# Dantzig-Wolfe and block coordinate-descent decomposition in large-scale integrated refinery-planning

Adebayo Alabi, Jordi Castro
Dept. of Statistics and Operations Research
Universitat Politècnica de Catalunya
Jordi Girona 1–3, 08034 Barcelona
adebayo.alabi@upc.edu, jordi.castro@upc.edu

Research Report DR 2008-01
January 2008, updated July 2008

Report available from http://www-eio.upc.es/~jcastro

# Dantzig-Wolfe and block coordinate-descent decomposition in large-scale integrated refinery-planning

Adebayo Alabi, Jordi Castro*
Department of Statistics and Operations Research
Universitat Politècnica de Catalunya, Jordi Girona 1–3
08034 Barcelona, Catalonia, Spain

### Abstract

The integrated refinery-planning (IRP), an instrumental problem in the petroleum industry, is made of several subsystems, each of them involving a large number of decisions. Despite the complexity of the overall planning problem, this work presents a mathematical model of the refinery operations characterized by complete horizontal integration of subsystems from crude oil purchase through to product distribution. This is the main contribution from a modelling point of view. The IRP, with a planning horizon ranging from 2 to 300 days, results in a large-scale linear programming (LP) problem of up to one million constraints, 2.5 million variables and 59 millions of nonzeroes in the constraint matrix. Large instances become computationally challenging for generic state-of-the-art LP solvers, such as CPLEX. To avoid this drawback, after the identification of the nonzero structure of the constraints matrix, structure-exploiting techniques such as Dantzig-Wolfe and block coordinate-descent decomposition were applied to IRP. It was also observed that interior-point methods are far more efficient than simplex ones in large IRP instances. These were the main contributions from the optimization viewpoint. A set of realistic instances were dealt with generic algorithms and these two decomposition methods. In particular the block coordinate-descent heuristic, with a reverse order of the subsystems, appeared as a promising approach for very large integrated refinery problems, obtaining either the optimal or an approximate feasible solution in all the instances tested.

**Keywords.** Planning, petroleum industry, large-scale linear programming, decomposition techniques.

## 1 Introduction

The primary goal of refiners in transforming crude oil into products is to maximize profit while keeping the pollution level within acceptable limits. A typical refinery operation involves a wide spectrum of activities, starting from crude oil purchase and transportation to refineries, refining, blending and transportation of products in demand to depots. The economics is extremely complex and heavily linked. Besides, the whole process is usually described by massive amount of operational data and decision-making processes. These situations, therefore, call for detailed planning over a specific period of time - typically one year. Often, the production plans are further broken down into feasible operations throughout time with detailed schedule of each activity and event in the refining operations. The scheduling horizons span from a few days to weeks, depending on information availability and uncertainty, and decisions are taken on hourly basis.

The use of mathematical programming in refinery operations spans well over half a century. Since the invention of the simplex algorithm in 1947 by G.B. Dantzig, linear programming (LP) has been routinely used in the petroleum industry not only for blending of gasoline, but also for configuration selection, capital investment analysis, long-range operations planning, supply and distribution planning. The limitations to

*Corresponding author

progress are the matrix size capacity of the computer, the time required to get a solution, and the accuracy of LP results which depend on the validity of input data. Refiners have therefore solved complex refinery-planning problems by decomposing into subproblems and, to date, this is reflected in the way some refineries operate its planning, central engineering, upstream operations, refining, supply and transportation.

In most cases, the integrated refinery-planning (IRP) problem is decomposed into three subproblems: crude oil supply, refining and blending, and product distribution. As it will be reviewed in next paragraphs, most of previous approaches dealt with these three subsystems independently, although few of them also considered the integrated problem. The main contribution from a modelling point of view is that, unlike earlier publications, all the three subproblems are modeled in detail and effectively integrated, i.e., several groups of refineries are considered, instead of a single one. From an optimization viewpoint, the main contribution is an extensive computational implementation and experimentation with large-scale realistic instances applying generic algorithms and two decomposition approaches: Dantzig-Wolfe [10] and block coordinate-descent heuristic [20]. As far as we know, this is the first attempt to apply the block coordinate-descent method in this context, although it was applied in other production planning problems. It was observed that interior-point methods are far more efficient than simplex ones in large IRP instances. However, the largest instances were challenging for generic state-of-the-art LP solvers (such as CPLEX) even in powerful computers. Using the reverse order of subsystems, the block coordinate-descent heuristic appeared as a promising approach for very large integrated refinery subsystems, obtaining either the optimal or an approximate feasible solution in all the instances tested.

Considerable research efforts have been directed towards the modeling of each subsystem. The planning and scheduling of exploration and production of crude oil in oil fields were formulated as a Mixed-Integer Linear Programming (MILP) problem [15], Mixed-Integer Nonlinear Programming (MINLP) problem [17], or a Nonlinear Programming (NP) problem [39, 40]. Meister, Clark and Shah [24] described the optimal selection of information-gathering process during the oil exploration phase which, at the same time, permits the calculation of optimal operating policies. Daily operations of crude oil shipment from oil fields to storage tanks in the wharf, unloading, transfer to storage tanks and inventory management are described as MILP problem in Lee et al. [19], Maglhães and Shah [22] and Neiro and Pinto [29]. The MILP problem describing the crude oil supply subproblem was further decomposed into two MILP problems in Más and Pinto [23].

In the area of refining and blending subsystem, examples of problems presented in literature are: NP model of diesel production planning [26], production scheduling of several processes such as fuel oil, asphalt and liquefied petroleum gas production [32], a NP model for obtaining increased gasoline yield in a group of crude distillation units (CDUs) in the context of reducing energy requirements [30], and an LP model for multi-period planning of oil refinery [33]. Recently, Méndez et al. [25] presented an MILP-based method that addressed the simultaneous optimization of the off-line blending and short-term scheduling problem in oil-refinery applications.

In the distribution subsystem, Van der Bruggen, Gruson and Salomon [38] presented a hierarchal approach for the operations related to the delivery of gasoline and diesel oil between depots and clients while Persson and Göthe-Lundgren [31] suggested an optimization model and a solution method for a shipment planning problem. Rejowski and Pinto [34] presented a model composed of petroleum refinery, one multi-product pipeline and several depots that were connected to local consumer markets.

The trend is not only in integrating the three subproblems but expanding the refining and blending processes to include a group of refineries instead of one, and the inclusion of discrete and binary variables—continuous and batch operations in multi-period planning [27]. Wenkai et al. [41] presented a solution algorithm that iteratively solved two mixed-integer programming (MIP) problems for short-term scheduling of crude oil unloading, storage, and processing with multiple oil types, multiple berths, and multiple processing units. Buchanan et al. [5] developed a system capable of simultaneously considering multiple refineries, multiple time periods, transportation of finished products to regional terminals, export and import options, and capital investments.

In summary, subsystems of a refinery network with considerable simplifications have been studied at reasonable level of details [29]. Obviously the idea of modeling the refinery network and subsequently solving the subsystems sequentially does not guarantee the globally optimal operation of the system, because the several problem features were not considered in a single and integrated model [23, 14]. In one of the leading multinational oil company operating today, different decisions are supported by sets of different tools and work is done continuously to integrate those tools to support the entire decision process [18]. Jia and Ierapetritou [16] successfully solved each of the subproblems and the integration of all the three subproblems is work in progress. Escudero, Quintana and Salmerón [11] presented a modeling framework for a real-life and stochastic supply, transformation and distribution scheduling problem for the oil industry, without attempting the solution of the resulting large-scale formulation. Unlike those approaches, this work presents a detailed integrated model, which is successfully solved by two decomposition methods.

The paper is organized as follows. Section 2 introduces the IRP problem and its general LP formulation. Section 3 describes the set of realistic test cases considered and the nonzero structure of the constraints matrix of IRP. Section 4 proposes the Dantzig-Wolfe and block coordinate-descent decomposition as solution methods for IRP. Finally, the numerical results of Section 5 show the effectiveness of the proposed model and solution methods.

## 2 Integrated refinery-planning problem

The system configuration of this planning problem is illustrated in Fig. 1. The system consists of a set of crude oils and corresponding crude oil tankers that arrive in the refinery docking station and sends crude oil to a set storage tanks from where crude oil is transferred to a set of charging tanks. In the charging tanks, different crude oils are mixed according to component compositions and transfered to CDUs in a set of refineries. In the CDUs different crude mixes are separated in processing units into different oil fractions as illustrated in Fig. 2. These fractions are further processed or blended to obtain other products that are stored or transported to a set of depots. Given necessary data, such as the cost of raw materials, physical properties of materials, capacity of processing units, demand of products, initial inventory level and capacity of tanks, the problem is to establish inventory levels in each tank, material flow rates between processing units, amount of final products that should be made, revenue from all final product, and cost of all purchased materials that translate to the maximum profit that can be made during a given planning horizon. The operating rules that have to be obeyed are presented in subsection 2.1.

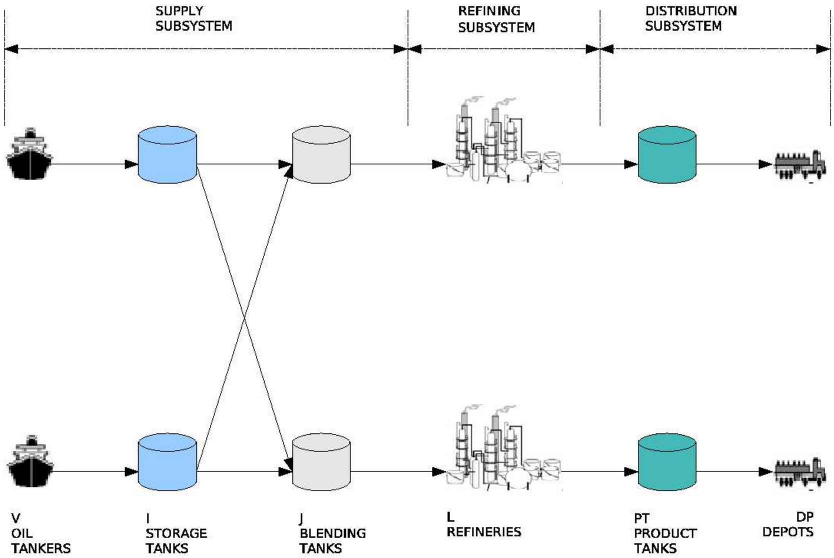

Figure 1: Graphical overview of the integrated subsystems

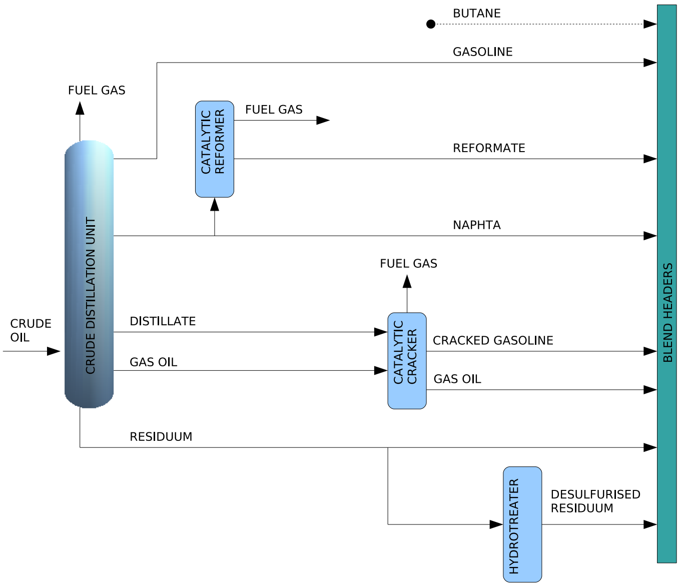

Figure 2: Illustration of the refinery subsystem

## 2.1 Mathematical model

For simplicity, it is assumed that: (a) only mass balance of materials in the tanks and processing units and inventory management of the tanks in the integrated system are included in the model; energy balance of processing units is not considered; (b) only continuous variables that are linear are considered; decisions that involve discrete variables like deciding which processing unit serves a given processing unit are determined at the beginning and so are taken as parameters–for instance, each oil tanker delivers a specific crude oil to an assigned storage tank; (c) the LP problem is deterministic within the planning horizon; (d) only specific key component in crude or blended oil fix the property of crude and blended oil; (e) the flow rate of materials is constant during the allocated time slot; (f) the mathematical formulations are based on discrete time domain allowing processing tasks to take place at certain time points. Even with these assumptions, the resulting LP problem is very large. The indices, sets, parameters, and variables used to model the IRP problem are listed and defined in the Appendix. Note that, although the same name was used to refer to different groups of variables, they can be distinguished by different subscripts (both in number and meaning). There is no ambiguity since each index iterates over a different set of values (e.g., crude oil type, depot, storage tank, blending tank, etc.). This license in notation was taken to improve the readability of the model.

### 2.1.1 Crude oil-supply model

The model presented in this section is taken from Song et al. [35]. The onset is the unloading of crude oil from crude oil tankers to specified storage tanks. The flow rate, cumulative flow rate of a given crude oil from corresponding crude oil tanker to storage tanks, the total volume and cost of crude oil consumed at a given time interval and during the planning horizon are defined in (1)–(8).

$$CF_{c,i,t} = CF_{c,i,t-1} + F_{c,i,t} \qquad \forall (c,i) \in CI, t \in T \mid t \neq 1 \tag{1}$$

$$CF_{k,i,t} = CF_{k,i,t-1} + F_{k,i,t} \qquad \forall k \in K, i \in I, t \in T \mid t \neq 1 \tag{2}$$

$$F_{k,i,t} = \sum_{c \in C \mid (c,i) \in CI} F_{c,i,t} \cdot \frac{\rho_c}{\rho_k} \cdot \phi_{k,c} \qquad \forall k \in K, i \in I, t \in T \tag{3}$$

$$C_t^{CR} = \sum_{c \in C} V_{c,t} \cdot cost_{c,t} \qquad \forall t \in T \tag{4}$$

$$V_c = \sum_{t \in T} F_{c,i,t} \qquad \forall (c,i) \in CI \tag{5}$$

$$V_{c,t} = F_{c,i,t} \qquad \forall (c,i) \in CI, t \in T \tag{6}$$

$$V_t^{CR} = \sum_{(c,i) \in CI} F_{c,i,t} \qquad \forall t \in T \tag{7}$$

$$V^{CR} = \sum_{(c,i) \in CI} \sum_{t \in T} F_{c,i,t} \tag{8}$$

From a given storage tank, crude oil is sent to a group of assigned blending tanks where different crude oils are mixed according to the concentration of sulfur in the crude oils. The flow rates and cumulative flow rates of crude oil from the storage tanks to blending tanks are defined in (9)–(11). Constraints (12)–(17) correspond to the volume and inventory costs of crude oil in the storage tanks and blending tanks; maximal/minimal volume of crude in these tanks are written in (18) to (21). Constraints (22) impose the mass balance of crude oil in and out of storage tank.

$$F_{k,i,j,t} = \sum_{c \in C | (c,i) \in CI} F_{c,i,j,t} \cdot \frac{\rho_c}{\rho_k} \cdot \phi_{k,c} \qquad \forall k \in K, i \in I, j \in J, t \in T \tag{9}$$

$$CF_{c,i,j,t} = CF_{c,i,j,t-1} + F_{c,i,j,t} \qquad \forall (c,i) \in CI, j \in J, t \in T \mid t \neq 1 \tag{10}$$

$$CF_{k,i,j,t} = CF_{k,i,j,t-1} + F_{k,i,j,t} \qquad \forall k \in K, i \in I, j \in J, t \in T \mid t \neq 1 \tag{11}$$

$$CINV_t^{ST} = \sum_{i \in I} cinv_i \cdot V_{i,t} \qquad \forall t \in T \tag{12}$$

$$CINV_t^{BT} = \sum_{j \in J} cinv\_j_j \cdot V_{j,t} \qquad \forall t \in T \tag{13}$$

$$V_{i,t} = \sum_{c \in C : (c,i) \in CI} V_{c,i,t} \qquad \forall i \in I, t \in T \tag{14}$$

$$V_{k,i,t} = \sum_{c \in C | (c,i) \in CI} V_{c,i,t} \cdot \frac{\rho_c}{\rho_k} \cdot \phi_{k,c} \forall k \in K, i \in I, t \in T \tag{15}$$

$$V_{j,t} = \sum_{c \in C} V_{c,j,t} \qquad \forall j \in J, t \in T \tag{16}$$

$$V_{k,j,t} = \sum_{c \in C} V_{c,j,t} \cdot \frac{\rho_c}{\rho_k} \cdot \phi_{k,c} \qquad \forall k \in K, j \in J, t \in T \tag{17}$$

$$\sum_{c \in C | (c,i) \in CI} V_{c,i,t} \leq v_i^{max} \qquad \forall i \in I, t \in T \tag{18}$$

$$\sum_{c \in C | (c,i) \in CI} V_{c,i,t} \geq v_i^{min} \qquad \forall i \in I, t \in T \tag{19}$$

$$\sum_{c \in C} V_{c,j,t} \leq v_j^{max} \qquad \forall j \in J, t \in T \tag{20}$$

$$\sum_{c \in C} V_{c,j,t} \geq v_j^{min} \qquad \forall j \in J, t \in T \tag{21}$$

$$V_{c,i,t} = v0_{c,i} + \sum_{c \in C | (c,i) \in CI} \sum_{t \in T} F_{c,i,t} \quad - \quad \sum_{j \in J} \sum_{t \in T} F_{c,i,j,t} \qquad \forall (c,i) \in CI, t \in T \tag{22}$$

### 2.1.2 Refining process model

The constraints in this subsystem are adapted from Aronofsky, Dutton and Tayyabkhan [2]. Crude blend from a given blending tank is sent to an assigned CDU in the refinery where the crude blend is separated into different intermediate fractions. Some of these intermediate fractions are further processed and blended in processing units. The processing units in the refinery are catalytic reformer (CR), catalytic cracker (CC), hydro treater (HT), and blend header (BH). Use of these processing units should not exceed specified capacity (23). Besides, the demand of blended crude by all processing units must be covered by the amount flowing from all the blending tanks (24) and the amount of intermediate fraction must cover the amount required for blending to make the products (25).

$$\sum_{pr \in PR} util_{u,pr} \cdot \sum_{c \in C} LEV^{PR}_{pr,c,t} \leq cap_u \qquad \forall u \in U, t \in T \tag{23}$$

$$\sum_{pr \in PR} io_{mr,c,pr} \cdot LEV^{PR}_{pr,c,t} + F_{c,j,l,t} \geq 0 \qquad \forall mr \in MR, c \in C, (j,l) \in JL, t \in T \tag{24}$$

$$\sum_{pr \in PR} io_{mi,c,pr} \cdot LEV^{PR}_{pr,c,t} + (F_{mi,c,t} \mid mi \in MP) \geq \sum_{(p,mi) \in BL} LEV^{BL}_{p,mi,c,t} \qquad \forall mi \in MI, c \in C, t \in T \tag{25}$$

Constraints (26)–(28) define the volume of product that must equal the volume of all intermediate fractions blended to make the product, and the minimum/maximum quality attributes of the products that must be met. The cost of all intermediate fractions purchased from a third party and the total cost of operating all the operating units are computed in (29)–(30).

$$F_{p,pt,t} = \sum_{(p,mi) \in BL} \sum_{c \in C} LEV^{BL}_{p,mi,c,t} \qquad \forall (p,pt) \in PPT, t \in T \tag{26}$$

$$\sum_{mi \in MI | (p,mi) \in BL} \sum_{c \in C} attr_{mi,c,q} \cdot LEV^{BL}_{p,mi,c,t} \geq attr^{min}_{p,q} \cdot F_{p,pt,t} \qquad \forall (p,pt) \in PPT, t \in T, q \in Q \mid attr^{min}_{p,q} \neq 0 \tag{27}$$

$$\sum_{mi \in MI | (p,mi) \in BL} \sum_{c \in C} attr_{mi,c,q} \cdot LEV^{BL}_{p,mi,c,t} \leq attr^{max}_{p,q} \cdot F_{p,pt,t} \qquad \forall (p,pt) \in PPT, t \in T, q \in Q \mid attr^{max}_{p,q} \neq 0 \tag{28}$$

$$C^{PUR}_t = \sum_{mp \in MP} \sum_{c \in C} cost_{mp,t} \cdot F_{mp,c,t} \qquad \forall (j,l) \in JL, t \in T \tag{29}$$

$$C^{OP}_t = \sum_{pr \in PR} (cost_{pr} \cdot \sum_{c \in C} LEV^{PR}_{pr,c,t}) \qquad \forall t \in T \tag{30}$$

### 2.1.3 Product distribution model

The final products from the blend headers are stored in corresponding product tanks from where the products are transported to depots. In this subsystem, the constraints are simply defined as follows: the volume of products in the product tanks should fall within the specified minimum/maximum capacity of the product tanks—constraints(32)–(33), and the total inventory cost of the product tanks is defined in (31). The total demand of a given product at a given depot should be satisfied by the flow of the product from all the product tanks—constraints (34)–(35). The cost of transportation of products to depots and the total revenue from all products are defined by (36) and (37) respectively. The material balance constraint for a given product in a given product tank is written in (38).

$$CINV^{PT}_t = \sum_{(p,pt) \in PPT} V_{p,pt,t} \cdot cinv_{pt} \qquad \forall t \in T \tag{31}$$

$$\sum_{p \in P | (p,pt) \in PPT} V_{p,pt,t} \leq v_{pt}^{max} \qquad \forall pt \in PT, t \in T \tag{32}$$

$$\sum_{p \in P | (p,pt) \in PPT} V_{p,pt,t} \geq v_{pt}^{min} \qquad \forall pt \in PT, t \in T \tag{33}$$

$$DEM_{p,pt,t} = \sum_{dp \in DP | (p,pt,dp) \in PPTDP} F_{p,pt,dp,t} \qquad \forall (p, pt) \in PPT, t \in T \tag{34}$$

$$F_{p,pt,dp,t} = dem_{p,pt,dp,t} \qquad \forall (p, pt, dp) \in PPTDP, t \in T \tag{35}$$

$$C_t^{TR} = \sum_{(p,pt,dp) \in PPTDP} F_{p,pt,dp,t} \cdot ct_{p,pt,dp} \qquad \forall t \in T \tag{36}$$

$$REV_t = \sum_{(p,pt,dp) \in PPTDP} \sum_{t \in T} rev_{p,t} \cdot F_{p,pt,dp,t} \qquad \forall t \in T \tag{37}$$

$$V_{p,pt,t} = v0_{p,pt} + F_{p,pt,t} - DEM_{p,pt,t} \qquad \forall (p, pt) \in PPT, t \in T \tag{38}$$

### 2.1.4 Linking constraints

Material balance around units that are situated at the border of two subsystems generates linking constraints. The volume of crude in blending tanks and the flow rates of the material from a given blending tank to a specified CDU in the refinery are given by (39)–(41) and the material balance for crude in a particular blending tank corresponds to (42).

$$CF_{c,j,l,t} = CF_{c,j,l,t-1} + F_{c,j,l,t} \qquad \forall c \in C, (j, l) \in JL, t \in T \mid t \neq 1 \tag{39}$$

$$CF_{k,j,l,t} = CF_{k,j,l,t-1} + F_{k,j,l,t} \qquad \forall k \in K, (j, l) \in JL, t \in T \mid t \neq 1 \tag{40}$$

$$F_{k,j,l,t} = \sum_{c \in C} F_{c,j,l,t} \cdot \frac{\rho_c}{\rho_k} \cdot \phi_{k,c} \forall k \in K, (j, l) \in JL, t \in T \tag{41}$$

$$V_{c,j,t} = v0_{c,j} + \sum_{i \in I | (c,i) \in CI} \sum_{t \in T} F_{c,i,j,t} - \sum_{l \in L | (j,l) \in JL} \sum_{t \in T} F_{c,j,l,t} \qquad \forall c \in C, j \in J, t \in T \tag{42}$$

### 2.1.5 Objective function

The objective is to minimize costs and maximize profit over the entire planning horizon. It is formulated as follows:

$$\begin{aligned} \min \quad & \sum_{t \in T} C_t^{OP} + \sum_{t \in T} C_t^{CR} + \sum_{t \in T} CINV_t^{ST} + \sum_{t \in T} CINV_t^{BT} + \\ & \sum_{t \in T} C_t^{PUR} + \sum_{t \in T} CINV_t^{PT} + \sum_{t \in T} C_t^{TR} - \sum_{t \in T} REV_t + \end{aligned} \tag{43}$$

## 3 Case studies and problem structure

The IRP model described by (1)–(43) was implemented using the AMPL modeling language [1]. A suite of seven IRP instances with planning horizons $T$ ranging from $T = 2$ to $T = 300$ days were considered. The system consists of 10 crude oil and crude tankers, 10 storage tanks, 5 blending tanks, a network of 4 refineries, 5 final products, and 60 depots. The realistic input data were taken from Song et al. [35], Aronofsky, Dutton and Tayyabkhan [2] and U.S. Department of Energy [37]. Following the deterministic assumption (c) of Subsection 2.1, some data that are stochastic in nature were generated randomly. For instance, the lowest

Table 1: Dimensions for the IRP problem suite

| $T$ in days | $m$ | $n$ | $nz$ | $m'$ | $n'$ | $nz'$ |
|---|---|---|---|---|---|---|
| 2 | 7211 | 13697 | 45389 | 3012 | 5798 | 24996 |
| 3 | 10883 | 20540 | 70049 | 4838 | 9022 | 40374 |
| 60 | 220187 | 410591 | 3409679 | 97520 | 181390 | 2809890 |
| 120 | 440507 | 821171 | 11031779 | 195080 | 362830 | 9831870 |
| 180 | 660827 | 1231751 | 22865879 | 281840 | 533470 | 21044250 |
| 240 | 881147 | 1642331 | 38911979 | 375800 | 711310 | 36483030 |
| 300 | 1101467 | 2052911 | 59170079 | 469760 | 889150 | 56133810 |

and highest cost of “Arabian light” crude oil from January to December 2005 was 33.80 and 60.32 \$/bbl respectively [37]; the daily prices of this crude oil were randomly generated within this range of values. Dimensions for these seven instances are reported in Table 1. Columns $m$, $n$ and $nz$ show respectively the number of constraints, columns and nonzeroes of the constraints matrix. Columns $m'$, $n'$ and $nz'$ show dimensions after applying the problem presolving of the generic solver CPLEX. Although it significantly reduced the size of the model, the memory requirements were still too large for standard computers and 32 bits version of CPLEX. For instance, the interior-point approach (the most efficient one, as it will be shown in Section 5) required more than 4, 7 and 10 GB of memory for the three largest instances, respectively. MPS files for the seven instances are available from `http://www-eio.upc.es/~jcastro/srd_data.html`.

It is worth noting that many constraints of the model of Section 2 define a positive variable as a sum of other positive variables, and thus it would be possible to substitute them, reducing the number of columns and rows of constraints matrix. However, this was not done for three reasons. First, if necessary, this substitution is automatically performed by the preprocessing of the solver, as shown in Table 1. Second, substitutions would hinder the readability of the current model. And third, for the interior-point solver, which as shown in Section 5 is more efficient than the simplex method in this problem, it is not clear whether these substitutions are helpful. Indeed, the cost of an interior-point iteration is not related with the number of constraints and/or variables, but with the density of $AA^T$, $A$ being the constraint matrix. This density may increase if substitutions of variables are performed. Indeed, in other block angular problems (e.g. [21]), it was shown that the addition of new redundant variables and constraints improved the performance of an interior-point solver.

As shown in Section 5, a 32 bits version of CPLEX-9.1 could only solve the first four instances and failed in the remaining three by memory limitations on a SUN Fire V20Z server with two AMD Opteron 250 2.46 GHZ processors—without exploitation of parallelism capabilities—and 8 GB of RAM. The three largest instances needed a 64 bits version of CPLEX-11 on a Dell PowerEdge 6950 server with four dual core AMD Opteron 8222 3.0 GHZ processors —without exploitation of parallelism capabilities—and 64 GB of RAM. As far as we know, such large IRP instances have not been tried before in the literature. Because of the difficulty of solving real large-scale IRP instances by state-of-the-art generic solvers, even with a well equipped server, we exploited the constraints structure to apply some suitable decomposition procedure.

From the AMPL model, an MPS file format was generated for each instance. This results in an unstructured general LP problem

$$\begin{array}{rrcl} z^* = \min & cx & = & z \\ \text{s.t.} & Ax & \bowtie & b \\ & x & \geq & 0 \end{array} \tag{44}$$

of $m$ constraints and $n$ variables, the symbol $\bowtie$ representing an arbitrary set of inequality and equality symbols $(\leq, =, \geq)$. Using an in-house code that read the MPS file and extracted the IRP constraints

Table 2: Dimensions of IRP problem and subproblems, instance $T = 2$ days

| Problem | range rows | n. rows | range columns | n. columns |
|---|---|---|---|---|
| IRP problem (matrix $A$) | 1:7211 | 7211 | 1:13697 | 13697 |
| Supply subproblem (matrix $B_1$) | 1:1291 | 1291 | 1:1931 | 1931 |
| Refining subproblem (matrix $B_2$) | 1292:4919 | 3628 | 1932:11765 | 9834 |
| Distribution subproblem (matrix $B_3$) | 4920:6851 | 1932 | 11766:13697 | 1932 |
| Linking constraints (matrix $[A_1\ A_2\ A_3]$) | 6852:7211 | 360 | 1:13697 | 13697 |

structure of (44), the following formulation was obtained:

$$\begin{array}{rccccccc} z^* = \min & c_1x_1 & + & c_2x_2 & + & c_3x_3 & = & z \\ \text{s.t.} & B_1x_1 & & & & & \bowtie & b_1 \\ & & & B_2x_2 & & & \bowtie & b_2 \\ & & & & & B_3x_3 & \bowtie & b_3 \\ & A_1x_1 & + & A_2x_2 & + & A_3x_3 & \bowtie & b_0 \\ & x_1 & , & x_2 & , & x_3 & \geq & 0. \end{array} \tag{45}$$

Problem (45) shows a primal block-angular structure, each block $B_i \in \mathbb{R}^{m_i \times n_i}, i = 1, 2, 3$, related to respectively the crude supply, refining and distribution subsystems. Linking constraints defined by $A_i \in \mathbb{R}^{m_0 \times n_i}, i = 1, 2, 3$ correspond to equations (39)–(41). For instance, Figure 3 shows the structure of the constraints matrix for the smallest instance of $T = 2$ days. Submatrices $B_i, i = 1, 2, 3$, for the three subsystems of this particular instance are detailed in Figures 4, 5 and 6, respectively. Figure 7 shows the structure of the linking constraints matrix $[A_1\ A_2\ A_3]$. Table 2 summarizes the range of rows in matrix $A$ ("range rows"), number of rows ("n. rows"), range of columns in matrix $A$ ("range columns"), and number of columns ("n. columns") for the submatrices of instance $T = 2$. It can be observed that the number of linking constraints is small compared to constraints of $B_i, i = 1, 2, 3$. It is also observed that the three subsystems are loosely coupled, since, from Figure 7, linking constraints only involve variables in the border of subsystems 1 and 2, and 2 and 3.

# 4 Decomposition techniques

Primal block-angular problems such as (45) can be solved by several effective decomposition techniques (see Conejo et al. [8] for a comprehensive and practical description of them). In this work we focused on Dantzig-Wolfe and block coordinate-descent decomposition. This choice is justified by the relative small number of linking constraints of IRP. Dantzig-Wolfe is equivalent to a particular implementation of the Lagrangian relaxation algorithm; they only differ in the type of variables used in the master problem: dual variables in Lagrangian relaxation, primal ones in Dantzig-Wolfe decomposition. Unlike Dantzig-Wolfe, block coordinate-descent does not guarantee convergence to an optimum for linear problems. However, it was able to provide either and optimal or an approximate feasible solution in all the instances tested, resulting in the fastest approach. Some recent specialized interior-point decomposition approaches have not been considered because, either they have shown to be more efficient for problems with many (even nonlinear) subproblems [9], or they rely on some specific properties of the linking constraints [6, 7]. Attempting the solution of IRP by these recent interior-point approaches is, however, part of the future work to be done. We briefly outline Dantzig-Wolfe and block coordinate-descent decomposition in next two subsections.

## 4.1 Dantzig-Wolfe decomposition

Dantzig-Wolfe decomposition is a classic solution approach for structured models with linking constraints, such as (45), which can not be solved by the standard simplex algorithm due to its large dimension. Implementations of the Dantzig-Wolfe decomposition algorithm have been described in, among others, Ho and Loute [12, 13], Tebboth [36].

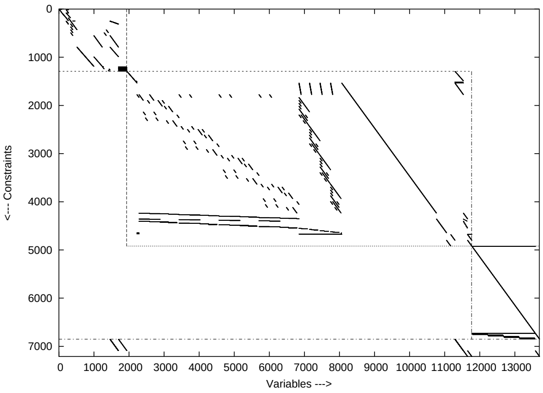

Figure 3: Nonzero structure of the constraints matrix of the integrated system ($T = 2$ days)

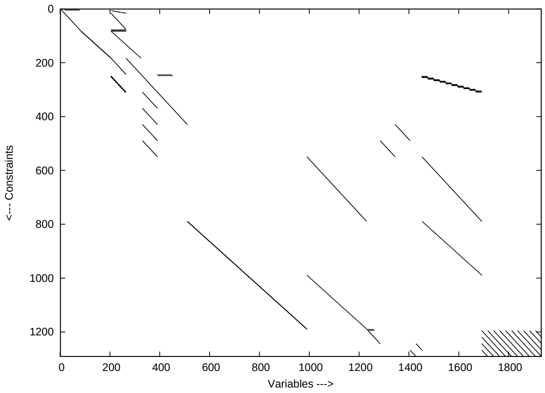

Figure 4: Nonzero structure of the constraints matrix of the supply subsystem ($T = 2$ days)

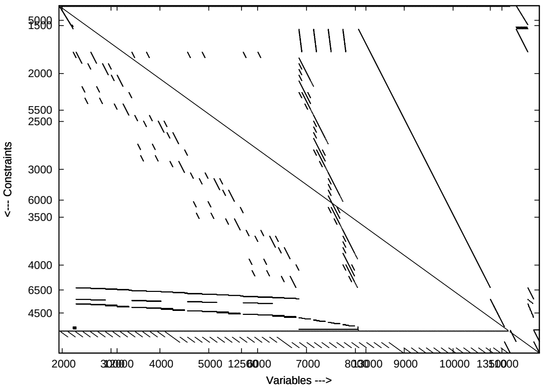

Figure 5: Nonzero structure of the constraints matrix of the refining subsystem ($T = 2$ days)

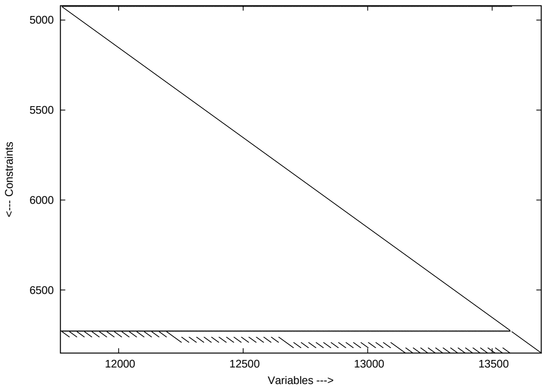

Figure 6: Nonzero structure of the constraints matrix of the distribution subsystem ($T = 2$ days)

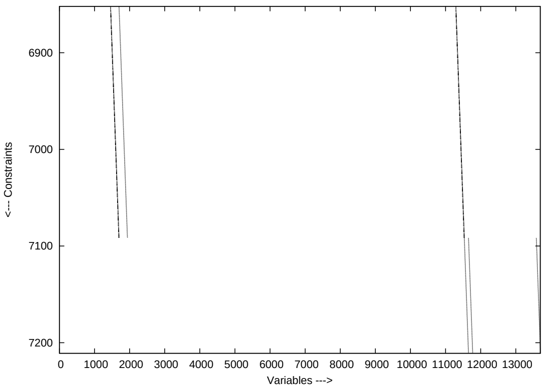

Figure 7: Nonzero structure of the constraints matrix of linking constraints ($T = 2$ days)

Briefly, the feasible polyhedron $P$ of (45) decomposes into three polyhedra $P_k,\ k = 1, \ldots, 3$, of smaller dimension representing each subsystem block and a polyhedron $P_0$ representing the linking constraints of the problem. Minkowski's representation theorem states that every point in a given polyhedron can be expressed as a linear combination of its extreme points, $\{x^i\}_{i \in I}$, and extreme rays, $\{r^j\}_{j \in J}$. For simplicity let $x^i$ represent both extreme points and extreme rays, where $i$ ranges over $E = I \cup J$ and let $\lambda_i$ represent the coefficient of $x^i$ in Minkowski's representation. The convexity constraint is written as $\sum_i \delta_i \lambda_i = 1$, where the constant $\delta_i$ is one if $x^i$ is an extreme point, and zero if $x^i$ is an extreme ray. Minkowski's representation of the three polyhedra $P_k,\ k = 1, \ldots, 3$, is therefore defined as: $P_k = \{x_k \geq 0 : x_k = \sum_{i_k \in E_k} (\lambda_k)_{i_k} (x_k^{i_k}); \sum_{i_k \in E_k} (\delta_k)_{i_k} (\lambda_k)_{i_k} = 1; (\lambda_k)_{i_k} \geq 0, i_k \in E_k\}$. Substituting the linear combination into (45), the master problem is then expressed as:

$$
\begin{array}{rcccccccl}
z^* = \min & \sum_{i_1 \in E_1} (c_1 x_1^{i_1})(\lambda_1)_{i_1} & + & \sum_{i_2 \in E_2} (c_2 x_2^{i_2})(\lambda_2)_{i_2} & + & \sum_{i_3 \in E_3} (c_3 x_3^{i_3})(\lambda_3)_{i_3} & = z \\
\text{s.t.} & \sum_{i_1 \in E_1} (\delta_1)_{i_1} (\lambda_1)_{i_1} & & & & & = 1 \\
& & & \sum_{i_2 \in E_2} (\delta_2)_{i_2} (\lambda_2)_{i_2} & & & = 1 \\
& & & & & \sum_{i_3 \in E_3} (\delta_3)_{i_3} (\lambda_3)_{i_3} & = 1 \\
& \sum_{i_1 \in E_1} (A_1 x_1^{i_1})(\lambda_1)_{i_1} & + & \sum_{i_2 \in E_2} (A_2 x_2^{i_2})(\lambda_2)_{i_2} & + & \sum_{i_3 \in E_3} (A_3 x_3^{i_3})(\lambda_3)_{i_3} & \bowtie b_0 \\
& (\lambda_1)_{i_1 \in E_1} & , & (\lambda_2)_{i_2 \in E_2} & , & (\lambda_3)_{i_3 \in E_3} & \geq 0,
\end{array}
\tag{46}
$$

The master problem and the original problem are equivalent. Although the number of rows is reduced in the new formulation, the number of columns is substantially increased. However, the method does not compute and store all these columns, but generates only very small subset of the more "attractive" columns of the master problem—restricted master problem—, as follows:

**Step 1.** Initialization: choose $E'_k \subseteq E_k$, $k = 1, \ldots, 3$, an attractive subset of columns leading to a basic feasible solution for the master problem. This could be done for IRP exploiting the particular information we have about the problem. Otherwise, it that was not possible, any standard procedure should be used: phase-I [4] or big-M [8]. Initialize the best known lower bound $\underline{z}^* = \infty$ of the unknown optimal objective value $z^*$.

**Step 2.** Solve the restricted master problem. Let $\mu \in \mathbb{R}^3$ and $\pi \in \mathbb{R}^{m_0}$ be respectively the dual variables for the convexity and linking constraints at the optimal solution, and $\bar{z}^*$ the objective function value (the best known upper bound of $z^*$).

If the restricted master problem is unbounded, then the problem is unbounded, and any unbounded ray identified in the restricted master is an unbounded ray of the problem. STOP.

**Step 3** Solve the pricing problem for column generation. Solve iteratively for $k = 1, 2, 3$ the subproblems

$$
\begin{array}{rrcl}
z_k = \min & (c_k - \pi A_k) x_k & = & z \\
\text{s.t.} & B_k x_k & \bowtie & b_k \\
& x_k & \geq & 0,
\end{array}
$$

until at least one basic feasible solution is found with $z_k < \mu_k$ or an unbounded extreme ray is found with $z_k = -\infty$. Any such solutions found correspond to $\lambda_k$ columns with negative reduced cost, and thus candidates to enter the basis of the current restricted master problem.

**Step 4.** Optimality test. If $z_k \geq \mu_k$ for each $k = 1, \ldots, 3$, then no column prices out negative to enter the basis of the restricted master problem. The current restricted master problem solution is an optimal solution to the problem. STOP.

**Step 5.** Near-optimality test. If the restricted master problem is feasible, and the pricing problems are all feasible and bounded, compute a new lower bound (see, e.g., Bertsimas and Tsitsiklis [4])

$$\underline{z} = \bar{z}^* + \sum_{k=1}^{3}(z_k - \mu_k).$$

If $\underline{z} > \underline{z}^*$, update the best known lower bound $\underline{z}^* = \bar{z}$.

Compute the absolute optimality gap $d_A = \bar{z}^* - \underline{z}^*$ and the relative optimality gap $d_R = d_A/(1+|\bar{z}^*|)$. If they are within the specified optimality tolerances, then the current restricted master problem solution solves the master problem to the requested tolerances. STOP.

**Step 6** Update the restricted master problem. As the optimality test failed, at least one column $(\lambda_k)_{i_k}$ was identified in Step 3 that is a candidate to enter the basis of the restricted master problem. Choose at least one to add to the current restricted master problem, i.e., update $E'_k$ by adding the chosen $i_k$.

Any current non-basic variable $(\lambda_k)_{i_k}$ of the restricted master problem with positive reduced costs may be removed, i.e., update $E'_k$ by removing the chosen $i_k$.

Return to Step 2 with the updated $E'_k$.

Since, as shown in Section 5, the number of iterations performed by the algorithm (i.e., added columns to the restricted master problem) was not excessive, no removing of inactive columns was performed at Step 6 above in the implementation developed.

## 4.2 Block coordinate-descent decomposition

The block coordinate-descent method optimizes the objective function with respect to columns in a given subproblem or block, while the columns in the remaining block(s) are kept constant. This is iteratively repeated until the difference between subsequent and current objective function is less than a specified optimality tolerance. This algorithm is appropriate in our situation because subsystems are loosely coupled by a few variables in the linking constraints. The convergence of this algorithm to a stationary point of a convex optimization problem is guaranteed if the minimum for each subproblem (associated to each subsystem, see step 2 of the algorithm below) is uniquely attained [3, Prop. 2.7.1]. Strict convexity of the objective function guarantees such uniqueness, thus it is sufficient, but not necessary condition. On the other hand, subproblems of linear objectives may have alternative solutions, and thus convergence to an optimal solution is not guaranteed. However, it usually behaves properly in practical applications [8].

For strictly convex functions, the order of blocks is irrelevant, since convergence to the unique optimizer is guaranteed. For linear functions, in principle, there is no rule, and one can start by minimizing the objective function with respect to columns of any given subproblem. However, for all the instances of the IRP problem considered in this work, the block coordinate-descent method failed to converge when the objective function was first minimized with respect to columns in the crude supply subproblem (subproblem 1 in (45)) keeping constant the columns in the refining and distribution subproblems (subproblems 2 and 3 in (45), respectively). After a careful analysis, it was observed that most parameters in the IRP problem occur in the distribution subsystem. Parameters such as demand for products in the distribution subsystem act as the driving force that determines the value of most of the variables in the other subsystems. Therefore, the algorithm was implemented by first minimizing the objective function with respect to the distribution variables, next with respect to the refining ones, and finally with respect the crude supply decisions. With this subsystem reverse order, block coordinate-descent was able to provide the optimal or an approximate feasible solution solution for all the instances. The algorithm implemented is outlined below:

**Step 1.** Initialization: let $\nu = 0$; assign arbitrary values of the optimal solution $x_k^{*(\nu)}$, for $k = 1, 2$.

**Step 2.** Let $\nu = \nu + 1$. Solve iteratively for $k = 3, 2, 1$, the subproblems:

$$\begin{array}{rrcl} x_k^{*(\nu)} = \arg\min & c_k x_k & & \\ \text{s.t.} & B_k x_k & \bowtie & b_k \\ & A_k x_k & \bowtie & b_0 - b_k^{'} \\ & x_k & \geq & 0 \end{array}$$

where

$$b_k^{'} = \begin{cases} A_1 x_1^{*(\nu-1)} + A_2 x_2^{*(\nu-1)} & \text{if } k = 3; \\ A_1 x_1^{*(\nu-1)} + A_3 x_3^{*(\nu)} & \text{if } k = 2; \\ A_2 x_2^{*(\nu)} + A_3 x_3^{*(\nu)} & \text{if } k = 1. \end{cases}$$

**Step 3.** Compute the objective function of the primal block angular LP problem for the current iteration, $z^{*(\nu)} = \sum_{k=1}^{3} c_k x_k^{*(\nu)}$

**Step 4.** Compute the optimality gap $|z^{*(\nu)} - z^{*(\nu-1)}|$. If the value is less than a specified optimality tolerance, then the current solution is reported as the solution of the LP problem. STOP.

Otherwise, return to Step 2.

Note that safeguards for infeasible subproblems were not needed: since the number of variables shared (thus fixed) between subproblems is small, subproblems were not over-constrained and, therefore, a feasible solution was always obtained for all the instances tested.

It is worth noting that the sequential solution of subsystems, instead of the solution of the integrated model, is equivalent to applying just one iteration of the above algorithm. From the computational results of Section 5 it was observed that block coordinate-descent stops after many iterations (from 10 to 32) in large instances, and this number increases with the size of the problem. Thus, it outperforms the simple sequential solution of subsystems. In addition, as stated above, it was found that the order of subsystems in the sequence is instrumental. The block coordinate-descent approach in this work uses a reverse order. If subsystems are solved in forward order they may lead to infeasible or poor solutions.

# 5 Computational results

Both block coordinate-descent and Dantzig-Wolfe decomposition algorithms, as described in previous section, were implemented in Matlab. Subproblems were solved with both the interior-point and dual simplex CPLEX options. These two generic algorithms, barrier and dual simplex, without exploiting the structure of the problem, were also considered. This amounts to six solution approaches for each IRP instance. Table 3 shows the results obtained using the generic barrier (columns "IP") and dual simplex (columns "dual simplex") of CPLEX. For each algorithm we provide the number of iterations and CPU time in seconds. The optimal objective function found is given in column $f^*$. Runs for the smallest four instances were performed with a 32 bits versions of CPLEX-9.1 on a "standard" computing environment: a SUN Fire V20Z server with two AMD Opteron 250 2.46 GHZ processors and 8 GB of RAM (without exploiting the parallelism capabilities). The three largest instances could not be solved in that environment by memory limitations, e.g., they required, respectively, more than 4, 7 and 10 GB of RAM with the interior-point algorithm. For these instances we used a 64 bits version of CPLEX-11 on a Dell PowerEdge 6950 server with four dual core AMD Opteron 8222 3.0 GHZ processors —without exploitation of parallelism capabilities—and 64 GB of RAM. Processors of both machines are similar in performance, and we performed no transformation of CPU time. Clearly, the interior-point algorithm outperformed the dual simplex. For instance, for the largest instance the CPLEX interior-point algorithm took 215531 seconds to find the optimal solution, whereas the dual simplex, after more than 842000 seconds (about 10 days of CPU), was still very far from optimality. This is partly explained by the high degeneracy of this problem. In particular, the objective function did not change after the first

Table 3: Results with CPLEX for the IRP problem suite

| Instance | dual simplex | | IP | | |
|---|---|---|---|---|---|
| $T$ (days) | iter. | CPU (sec.) | iter. | CPU (sec.) | $f^*$ |
| 2 | 4933 | 0.4 | 28 | 0.5 | 146320.4 |
| 3 | 9070 | 1.8 | 26 | 0.9 | 290699.5 |
| 60 | 383629 | 2834.8 | 28 | 1798.8 | 4232012.8 |
| 120 | 844162 | 37337.5 | 30 | 14689.9 | 8655432.5 |
| 180 | 1388429 | 235214.0 | 29 | 37432.1 | 12625528.7 |
| 240 | > 1081000 † | > 878081.3 † | 31 | 99907.6 | 16731729.3 |
| 300 | > 540000 ‡ | > 842000.3 ‡ | 35 | 215531.9 | 20888039.6 |

† stopped very far from optimality, with an objective value of 3887976.3

‡ stopped very far from optimality, with an objective value of -4596190.7

Table 4: Results with decomposition approaches for the IRP problem suite

| Instance | IP | BCD simplex | BCD IP | DW simplex | DW IP |
|---|---|---|---|---|---|
| $T$ (days) | CPU (sec.)<br>iter. | CPU (sec.)<br>iter. | CPU (sec.)<br>iter. | CPU (sec.)<br>iter. | CPU (sec.)<br>iter. |
| 2 | 0.3<br>16 | 1.2<br>3 | 0.8<br>2 | 7.5<br>7 | 5.4<br>5 |
| 3 | 0.9<br>26 | 3.8<br>12 | 1.3<br>10 | 9.7<br>170 | 8.2<br>91 |
| 60 | 1798.8<br>28 | 3569.5<br>15 | 1787.9<br>12 | 5758.9<br>182 | 3183.9<br>154 |
| 120 | 8814.0<br>18 | 22903.6<br>22 | 14596.4<br>18 | 25783.0<br>194 | 22440.7<br>169 |
| 180 | 24523.8<br>19 | 26487.8<br>28 | 18701.2<br>20 | 31783.6<br>205 | 28966.2<br>182 |
| 240 | 58010.5<br>18 | 32543.1<br>30 | 23194.8<br>27 | 33981.6<br>209 | 29255.9<br>193 |
| 300 | 92370.4<br>15 | 34009.5<br>35 | 26369.5<br>32 | 39128.9<br>219 | 31014.6<br>205 |

100 iterations using the dual simplex method. This forced CPLEX to perturb the problem coefficients after that iteration. However, even with that perturbation, the number of degenerate steps was significant. For instance, for the four smallest instances the percentage of degenerate steps was, respectively of 17.59%, 17.04%, 14.22% and 21.24%.

Table 4 reports the results obtained with the decomposition approaches. Results are provided for block coordinate-descent decomposition using either dual simplex or interior-point algorithm for the subproblems (columns "BCD simplex" and "BCD IP"), Dantzig-Wolfe decomposition using either dual simplex or interior-point algorithm for the subproblems and restricted master (columns "DW simplex" and "DW IP"), and for the interior-point algorithm applied to the whole LP problem (column "IP") which was the most efficient generic algorithm according to Table 3. For each (instance, solution approach) pair, Table 4 shows two values: the CPU time in seconds (top entry) and the number of iterations (entry "iter."). For column "IP", entry "iter" provides the number of interior-point iterations; for columns "BCD simplex" and "BCD IP", entry "iter" gives the number of iterations of the block coordinate-descent algorithm (i.e., the number of times Step 2 of the algorithm of Subsection 4.2 is performed); and for columns "DW simplex" and "DW IP", entry "iter" gives the number of iterations of the Dantzig-Wolfe method (i.e., the number of times Step 2 of the algorithm of Subsection 4.1 is executed). The CPU time reported for the decomposition algorithms corresponds to the CPU time needed by CPLEX for the solution of all the LP subproblems, including the significant overhead due to pass of parameters between Matlab and CPLEX, but not the time spent in the Matlab language (which is meaningless, since it is an interpreted language).

Block coordinate-descent provided the optimal solution for the three smaller instances, while for the four largest ones it stopped in suboptimal feasible solutions of objective values 11.8e+6, 17.9e+6, 36.0e+6 and 48.2e+6, respectively. However it was the fastest method to reach such a feasible solutions. Therefore, for a fair comparison, Table 4 provides the CPU time needed by the several algorithmic approaches to reach such feasible solutions. The block coordinate-descent was always superior to Dantzig-Wolfe. Indeed, Dantzig-Wolfe never outperformed the standard algorithm in the smaller instances, where optimality was reached; this is consistent with the observed behaviour of Dantzig-Wolfe in other applications [4]. As a summary, it could be stated that, in general, interior-point methods should be used for optimal solutions of IRP, while block coordinate-descent is the most effective choice for approximate feasible solutions of very large instances (to be used as upper bounds or for warm-starting any other algorithm).

## 6 Conclusions

The IRP problem has been modeled and implemented as a large primal three-blocks-angular LP problem. It has been effectively approached by interior-point algorithms and two decomposition techniques, Dantzig-Wolfe and block coordinate-descent. Interior-points and block coordinate-descent with a particular order of the variables (i.e., distribution-refining-supply) resulted in the most effective approaches for respectively optimal solutions and approximate feasible ones.

Among the future tasks to be done, we mention two of them. First, since both interior-point-based Dantzig-Wolfe and block coordinate-descent outperformed their simplex-based variants, it is worthwhile to study the use of alternative interior-point decomposition techniques. Among them we find, for instance, those of Conejo, Nogales and Prieto [9] and Castro [7]. A second task is the inclusion of binary decisions in the models for the three subsystems (e.g., to which storage tank is delivered a specific crude oil from some oil tanker). Even if the planning horizon is significantly reduced (i.e., $T \ll 120$ days) the resulting MILP problem is still very large, and additional decomposition approaches should be applied (e.g., Benders decomposition).

## 7 Acknowledgments

The first author was supported by a grant of the Generalitat de Catalunya under the program International Graduate School of Catalonia for fulfillment of his PhD thesis. This work has been supported by Spanish MEC grant MTM2006-05550. The authors thank two anonymous referees, whose comments greatly improved the presentation of the paper.

## Appendix: Nomenclature

### Indices, Sets

(Note: to simplify notation, numeric sets, e.g. $\{1,\ldots,T\}$, are denoted by its cardinality, i.e. $T$.)

| | |
|---|---|
| $bl, BL$ | product-intermediate material blend |
| $c, C$ | crude oil type |
| $ci, CI$ | crude oil type-storage tank pair |
| $dp, DP$ | depot |
| $i, I$ | storage tank |
| $j, J$ | blending tank |
| $jl, JL$ | blending tank-CDU pair |
| $k, K$ | component |
| $l, L$ | CDU in refinery |

| | |
|---|---|
| $mi, MI$ | intermediate material |
| $mp, MP$ | purchased material |
| $mr, MR$ | raw material |
| $p, P$ | final product |
| $ppt, PPT$ | product-product tank pair |
| $pt, PT$ | product tank |
| $pptdp, PPTDP$ | product-product tank-depot pair |
| $pr, PR$ | refinery process |
| $q, Q$ | quality attribute |
| $t, T$ | time interval,planning horizon |
| $u, U$ | unit |

**Decision variables**

| | |
|---|---|
| $C_t^{CR}$ | cost of all crude consumed during time period $t$ |
| $CF_{c,i,j,t}$ | cumulative flow rate of crude $c$ from storage tank $i$ to blending tank $j$ during time period $t$ |
| $CF_{c,i,t}$ | cumulative flow rate of crude $c$ into storage tank $i$ during time period $t$ |
| $CF_{c,j,l,t}$ | cumulative flow rate of crude $c$ from blending tank $j$ to CDU $l$ during time period $t$ |
| $CF_{k,i,j,t}$ | cumulative flow rate of component $k$ from storage tank $i$ to blending tank $j$ during time period $t$ |
| $CF_{k,i,t}$ | cumulative flow rate of component $k$ into storage tank $i$ during time period $t$ |
| $CF_{k,j,l,t}$ | cumulative flow rate of component $k$ from blending tank $j$ to CDU $l$ during time period $t$ |
| $CINV_t^{BT}$ | inventory cost of all blending tank during time period $t$ |
| $CINV_t^{PT}$ | inventory cost of all product tanks during time period $t$ |
| $CINV_t^{ST}$ | inventory cost of all storage tank during time period $t$ |
| $C_t^{OP}$ | cost of all refining operations during time period $t$ |
| $C_t^{PUR}$ | cost of all purchased materials during time period $t$ |
| $C_t^{TR}$ | transportation cost of all product trucks or pipeline during time period $t$ |
| $DEM_{p,pt,t}$ | demand of product $p$ from product tank $pt$ during time period $t$ |
| $F_{c,i,j,t}$ | flow rate of crude $c$ from storage tank $i$ to blending tank $j$ during time period $t$ |
| $F_{c,i,t}$ | flow rate of crude $c$ into storage tank $i$ during time period $t$ |
| $F_{c,j,l,t}$ | flow rate of crude $c$ from blending tank $j$ to CDU $l$ during time period $t$ |
| $F_{k,i,j,t}$ | flow rate of component $k$ from storage tank $i$ to blending tank $j$ during time period $t$ |
| $F_{k,i,t}$ | flow rate of component $k$ into storage tank $i$ during time period $t$ |
| $F_{k,j,l,t}$ | flow rate of component $k$ from blending tank $j$ to CDU $l$ during time period $t$ |
| $F_{mp,c,t}$ | flow rate of purchased material $mp$ derived from crude $c$ during time period $t$ |
| $F_{p,pt,dp,t}$ | flow rate of product $p$ from product tank $pt$ to depot $dp$ during time period $t$ |
| $F_{p,pt,t}$ | flow rate of product $p$ from product tank $pt$ during time period $t$ |
| $LEV_{p,mi,c,t}^{BL}$ | level of intermediate material $mi$ from crude $c$ blended into final product $p$ during time period $t$ |
| $LEV_{pr,c,t}^{PR}$ | level of process $pr$ using crude $c$ during time period $t$ |
| $REV_t$ | revenue from all products during time period $t$ |
| $V_c$ | volume of crude $c$ consumed during planning horizon |
| $V_{c,i,t}$ | volume of crude $c$ in storage tank $i$ during time period $t$ |

| | |
|---|---|
| $V_{c,j,t}$ | volume of crude $c$ in blending tank $j$ during time period $t$ |
| $V_{c,t}$ | volume of crude $c$ consumed during time period $t$ |
| $V^{CR}$ | volume of all crude consumed during planning horizon |
| $V_t^{CR}$ | volume of all crude consumed during time period $t$ |
| $V_{i,t}$ | volume of all crude in storage tank $i$ during time period $t$ |
| $V_{j,t}$ | volume of all crude in blending tank $j$ during time period $t$ |
| $V_{k,i,t}$ | volume of component $k$ in storage tank $i$ during time period $t$ |
| $V_{k,j,t}$ | volume of component $k$ in blending tank $j$ during time period $t$ |
| $V_{p,pt,t}$ | volume of product $p$ in product tank $pt$ during time period $t$ |

**Parameters**

| | |
|---|---|
| $attr_{mi,q}^{both}$ | quality attribute $q$ of intermediate material $mi$ |
| $attr_{mi,c,q}^{cr}$ | amount of quality attribute $q$ contributed<br>to blend by intermediate $mi$ that is derived from crude $c$ |
| $attr_{p,q}^{max}$ | upper bound specification of quality attribute $q$ for final product $p$ |
| $attr_{mi,c,q}$ | $attr_{mi,c,q}^{cr}$ if $attr_{mi,c,q}^{cr} > 0$, $attr_{mi,q}^{both}$ otherwise |
| $attr_{p,q}^{min}$ | lower bound specification of quality attribute $q$ for final product $p$ |
| $cap_u$ | processing capacity of unit $u$ |
| $cinv_i$ | inventory cost of storage tank $i$ |
| $cinv_j$ | inventory cost of blending tank $j$ |
| $cinv_{pt}$ | inventory cost of product tank $pt$ |
| $cost_{c,t}$ | cost of crude $c$ during time period $t$ |
| $cost_{mp,t}$ | cost of purchased material $mp$ during time period $t$ |
| $cost_{pr}$ | cost of operating process $pr$ |
| $ct_{p,pt,dp}$ | transportation cost of product $p$ from product tank $pt$ to depot $dp$ |
| $dem_{p,pt,dp,t}$ | demand of product $p$ from product tank $pt$ at depot $dp$ during time period $t$ |
| $io_{mr,c,pr}$ | input or output of raw material or intermediate material $mr$, when using<br>crude $c$, per unit level of operation |
| $\phi_{k,c}$ | of process $pr$ wt.% of component $k$ in crude $c$ |
| $rev_{p,t}$ | revenue from product $p$ during time period $t$ |
| $\rho_c$ | density of crude $c$ |
| $\rho_k$ | density of component $k$ |
| $util_{u,pr}$ | 1 if process $pr$ uses unit $u$, 0 otherwise |
| $v0_{c,i}$ | initial volume of crude $c$ in storage tank $i$ |
| $v0_{c,j}$ | initial volume of crude $c$ in blending tank $j$ |
| $v0_{p,pt}$ | initial volume of product $p$ in product tank $pt$ |
| $v_i^{max}$ | maximum capacity of storage tank $i$ |
| $v_j^{max}$ | maximum capacity of blending tank $j$ |
| $v_{pt}^{max}$ | maximum capacity of product tank $pt$ |
| $v_i^{min}$ | minimum capacity of storage tank $i$ |
| $v_j^{min}$ | minimum capacity of blending tank $j$ |
| $v_{pt}^{min}$ | minimum capacity of product tank $pt$ |

# References

[1] Fourer, R., Gay, D.M., Kernighan, D.W., 2002. AMPL: A Modeling Language for Mathematical Programming. *Duxbury Press.*

[2] Aronofsky, J.S., Dutton, J.M., Tayyabkhan, M.T., 1978. Managerial planning with linear programming: in process industry operations. *John Wiley and Sons, Inc.*

[3] Bertsekas D.P., 1999. *Nonlinear Programming, 2nd ed.*, Athena Scientific.

[4] Bertsimas D., Tsitsiklis, J.N., 1997. *Introduction to Linear Optimization*, Athena Scientific.

[5] Buchanan, J.E., Gavern, S.C., Genis, O., Shapiro, J.F., Singhal, V., Thomas, J.M., Torpis, S., 1990. A multi-refinery, multi-period modelling system for Turkish petroleum refining industry. *Interfaces*, 20(4) 48–60.

[6] Castro, J., 2000. A specialized interior-point algorithm for multicommodity network flows, *SIAM Journal on Optimization*, 10 852–877.

[7] Castro, J., 2007. An interior-point approach for primal block-angular problems, *Computational Optimization and Applications*, 36 195–219.

[8] Conejo, A.J., Castillo, E., Minguez, R., Garcia-Bertrand, R., 2006. Decomposition Techniques in Mathematical Programming: Engineering and Science Applications. *Springer.*

[9] Conejo A.J., Nogales, F.J., Prieto, F.J., 2002. A decomposition procedure based on approximate Newton directions, *Mathematical Programming*, 93 495–515.

[10] Dantzig, G.B., Wolfe, P., 1960. Decomposition principle for linear programs, *Operations Research*, 8, 101–111.

[11] Escudero, L.F., Quintana, F.J., Salmerón, J., 1999. CORO, a modelling and an algorithmic framework for oil supply, transformation and distribution optimization under uncertainty. *European Journal of Operational Research*, 114, 638–656.

[12] Ho, J.K., Loute, E., 1981. An advanced implementation of the Dantzig-Wolfe decomposition algorithm for linear programming. *Mathematical Programming*, 20, 303–326.

[13] Ho, J.K., Loute, E., 1983. Computational experience with advanced implementation of decomposition algorithms for linear programming. *Mathematical Programming*, 27, 283–290.

[14] Honkomp, S.J., Lombardo, S., Rosen, O., Pekny, J.F., 2000. The curse of reality - why process scheduling optimization problems are difficult in practice. *Computers and Chemical Engineering*, 24, 323–328.

[15] Iyer, R.R., Grossmann, I.E., Vasantharajan, S., Cullik, A.S., 1998. Optimal planning and scheduling of offshore oil field infrastructure investment and operations. *Industrial and Engineering Chemistry Research*, 37, 1380–1397.

[16] Jia, Z., Ierapetritou, M., 2004. Efficient short-term scheduling of refinery operations based on a continuous time formulation. *Computers and Chemical Engineering*, 28, 1001–1019.

[17] Kosmidis,V.D., Perkins,J.D., Pistikopoulos,E.N., 2005. A mixed integer optimization formulation for the well scheduling problem on petroleum fields. *Computers and Chemical Engineering*, 29, 1523–1541.

[18] Lasschuit, W., Thijssen, N., 2003. Supporting supply chain planning and scheduling decisions in the oil and chemical industry. *In I.E. Grossmann and C.M. McDonald (Eds.), Proceedings of fourth international conference on foundations of computer-aided process operations*, pp. 37–44, Coral Springs, CAChE.

[19] 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, 1630–1641.

[20] Luo, Z.Q., Tseng, P., 1992. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72, 7–35.

[21] Lustig, I.J., Mulvey, J.M, Carpenter, T., 1991. Formulating two-stage stochastic programs for interior point methods. *Operations Research*, 39, 757–770.

[22] Maglhães, M.V., Shah, N., 2003. Crude oil scheduling. *In I.E. Grossmann and C.M. McDonald (Eds.), Proceedings of fourth international conference on foundations of computer-aided process operations*, pp. 323–326, Coral Springs, CAChE.

[23] Más, R., Pinto, J.M., 2003. A mixed-integer optimization strategy for oil supply in distribution complexes. *Optimization and Engineering*, 4, 23–64.

[24] Meister, B., Clark, J.M.C., Shah, N., 1996. Optimization of oilfield exploitation under uncertainty. *Computers and Chemical Engineering*, 20, Suppl., S1251–S1256.

[25] Méndez, C.A., Grossmann, I.E., Harjunkoski, I., Kaboré, P., 2006. A simultaneous optimization approach for off-line blending and scheduling of oil-refinery operations. *Computers and Chemical Engineering*, 30, 614–634.

[26] Moro, L.F.L., Zanin A.C., Pinto, J.M., 1998. A planning model for refinery diesel production. *Computers and Chemical Engineering*, 22, Suppl., S1039–S1042.

[27] Moro, L.F.L. 2003. Process technology in the petroleum refining industry—current situation and future trends. *Computers and Chemical Engineering*, 27, 1303–1305.

[28] Moro, L.F.L., Pinto, J.M., 2004. Mixed-Integer Programming Approach for Short-Term Crude Oil Scheduling. *Ind. Eng. Chem. Res.*, 43(1), 85–94.

[29] Neiro, S.M.S., Pinto, J.M., 2004. A general modelling framework for the operational planning of petroleum supply chains. *Computers and Chemical Engineering*, 28, 871–896.

[30] Okeke, E.O., Osakwe-Akofe, A.A., 2003. Optimization of a refinery crude distillation unit in the context of total energy requirement, Technical Report, NNPC R&D Division, Port Harcourt, Nigeria.

[31] Persson, J.A., Göthe-Lundgren, M., 2005. Shipment planning at oil refineries using column generation and valid inequalities. *European Journal of Operational Research*, 163, 631–652.

[32] Pinto, J.M., Joly, M., Moro, L.F.L., 2002. Planning and scheduling models for refinery operations. *Computers and Chemical Engineering*, 24, 2259–2276.

[33] Ponnambalam, K., Vannelli, A., Woo, S., 1992. An interior point method implementation for solving large planning problems in the oil refinery industry. *The Canadian Journal of Chemical Engineering*, 70, 368–374.

[34] Rejowski, R. Jr., Pinto, J.M., 2003. Scheduling of a multi-product pipeline system. *Computers and Chemical Engineering*, 27, 1229–1246.

[35] Song, J., Park, H., Lee, D.-Y., Park, S., 2002. Scheduling of actual size refinery processes considering environmental impacts with multi-objective optimization. *Industrial and Engineering Chemistry Research*, 41, 4794–4806.

[36] Tebboth, J.R., 2001. *A Computational Study of Dantzig-Wolfe Decomposition*, Ph.D. Thesis, University of Buckingham.

[37] U.S. Department of Energy, Energy Information Administration (URL http://www.eia.doe.gov/).

[38] Van der Bruggen, L., Gruson, R., Salomon, M., 1995. Reconsidering the distribution of gasoline products for a large oil company. *European Journal of Operational Research*, 81, 460–473.

[39] Van der Heever, S.A., Grossmann, I.E., 2000. An iterative aggregation/disaggregation approach for the solution of a mixed-integer nonlinear oilfield infrastructure planning model. *Industrial and Engineering Chemistry Research*, 39, 1955–1971.

[40] Van der Heever, S.A., Grossmann, I.E., Vasantharajan, S., Edwards, K. 2000. Integrating complex economic objectives with the design and planning of offshore oilfield infrastructure. *Computers and Chemical Engineering*, 24, 1049–1055.

[41] Wenkai, L., Hui, C-W., Hua, B., Tong, Z., 2002. Scheduling crude oil unloading, storage, and processing. *Industrial and Engineering Chemistry Research*, 41, 6723–6734.