constraints (e.g. quality balance constraints)

2.2. Crude-Oil Scheduling Problem

The crude-oil scheduling problem deals with the unloading, transfer and blending operations executed on crude-oil tankers and crude-oil inventories. The goal is to sequentially prepare multiple crude blends, which are defined by specific property requirements. Each type of crude blend corresponds to a specific CDU operating mode. Different objectives have been studied, namely minimization of logistics costs (see Lee et al., 1996) or maximization of profit (see Mouret et al., 2009, 2010). In this work, the objective is to minimize the total replacement cost of the crudes that are selected for distillation. The replacement cost is the cost of replacing the crude once it has been processed. The crude-oil schedule must satisfy inventory capacity limitations, crude tankers arrival dates as well as the following logistics constraints:

- Only one berth is available at the docking station for crude tanker unloadings,
- inlet and outlet transfers on tanks must not overlap,
- a tank may charge only one CDU at a time,
- a CDU can be charged by only one tank at a time,
- CDUs must be operated continuously throughout the scheduling horizon.

Figure 3 shows the refinery system corresponding to problem 1 introduced in Lee et al. (1996). Table 1 displays the dimensionless data for this example. Besides a different objective function, the example is modified by introducing a minimum duration of one day for distillation operations. Therefore, due to crude blend alternative sequencing, at most 4 batches of each crude mix can be processed in 8 days. The scheduling problem is formulated using the Multi-Operation Sequencing (MOS) time representation as introduced in Mouret et al. (2010) (see Appendix B).

In the remainder of the paper, we consider the following MINLP, which is a simplified version of the scheduling model P_S .

$$(P_S) \begin{cases} \max & -V_C^T y_F \\ \text{s.t.} & f_S(y_B, y_C, y_F) \leq 0 \\ & g_S(y_C) \leq 0 \\ & y_B \in \{0, 1\}^{|B|}, y_C \in \mathbb{R}^{|C|}, y_F \in \mathbb{R}^{|F|} \end{cases}$$

The nomenclature used is as follows:

- V_C is the replacement cost of crude-oils (usually based on market value)
- y_F is a set of continuous variables representing total CDU feedstock quantities over the scheduling horizon
- y_C is a set of continuous variables representing other continuous decisions (e.g. timing decisions)

Table 1: Crude-oil scheduling data for case-Study

Scheduling horizon			8 days
Vessels	Arrival time	Composition	Amount of crude
Vessel 1	0	100% A	100
Vessel 2	4	100% B	100
Storage tanks	Capacity	Initial composition	Initial amount of crude
Tank 1	[0, 100]	100% A	25
Tank 2	[0, 100]	100% B	75
Charging tanks	Capacity	Initial composition	Initial amount of crude
Tank 1 (mix X)	[0, 100]	80% A, 20% B	50
Tank 2 (mix Y)	[0, 100]	20% A, 80% B	50
Crudes	Property 1 (sulfur concentration)		Crude unit cost
Crude A	0.01		7
Crude B	0.06		6
Crude mixtures	Property 1 (sulfur concentration)		Maximum number of batches
Crude blend X	[0.015, 0.025]		4
Crude blend Y	[0.045, 0.055]		4
Unloading flowrate	[0, 50]	Transfer flowrate	[0, 50]
Distillation flowrate	[5, 50]	Minimum duration of distillations	1 day

- y_B is a set of binary variables representing sequencing decisions (see Mouret et al., 2010)
- $f_S(y_B, y_C, y_F) \leq 0$ is the set of *linear* constraints (e.g. scheduling constraints)
- $g_S(y_C) \leq 0$ is the set of *nonlinear* stream composition constraints (see Mouret et al., 2009)

2.3. Full-Space Problem

Given the importance of crude selection for refinery optimization, the refinery planning problem and the crude-oil scheduling problem should ideally be optimized simultaneously. This can only be done by solving an integrated *full-space* MINLP problem, denoted (P) , which aims at optimizing all refinery decisions subject to planning, scheduling, and linking constraints.

$$(P) \left\{ \begin{array}{l} \max \quad V_P^T x_S - V_C^T y_F \\ \text{s.t.} \quad f_P(x_F, x_I, x_S) \leq 0 \\ \quad \quad g_P(x_I) \leq 0 \\ \quad \quad f_S(y_B, y_C, y_F) \leq 0 \\ \quad \quad g_S(y_C) \leq 0 \\ \quad \quad y_F - x_F = 0 \\ \quad \quad x_F \in \mathbb{R}^{|F|}, x_I \in \mathbb{R}^{|I|}, x_S \in \mathbb{R}^{|S|} \\ \quad \quad y_B \in \{0, 1\}^{|B|}, y_C \in \mathbb{R}^{|C|}, y_F \in \mathbb{R}^{|F|} \end{array} \right.$$

The integrated objective is to maximize profit defined by final product sales revenues minus crude-oil replacement costs. The linking constraint $y_F - x_F = 0$ ensures consistency of planning and scheduling decisions in terms of CDU feedstock quantities. More precisely, it ensures that the amounts of crudes selected for distillation are identical in the planning and scheduling solutions. Also, to be consistent in time, it is considered that the planning and scheduling horizons have identical lengths.

3. Lagrangian Decomposition Scheme

The full-space problem (P) is a large-scale MINLP, which contains many binary variables from the crude-oil scheduling problem and many non-convex constraints from the refinery planning model. Due to convergence issues and the presence of many potential local optima, standard MINLP solvers for convex optimization, such as AlphaECP, Bonmin, DICOPT,

KNITRO, or SBB, may fail solving the model or return poor solutions. Global MINLP solvers, such as BARON, Couenne, or LINDOGLOBAL, are in principle able to solve the problem but they may require prohibitive computational times. Therefore, a specific solution strategy needs to be developed to address this problem.

Robertson et al. (2010) proposed a multi-level approach consisting of approximating the refinery planning model by multiple linear regressions that are then used in the crude-oil scheduling problem for the minimization of the total logistics and production costs. The method is applied to a case-study comprising two different crudes. Although computationally effective, the use of linear regressions may not be sufficient for the the global optimization of highly nonlinear refinery planning models.

In this work, we present an integration approach based on *Lagrangian decomposition*, which is a special case of *Lagrangian relaxation* (Guignard, 2003). The idea is to build a relaxed version of the full-space problem, which is decomposable, and therefore much easier to solve. In particular, the decomposition procedure is based on the dualization of the linking constraint $y_F - x_F = 0$. The relaxed problem $(P_R(\lambda))$, composed of NLP and MINLP models, is defined by removing this constraint and penalizing its violations by adding the term $\lambda^T(y_F - x_F)$ to the objective function. The parameter λ is a *Lagrange multiplier* whose value is fixed prior to solving the model and adjusted iteratively.

$$(P_R(\lambda)) \left\{ \begin{array}{l} \max \quad V_P^T x_S - \lambda^T x_F + (\lambda - V_C)^T y_F \\ \text{s.t.} \quad f_P(x_F, x_I, x_S) \leq 0 \\ \quad \quad g_P(x_I) \leq 0 \\ \quad \quad f_S(y_B, y_C, y_F) \leq 0 \\ \quad \quad g_S(y_C) \leq 0 \\ \quad \quad x_F \in \mathbb{R}^{|F|}, x_I \in \mathbb{R}^{|I|}, x_S \in \mathbb{R}^{|S|} \\ \quad \quad y_B \in \{0, 1\}^{|B|}, y_C \in \mathbb{R}^{|C|}, y_F \in \mathbb{R}^{|F|} \end{array} \right.$$

As already mentioned, problem $(P_R(\lambda))$ is easier to solve as it can be decomposed into two subproblems $(P_P(\lambda))$ and $(P_S(\lambda))$.

$$v(P_R(\lambda)) = v(P_P(\lambda)) + v(P_S(\lambda))$$

The subproblem $(P_P(\lambda))$, an NLP, is a modification of the original refinery planning problem (P_P) as it consists of assigning crude costs λ to the CDU feedstock variables x_F . For a given crude i , increasing λ^i will decrease the incentive to select this crude for distillation processing. On the other hand, decreasing λ^i will increase the incentive to select it.

$$(P_P(\lambda)) \left\{ \begin{array}{l} \max \quad V_P^T x_S - \lambda^T x_F \\ \text{s.t.} \quad f_P(x_F, x_I, x_S) \leq 0 \\ \quad \quad g_P(x_I) \leq 0 \\ \quad \quad x_F \in \mathbb{R}^{|F|}, x_I \in \mathbb{R}^{|I|}, x_S \in \mathbb{R}^{|S|} \end{array} \right.$$

The subproblem $(P_S(\lambda))$, an MINLP, is a modification of the original crude-oil scheduling problem (P_S) as it consists of assigning crude values λ to the CDU feedstock variables y_F . For a given crude i , increasing λ^i will increase the incentive to select this crude for blending and distillation processing. On the other hand, decreasing λ^i will decrease the incentive to select it.

$$(P_S(\lambda)) \left\{ \begin{array}{l} \max \quad (\lambda - V_C)^T y_F \\ \text{s.t.} \quad f_S(y_B, y_C, y_F) \leq 0 \\ \quad \quad g_S(y_C) \leq 0 \\ \quad \quad y_B \in \{0, 1\}^{|B|}, y_C \in \mathbb{R}^{|C|}, y_F \in \mathbb{R}^{|F|} \end{array} \right.$$

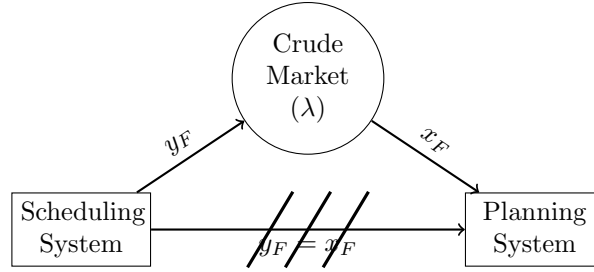


Figure 4: Economic interpretation of the Lagrangian decomposition.

On the whole, the Lagrange multiplier λ can be seen as a crude purchase cost for the planning system, and as a crude sales value for the scheduling system. Therefore, the spatial Lagrange decomposition procedure applied to this problem can be seen as introducing a crude market between the planning and scheduling systems (see Fig. 4). The planning system acts as a consumer who buys crude from the market, while the scheduling system acts as a producer who sells it to the market. It is clear that for fixed prices both actors are independent, which explains why the two corresponding subproblems can be solved in parallel.

Although computationally convenient, this decomposition procedure does not solve the original full-space problem. In particular, it is well-known that $(P_R(\lambda))$ is a relaxation of the full-space problem, thus $v(P_R(\lambda)) > v(P)$. However, one can search for a Lagrange multiplier λ that minimizes $v(P_R(\lambda))$ in order to get as close as possible to $v(P)$. This problem is called the *dual problem* and the function $\lambda \mapsto v(P_R(\lambda))$ is often called the *Lagrangian function*.

$$(P_D) \quad \min_{\lambda} v(P_R(\lambda))$$

Duality theory establishes that $v(P_D) - v(P) \geq 0$ (see Guignard, 2003). In some cases (e.g. nonconvex models), we may have $v(P_D) - v(P) > 0$ and this difference is called *dual gap*. Our hope is that this dual gap is small enough so that valuable information can be inferred to generate near-optimal heuristic solutions (see Section 5).

4. Solution of the Dual Problem

Several approaches have been proposed in the literature in order to solve the dual problem associated with the Lagrangian relaxation. A classical approach is the *subgradient method* proposed by Held and Karp (1971) and Held and Karp (1974). Many researchers have used and improved this technique over the years (see Camerini et al., 1975; Bazararaa and Sherali, 1981; Fisher, 1981). This approach is preferred as it usually predicts very good Lagrange multiplier updates. However, special care must be taken in order to insure convergence and it requires a good strategy for defining and updating the subgradient step size. Another approach that theoretically displays better convergence properties has been introduced by Cheney and Goldstein (1959) and Kelley (1960). It is often denoted as the *cutting plane method*. In practice, this approach usually takes a long time to converge as many iterations are necessary in order to obtain good Lagrange multiplier updates. A refinement of this approach is the *boxstep method* (also called *trust-region method*) presented in Marsten et al. (1975). It allows obtaining better updates for the Lagrange multiplier during early iterations while keeping the same convergence properties. Other refinements of previous approaches include the *bundle method* (Lemaréchal, 1974), the *volume algorithm* (Barahona and Anbil, 2000), and the *analytic center cutting plane method* (Goffin et al., 1992).

All the above approaches are based on an iterative solution procedure between the primal and the dual world. Figure 5 gives a schematic description of this algorithm. The first step consists of initializing the Lagrange multipliers. Problem-specific strategies, often based on the economic interpretation of λ , exist in order to provide good initial values. Then, at

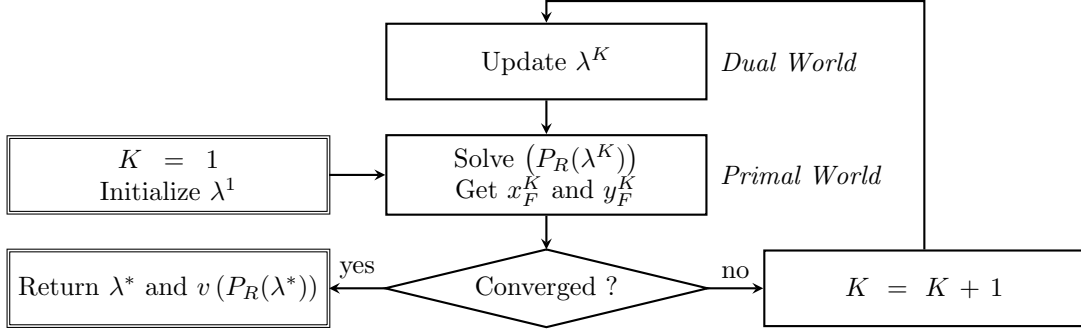


Figure 5: General iterative primal-dual algorithm.

each iteration the relaxed problem is solved and a primal solution is obtained. If a stopping criterion is satisfied, the algorithm converges. Otherwise, the Lagrange multipliers are updated for the next iteration. The definition of the stopping criterion depends on the approach used and can, in certain cases, ensure convergence to an optimal dual solution λ^* .

In this work, we introduce a new hybrid method to update the Lagrange multipliers. It is based on the three concepts of cutting planes, subgradient and trust-region. Cutting planes are valid constraints for the dual problem that are generated at each iteration. They are used to record valuable dual information to be used during later iterations. A subgradient defines a descent direction for the dual problem, while a trust-region allows to deviate from this direction within a specified domain. The combination of these techniques ensures good convergence properties while providing efficient Lagrange multiplier updates. At iteration $K + 1$, the Lagrange multiplier is updated to the solution of the following restricted LP dual problem with subgradient-based trust-region.

$$(\hat{P}_D^{K+1}) \quad \left\{ \begin{array}{l} \min \quad \eta \\ \text{s.t.} \quad \eta \geq V_P^T x_S^k - V_C^T y_F^k + \lambda^T (y_F^k - x_F^k) \quad \forall k = 1 \dots K \quad (\text{CP}^k) \\ \lambda = \lambda^K + \alpha \frac{v(P_D) - v(P_R(\lambda^K))}{\|y_F^K - x_F^K\|^2} (y_F^K - x_F^K) + \delta \quad (\text{SG+TR}) \\ \eta \in \mathbb{R}, \lambda \in \mathbb{R}^{|F|} \\ \alpha \in]-\infty, \bar{\alpha}], \delta \in [-\bar{\delta}, \bar{\delta}]^{|F|} \end{array} \right.$$

The variables λ and η are classically used in the pure cutting plane approach. The pure cutting plane restricted LP dual problem (P_D^{K+1}) consists of minimizing η subject to constraints (CP^k) , $k = 1 \dots K$ only:

$$(P_D^{K+1}) \quad \left\{ \begin{array}{l} \min \quad \eta \\ \text{s.t.} \quad \eta \geq V_P^T x_S^k - V_C^T y_F^k + \lambda^T (y_F^k - x_F^k) \quad \forall k = 1 \dots K \quad (\text{CP}^k) \\ \eta \in \mathbb{R}, \lambda \in \mathbb{R}^{|F|} \end{array} \right.$$

This problem is always unbounded during early iterations (i.e. $v(P_D^K) = -\infty$) so it cannot be used directly to update the Lagrange multipliers. This issue can be solved by defining a bounded feasible set for the multipliers based on their interpretation (e.g. lower and upper bounds). However, in this work, an iteratively updated trust-region based on the subgradient step is used so that the restricted dual problem (\hat{P}_D^{K+1}) is bounded.

The subgradient step is classically defined as $\lambda = \lambda^K + \alpha (v(P_D) - v(P_R(\lambda^K))) / \|y_F^K - x_F^K\|^2 (y_F^K - x_F^K)$ where $v(P_D)$ can be estimated using a heuristic solution for (P) . However, instead of heuristically updating the step size, it is optimized using variable α , which is bounded by the parameter $\bar{\alpha} > 0$. Note that α is allowed to take negative values. Variable δ defines a deviation from the subgradient step. The parameter $\bar{\delta} > 0$ is the maximum deviation in each multiplier direction and defines a full-dimensional trust-region. Both subgradient and trust-region concepts are simultaneously embedded in

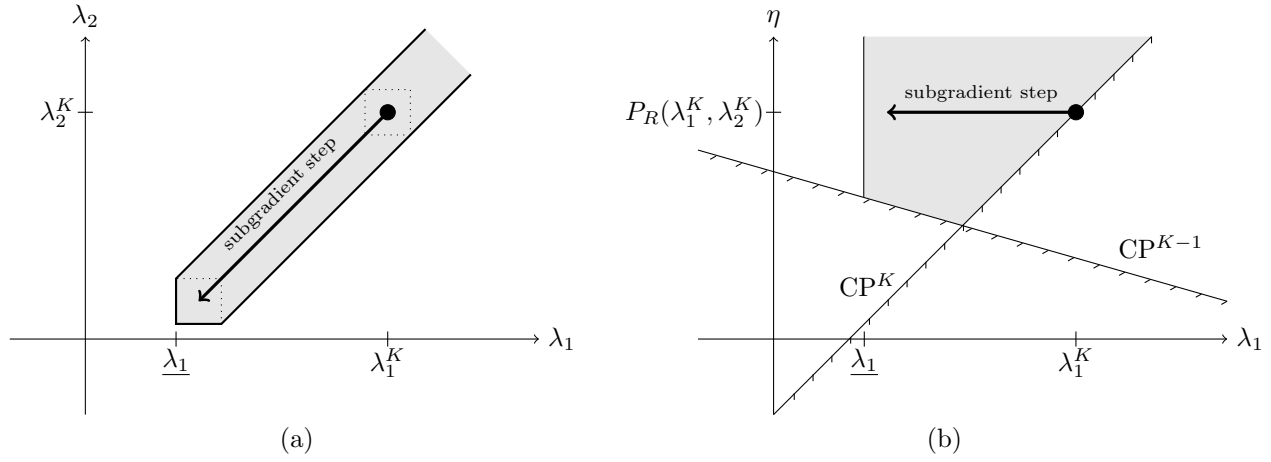


Figure 6: Plots of the feasible space of (\hat{P}_D^{K+1}) .

constraint (SG+TR). Note that the parameters $\bar{\alpha}$ and $\bar{\delta}$ can be heuristically updated at each iteration. In practice, our computational experiments have shown that using fixed values for $\bar{\alpha}$ and $\bar{\delta}$ is a reasonable choice.

Figure 6 displays the feasible space (grey area) of (\hat{P}_D^{K+1}) . The projection on the space of Lagrange multipliers (λ_1, λ_2) depicts the shape of the subgradient-based trust-region. In the space of (λ_1, η) , CP^k represents the projection of the cutting plane generated at iteration k . Note that the feasible space of (\hat{P}_D^{K+1}) contains $(\lambda_1 = \lambda_1^K, \lambda_2 = \lambda_2^K, \eta = v(P_R(\lambda_1^K, \lambda_2^K)))$. In both plots (a) and (b), $\underline{\lambda}_1$ corresponds to a lower bound on multiplier λ_1 induced by the trust-region constraints.

The stopping criterion for this hybrid strategy is identical to the pure cutting plane method and is based on the *Lagrangian gap* between the relaxed primal problem and the restricted dual problem:

$$v(P_R(\lambda^K)) - v(P_D^K) \leq \varepsilon \quad (1)$$

In the pure cutting plane approach (without constraint (SG+TR)), the optimal value of the restricted dual problem $v(P_D^K)$ iteratively approximates $v(P_D)$ from below since it involves the minimization of a relaxation of $v(P_D)$.

$$v(P_D^K) \leq v(P_D) \quad \forall K \quad (2)$$

Therefore, the stopping criterion (1) ensures convergence to an ε -optimal solution of the dual problem (P_D) as:

$$v(P_R(\lambda^K)) - v(P_D^K) \leq \varepsilon \Rightarrow v(P_R(\lambda^K)) - v(P_D) \leq \varepsilon \quad (3)$$

In the proposed approach, when the pure restricted dual problem (P_D^K) becomes bounded, the stopping criterion (1) is used to check convergence while the hybrid restricted dual problem (\hat{P}_D^K) is used to update the Lagrange multipliers. Finite convergence properties for the pure cutting plane and trust-region methods in the context of mixed-integer linear programming can be obtained in Frangioni (2005) and Amor and Desrosiers (2006), respectively. For the rest of the paper, it is assumed that *practical* convergence of the proposed hybrid method can be achieved in the context of integrated refinery planning and scheduling.

5. Heuristic Solutions

In this section, a classical adaptation of the primal-dual algorithm is presented in order to obtain solutions that satisfy all constraints of the full-space problem, including linking constraints. As explained by Frangioni (2005), the solution of the Lagrangian dual problem yields primal information that can be used to generate good heuristic solutions for (P) . In this

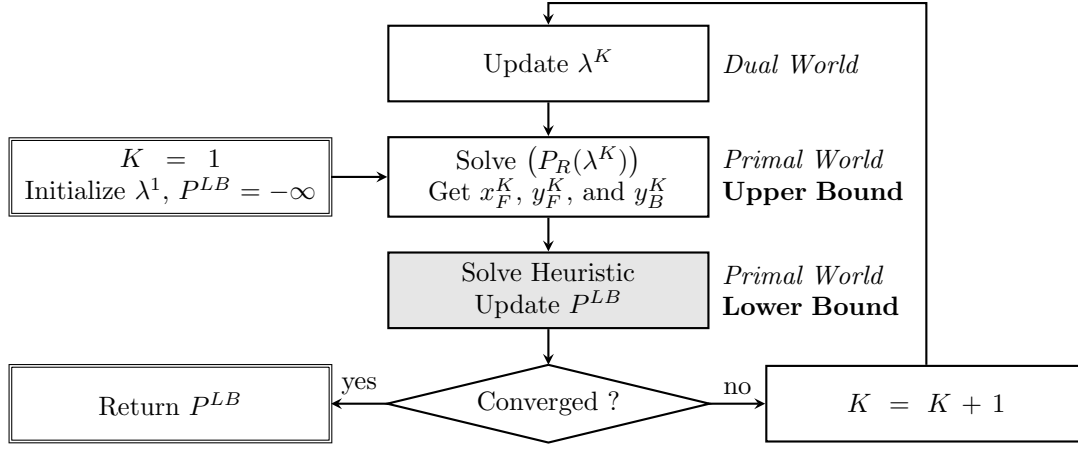


Figure 7: Iterative primal-dual algorithm with heuristic step.

paper, a heuristic step is introduced in the iterative algorithm to produce valid lower bounds P^{LB} (see Fig. 7). This induces a second stopping criterion based on the dual gap: $v(P_R(\lambda^K)) - P^{LB} \leq \epsilon'$. If either of the two stopping criteria is satisfied, the algorithm converges and returns P^{LB} .

The heuristic algorithm consists of fixing binary variables y_B from the crude-oil scheduling formulation to their values y_B^K in the solution of the relaxed problem $(P_R(\lambda^K))$. As a consequence, the full-space problem (P) reduces to a continuous NLP, denoted $(P_H(y_B^K))$, and can then be efficiently solved. If it is feasible and its (local) optimal solution is better than the previous incumbent, P^{LB} is updated. Otherwise, P^{LB} is left unchanged.

$$(P_H(y_B^K)) \left\{ \begin{array}{l} \max \quad V_P^T x_S - V_C^T y_F \\ \text{s.t.} \quad f_P(x_F, x_I, x_S) \leq 0 \\ \quad \quad g_P(x_I) \leq 0 \\ \quad \quad f_S(y_B^K, y_C, y_F) \leq 0 \\ \quad \quad g_S(y_C) \leq 0 \\ \quad \quad y_F - x_F = 0 \\ \quad \quad x_F \in \mathbb{R}^{|F|}, x_I \in \mathbb{R}^{|I|}, x_S \in \mathbb{R}^{|S|} \\ \quad \quad y_C \in \mathbb{R}^{|C|}, y_F \in \mathbb{R}^{|F|} \end{array} \right.$$

In this heuristic solution, fixing binary variables y_B to y_B^K corresponds to fixing the selection and sequencing of operations for the crude-oil scheduling system. Therefore, when solving problem $(P_H(y_B^K))$, the nonlinear solver has the opportunity to re-optimize all other continuous decisions such as quantities, blend recipes, and timing of operations (start time, duration, and end time). It is crucial to note that the timing of operations can only be re-optimized if these decisions are handled by continuous variables. For instance, in Mouret et al. (2010), the discrete-time representation, denoted by MOS-FST, uses binary variables to determine the timing decisions whereas all other representations, denoted by MOS, MOS-SST, and SOS, use continuous variables instead. This shows a clear advantage, although not intuitive, of continuous-time scheduling formulations over discrete-time representations. In particular, the latter might be inefficient in the context of this work as it would decrease the flexibility of the heuristic algorithm to find good, or at least feasible solutions.

Overall, it is interesting to note that using this approach, the iterative primal-dual algorithm that solves the Lagrangian dual problem acts as a *discrete solution generator* that suggests potentially good discrete solutions for the full-space problem. In other words, it searches the optimal selection and sequencing of operations for the crude-oil scheduling system.

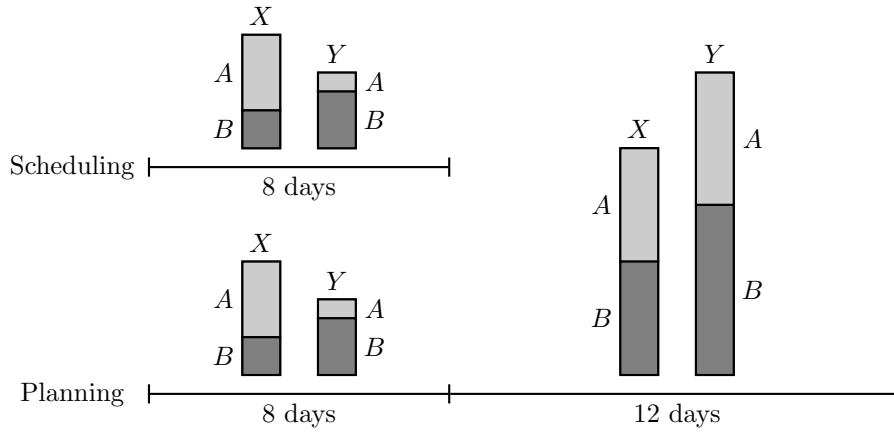


Figure 8: Crude-oil scheduling and multi-period refinery planning integration.

6. Remarks

6.1. CDU Feedstocks and Lagrange Multipliers

The number of Lagrange multipliers highly depends on the CDU feedstock possibilities. Exactly one multiplier is needed for feasible combination of crude and type of blends (or corresponding CDU operating mode). In the case-study, there are 2 different crudes which can both be blended in any of the 2 different types of blends, so 4 Lagrange multipliers are needed to solve the dual problem. Typical large-scale refineries may need 50 and up to 100 Lagrange multipliers.

The optimal value of the Lagrange multipliers correspond to the optimal marginal costs of the linking constraints for the convexified problem (for more details in the case of MILP models, see Frangioni, 2005). Therefore, the Lagrange multipliers can be seen as the optimal pricing strategy between the crude-oil scheduling system and the refinery planning system. In other words, the optimal Lagrange multipliers are crude prices for which it is economically equivalent to either exchange crudes between the two systems or sell and buy to the crude market (see Fig. 4). From this observation, it is natural to use the crude costs defined in the crude-oil scheduling problem as initialization for the Lagrange multipliers. For the case-study, we use the following initial values:

$$\begin{aligned}\lambda_{(X,A)}^1 &= \lambda_{(Y,A)}^1 = 7 \\ \lambda_{(X,B)}^1 &= \lambda_{(Y,B)}^1 = 6\end{aligned}$$

6.2. Multi-Period Refinery Planning

In this work, the refinery planning problem is expressed over a single period for which CDU feedstocks are synchronized with the crude-oil scheduling problem. Even though it is often computationally critical to increase the time horizon for scheduling problem, this can easily be done in refinery planning models by introducing additional time periods. Therefore, one can define 2 or more time periods for the refinery planning model and synchronize CDU feedstocks for the first period only, as shown in Figure 8. In this particular case, the refinery planning decisions for the second period, including CDU feedstock decisions x_F , are made without taking into account crude-oil scheduling constraints. This methodology allows making short-term scheduling decision while considering the long-term economic impacts of these decisions, which cannot be done with detailed long-term scheduling models due to their computational complexity.

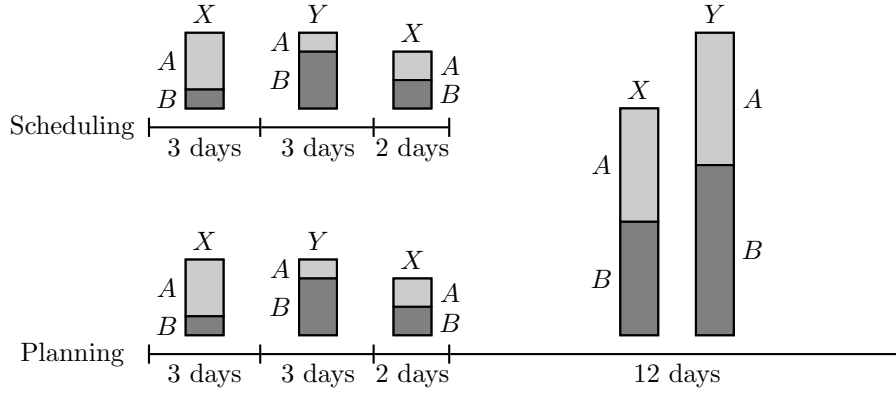


Figure 9: Disaggregated CDU feedstocks synchronization.

6.3. CDU Feedstocks Aggregation

An important issue with the proposed approach comes from the fact that the CDU feedstocks for the linking period are aggregated. In the optimal solution, the crude-oil operations schedule prepares several *batches* for each type of crude blends. Then, for each of these blend types, all the corresponding batches are accumulated and the refinery planning solution determines the processing decisions for the *aggregated batch*. This approximation may lead to sub-optimality or even technical infeasibility of the solutions obtained. This problem can be solved by postulating exactly one period for each batch and synchronizing all the corresponding CDU feedstocks. Figure 9 depicts the synchronization of disaggregated CDU feedstocks.

6.4. Handling Nonlinearities in Crude-Oil Scheduling Model

Although, the relaxed problem $(P_R(\lambda))$ is decomposable, it is not easy to solve it to global optimality. In particular, two major issues arise. First, the crude-oil scheduling problem $(P_S(\lambda))$ corresponds to a MINLP due to the presence of nonlinear composition constraints. In Mouret et al. (2009) an MILP relaxation is derived by simply dropping these nonlinear constraints. The solution can then be refined by fixing the binary variables and solving the reduced NLP, similarly to the heuristic approach presented in Section 5. Results show that the solution obtained is close to the global optimum as it tends to satisfy the relaxed nonlinear constraints. Therefore a similar methodology is used in this work. Instead of simply dualizing the linking constraints, the nonlinear scheduling constraints $g_S(y_C) \leq 0$ are also relaxed (i.e. dropped). The modified relaxed MINLP problem (NLP + MILP) is denoted by $(\tilde{P}_R(\lambda))$:

$$(\tilde{P}_R(\lambda)) \left\{ \begin{array}{l} \max \quad V_P^T x_S - \lambda^T x_F + (\lambda - V_C)^T y_F \\ \text{s.t.} \quad f_P(x_F, x_I, x_S) \leq 0 \\ \quad \quad g_P(x_I) \leq 0 \\ \quad \quad f_S(y_B, y_C, y_F) \leq 0 \\ \quad \quad x_F \in \mathbb{R}^{|F|}, x_I \in \mathbb{R}^{|I|}, x_S \in \mathbb{R}^{|S|} \\ \quad \quad y_B \in \{0, 1\}^{|B|}, y_C \in \mathbb{R}^{|C|}, y_F \in \mathbb{R}^{|F|} \end{array} \right.$$

The corresponding modified crude-oil scheduling MILP subproblem is denoted by $(\tilde{P}_S(\lambda))$:

$$(\tilde{P}_S(\lambda)) \left\{ \begin{array}{l} \max \quad (\lambda - V_C)^T y_F \\ \text{s.t.} \quad f_S(y_B, y_C, y_F) \leq 0 \\ \quad \quad y_B \in \{0, 1\}^{|B|}, y_C \in \mathbb{R}^{|C|}, y_F \in \mathbb{R}^{|F|} \end{array} \right.$$

The decomposability property is preserved:

$$v\left(\tilde{P}_R(\lambda)\right) = v\left(P_P(\lambda)\right) + v\left(\tilde{P}_S(\lambda)\right)$$

Finally, the modified dual problem (\tilde{P}_D) can be defined as:

$$(\tilde{P}_D) \quad \min_{\lambda} v\left(\tilde{P}_R(\lambda)\right)$$

This modified dual problem still provides a valid upper bound for the original full-space problem (P) . The following modified heuristic problem $(\tilde{P}_H(y_B^K))$ is also defined. It is obtained from the original heuristic problem $(P_H(y_B^K))$ by dropping the nonlinear scheduling constraints. It is used as a first heuristic step to get a good initial point before solving the original heuristic NLP problem $(P_H(y_B^K))$.

$$(\tilde{P}_H(y_B^K)) \quad \left\{ \begin{array}{l} \max \quad V_P^T x_S - V_C^T y_F \\ \text{s.t.} \quad f_P(x_F, x_I, x_S) \leq 0 \\ \quad \quad g_P(x_I) \leq 0 \\ \quad \quad f_S(y_B^K, y_C, y_F) \leq 0 \\ \quad \quad y_F - x_F = 0 \\ \quad \quad x_F \in \mathbb{R}^{|F|}, x_I \in \mathbb{R}^{|I|}, x_S \in \mathbb{R}^{|S|} \\ \quad \quad y_C \in \mathbb{R}^{|C|}, y_F \in \mathbb{R}^{|F|} \end{array} \right.$$

6.5. Handling Nonlinearities in the Refinery Planning Model

In order to obtain valid upper bounds when solving $(P_R(\lambda))$ or $(\tilde{P}_R(\lambda))$, the refinery planning problem $(P_P(\lambda))$ has to be solved to global optimality. Although global optimization of industrial large-scale refinery planning models is still unachievable, the case-study on the refinery planning model presented in Section 2 is solvable by the global NLP solver BARON in reasonable time. However, it should be noted that it is critical to provide tight bounds for the quality variables $q_2^{j_{kp}}$. In particular, based on the structure of the pooling system, we use the following bounds:

$$\min_{i \in I} q_1^{ijk} \leq q_2^{j_{kp}} \leq \max_{i \in I} q_1^{ijk}$$

6.6. Detailed Implementation

Based on previous remarks to handle nonlinearities, the complete heuristic algorithm is developed as depicted in Figure 10. Although global optimality cannot be ensured, the dual gap can be estimated using the upper bound provided by $v\left(\tilde{P}_R(\lambda^*)\right)$. Local NLP solvers, such as CONOPT, are used for the heuristic steps as the solution time is more critical than global optimality for these problems. The hybrid restricted dual problem (\hat{P}_D^K) is solved using the best solution of the modified heuristic problems $(\tilde{P}_H(y_B^K))$ to estimate the optimal dual solution $v(P_D)$. The stopping criterion is based on relative gaps but can also be expressed in terms of absolute gaps. Converging on the Lagrangian gap means that no further improvement of the upper bound can be achieved and the current Lagrange multipliers are optimal.

7. Numerical Illustration

In this section, several computational results are presented for two approaches: the direct MINLP approach and the proposed Lagrangian decomposition method. Experiments have been performed on an Intel Xeon 1.86GHz processor using GAMS as the modeling and algorithmic language. A 1000 seconds time limit has been used for each run. The following

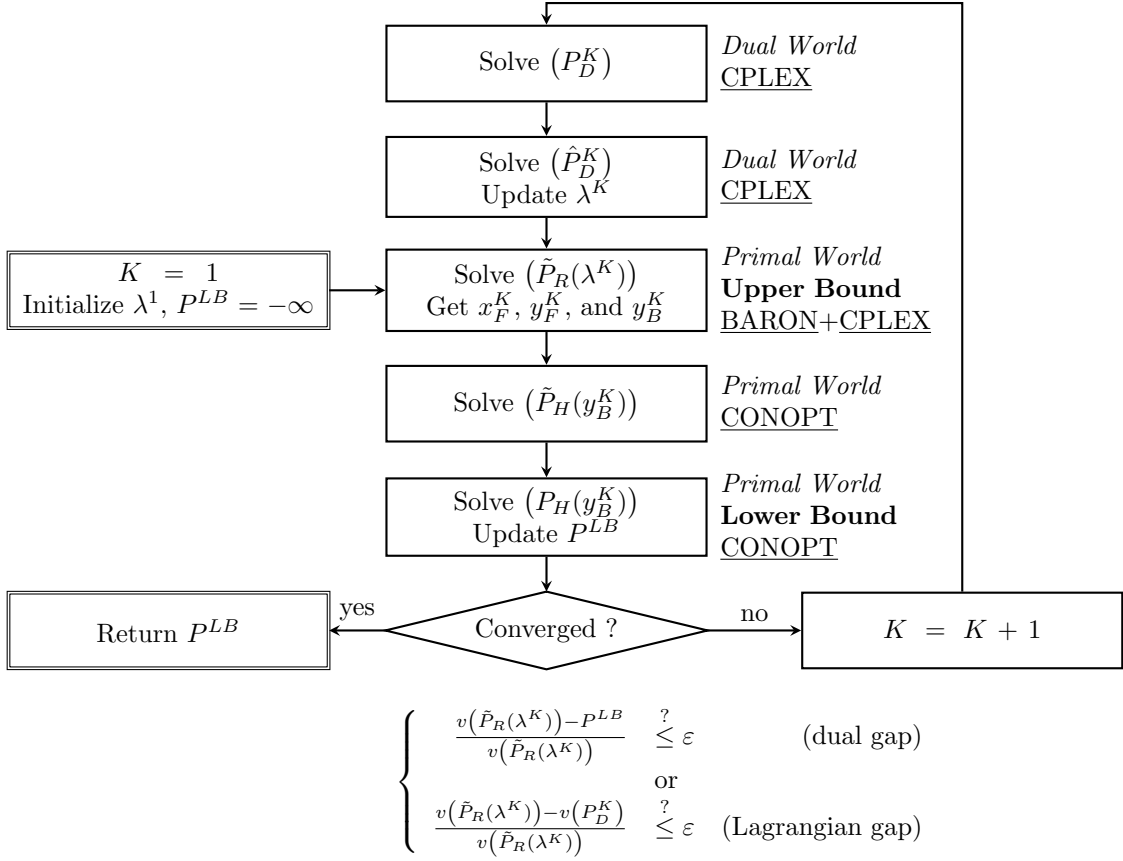


Figure 10: Complete algorithm implementation.

local NLP solvers have been used: CONOPT, SNOPT and IPOPT. In our experiments, the convergence tolerance ε is set to 0.0001, the maximum step size parameter $\bar{\alpha}$ is set to 1 and the step bound parameter $\bar{\delta}$ to 0.05.

The number of priority-slots for the crude-oil scheduling model is set to 6 and 7 (see Mouret et al., 2010). Tables 2 and 3 show iteration statistics for the Lagrangian decomposition method using SNOPT as the heuristic NLP solver. In particular, the optimal value of each problem solved is given as well as the optimal step size calculated by the hybrid restrict dual problem and the cumulative CPU time at the end of each iteration (e.g., for 6 priority-slots, the first iteration took 3 seconds). Dashes are used when the information is not available (Hybrid Dual and Step Size for the first iteration), when the problem is unbounded (Pure Dual during the first few iterations), or when it is locally infeasible (Original Heuristic at some iterations).

In each case, the global optimal solution is found (see underlined entries in the *Original Heuristic* column) and proved optimal. It can be noted that in some iterations the step size variable α is strictly lower than 1, which corresponds to cases where the pure subgradient multiplier update would violate some cutting planes generated at previous iterations. The proposed approach automatically overcomes this issue. The increase of CPU time between 6 and 7 priority-slots is mostly explained by the increase in size of the MILP scheduling model. Figures 11 and 13 plot the evolution of the objective value of various problems solved during the Lagrangian iterations.

The optimal value of the Lagrange multipliers is $\lambda_{(X,A)}^* = \lambda_{(X,B)}^* = \lambda_{(Y,A)}^* = \lambda_{(Y,B)}^* = 7$. Figures 12 and 14 plot the evolution of the Lagrange multipliers during the Lagrangian iterations. The proposed approach demonstrates its efficiency through stable updates and fast convergence of the Lagrange multipliers.

A basic sequential procedure is introduced to compare with the Lagrangian decomposition approach. First, the modified crude-oil scheduling problem (\tilde{P}_S) is solved and the binary variables are fixed to their solution value. Then, the modified and original heuristic problems ($\tilde{P}_H(y_B^0)$) and ($P_H(y_B^0)$) are successively solved. This procedure is not computationally expensive

Table 2: Lagrangian iterations statistics (6 priority-slots, NLP=SNOPT)

Iteration	Pure Dual $v(P_D^K)$	Hybrid Dual $v(\hat{P}_D^K)$	Step Size α^K	Modified Relaxation $v(\tilde{P}_R(\lambda^K))$	Modified Heuristic $v(\tilde{P}_H(y_B^K))$	Original Heuristic $v(P_H(y_B^K))$	CPU Time
1	— ^a	— ^b	— ^b	645.000	400.942	— ^c	3s
2	— ^a	389.942	1	689.049	564.000	560.500	9s
3	— ^a	547.929	1	637.063	592.368	<u>592.368</u>	13s
4	— ^a	582.790	1	603.377	592.368	<u>592.368</u>	18s
5	— ^a	591.172	1	609.526	586.191	— ^c	24s
6	— ^a	590.629	0.851	598.380	562.881	— ^c	29s
7	580.717	591.324	-0.617	600.171	586.191	— ^c	33s
8	592.368	592.785	0.390	594.774	586.191	— ^c	46s
9	592.369	592.369	1	592.372	592.368	<u>592.368</u>	50s

^a unbounded LP

^b not available

^c locally infeasible NLP

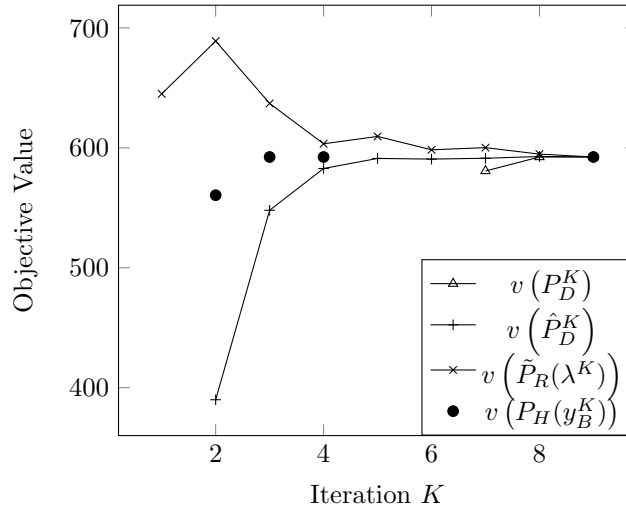


Figure 11: Lagrangian iteration objective values (6 priority-slots, NLP=SNOPT).

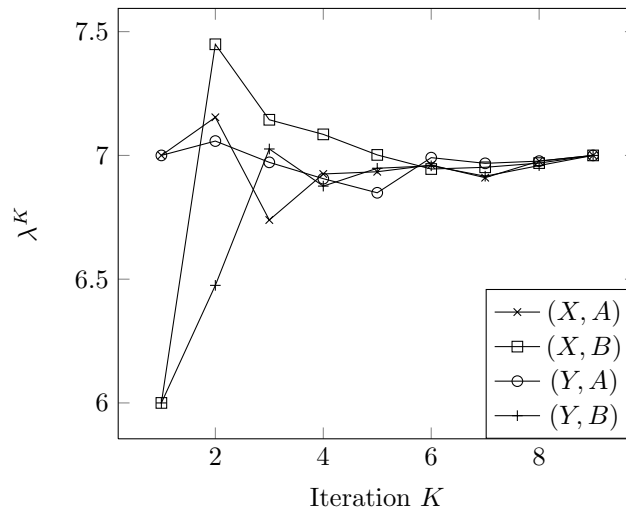


Figure 12: Lagrange multiplier updates (6 priority-slots, NLP=SNOPT).

Table 3: Lagrangian iterations statistics (7 priority-slots, NLP=SNOPT)

Iteration	Pure Dual $v(P_D^K)$	Hybrid Dual $v(\hat{P}_D^K)$	Step Size α^K	Modified Relaxation $v(\tilde{P}_R(\lambda^K))$	Modified Heuristic $v(\tilde{P}_H(y_B^K))$	Original Heuristic $v(P_H(y_B^K))$	CPU Time
1	—	—	—	645.000	393.470	—	3s
2	—	381.562	1	751.494	568.077	568.077	10s
3	—	552.076	1	622.785	592.368	—	16s
4	—	574.735	1	614.178	592.368	<u>592.368</u>	22s
5	—	583.166	1	602.218	592.368	592.079	50s
6	—	588.279	1	617.268	592.368	592.079	56s
7	—	592.856	0.911	600.752	592.368	<u>592.368</u>	66s
8	—	592.835	0.517	595.264	592.368	—	81s
9	—	592.288	0.357	595.292	592.368	<u>592.368</u>	95s
10	592.368	592.369	1	592.369	592.368	592.079	101s

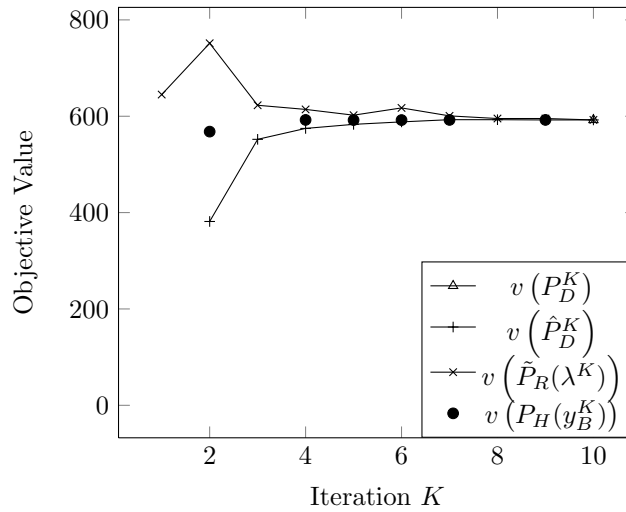


Figure 13: Lagrangian iteration objective values (7 priority-slots, NLP=SNOPT).

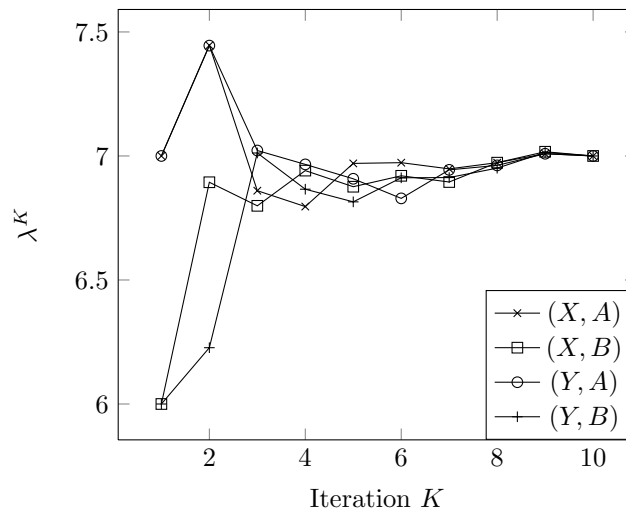


Figure 14: Lagrange multiplier updates (7 priority-slots, NLP=SNOPT).

Table 4: Comparative performance of several MINLP algorithms

MINLP Solver	6 priority-slots			7 priority-slots		
	Objective Value	CPU Time	Optimality Gap	Objective Value	CPU Time	Optimality Gap
Proposed (CONOPT)	<u>592.368</u>	37s	0%	<u>592.368</u>	94s	0%
Proposed (SNOPT)	<u>592.368</u>	50s	0%	<u>592.368</u>	101s	0%
Proposed (IPOPT)	<u>592.368</u>	244s	0%	<u>592.368</u>	833s	0%
Sequential (BARON)	545.000	9s	—	545.000	10s	—
DICOPT (CONOPT)	545.000	5s	—	<u>592.368</u>	7s	—
DICOPT (SNOPT)	<u>592.368</u>	429s	—	<u>592.368</u>	6s	—
DICOPT (IPOPT)	568.077	54s	—	<u>592.368</u>	44s	—
AlphaECP (CONOPT)	512.324	67s	—	545.000	120s	—
AlphaECP (SNOPT)	512.324	67s	—	545.000	395s	—
AlphaECP (IPOPT)	512.324	69s	—	545.000	175s	—
SBB (CONOPT)	<u>592.368</u>	267s	—	<u>592.368</u>	+1,000s	—
SBB (SNOPT)	—	+1,000s	—	—	+1,000s	—
SBB (IPOPT)	—	+1,000s	—	493.536	+1,000s	—
LINDOGLOBAL	568.077	+1,000s	11.9%	532.857	+1,000s	17.1%
BARON	592.170	+1,000s	7.3%	400.000	+1,000s	37.5%

as it requires solving only one MILP, solved with CPLEX, and two NLPs, both solved with BARON. However, the solution obtained, if any, might not be optimal.

Additionally, the Lagrangian decomposition approach is compared to the direct approach which consists of solving the full-space problem (P) with various MINLP solvers. Table 4 shows the computational performance of these MINLP algorithms. The sequential approach quickly provides a feasible solution which is 8.0% lower than the global optimum (545.000 against 592,368). The solvers DICOPT, AlphaECP and SBB cannot guarantee global optimality of the solution returned while LINDOGLOBAL and BARON are actual global MINLP solvers. DICOPT is able to find the global optimal solution in many cases in reasonable CPU times. AlphaECP never finds the global optimum. SBB seems to return the best solutions when CONOPT is used as NLP solver but it requires large CPU times. Neither LINDOGLOBAL or BARON have found the global optimum in the specified time limit.

In comparison to these solvers, the proposed Lagrangian decomposition approach proves to be very effective for the following reasons:

- it is computationally effective (although DICOPT is faster);
- it always returns the global optimum;
- it is very robust with the choice of NLP solver (although IPOPT is significantly slower).

Figure 15 depicts the composition of each crude blend in four different solutions. Crude blend X is mostly composed of crude A while crude blend Y is mostly composed of crude B . If all solutions except the second one (with objective value 568.077) are considered, one may conclude that blend Y is "more profitable" than blend X because the objective value increases when processing larger amounts of blend Y and smaller amounts of blend X . Additionally, one could say that increasing the amount of distilled crude increases the overall profit (solution 1 processes 253.034 units of crude, solution 3, 224.49 units of crude, and solution 4, 212.867 units of crude). Interestingly, the second solution does not follow these observations. In this solution, the amount of blend X is larger than the amount of blend Y . Besides, the total amount of crude processed is the largest (276.923 units of crude). This shows how difficult it is to approximate the economic behavior of refinery operations, even for such a small case-study. Therefore, it is possible that linear approximations of refinery operations

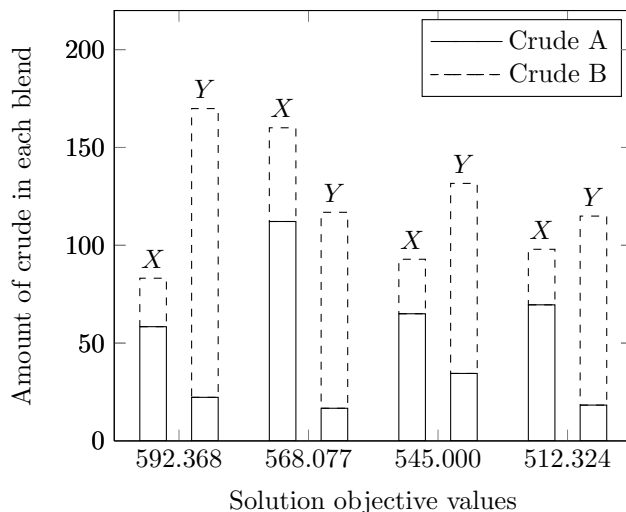


Figure 15: Blend compositions in solutions obtained with 6 priority-slots.

as proposed by Robertson et al. (2010) might not be able to correctly evaluate the economic value of some feasible solutions.

8. Larger Refinery Problem

In this section, the proposed Lagrangian decomposition approach is applied on a larger refinery example and compared to standard MINLP solvers. The refinery planning problem is based on a crude distillation simulation model developed by Gueddar and Dua (2010). This crude distillation model is based on a layered artificial neural network (ANN). This ANN is generated by solving an MINLP which aims at fitting empirical data for atmospheric distillation of several crudes (crude assays) for simplifying the calculations. The model obtained is able to predict cut yields and cut properties from the chosen cut points and properties of the inlet crude. Figure 16 depicts the full nonlinear planning model. Three different types of crude blends are processed in three different CDU operating modes and five crude cuts are produced: LPG (Liquefied Petroleum Gas), naphta, kerosene, diesel, and residue. Each discrete CDU mode is defined by the distillation cut point between diesel and residue fractions: 340, 360, or 380°C. The decision variables for each CDU mode are composed of individual crude flows and distillation cut points between naphta, kerosene and diesel fractions. The three streams produced for each fraction are then blended into cut pools. The bilinear pooling constraints discussed in Section 2 are classically used to calculate the properties of each cut pool. Table 5 shows market prices for each cut and corresponding property specifications. Crude availability constraints are also included in the model in accordance with the description of the crude-oil scheduling problem. This refinery planning model is composed of 1,831 variables and 1,817 constraints. The full mathematical model is detailed in Appendix A.

The crude-oil scheduling problem is based on example 3 from Lee et al. (1996) (see Figure 17) with the parameters described in Table 6. It consists of three crude arrivals, three storage tanks, three charging tanks (one for each type of crude mixture), and two identical CDUs whose respective scheduled feedstocks are merged when linked to the refinery planning problem. Seven different crudes are available, and fourteen transfer operations can be executed to prepare the different crude blends. When 6 priority-slots are used, the crude-oil scheduling problem is composed of 1,814 variables (84 binary) and 2,338 constraints.

Table 7 and Figure 18 show the iteration statistics for the Lagrangian decomposition method using 6 priority-slots for the crude-oil scheduling model. The maximum step size parameter $\bar{\alpha}$ is set to 1 and the step bound parameter $\bar{\delta}$ to 0.05. Because

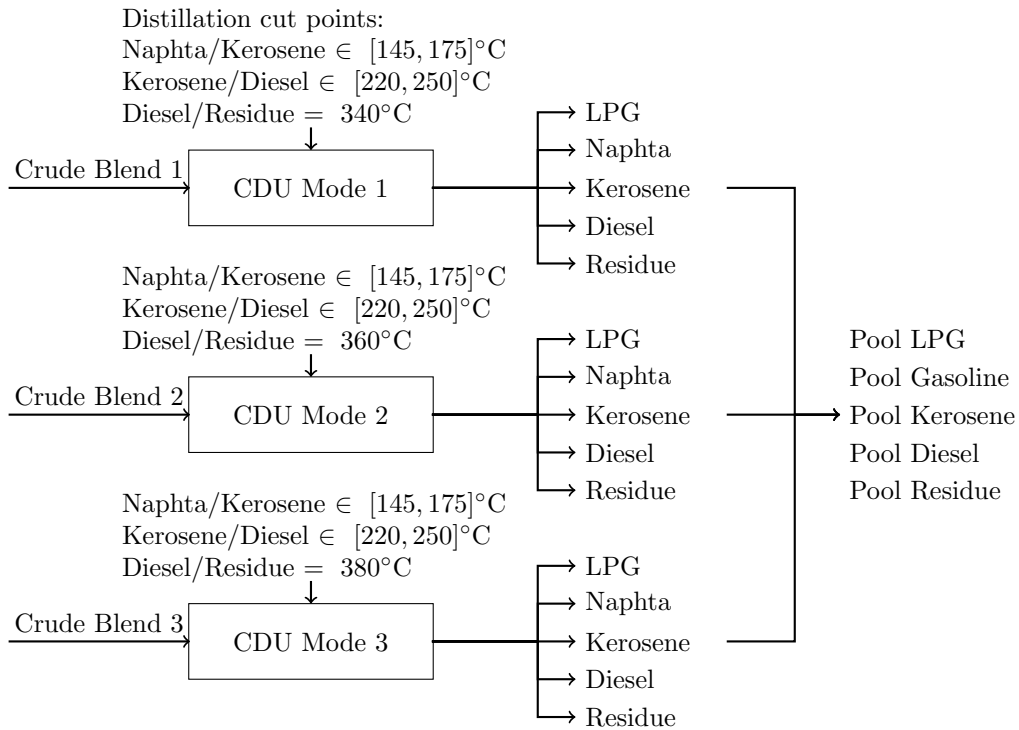


Figure 16: Planning model for larger refinery problem.

Table 5: Crude cut prices and specification for larger refinery problem

Crude Cut	Price	Specifications
LPG	8.5	None
Naphta	8.0	specific gravity $\in [0.72, 0.775]$ motor octane number ≥ 45 research octane number ≥ 45 sulfur weight content $\leq 120\text{ppm}$
Kerosene	7.0	specific gravity $\in [0.775, 0.84]$ freeze point $\leq -40^{\circ}\text{C}$
Diesel	8.0	specific gravity $\in [0.82, 0.86]$ cetane number ≥ 48 cloud point $\leq 4^{\circ}\text{C}$ sulfur weight content $\leq 2800\text{ppm}$
Residue	6.5	None

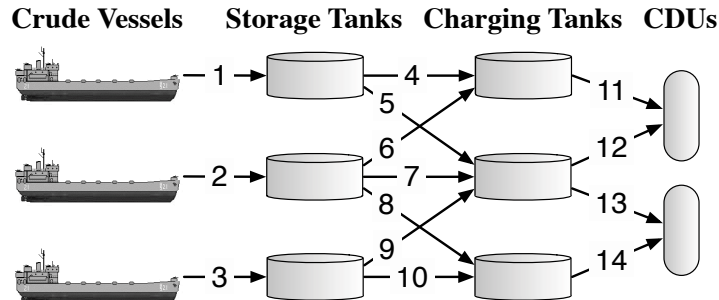


Figure 17: Crude-oil scheduling system 3 (Lee et al., 1996).

Table 6: Crude-oil scheduling data for larger refinery problem

Scheduling horizon			12 days
Vessels	Arrival time	Composition	Amount of crude
Vessel 1	0	100% A	500
Vessel 2	4	100% B	500
Vessel 3	8	100% C	500
Storage tanks	Capacity	Initial composition	Initial amount of crude
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	Initial composition	Initial amount of crude
Tank 1 (mix 1)	[0, 1,000]	100% G	300
Tank 2 (mix 2)	[0, 1,000]	100% E	500
Tank 3 (mix 3)	[0, 1,000]	100% F	300
Crudes	Property 1 (sulfur concentration)		Crude unit cost
Crude A	0.01		6
Crude B	0.085		6.5
Crude C	0.06		5.5
Crude D	0.02		7.2
Crude E	0.05		6.7
Crude F	0.08		6.2
Crude G	0.03		7.5
Crude mixtures	Property 1 (sulfur concentration)		Maximum number of batches
Crude mix 1	[0.025, 0.035]		6
Crude mix 2	[0.045, 0.065]		6
Crude mix 3	[0.075, 0.085]		6
Unloading flowrate	[0, 50]	Transfer flowrate	[0, 50]
Distillation flowrate	[5, 50]	Minimum duration of distillations	1 day

we were not able to solve the refinery planning problem to global optimality, we used CONOPT as the NLP solver. Other local NLP solvers as SNOPT and IPOPT did not perform as well (slow convergence, poor solutions or local infeasibilities). As the refinery planning problem is not solved to global optimality, it is not possible to rigorously estimate global optimality for the solution obtained for the full-space problem. Therefore, the convergence tolerance ε is set to 0.01 instead of 0.0001 as in the case-study. Furthermore, many iterations are needed to generate enough cutting planes and make the pure restricted dual problem bounded. In order to achieve convergence after a few iterations, the Lagrangian gap is calculated using the objective value of the hybrid dual problem, which is always bounded, as follows:

$$\frac{v\left(\tilde{P}_R(\lambda^K)\right) - v\left(\hat{P}_D^K\right)}{v\left(\tilde{P}_R(\lambda^K)\right)} \stackrel{?}{\leq} \varepsilon$$

The proposed approach converges on the Lagrangian gap in 15 iterations. The final dual gap, although it does not represent a valid optimality gap, is 3.8%. 83% of the time is spent on solving the crude-oil scheduling MILPs. The optimal value of the Lagrange multiplier is given in Table 8. With 7 priority-slots, the decomposition procedure converges in 18 iterations and 4,451 seconds, the increase of CPU time being explained mostly by the increased number of priority-slots. The solution obtained is slightly lower than the previous one because the algorithm did not fully converge within tight tolerances.

Table 9 presents computational performances of the different approaches. Clearly, the Lagrangian decomposition procedure is much more robust than the other algorithms. Only the sequential approach was able to deliver a solution with 6 priority-slots, but its objective value is much lower (50% reduction) than the best know solution. Indeed, the scheduling solution obtain during the first stage of the sequential procedure does not take into account the economic impact on the

Table 7: Lagrangian iterations statistics for larger refinery problem (6 priority-slots, NLP=CONOPT)

Iteration	Pure Dual $v(P_D^K)$	Hybrid Dual $v(\hat{P}_D^K)$	Step Size α^K	Modified Relaxation $v(\tilde{P}_R(\lambda^K))$	Modified Heuristic $v(\tilde{P}_H(y_B^K))$	Original Heuristic $v(P_H(y_B^K))$	CPU Time
1	—	—	—	266.226	—	—	43s
2	—	-2.166	1	417.311	257.943	222.339	102s
3	—	253.546	1	411.657	258.488	245.306	173s
4	—	281.076	0.857	325.906	258.106	—	235s
5	—	260.746	1	288.587	258.104	—	298s
6	—	256.301	1	273.197	258.516	245.467	372s
7	—	256.385	1	271.415	258.027	<u>250.989</u>	459s
8	—	258.091	1	264.974	257.961	233.713	539s
9	—	258.847	0.861	265.149	257.961	233.713	610s
10	—	257.623	1	267.541	257.740	228.230	682s
11	—	259.500	0.868	261.237	257.750	247.971	769s
12	—	259.170	1	261.512	257.750	247.971	850s
13	—	259.126	0.759	264.109	258.027	<u>250.989</u>	913s
14	—	259.121	1	263.708	258.306	—	979s
15	—	259.592	0.774	260.857	258.516	238.763	1,045s

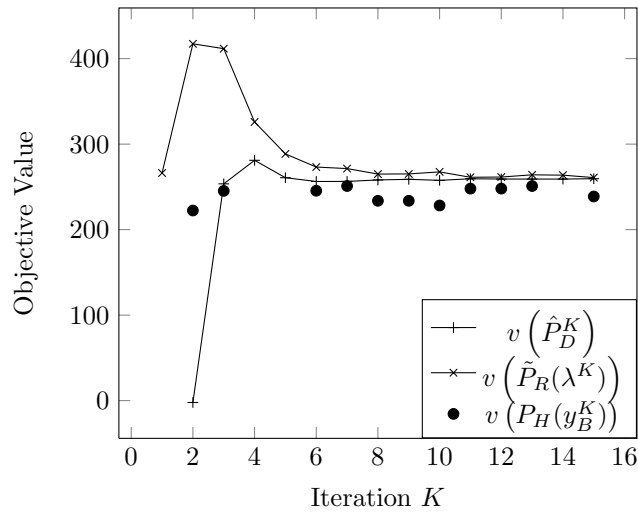


Figure 18: Lagrangian iteration objective values for larger refinery problem (6 priority-slots, NLP=CONOPT).

Table 8: Optimal Lagrange multipliers for each crude and each CDU mode

Crude	CDU Mode	Initial value (=Price)	Optimal value
A	1	6.0	6.169
	2	6.0	6.207
	3	6.0	7.695
B	1	6.5	6.971
	2	6.5	6.987
	3	6.5	7.026
C	1	5.5	6.062
	2	5.5	6.009
	3	5.5	6.054
D	1	7.2	7.186
	2	7.2	7.262
	3	7.2	7.962
E	1	6.7	6.859
	2	6.7	6.878
	3	6.7	6.898
F	1	6.2	6.340
	2	6.2	6.226
	3	6.2	6.293
G	1	7.5	6.930
	2	7.5	7.360
	3	7.5	7.360

Table 9: Comparative performance of several MINLP algorithms for larger refinery problem (NLP solver: CONOPT)

MINLP Solver	6 priority-slots		7 priority-slots	
	Objective Value	CPU Time	Objective Value	CPU Time
Proposed	250.989	1,045s	250.128	4,451s
Sequential	116.814	46s	—	79s
DICOPT	—	+3,600s	—	+3,600s
AlphaECP	—	+3,600s	—	+3,600s
SBB	—	+3,600s	—	+3,600s

refinery planning problem, which leads to poor decisions. The standard local MINLP solvers were not able to find a solution within one hour because the MINLP model is too large: 3,645 variables (84 binary) and 4,177 constraints.

Table 10 shows the crude compositions of the optimal solution with objective value 250.989. The two CDUs are mostly operated in mode 2 which corresponds to the average cut point for diesel and residue cuts.

9. Conclusion

In this work, a novel approach towards the integration of planning and scheduling has been developed in the context of oil refining. In particular, a precise crude-oil operations scheduling model and a coarse refinery planning model were optimized simultaneously using Lagrangian decomposition. It makes use of Lagrange multipliers as a way to communicate economic information between the two subsystems. The methodology leads to a classical primal-dual iterative algorithm to solve the Lagrangian dual problem. The critical multiplier update step is performed by solving a new hybrid restricted dual problem. This approach combines the strengths of cutting planes and subgradient steps and does not require to define heuristic updates of parameters during iterations.

Although it is not guaranteed, our results achieved a 0% dual gap for the smaller case-study. It is well-known that augmented Lagrangian techniques (see Li and Ierapetritou, 2009) can ensure to close the dual gap for any instance. However, this would require to solve the refinery planning subproblem to global optimality, which is not yet achievable in an industrial

Table 10: Blend compositions in the optimal solution of larger refinery problem

Crude	Blend 1	Blend 2	Blend 3
A	0.164	49.480	
B		33.293	16.707
C		50.000	
D	0.066	19.792	
E	3.500	55.943	2.982
F			50.000
G	9.480		

context. It is therefore more practical to use standard duality in order to obtain feasible heuristic solutions for the integrated problem.

The proposed Lagrangian decomposition procedure has been applied to a larger and more complex refinery problem. The refinery planning problem is based on a crude distillation simulation model that has been developed by Gueddar and Dua (2010) using an artificial neural network to fit empirical data. The crude-oil scheduling problem is the third example of Lee et al. (1996). The results have shown that the proposed approach remains very competitive compared to other MINLP algorithms, namely the two-step MILP-NLP sequential procedure and standard MINLP solvers.

Acknowledgment

The authors are grateful to Total Raffinage & Marketing for financial support of this project.

Appendix A. Mathematical Model for the Refinery Planning Problem

In this section, the ANN model developed in Gueddar and Dua (2010) for CDU simulations is presented. It is based on a layered directed graph which represents the model calculations (see Fig. A.19). Each node in the input/output layers correspond to one input/output variable. Each node $j = 1, \dots, N_n$ in the intermediate layer $l = 1, \dots, N_l$ corresponds to an *activation variable* a_j^l and a *transformed variable* h_j^l . The activation variables are calculated from the transformed variables of the previous layer using an affine expression while the transformed variables are calculated by applying the hyperbolic tangent to their associated activation variable. The ANN equations are expressed as follows.

$$a_j^1 = \sum_{i=1}^{N_x} w_{ji}^1 x_i + b_j^1 \quad j = 1, \dots, N_n \quad (\text{A.1})$$

$$h_j^l = \tanh a_j^l \quad l = 1, \dots, N_h, j = 1, \dots, N_n \quad (\text{A.2})$$

$$a_j^l = \sum_{i=1}^{N_n} w_{ji}^l h_i^{l-1} + b_j^l \quad l = 2, \dots, N_h, j = 1, \dots, N_n \quad (\text{A.3})$$

$$u_k = \sum_{i=1}^{N_n} W_{ki} h_i^{N_h} + B_k \quad k = 1, \dots, N_o \quad (\text{A.4})$$

The model uses the following parameters:

- N_x is the number of inputs
- N_o is the number of outputs
- N_h is the number of intermediate layers

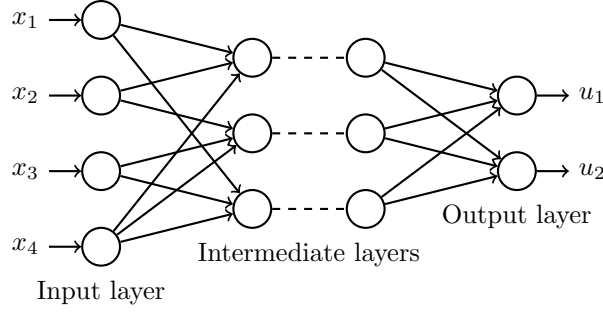


Figure A.19: Layered Artificial Neural Network.

- N_n is the number of nodes in each intermediate layer
- $w_{ji^l}, b_j^l, W_{ki}, B_k$ are parameters specific to the ANN

Dua (2010) demonstrates how to tune the ANN parameters by minimizing the total prediction error as well as the ANN complexity. This tuning step is performed by solving a *training* MINLP. We denote $\text{CDUANN}(x, u)$ the set of constraints defining the relation between the ANN inputs x and outputs u . In particular, the model inputs include crude properties q_C^{ip} and CDU cut points τ^k ($k \in \{\text{naphta, kerosene, diesel}\}$), and the model outputs include cut yields α^{ijk} and crude cut properties q_1^{ijkp} . All crude property inputs are fixed while CDU cut points are variable. The cut point between diesel and residue cuts can take three discrete values defining the three CDU operating modes. All the outputs are variable. The full refinery planning model is expressed as follows.

$$\begin{aligned}
\max \quad & \sum_{l \in L} p^l x_S^l && \text{(sales revenue maximization)} \\
\text{s.t.} \quad & 0 \leq \sum_{j \in J} x_F^{ij} \leq C^i && i \in I \quad \text{(crude availability)} \\
& \underline{\tau}^k \leq \tau^k \leq \overline{\tau}^k && k \in K \quad \text{(CDU cut point limits)} \\
& \text{CDUANN} \left(\left\{ q_C^{ip}, \tau^k \right\}, \left\{ \alpha^{ijk}, q_1^{ijkp} \right\} \right) && \text{(CDU model)} \\
& \underline{FR} \cdot H \leq \sum_{i \in I} \sum_{j \in J} x_F^{ij} \leq \overline{FR} \cdot H && \text{(CDU flowrate limitations)} \\
& x_1^{ijk} = \alpha^{ijk} \cdot x_F^{ij} && (i, j, k) \in I \times J \times K \quad \text{(CDU yield calculation)} \\
& \sum_{i \in I} x_1^{ijk} = \sum_{l \in L} x_2^{jkl} && (j, k) \in J \times K \quad \text{(pool mass balance)} \\
& \sum_{i \in I} q_1^{ijkp} x_1^{ijk} = q_2^{jkp} \sum_{l \in L} x_2^{jkl} && (j, k, p) \in J \times K \times P \quad \text{(pool quality balance, nonlinear)} \\
& \sum_{j \in J} \sum_{k \in K} x_2^{jkl} = x_S^l && l \in L \quad \text{(product mass balance)} \\
& x_S^l \leq D^l && l \in L \quad \text{(maximum product demand)} \\
& \sum_{j \in J} \sum_{k \in K} q_2^{jkp} x_2^{jkl} \leq Z^{lp} x_S^l && (l, p) \in L \times P \quad \text{(product quality requirement, nonlinear)} \\
& x_F^{ij}, x_1^{ijk}, x_2^{jkl}, x_S^l, \alpha^{ijk} \geq 0, q_1^{ijkp}, q_2^{jkp}, \tau^k \in \mathbb{R}
\end{aligned}$$

Appendix B. Mathematical Model for the Crude-Oil Operations Scheduling Problem

In this section, the mathematical model for crude-oil operations scheduling problems corresponding to the MOS time representation (see Mouret et al., 2010) is presented. It makes use of the following sets and parameters.

- $T = \{1, \dots, n\}$ is a totally ordered set of priority-slots (indices i, j, i_1, i_2)
- W is the set of all operations (indices v, v', v_1, v_2)
- $P_{v_1 v_2} = 1$ denotes a precedence constraint between operations v_1 and v_2
- $P_{W_1 W_2} = 1$ denotes a precedence constraint between set of operations W_1 and W_2
- $W_U \subset W$ is the set of unloading operations ($W_U = \{1, 2, 3\}$ for COSP2)
- $W_T \subset W$ is the set of tank-to-tank transfer operations ($W_T = \{4, \dots, 10\}$ for COSP2)
- $W_D \subset W$ is the set of distillation operations ($W_D = \{11, \dots, 14\}$ for COSP2)
- R is the set of resources (i.e. tanks, units): $R = R_V \cup R_S \cup R_C \cup R_D$
- $R_V \subset R$ 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)
- H is the scheduling horizon
- $\underline{S}_v \in [0, H]$ is a lower bound on the start time of unloading operations $v \in W_U$
- $NO_{v_1 v_2}$ is 1 if operations v_1 and v_2 must not overlap, 0 if they are allowed to overlap
- G_{NO} is the non-overlapping graph (see Mouret et al., 2010) whose adjacency matrix is NO
- $[\underline{V}_v^t, \overline{V}_v^t]$ are bounds on the total volume transferred during transfer operation v ; in all instances, $\underline{V}_v^t = 0$ for all operations except unloadings for which $\underline{V}_v^t = \overline{V}_v^t$ is the volume of crude in the marine vessel
- $[\underline{N}_D, \overline{N}_D]$ are the bounds on the number of distillations
- $[\underline{FR}_v, \overline{FR}_v]$ are flowrate limitations for transfer operation v
- $[\underline{x}_{vk}, \overline{x}_{vk}]$ are the limits of property k of the blended products transferred during operation v
- x_{ck} is the value of the property k of crude c

- $[L_r^t, \overline{L}_r^t]$ are the capacity limits of tank r
- $[D_r, \overline{D}_r]$ are the bounds of the demand on products to be transferred out of the charging tank r during the scheduling horizon

The following variables are introduced in the model.

- **Assignment variables** $Z_{iv} \in \{0, 1\}$ $i \in T, v \in W$
 $Z_{iv} = 1$ if operation v is assigned to priority-slot i , $Z_{iv} = 0$ otherwise
- **Time variables** $S_{iv} \geq 0, D_{iv} \geq 0, E_{iv} \geq 0$ $i \in T, v \in W$
 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
 E_{iv} is the end time of operation v if it is assigned to priority-slot i , $E_{iv} = 0$ otherwise
- **Operation variables** $V_{iv}^t \geq 0$ and $V_{ivc} \geq 0$ $i \in T, v \in W, c \in C$
 V_{iv}^t is the total volume of crude transferred during operation v if it is assigned to priority-slot i , $V_{iv}^t = 0$ 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}^t and L_{irc} $i \in T, r \in R, c \in C$
 L_{ir}^t is the total *accumulated* level of crude in tank $r \in R_S \cup R_C$ before the operation assigned to priority-slot i
 L_{irc} is the *accumulated* level of crude c in tank $r \in R_S \cup R_C$ before the operation assigned to priority-slot i

The crude-oil scheduling MOS model is expressed as follows:

$$\begin{aligned}
\text{Maximize} \quad & - \sum_{c \in C} V_C^c \cdot y_F^c \\
\text{Subject to} \quad & \\
& y_F^c = \sum_{i \in T} \sum_{r \in R_D} \sum_{v \in I_r} V_{ivc} & c \in C \\
& S_{iv} \geq \underline{S}_v \cdot Z_{iv} & i \in T, v \in W_U \\
& E_{iv} \leq H \cdot Z_{iv} & i \in T, v \in W \\
& E_{iv} = S_{iv} + D_{iv} & i \in T, v \in W \\
& \sum_{i \in T} \sum_{v \in O_r} Z_{iv} = 1 & r \in R_V \\
& \underline{N}_D \leq \sum_{i \in T} \sum_{v \in W_D} Z_{iv} \leq \overline{N}_D \\
& \sum_{i \in T} \sum_{v \in O_{r_1}} E_{iv} \leq \sum_{i \in T} \sum_{v \in O_{r_2}} S_{iv} & r_1, r_2 \in R_V, r_1 < r_2 \\
& \sum_{\substack{j \in T \\ j < i}} \sum_{v \in O_{r_1}} Z_{jv} \geq \sum_{\substack{j \in T \\ j \leq i}} \sum_{v \in O_{r_2}} Z_{jv} & i \in T, r_1, r_2 \in R_V, r_1 < r_2 \\
& \sum_{i \in T} \sum_{v \in I_r} D_{iv} = H & r \in R_D \\
& \sum_{v \in W'} Z_{iv} \leq 1 & i \in T, W' \in \text{clique}(G_{NO}) \\
& \sum_{v \in W'} E_{i_1 v} + \sum_{\substack{i \in T \\ i_1 < i < i_2}} \sum_{v \in W'} D_{iv} \leq \sum_{v \in W'} S_{i_2 v} + H \cdot (1 - \sum_{v \in W'} Z_{i_2 v}) & i_1, i_2 \in T, i_1 < i_2, W' \in \text{clique}(G_{NO}) \\
& Z_{iv} \leq \sum_{\substack{v' \in W \\ NO_{vv'}=1}} Z_{(i-1)v'} & i \in T, i \neq 1, v \in W \\
& \sum_{v \in W} Z_{iv} \geq 1 & i \in T \\
& V_{iv}^t \leq \overline{V}_v^t \cdot Z_{iv} & i \in T, v \in W \\
& V_{iv}^t \geq \underline{V}_v^t \cdot Z_{iv} & i \in T, v \in W \\
& V_{iv}^t = \sum_{c \in C} V_{ivc} & i \in T, v \in W \\
& 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} & i \in T, r \in R, c \in C \\
& L_{ir}^t = \sum_{c \in C} L_{irc} & i \in T, r \in R \\
& \underline{FR}_v \cdot D_{iv} \leq V_{iv}^t \leq \overline{FR}_v \cdot D_{iv} & i \in T, v \in W \\
& \underline{x}_{vk} \cdot V_{iv}^t \leq \sum_{c \in C} x_{ck} V_{ivc} \leq \overline{x}_{vk} \cdot V_{iv}^t & i \in T, v \in W, k \in K \\
& \underline{L}_r^t \leq L_{ir}^t \leq \overline{L}_r^t & i \in T, r \in R_S \cup R_C \\
& 0 \leq L_{irc} \leq \overline{L}_r^t & i \in T, r \in R_S \cup R_C, c \in C \\
& \underline{L}_r^t \leq L_{0r}^t + \sum_{i \in T} \sum_{v \in I_r} V_{iv}^t - \sum_{i \in T} \sum_{v \in O_r} V_{iv}^t \leq \overline{L}_r^t & r \in R_S \cup R_C \\
& 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^t & r \in R_S \cup R_C, c \in C \\
& \underline{D}_r \leq \sum_{i \in T} \sum_{v \in O_r} V_{iv}^t \leq \overline{D}_r & r \in R_C \\
& S_{iv}, D_{iv}, E_{iv}, V_{iv}^t \geq 0 & i \in T, v \in W \\
& V_{ivc} \geq 0 & i \in T, v \in W, c \in C \\
& L_{ir}^t \geq 0 & i \in T, r \in R_S \cup R_C \\
& L_{irc} \geq 0 & i \in T, r \in R_S \cup R_C, c \in C \\
& Z_{iv} \in \{0, 1\} & i \in T, v \in W
\end{aligned}$$

Adhya, N., Tawarmalani, M., Sahinidis, N. V., 1999. A Lagrangian approach to the pooling problem. *Industrial and Engineering Chemistry Research* 38 (5), 1956–1972.

Alhajri, I., Elkamel, A., Albahri, T., Douglas, P. L., 2008. A nonlinear programming model for refinery planning and

- optimisation with rigorous process models and product quality specifications. *International Journal of Oil, Gas and Coal Technology* 1 (3), 283–307.
- Amor, H. B., Desrosiers, J., 2006. A proximal trust-region algorithm for column generation stabilization. *Computers and Chemical Engineering* 22 (4), 910–927.
- Audet, C., Brimberg, J., Hansen, P., Le Digabel, S., Mladenović, N., 2004. Pooling problem: Alternate formulations and solution methods. *Management Science* 50 (6), 761–776.
- Barahona, F., Anbil, R., 2000. The volume algorithm: producing primal solutions with a subgradient method. *Mathematical Programming* 87 (3), 385–399.
- Bazaraa, M. S., Sherali, H. D., 1981. On the choice of step size in subgradient optimization. *European Journal of Operational Research* 7 (4), 380–388.
- Bodington, C. E., Baker, T. E., 1990. A history of mathematical programming in the petroleum industry. *Interfaces* 20 (4), 117–127.
- Camerini, P. M., Fratta, L., Maffioli, F., 1975. On improving relaxation methods by modified gradient techniques. *Mathematical Programming Studies* 3, 26–34.
- Cheney, E. W., Goldstein, A. A., 1959. Newton's method for convex programming and Tchebycheff approximation. *Numerische Mathematik* 1 (1), 253–268.
- Dua, V., 2010. A mixed-integer programming approach for optimal configuration of artificial neural networks. *Chemical Engineering Research and Design* 88 (1), 55–60.
- Erdirik-Dogan, M. E., Grossmann, I. E., 2008. Simultaneous planning and scheduling of single-stage multi-product continuous plants with parallel lines. *Computers and Chemical Engineering* 32 (11), 2664–2683.
- Fisher, M. L., 1981. The Lagrangian relaxation method for solving integer programming problems. *Management Science* 27 (1), 1–18.
- Fisher, M. L., 1985. An application oriented guide to Lagrangian relaxation. *Interfaces* 15, 10–21.
- Floudas, C. A., Visweswaran, V., 1993. A primal-relaxed dual global optimization approach. *Journal of Optimization Theory and Applications* 78 (2), 187–225.
- Foulds, L. R., Haugland, D., Jörnsten, K., 1992. A bilinear approach to the pooling problem. *Optimization* 24 (1 & 2), 165–180.
- Frangioni, A., 2005. About Lagrangian methods in integer optimization. *Annals of Operations Research* 139 (1), 163–193.
- Goffin, J. L., Haurie, A., Vial, J. P., 1992. Decomposition and nondifferentiable optimization with the projective algorithm. *Management Science* 38 (2), 284–302.
- Gueddar, T., Dua, V., 2010. Novel model reduction techniques for refinery-wide energy optimization. *Applied Energy Journal*.
- Guignard, M., 2003. Lagrangean relaxation. *Top* 11 (2), 151–228.

- Held, M., Karp, R. M., 1971. The traveling-salesman problem and minimum spanning trees: Part ii. *Mathematical Programming* 1 (1), 6–25.
- Held, M., Karp, R. M., 1974. Validation of subgradient optimization. *Mathematical Programming* 6 (1), 62–88.
- Jia, Z., Ierapetritou, M. G., Kelly, J. D., 2003. Refinery short-term scheduling using continuous time formulation: Crude-oil operations. *Industrial and Engineering Chemistry Research* 42 (13), 3085–3097.
- Kelley, J. E., 1960. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics* 8 (4), 703–712.
- Kelly, J. D., Mann, J. L., 2003. Crude oil blend scheduling optimization: an application with multimillion dollar benefits. *Hydrocarbon Processing* 82 (6), 47–53.
- Lee, H., Pinto, J. M., Grossmann, I. E., Park, S., 1996. Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management. *Industrial and Engineering Chemistry Research* 35 (5), 1630–1641.
- Lemaréchal, C., 1974. An algorithm for minimizing convex functions. In: *Proceedings IFIP'74 Congress*. pp. 552–556.
- Li, W., Hui, C., Li, A., 2005. Integrating cdu, fcc and product blending models into refinery planning. *Computers and Chemical Engineering* 29 (9), 2010–2028.
- Li, Z., Ierapetritou, M. G., 2009. Production planning and scheduling integration through augmented Lagrangian optimization. *Computers and Chemical Engineering*.
- Maravelias, C. T., Sung, C., 2009. Integration of production planning and scheduling: Overview, challenges and opportunities. *Computers and Chemical Engineering* 33 (12), 1919–1930.
- Marsten, R. E., Hogan, W. W., Blankenship, J. W., 1975. The boxstep method for large-scale optimization. *Operations Research* 23 (3), 389–405.
- Misener, R., Floudas, C. A., 2009. Advances for the pooling problem: Modeling, global optimization, and computational studies. *Applied and Computational Mathematics* 8 (1), 3–22.
- Moro, L. F. L., Pinto, J. M., 2004. Mixed-integer programming approach for short-term crude oil scheduling. *Industrial and Engineering Chemistry Research* 43, 85–94.
- Mouret, S., Grossmann, I. E., Pestiaux, P., 2009. A novel priority-slot based continuous-time formulation for crude-oil scheduling problems. *Industrial and Engineering Chemistry Research* 48 (18), 8515–8528.
- Mouret, S., Grossmann, I. E., Pestiaux, P., 2010. Time representations and mathematical models for process scheduling problems. Submitted in *Computers and Chemical Engineering*.
- Neiro, S. M. S., Pinto, J. M., 2006. Lagrangean decomposition applied to multiperiod planning of petroleum refineries under uncertainty. *Latin American Applied Research* 36 (4), 213–220.
- Pinto, J. M., Moro, L. F. L., 2000. A planning model for petroleum refineries. *Brazilian Journal of Chemical Engineering* 17 (4-7), 575–586.

- Quesada, I., Grossmann, I. E., 1995. A global optimization algorithm for linear fractional and bilinear programs. *Journal of Global Optimization* 6 (1), 39–76.
- Robertson, G., Palazoglu, A., Romagnoli, J. A., 2010. Refinery scheduling of crude oil unloading, storing, and processing considering production level cost. In: 20th European Symposium on Computer Aided Process Engineering - ESCAPE20. Vol. 28 of *Computer Aided Chemical Engineering*. pp. 1159–1164.
- Zhang, J., Kim, N., Lasdon, L., 1985. An improved successive linear programming algorithm. *Management Science* 31 (10), 1312–1331.