ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
CONSIGLIO NAZIONALE DELLE RICERCHE

G. Felici, C. Gentile, G. Rinaldi

SOLVING LARGE MIP MODELS IN SUPPLY CHAIN MANAGEMENT BY BRANCH & CUT[1]

R. 522 Gennaio 2000

Giovanni Felici – Istituto di Analisi dei Sistemi ed Informatica del CNR, Viale Manzoni 30 - 00185 Roma, Italy. Email : felici@iasi.rm.cnr.it.

Claudio Gentile – Istituto di Analisi dei Sistemi ed Informatica del CNR, Viale Manzoni 30 - 00185 Roma, Italy. Email : gentile@iasi.rm.cnr.it.

Giovanni Rinaldi – Istituto di Analisi dei Sistemi ed Informatica del CNR, Viale Manzoni 30 - 00185 Roma, Italy. Email : rinaldi@iasi.rm.cnr.it.

ISSN: 1128–3378

[1]Research partially supported by ESPRIT project 24897: Hydrocarbon and Chemical Logistics Optimization under Uncertainty via Stochastic Optimization (HChLOUSO)

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica, CNR

viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.rm.cnr.it
URL: http://www.iasi.rm.cnr.it

Abstract

We consider supply chain management problems where a number of commodities are produced and consumed in different bases in a given time horizon. A set of transport means is used to move products between bases in order to respect the daily stock constraints of each product. Such problems result in very large scale mixed integer problems with block structure. We propose a solution algorithm that integrates column generation techniques and polyhedral approaches in a Branch & Cut scheme. Special classes of cuts for mixed integer problems are introduced. Finally, we discuss an application to ship scheduling for hydrocarbon products distribution in an oil industry.

Key words: Mixed Integer Programming, Polyhedral Methods, Column Generation, Supply Chain Management.

1. Introduction

Optimization techniques in industrial production and distribution systems have been extensively studied and applied by many reseacheres. These efforts have led to a consistent body of literature addressing the Supply Chain Management problem. In [39] an accurate overview of many different approaches to this problem is given; previously, interesting reviews have been written by [42] and [40].

With *Supply Chain* it is intended the union of all processes that contribute to the creation of finished products and to their consumption. The objective of *Supply Chain Management* is to optimize and control the flow of materials and information among all the components of the Supply Chain system. Traditionally, three stages are identified in the Supply Chain (e.g., see [40]): *procurement*, *production*, and *distribution*. In real situations, each stage may present different characteristics, dimensions, and complexity and thus require specific modeling and solution approaches. In recent works, relevant efforts are being made to provide a complete approach that coordinates the three stages of the Supply Chain; nevertheless, the specific characteristics of real problems may constraint the extent to which coordination and integration of Supply Chain Management take place.

According to the time horizon adopted, the three different planning levels listed below are identified:

- The *strategic* level considers network design and mid-long term planning. Network design concerns the location of the production and consumption facilities, the selection of the suppliers, and the identification and the sizing of the different transport means; mid or long term planning deals with the different operations addressing a period of one or more years. In this level only aggregated data are required.

- The *tactical* level takes into account a time horizon of one or two months; detailed schedules of all the operations that must take place are produced. Here, data accuracy and availability is crucial to obtain implementable solutions.

- The *operational* level is concerned with daily control of last minute information and recovery from unexpected events. The required data are of a great level of detail, and are often restricted to a subsystem. Quick decision making is needed; the recovery of the schedules planned at the tactical level should not interfere with the rest of the system.

The most recent body of literature concering the strategic level of the Supply Chain proposes models that address both the network design problems and the mid-long term planning. In [20], [21], and [22] a MIP model that solves a multicommodity single period production-distribution problem with location selection is presented. The model is solved using Benders Decomposition and binary variables fixings. Extensions of these models are proposed in [11], where a deterministic and multi-period production-distribution model with a nonlinear objective function is described and solved with a heuristic approach. In [13] it is discussed a mixed integer model to optimize production-distribution flows with piecewise linear concave costs where the location of facilities is fixed. In [27] is described a plant location model that takes into account exchange rate fluctuations, market prices, and international interest rates. This model results in a large nonlinear MIP, approached by approximation techniques. Stochasticity aspects of Supply Chain Management are also considered, amongst other issues, in [10], [12], [18] and [17]. Strategic design of production-distribution systems is modeled as a general framework in [24] and [23];

in [41] also supplier reliability is taken into account by means of a set of constraints on the probability of being in time for the suppliers shippings.

The optimization of the tactical and the operational levels of Supply Chain Management is characterized by large MIP models that can differ largely from application to application, and require sophisticated optimization techniques or heuristics. For production optimization, a typical mathematical approach is represented by Lot Sizing models (see [36] for a review).

In this paper we present a general framework for the distribution stage and its application to oil industries. In this field the procurement stage consists in the transportation of crude oil from suppliers located in different countries to a set of refineries, where the oil is tranformed into several finished or semi-finished products (production stage). The distribution stage is divided into primary and secondary. The primary distribution is related to the transportation of the products among the refineries, from refineries to depots, and among the depots. Typical transport means for primary distribution are ships, pipelines, and truck-wagons. The secondary distribution is concerned with the trasportation of the finished products from the depots to the gas stations. In that case transportation is mainly performed by trucks.

Amongst the main elements that characterize the Supply Chain in Oil Industry, we note that:

a) production is driven by market demand and oil price;

b) the process of transformation of crude oil into market products imposes strong constraints both on the procurement and on the distribution stages, as refineries have to work at given rates and are not flexible in the short term;

c) coordination between procurement, production, and distribution may take place only at the strategic level;

d) once a strategic plan has been made for a given time horizon, optimization at the tactical level results in large savings of the transportation costs associated with procurement and distribution.

Based on these premises, we believe that the tactical level plays a key role in the Supply Chain of Oil Industry: on the one hand, tactical models can be used to validate and control in a feedback process the decisions made at the strategic level, that are based on aggregate data; on the other hand, tactical optimization is typically more complex from the mathematical modeling standpoint, and leads to the formulation of challenging Operations Research problems.

We propose a model that operates at the tactical level, can interact with the strategic level, and can be focused on the operational level. For example, an integration has been realized between the proposed operational model and the strategic model described in [18]. We view the operational level as a specialization of the tactical one, where some variables are fixed and only the variables that have been modified from unexpected events are to be reoptimized. In our model, we assume that the quantities to be delivered by each transport mean are decided at the strategic level; the output of the model is the monthly schedule for the considered transport means. We present an application to primary distribution (amongst refineries and depots) in AgipPetroli. Here, a number of ships with different characteristics are used.

Similar problems are studied in the literature, the majority of which restrict the focus to the ship scheduling problem without considering its interactions with the rest of the Supply Chain. A very extensive survey on ship scheduling and ship routing problems is given in [37] and [38]. The author compares the ship scheduling problem with the more famous and largely studied vehicle routing or vehicle scheduling problems, pointing out that in the former there is a much larger variety in problem structure and in operating environments, that shipping operations are subject to a significant degree of uncertainty, and that the prices of products and shipping on the international market are very unstable.

Early works on ship sheduling include [15], which aims to the minimization of the number of used ships, and [28], where an LP model is proposed as an appromixation for the problem of allocating a total transportation capacity for each pair of origin-destination bases, and to determine the minimum number of vessels required. In [30] it is described a mathematical model to solve the tanker scheduling model of the Defence Fuel Center and the Military Sealift Command in the worldwide distribution of bulk petroleum products; the resulting integer programming formulation appears to be untractable by the computational resources then available, and an heuristic scheme based on rounding was adopted to obtain feasible solutions.

More recently, in [31], it is described an interactive computer system addressing daily scheduling issues as well as longer range planning problems; the method utilizes a network flow model and a mixed integer programming model. The time horizon is one year and a half, during which four ships can make up to five travels. The solution method adopted is a heuristic with special features that enhance the interactions between the user and the system. In [6] a crude oil tanker scheduling problem is considered. The constraints of the problem limit the number of all feasible schedules, which can be generated off-line, and appear as columns in a set partitioning model. The contained dimensions of the application problem allow to find optimal integer solutions within a reasonable computation time. In [19] the efficient scheduling of fleet of ships engaged in "pick-up and delivery" of bulk cargoes is presented. The system adopts off-line generation of ship schedules. According to the dimensions of the problem, the system can either generate all the feasible schedules or heuristically limit the generation process. The corresponding set packing problem is efficiently solved using a Lagrangean heuristic. Finally, in [9] a variant of the multi-vehicle "pick-up and delivery" problem with time windows combined with a multi-inventory model for a real ship planning problem is described. Here an on-line column generation approach is adopted and integrated in a standard Branch & Price algorithm (see [5]).

In this paper we describe a model for tactical Supply Chain Optimization that can be specialized for the operational level and interfaced with a strategic model. In Section 2 we describe the model with different formulations and possible extensions. In Section 3 we outline a general solution approach combining Dantzig-Wolfe decomposition and polyhedral methods. In Section 4 we present the class of Nested Knapsack Inequalities with some polyhedral properties and describe the application to the problem at hand. In Section 5 we discuss the application of cover inequalities. In Section 6 we present an application to a real life problem arising in oil industries.

2. Model Description

With the objective of extending the previous work above described, we consider a general model that can be used for tactical Supply Chain Optimization. The model addresses the problem of the distribution of a set of *commodities* amongst a set of *bases*. In each base, commodities can be produced or consumed; constraints on the minimum and maximum amount of stock are then given for each commodity and for each base. A number of transport means is used to move the different commodities amongst the bases.

Given a planning period made of a finite number of time units (typically days), the problem is to move the commodities between the bases in order to obey the stock constraints with minimal transportation cost. Moreover, in many real situations, it may be the case that the available transport means are not capable of satisfying the given constraints; in these situations, extra transport capacity is to be obtained at an additional cost, or stock violations can be recoved by buying or selling the product on the market. For these reasons we also consider *violations variables*, associated with balance constraints.

The network data needed for formulating the model concerns the bounds on stock for each commodity, for each base, and for each time unit, and the production and the consumption rates for each commodity in the bases. The data related to the transport means may vary according to their type; generally, we assume the travel time and the travel cost amongst all bases to be known.

The basic element of the model is the *space-time* graph, depicted in Figure 1. A space-time graph is associated with a single transport mean or to a class of equivalent transport means. The nodes of this graph are disposed on a regular grid with the rows indexed by the bases and the columns indexed by the time units in the planning horizon. Each node is then associated with a pair (base, time unit). An arc between two nodes (b_1, t_1) and (b_2, t_2) is present if the distance between b_1 and b_2 can be covered in time $t_2 - t_1$. The arcs in the space-time graph can be of three different types: *loaded arcs*, i.e., arcs that indicate the moving of commodities from production to consumption bases (thick arcs in the figure); *unloaded arcs*, indicating the travels of the transport mean from a consumption base to the next production base (thin arcs in the figure); and *still* arcs, that are present when the transport mean stays in the same base for one or more time units (dashed arcs in the figure). Such a graph is obviously acyclic.

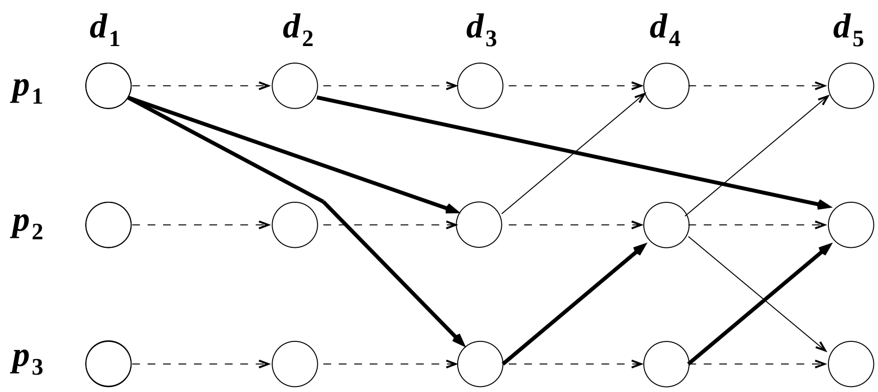

Figure 1: Example of space-time graph for a transport mean starting at base 1 on day 1

In the following sections we present two alternative formulations for the Tactical Distribution Problem in Supply Chain Management:

1. The *Schedule Formulation*: a column generation based formulation, with binary variables associated with paths on the space-time graphs of the type of Figure 1.

2. The *Arc Formulation*: a direct formulation, with binary variables associated with arcs of the same graphs.

Henceforth, we assume a time unit of one day. Before introducing the two formulations we introduce the common notation:

- the index denoting the transport mean is $\sigma \in \mathcal{S}$, where $\mathcal{S}$ is the set of all transport means;

- $SG(\sigma)$ is the space-time graph associated with transport mean σ;
- the index denoting the commodity is $k \in K$, where K is the set of all commodities;
- the index denoting the base is $p \in P \supseteq \Pi_k \cup \Omega_k$, where:
  - P is the set of all bases;
  - Π_k is the set of production bases for commodity $k \in K$;
  - Ω_k is the set of consumption bases for commodity $k \in K$;
- the index denoting the day is $d \in D$, where $D = \{1, \ldots, d_m\}$ is the set of time units;
- an arc of the space-time graph is denoted by $e \in E$, where:
  - E is the union of the set of arcs of all the space-time graphs;
  - E_σ is the set of arcs in $SG(\sigma)$;
  - FS_σ^{pd} is the set of travel arcs with tail in node (p, d) associated with transport mean σ;
  - BS_σ^{pd} is the set of travel arcs with head in (p, d) associated with transport mean σ;
- b_{kpd} is the amount of commodity k produced or consumed on day d in base p (production $b_{kpd} > 0$, consumption $b_{kpd} < 0$);
- z_{kpd} is the continuous variable representing the stock level of commodity k at base p on day d;
- $\check{Z}_{kpd}$ and $\hat{Z}_{kpd}$ are the lower and the upper bounds for z_{kpd}, respectively;
- $\check{s}_{kpd}$ and $\hat{s}_{kpd}$ are the continuous variables representing violation of lower and upper bounds of the stock level of commodity k at base p on day d, respectively.

2.1. The Schedule Formulation

This model is a mixed integer program, where continuous variables are associated with the stock level of each commodity, on each day, in each base, and binary variables are associated with paths of the transport means. Each binary variable represents a path on the space-time graph for a transport mean, and the corresponding loads for each travel in the path. Such path spans all the planning horizon. We call such a combination of a path and loads a *schedule*.

The constraints in this formulation are of three types:

- **Balance Constraints**: say that the stock level on each day d, for each commodity and for each production (consumption) base is equal to the stock on day $d-1$ minus the quantities delivered (plus quantities received) on day d plus production (minus consumption) on day d, plus and minus the corresponding violations variables.
- **Stock Constraints**: say that the stock level on each day, in each base and for each commodity, is bounded by a minimum and a maximum stock level.
- **Schedule Constraints**: say that for each transport mean at most one schedule can be selected.

The Schedule Formulation can be summarized as follows:

$$
\begin{aligned}
\min \quad & \sum_{s\in S} c_s x_s + \sum_{k\in K}\sum_{d\in D}\sum_{p\in P} \hat{w}_{kpd}\hat{s}_{kpd} + \sum_{k\in K}\sum_{d\in D}\sum_{p\in P} \check{w}_{kpd}\check{s}_{kpd} \\
& z_{kpd} - z_{kp,d-1} + \sum_{s\in S}\sum_{e\in BS^{pd}_{\sigma(s)}} q^k_{se}\, x_s + \hat{s}_{kpd} - \check{s}_{kpd} = b_{kpd} && \text{for all } k\in K,\ p\in \Pi_k,\ d\in D \\
& z_{kpd} - z_{kp,d-1} - \sum_{s\in S}\sum_{e\in FS^{pd}_{\sigma(s)}} q^k_{se}\, x_s + \hat{s}_{kpd} - \check{s}_{kpd} = b_{kpd} && \text{for all } k\in K,\ p\in \Omega_k,\ d\in D \\
& \check{Z}_{kpd} \le z_{kpd} \le \hat{Z}_{kpd} && \text{for all } k\in K,\ p\in P,\ d\in D \\
& \sum_{s\in S_\sigma} x_s \le 1 && \text{for all } \sigma\in \mathcal{S} \\
& \hat{s}_{kpd} \ge 0, \check{s}_{kpd} \ge 0 && \text{for all } k\in K,\ p\in P,\ d\in D \\
& x_s \in \{0,1\} && \text{for all } s\in S
\end{aligned}
\tag{1}
$$

where

- S is the set of all possible schedules; for each schedule $s \in S$ we denote by $E(s)$ the set of arcs composing the path of s and by x_s the corresponding binary variable;
- S_σ is the set of schedules of transport mean σ;
- transport mean $\sigma(s)$ is the one associated with schedule $s \in S$;
- c_s is the cost of schedule s; we suppose that it can be obtained as the sum of the costs of the arcs that compose the path of s, that is $c_s = \sum_{e\in E(s)} \tilde{c}_{\sigma(s)e}$;
- q^k_{se} is the quantity of commodity k loaded on transport mean $\sigma(s)$ in travel arc e of schedule s (zero if arc e is not in the path represented by schedule s).

The Schedule Formulation has a contained number of constraints but a very large number of variables, as the number of possible schedules for a given transport mean in a real size space-time graph can explode to untractable numbers. Nevertheless, the Schedule Formulation can be tackled with decomposition techniques, such as Dantzig-Wolfe decomposition and Column Generation. In these settings, we consider only a subset of all the feasible schedules and generate on the fly new schedules that reduce the value of the optimal solution. Such techniques can efficiently lead to the optimal solution of the linear relaxation of (1) under some conditions, that will be analyzed in details in Section 3.

2.2. The Arc Formulation

This model is a mixed integer program, where the continuous variables are associated with the stock level of each commodity, on each day, in each base, and the binary variables are associated with each arc of the space-time graphs and with all pairs (segregation, commodity) for each travel arc of the graphs. These variables are linked by the following sets of constraints:

- **Balance Constraints**: say that the stock level on each day d, for each commodity and for each production (consumption) base is equal to the stock on day $d-1$ minus the quantities delivered (plus quantities received) on day d plus production (minus consumption) on day d, plus and minus the corresponding violation variables.
- **Stock Constraints**: say that the stock level on each day, in each base and for each commodity, is bounded by a minimum and a maximum stock level.
- **Commodity Assignment Constraints:** identify the commodity (or the commodities) that are loaded on the transport means.
- **Path Constraints**: say that the arcs chosen for a given transport mean must form a path from its initial position to one of the bases in the last day of the planning period.

In the formulation below we make the simplifying assumption that each transport mean can be loaded with only one commodity in each travel. This assumption can be removed by the addition of extra variables and extra constraints, as explained in the Section 2.3. We now state the Arc Formulation as follows:

$$
\begin{aligned}
&\min \sum_{\sigma\in\mathcal{S}}\sum_{e\in E}\tilde{c}_{\sigma e}v_{\sigma e} + \sum_{k\in K}\sum_{d\in D}\sum_{p\in P}\hat{w}_{kpd}\hat{s}_{kpd} + \sum_{k\in K}\sum_{d\in D}\sum_{p\in P}\check{w}_{kpd}\check{s}_{kpd} \\
&z_{kpd} - z_{kp,d-1} + \sum_{\sigma\in\mathcal{S}}\sum_{e\in FS_\sigma^{pd}} q_{\sigma e}\, y_{\sigma e}^k + \hat{s}_{kpd} - \check{s}_{kpd} = b_{kpd} && \text{for all } k\in K,\ p\in\Pi_k,\ d\in D \\
&z_{kpd} - z_{kp,d-1} - \sum_{\sigma\in\mathcal{S}}\sum_{e\in BS_\sigma^{pd}} q_{\sigma e}\, y_{\sigma e}^k + \hat{s}_{kpd} - \check{s}_{kpd} = b_{kpd} && \text{for all } k\in K,\ p\in\Omega_k,\ d\in D \\
&\check{Z}_{kpd} \le z_{kpd} \le \hat{Z}_{kpd} && \text{for all } k\in K,\ p\in P,\ d\in D \\
&\sum_{k\in K} y_{\sigma e}^k = v_{\sigma e} && \text{for all } \sigma\in\mathcal{S},\ e\in E_\sigma \\
&\mathcal{N}_\sigma\, v_\sigma = \beta_\sigma && \text{for all } \sigma\in\mathcal{S} \\
&v_{\sigma e}\in\{0,1\}, y_{\sigma e}^k\in\{0,1\}, \hat{s}_{kpd}\ge 0, \check{s}_{kpd}\ge 0
\end{aligned}
\tag{2}
$$

where

- $v_\sigma = [v_{\sigma e}]_{e\in E_\sigma}$ is a vector of 0-1 variables, where $v_{\sigma e} = 1$ means that arc e is in the path of transport mean σ, and $v_{\sigma e} = 0$ otherwize;
- the constraints $\mathcal{N}_\sigma\, v_\sigma = \beta_\sigma$ represent the path constraints for ship σ, where $\mathcal{N}_\sigma$ is the node-arc incidence matrix of the space-time graph for transport mean σ;
- $y_{\sigma e}^k$ is a $0-1$ variable that takes value 1 if and only if the corresponding travel is performed ($v_{\sigma e} = 1$), and the commodity k is loaded on the trasport mean σ;
- $\tilde{c}_{\sigma e}$ is the cost of travel on arc e of ship σ;
- $q_{\sigma e}$ is the quantity that transport mean σ can carry in travel represented by arc e.

2.3. Extensions

Here we discuss some extensions to the basic model associated with the two formulations presented above. These extensions refer to the way transport means are loaded, and thus depend on the specific type of transport means considered. Above we have described a model where a transport mean can be loaded with only one of the commodities produced and consumed. Indeed, some real applications allow for more than one commodity to be loaded on the same transport mean, for example, when transport means are ships or trucks. We consider two possible extensions: the first allows to pack the capacity of a transport mean with several commodities, with the only restriction that the sum of the load of each commodity does not exceed the capacity of the transport mean; the second assumes the transport mean to have a fixed number of compartments, called *segregations*, with given capacity, each of which may be loaded with a different commodity.

The extensions have a larger impact on the Arc Formulation, where additional variables are to be considered, while in the Schedule Formulation the new requirements may be dealt with in the column generation subproblem; the reader may recall that a generated schedule consists in a path on the space-time graph with associated loads on each travel.

Multiple loads on the same Transport mean

If the transport means can carry more than one commodity in the same travel, we relax the integrality constraints on the variables $y^k_{\sigma e}$, that now indicate the proportion of commodity k loaded on transport means σ on travel arc e. In the Arc Formulation it is sufficient to substitute $y^k_{\sigma e} \in \{0,1\}$ with

$$0 \le y^k_{\sigma e} \le 1 \qquad \text{for all } k \in K,\ \sigma \in \mathcal{S},\ e \in E_\sigma.$$

This modification models the case of transport means where the different commodities can be loaded in any possible way.

In the Schedule Formulation the relaxation of the integrality constraints on the y variables is expressed by substituing $x_s \in \{0,1\}$ with the constraints

$$\sum_{s \in S(\sigma,e)} x_s \in \{0,1\} \qquad \text{for all } \sigma \in \mathcal{S},\, e \in E_\sigma,$$

where $S(\sigma, e)$ is the set of schedules of σ containing arc e; such constraints correspond to the integrality of the v variables.

Finally, note that so far we have assumed that each transport mean travels at full load. With the addition of a *dummy commodity*, respresenting the empty space on the transport mean, it is possible to consider also partial loads.

Segregations with Fixed Capacity

In this case, the transport means are allowed to carry more than one commodity but have their full capacity divided into a fixed number of segregations, with given capacity. The new variables and parameters have the following meaning:

- the index denoting a segregation is $g \in G_\sigma$, where G_σ is the set of segregations of transport mean σ;

- $y^{gk}_{\sigma e}$ has value 1 if commodity k is transported in segregation g of transport mean σ in travel arc e, and zero otherwise (segregation-commodity variables);
- $\tilde{q}^{g}_{\sigma e}$ is the quantity that segregation g of transport mean σ can carry in travel represented by arc e, and $\sum_{g \in G_\sigma} \tilde{q}^{g}_{\sigma e} = q_{\sigma e}$.

The changes in the Arc Formulation concern the Balance Constraints and the Commodity Assignment Constraints:

$$z_{kpd} - z_{kp,d-1} + \sum_{\sigma \in \mathcal{S}} \sum_{e \in FS^{pd}_{\sigma}} \sum_{g \in G_\sigma} \tilde{q}^{g}_{\sigma e}\, y^{gk}_{\sigma e} + \hat{s}_{kpd} - \check{s}_{kpd} = b_{kpd} \qquad \text{for all } k \in K,\ p \in \Pi_k,\ d \in D$$

$$z_{kpd} - z_{kp,d-1} - \sum_{\sigma \in \mathcal{S}} \sum_{e \in BS^{pd}_{\sigma}} \sum_{g \in G_\sigma} \tilde{q}^{g}_{\sigma e}\, y^{gk}_{\sigma e} + \hat{s}_{kpd} - \check{s}_{kpd} = b_{kpd} \qquad \text{for all } k \in K,\ p \in \Omega_k,\ d \in D$$

$$\sum_{k \in K} y^{gk}_{\sigma e} = v_{\sigma e} \qquad \text{for all } \sigma \in \mathcal{S},\ e \in E_\sigma,\ g \in G_\sigma$$

$$y^{gk}_{\sigma e} \in \{0,1\} \qquad \text{for all } \sigma \in \mathcal{S},\ e \in E_\sigma,\ g \in G_\sigma,\ k \in K$$

This extension has a relevant impact on the complexity and on the dimension of the Arc Formulation, as new binary variables and new constraints are introduced in the model. On the other hand, the Schedule Formulation has the same representation, but the number of possible columns in the set S is increased. This fact must be taken into account in the Column Generation Subproblem where the loads on the different segregations must also be generated.

3. Solution algorithm

Real instances of the tactical distribution problem result in models with very large dimensions. A typical example, coming from an italian oil company with 19 bases, 10 different commodities, one month of planning, and 10 transport means, corresponds to an Arc Formulation with over 1 million constraints, over 34.000 continuous violations variables, and over 100.000 integer variables.

Even with the latest versions of the commercial solvers for Linear Programming (e.g., CPLEX 6.5, XPRESS-MP 11) it is difficult to handle problems with such dimensions; hence, decomposition approaches must be used. In the present work we choose the *Column Generation* technique that enables to formulate the problem with a very general model unifying the representation of different transport means.

In this section we design a solution method based on two techniques and on their integration: the Column Generation and the Branch & Cut techniques. In Section 3.1 we analyse the issue of the integration of Column Generation and Branch & Cut and propose a general framework that can be adopted for large classes of problems with integer variables; in Section 3.2 we expose the structure of the algorithm for the generation of the columns for the Schedule Formulation; in Section 3.3 we discuss the valid inequalities which will be applied to the problem.

3.1. Integration of the Dantzig-Wolfe decomposition and of the polyhedral methods

The Dantzig-Wolfe decomposition and the connected Column Generation technique can be deployed to efficiently solve the linear relaxation of large block structured problems; in fact, these methods take advantage from the peculiar structure of the subproblems associated with each block of contraints. Although, to obtain an optimal integer solution for an IP problem, additional machinery is needed.

In the literature, large integer and mixed-integer problems have been solved using the Column Generation approach combined with Branch & Bound; at each node of the search tree new columns are generated by the solution of the column generation subproblem, also referred to as the *pricing problem*. The resulting approach is thus called Branch & Price (see [5] and for more details).

The literature also proposes other techniques for the solution of integer and mixed-integer problems, amongst which Cutting Planes, or Polyhedral techniques, has proven to be extremely successful; these methods are based on the reinforcement of the problem formulations by new inequalities, or *cuts*. Cutting Planes methods, combined with Branch & Bound, yield to Branch & Cut methods. Many theoretical and real life integer problems have been successfully solved using Branch & Cut (e.g., see [35] for the original reference on the method, or [7] for an extensive list of references), so that this technique has become the most established approach to solve integer problems to optimality.

Branch & Price approaches have been successfully used to solve problems arising in airline companies, such as the Crew Scheduling Problem (e.g., [4],[16], and [29]), and challenging theoretical problems such as the Capacitated Vehicle Routing Problem (e.g., [3],[1], and [25]); unfortunately in all these papers no cuts are added to strengthen the LP formulation.

In the following, we describe some conditions and state some results showing how it is possible to combine the Column Generation and the polyhedral techniques, without affecting the efficiency of the former and gaining the benefits of the latter. We describe the complete cycle of the process from the solution of the relaxation of the problem to the identification and the addition of violated cuts.

Given a set of elements E, we consider a class of optimization problems, where each feasible solution is represented by a number $k \geq 1$ of subsets of E. Each subset is defined by some structural constraints while the k subsets satisfy some compatibility requirements.

We consider the following general formulation of the problem, that we refer to as the *Block Formulation*:

$$\begin{array}{lcccccccl} \min & \tilde{c}^1 y^1 & + & \tilde{c}^2 y^2 & + & \cdots & + & \tilde{c}^k y^k & \\ & \tilde{A}^1 y^1 & + & \tilde{A}^2 y^2 & + & \cdots & + & \tilde{A}^k y^k & \leq \tilde{b} \\ & B^1 y^1 & & & & & & & \leq d^1 \\ & & & B^2 y^2 & & & & & \leq d^2 \\ & & & & & \ddots & & & \vdots \\ & & & & & & & B^k y^k & \leq d^k \\ & \multicolumn{8}{l}{y^i_e \in \{0,1\} \text{ for all } i = 1, \ldots, k, \ e \in E.} \end{array} \tag{BF}$$

The variable y^i_e takes value 1 if the element $e \in E$ is selected for the i-th subset. The constraints $B^i y^i \leq d^i$ are the structural constraints of the i-th subset, while

$$\tilde{A}^1 y^1 + \tilde{A}^2 y^2 + \cdots + \tilde{A}^k y^k \leq \tilde{b}$$

are the compatibility constraints among the different subsets.

When the Dantzig-Wolfe decomposition is applied, one obtains another formulation of the problem, where the variables are directly associated with the subsets of E satisfying the structural constraints, and one such variable for each block is to be selected. Let the following be this alternative formulation that we refer to as the *Subsets Formulation*:

$$\begin{aligned} \min \quad & \sum_{i=1}^{k} \sum_{s \in S^i} c_s^i x_s^i \\ & Ax \le b \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(SF)} \\ & x_s^i \in \{0,1\} \text{ for all } s \in S^i,\ i = 1,\ \ldots,\ k. \end{aligned}$$

The constraints $Ax \le b$ represent a reformulation of the compatibility constraints, and contain also the requirement that only one subset can be selected for each block i. The sets S^i are the families of all subsets which are feasible for the structural constraints of block i.

The Subsets Formulation has in general an exponential number of columns, but it has a smaller number of rows as the structural constraints are no more needed. In fact, they are already satisfied by the definition of the variables.

Now we introduce some notation and summarize the one previously exposed:

- E is the set of elements that can be selected;
- S^i is the family of subsets of E which are feasible for the i-th block; hereafter we refer to these subsets as *configurations*;
- S is the union of the families S^i over all blocks i;
- y_e^i is the variable associated with element $e \in E$ for block i in (BF);
- x_s^i is the variable associated with configuration $s \in S^i$ in (SF);
- $S^i(e)$ is the subset of configurations $s \in S^i$ containing the element $e \in E$;
- $E(s)$ is the subset of elements in E contained in the configuration $s \in S$.

As (SF) and (BF) are two formulations of the same problem, the cost c_s^i of x_s^i is equal to the cost of the elements in the set $E(s)$, i.e., $c_s^i = \sum_{e \in E(s)} \tilde{c}_e^i$.

The following proposition establishes the connection between the feasible solutions of the two formulations (SF) and (BF).

Proposition 3.1. *Let x be an integer feasible solution of* (SF)*, then the solution y, where*

$$\{y_e^i = \textstyle\sum_{s \in S^i(e)} \quad x_s^i \mid e \in E,\ i \in \{1,\ \ldots,\ k\}\}, \tag{3}$$

is an integer feasible solution for (BF).

Proof: Follows straightforwardly from the fact that x defines a set of feasible configurations. □

With the following proposition we transform valid inequalities for formulation (BF) into valid inequalities for formulation (SF).

Proposition 3.2. *If the inequality $\tilde{\alpha} y \le \tilde{\alpha}_0$ is valid for the original Block Formulation* (BF)*, then the inequality $\alpha x \le \alpha_0$ defined by*

$$\alpha_s^i = \sum_{e \in E(s)} \tilde{\alpha}_e^i \qquad \forall\, s \in S^i \ \ \forall\, i = 1,\ \ldots\ ,k \tag{4}$$

$$\alpha_0 = \tilde{\alpha}_0, \tag{5}$$

is valid for the Subsets Formulation (SF).

Proof: Let x be a feasible solution for (SF) and y be the corresponding feasible solution for (BF) obtained by (3), then

$$\begin{aligned}\alpha x &= \sum_{i=1}^{k}\sum_{s\in S^i}\alpha_s^i x_s^i = \sum_{i=1}^{k}\sum_{s\in S^i}x_s^i\sum_{e\in E(s)}\tilde{\alpha}_e^i = \\ &= \sum_{i=1}^{k}\sum_{e\in E}\tilde{\alpha}_e^i\sum_{s\in S^i(e)}x_s^i = \sum_{i=1}^{k}\sum_{e\in E}\tilde{\alpha}_e^i y_e^i = \tilde{\alpha}y \leq \tilde{\alpha}_0 = \alpha_0.\end{aligned}$$

Hence, the inequality $\alpha x \leq \alpha_0$ is valid for (SF). □

The following result permits to combine the Column Generation technique with the polyhedral methods to solve linear programming problems with integer variables.

Theorem 3.3. *If an inequality $\alpha x \leq \alpha_0$ valid for* (SF) *obtained by (4) and (5) is added to* (SF), *then the structure of the pricing subproblem is not modified.*

Proof: We show that only a change of objective function coefficients occurs in the pricing subproblem as a consequence of adding new inequalities. Let (π, ξ) be a dual optimal solution of the relaxation of (SF) together with a valid inequality $\alpha\, x \leq \alpha_0$, where ξ is the dual price of the added constraint.

If $\bar{c}_s^i$ was the reduced cost of configuration s disregarding ξ, then the reduced cost in the new problem can be expressed as $\bar{c}_s^i - \xi\alpha_s^i$.

From the definition of c_s^i, we derive that $\bar{c}_s^i$ may be written as $\sum_{e\in E(s)} w_e^i$, where w_e^i depends on $\tilde{c}$ and π. As for (4) $\alpha_s^i = \sum_{e\in E(s)}\tilde{\alpha}_e^i$, then the new reduced cost is equal to $\sum_{e\in E(s)} w_e^i - \xi\sum_{e\in E(s)}\tilde{\alpha}_e^i = \sum_{e\in E(s)} w_e^i - \xi\tilde{\alpha}_e$. As in the pricing subproblem only the objective function is modified when adding a new inequality, the same algorithm can be used. □

In the following proposition, we state another property that allows to use a separation algorithm for valid inequalities in the original formulation (BF) and then use these inequalities in (SF).

Proposition 3.4. *Let x be a solution for* (SF). *If the inequality $\tilde{\alpha}y \leq \tilde{\alpha}_0$ is violated by the solution y for* (BF) *obtained from x by (3), then the corresponding inequality $\alpha x \leq \alpha_0$ defined by (4) and (5) is violated by x.*

Proof: We have to prove that if $\tilde{\alpha}y > \tilde{\alpha}_0$ then $\alpha x > \alpha_0$. By propositions 3.1 and 3.2

$$\begin{aligned}\alpha_0 = \tilde{\alpha}_0 < \tilde{\alpha}y &= \sum_{i=1}^{k}\sum_{e\in E}\tilde{\alpha}_e^i y_e^i = \sum_{i=1}^{k}\sum_{e\in E}\tilde{\alpha}_e^i\sum_{s\in S^i(e)}x_s^i = \\ &= \sum_{i=1}^{k}\sum_{s\in S^i}x_s^i\sum_{e\in E(s)}\tilde{\alpha}_e^i = \sum_{i=1}^{k}\sum_{s\in S^i}\alpha_s^i x_s^i = \alpha x,\end{aligned}$$

hence $\alpha x > \alpha_0$. □

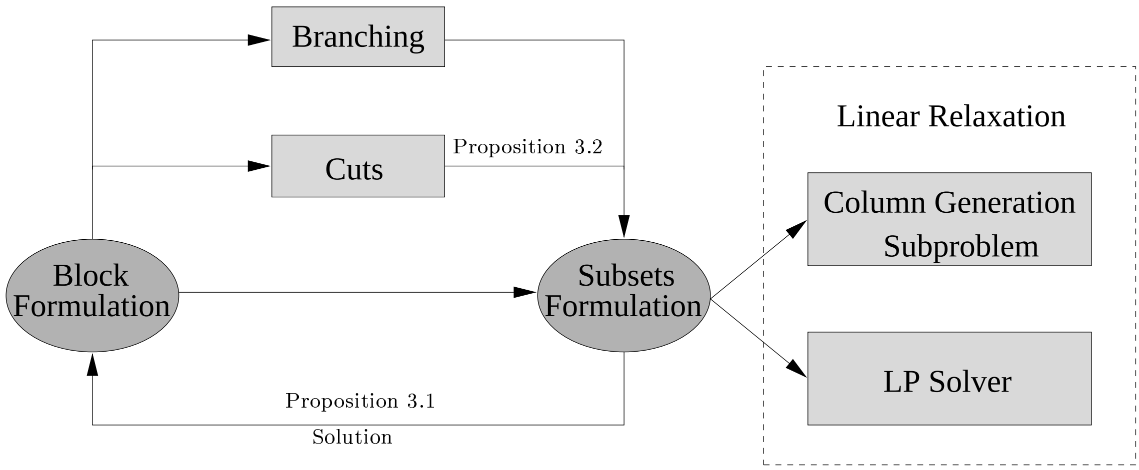

Figure 2: The combination scheme of Column Generation and Branch & Cut

Note that if all the cuts for (BF) necessary to prove that a solution y is optimal are available, then these cuts, transformed and added to (SF), would prove x to be optimal for (SF), as relation (3) translates solutions for (SF) into solutions for (BF) with equal cost. The technique to combine Column Generation and the methods to solve linear programming problems with integer variables is sketched in Figure 2 and summarized below:

- two formulations of the problem are used: the original Block Formulation (BF), and the alternative Subsets Formulation (SF);

- the linear relaxation of the Subsets Formulation is block structured and is efficiently solved by means of Column Generation; the Block Formulation is well characterized and has variables associated with elements in E;

- cuts that are valid for the Block Formulation can be taken into account in the generation of the variables with minimum reduced cost for the Subset Formulation (Theorem 3.3);

- solutions (possibly fractional) of the Subset Formulation are transformed into (possibly fractional) solutions for thre Block Formulation (Proposition 3.1);

- cuts determined in the Block Formulation are transformed and added to the Subset Formulation (Proposition 3.2).

Therefore, we use the Subsets Formulation to solve the linear relaxation of the problem using an LP solver and the pricing subproblem, and the Block Formulation to identify violated cuts and branching rules.

3.2. Column Generation

Here we recall how to solve the Column Generation Subproblem, i.e., how to find a column with minimum reduced cost for the Schedule Formulation. Let π and μ be the dual variables associated with the Balance Constraints and with the Schedule Constraints, respectively. The reduced cost of a variable x_s associated with schedule s is:

$$\bar{c}_s = c_s - \sum_{e=(i,j)\in E(s)} \sum_{k\in K(s,e)} (\pi_k(i) - \pi_k(j))q^k_{se} - \mu_{\sigma(s)},$$

where $K(s,e)$ is the set of commodities shipped by the schedule s on travel arc e. Due to the full load assumption, $\sum_{k\in K(s,e)} q^k_{se}$ is equal to the total capacity of the transport mean $\sigma(s)$ on that arc which we denote with $q_{\sigma(s)e}$.

Let (i,j) be an arc of the space-time graph $SG(\sigma)$, $k^* = \text{argmax}_{k\in K}\, \pi_k(i) - \pi_k(j)$, and s any schedule that uses arc (i,j), representing that mean σ is (totally or partially) loaded with a commodity $\tilde{k}$ such that $\pi_{k^*}(i) - \pi_{k^*}(j) > \pi_{\tilde{k}}(i) - \pi_{\tilde{k}}(j)$. We observe that the variable x_s has a reduced cost greater than the one of the variable associated with the schedule representing that transport mean σ is totally loaded with commodity k^*.

Therefore, as we supposed that c_s is the sum of the costs on the arcs in $E(s)$, it is possible to find the column with minimum reduced cost solving $|\mathcal{S}|$ shortest path problems on the space-time graphs, where costs on the arcs are given by

$$\bar{c}_{\sigma e} = c_{\sigma e} - \max_{k\in K}(\pi_k(i) - \pi_k(j))q_{\sigma e}$$

and then adding the value $-\mu_\sigma$ (note that $\mu_\sigma \leq 0$).

The above described weights are suitable for all the three models described in Section 2 if no inequalities are added to the problem. From the results of Section 3.1 we can derive the modifications to be applied when polyhedral techniques are integrated.

In the initial model (formulations (1) and (2)) and in the first extension of Section 2.3, that considers multiple loads without fixed capacity segregations, three steps must be followed: (a) for each commodity, add the contribution of all the dual variables associated with cuts where it is included the variable $y^k_{\sigma e}$ to the term $(\pi_k(i) - \pi_k(j))q_{\sigma e}$; (b) consider the commodity k maximizing the value of the sum computed in step (a); (c) add to the result of (b) the contribution of all the dual variables associated with cuts where the variable $v_{\sigma e}$ is included.

In the extension of the model described in Section 2.3, that takes into account segregations with fixed capacity, the above three steps must be applied to each segregation and then the results are to be summed. The commodity yielding the maximum value in step (b) is the one that must be loaded on the correponding segregation of the transport mean.

3.3. Knapsack subproblems and valid inequalities

In this section we derive knapsack constraints from the structure of the formulations described in Section 2. These knapsack problems are used, in the following sections, to determine valid inequalities.

We consider the more general case of the extension with fixed segregations presented in Section 2.3. The considerations made below obviously apply to the other cases described. We start from the following balance constraint of the Arc Formulation for a given commodity k, a consumption base p, and a day d:

$$z_{kpd} - z_{kp,d-1} - \sum_{\sigma\in\mathcal{S}} \sum_{e\in BS^{pd}_\sigma} \sum_{g\in G_\sigma} \tilde{q}^g_{\sigma e}\; y^{gk}_{\sigma e} + \hat{s}_{kpd} - \check{s}_{kpd} = b_{kpd}.$$

Adding the same relations for days $d-1,\ d-2,\ \dots,\ 1$, we obtain a formula for variable z_{kpd}; then, considering the maximum and the minimum stock levels, we obtain mixed integer knapsack problems. For instance, from $z_{kpd} \geq \check{Z}_{kpd}$ it derives:

$$z_{kpd} = \sum_{1\leq d'\leq d} \sum_{\sigma\in\mathcal{S}} \sum_{e\in BS_{\sigma}^{pd'}} \sum_{g\in G_{\sigma}} \tilde{q}_{\sigma e}^{g}\ y_{\sigma e}^{gk} - \sum_{1\leq d'\leq d} (\hat{s}_{kpd'} - \check{s}_{kpd'}) + \sum_{1\leq d'\leq d} b_{kpd'} \geq \check{Z}_{kpd}.$$

Writing all variables on the left hand side and all parameters on the right hand side, and dropping the continuous variables with negative sign, we get the following constraint:

$$\sum_{1\leq d'\leq d} \sum_{\sigma\in\mathcal{S}} \sum_{e\in BS_{\sigma}^{pd'}} \sum_{g\in G_{\sigma}} \tilde{q}_{\sigma e}^{g}\ y_{\sigma e}^{gk} + \sum_{1\leq d'\leq d} \check{s}_{kpd'} \geq \check{Z}_{kpd} - \sum_{1\leq d'\leq d} b_{kpd'}. \tag{6}$$

Assuming, for the moment, that the violation variables have value zero, the meaning of this constraint is that the total amount of commodity k delivered to base p before day d is greater than or equal to the minimum quantity that must be in p in that day plus the total amount consumed until day d (remember that $b_{kpd'}$ is negative for consumption).

Note that a process similar to the one discussed above can be applied to the case of production bases.

Besides the y variables considered in constraints (6), in the Arc Formulation are also present the v variables associated only with travels of the transport means, and not with commodities. We can derive knapsack constraints on v variables summing the knapsack constraints (6) on the different commodities that can be shipped from or towards the specified port. In fact, applying the above argument we obtain:

$$\sum_{k\in K} \sum_{1\leq d'\leq d} \sum_{\sigma\in\mathcal{S}} \sum_{e\in BS_{\sigma}^{pd'}} \sum_{g\in G_{\sigma}} \tilde{q}_{\sigma e}^{g}\ y_{\sigma e}^{gk} + \sum_{k\in K} \sum_{1\leq d'\leq d} \check{s}_{kpd'} \geq \sum_{k\in K} (\check{Z}_{kpd} - \sum_{1\leq d'\leq d} b_{kpd'})$$

Now we can modify the order of the sums getting the first sum on the commodities in the innermost position; considering the constraints which connect y variables to v variables in the Arc Formulation, that is $\sum_{k\in K} y_{\sigma e}^{gk} = v_{\sigma e}$, we derive the following knapsack constraint:

$$\sum_{1\leq d'\leq d} \sum_{\sigma\in\mathcal{S}} \sum_{e\in BS_{\sigma}^{pd'}} q_{\sigma e} v_{\sigma e} + \sum_{k\in K} \sum_{1\leq d'\leq d} \check{s}_{kpd'} \geq \sum_{k\in K} (\check{Z}_{kpd} - \sum_{1\leq d'\leq d} b_{kpd'}), \tag{7}$$

where $q_{\sigma e} = \sum_{g\in G_{\sigma}} \tilde{q}_{\sigma e}^{g}$ is the capacity of mean σ (for the travel denoted by arc e).

As the problem is a Mixed Integer Programming (MIP) problem, the cuts to be applied may be determined following two lines of search:

- find directly valid inequalities for Mixed Integer Programming problems;
- find valid inequalities for Integer Programming problems and then apply lifting procedures for the continuous variables.

In the second case there are three steps to consider: first, fix to zero the continuous variables; second, study the resulting integer program; finally, apply lifting procedures for the continuous variables to obtain inequalities which are valid for the original mixed integer problem.

In Section 4 we present a class of valid inequalities directly studied for MIP problems, while in Section 5 we analyse some issues on inequalities determined on Integer Programs, lifting procedures for the continuous variables, and their application to the problem discussed in this paper.

4. Nested Knapsack Inequalities

In this section we present a class of valid inequalities for mixed integer problems that, under certain hypotheses, has a facet-inducing property. We also discuss its application to the Tactical Distribution Problem in Supply Chain Management.

Definition 4.1. *A* Nested Knapsack System *is a set of knapsack constraints*

$$\begin{array}{llllll}
\sum_{i\in V_1} a_i x_i & & & +s_1 & & \geq b_1 \\
\sum_{i\in V_1} a_i x_i + \sum_{i\in V_2} a_i x_i & & & +s_1+s_2 & & \geq b_2 \\
\vdots & \ddots & & \vdots & \ddots & \vdots \\
\sum_{i\in V_1} a_i x_i + \sum_{i\in V_2} a_i x_i + & \cdots + \sum_{i\in V_n} a_i x_i & & +s_1+s_2 & \cdots + s_n & \geq b_n,
\end{array} \tag{8}$$

satisfying the following conditions: $b_0 = 0$, $b_h > b_{h-1}$ *for* $h = 1, \ldots, n$, $x_i \in \{0,1\}$ *for all* $i \in \bigcup_{h=1}^{n} V_h$, *and* $s_h \in \mathbb{R}_+$ *for all* $h = 1, \ldots, n$.

Note that the variables in constraint h appear with the same coefficient in constraints $k > h$.

Theorem 4.1. *The inequality*

$$\sum_{i\in V_1} \pi_i x_i + \sum_{i\in V_2} \pi_i x_i + \cdots + \sum_{i\in V_n} \pi_i x_i + s_1 + s_2 + \cdots + s_n \geq b_n,$$

where

$$\pi_i = \min\{a_i, b_n - b_{h-1}\} \text{ for all } i \in V_h,\ h = 1, \ldots, n.$$

is a facet inducing inequality for the Nested Knapsack System (8).

Proof: Validity of the inequality can be easily verified. The facet inducing property is proved by sequential lifting and induction on the number of binary variables. We refer to [33] and [32] for lifting procedures in 0-1 programming problems.

First, one can easily check that the theorem is true if $\sum_{h=1}^{n} |V_h| = 1$, as a basis for the induction process. Hence, let x_j with $j \in V_h$ be the variable to lift and, for each k, V_k' be the subset of V_k containing the variables already lifted at the current step. We have to show that the lifting coefficient is $\pi_j = \min\{a_j, b_n - b_{h-1}\}$. This coefficient can be found applying the following lifting procedure:

$$
\begin{array}{llllll}
z_j = \min & \sum_{i \in V_1'} \pi_i x_i + \sum_{i \in V_2'} \pi_i x_i + \cdots + \sum_{i \in V_n'} \pi_i x_i & +s_1 + s_2 & \cdots + s_n & \\
& \sum_{i \in V_1'} a_i x_i & +s_1 & & \geq b_1 \\
& \sum_{i \in V_1'} a_i x_i + \sum_{i \in V_2'} a_i x_i & +s_1 + s_2 & & \geq b_2 \\
& \vdots \quad \ddots & \vdots & \ddots & \vdots \\
& \sum_{i \in V_1'} a_i x_i + \sum_{i \in V_2'} a_i x_i + \cdots + \sum_{i \in V_{h-1}'} a_i x_i & +s_1 + s_2 + \cdots + s_{h-1} & & \geq b_{h-1} \\
& \sum_{i \in V_1'} a_i x_i + \sum_{i \in V_2'} a_i x_i + \cdots + \sum_{i \in V_h'} a_i x_i & +s_1 + s_2 + \cdots + s_h + & & \geq b_h - a_j \\
& \vdots & \vdots & & \vdots \\
& \sum_{i \in V_1'} a_i x_i + \sum_{i \in V_2'} a_i x_i + \cdots + \sum_{i \in V_n'} a_i x_i & +s_1 + s_2 + \cdots + s_n & & \geq b_n - a_j
\end{array}
$$

$$\pi_j = b_n - z_j.$$

As a first case, we suppose that $a_j \geq b_n - b_{h-1}$; then $b_n - a_j \leq b_{h-1}$ and, as $b_k - a_j \leq b_n - a_j$ for $h \leq k \leq n$, constraint $h-1$ dominates constraints from h to n. Thus a feasible solution for the lifting problem is $s_1 = b_{h-1}$, from which we obtain that $z_j \leq b_{h-1}$. Now we prove that $z_j = b_{h-1}$ holds, by showing that the following variables take value zero in the optimal solution of the lifting problem: (a) variables x_i for $i \in V_1'$ with $a_i \geq b_n$, and (b) variables x_i for $i \in V_k'$, $2 \leq k \leq h-1$ with $a_i \geq b_n - b_{k-1}$.

a) Suppose $x_i = 1$ for $i \in V_1'$ with $a_i \geq b_n$; then $z_j \geq \pi_i = b_n > b_{h-1}$; this contradicts what above stated, thus in every optimal solution $x_i = 0$ and $\sum_{i \in V_1'} \pi_i x_i + s_1 \geq b_1$.

b) Here we use induction on constraint k, $2 \leq k \leq h-1$. From the inductive hypothesis we know that, in every optimal solution, $x_i = 0$ for $i \in V_l'$ where $a_i \geq b_n - b_{l-1}$ for each $l < k$, and also that $\sum_{t \in V_1' \cup \ldots \cup V_{k-1}'} \pi_t x_t + s_1 + \ldots + s_{k-1} \geq b_{k-1}$. Now suppose $x_i = 1$ for $i \in V_k'$ with $a_i \geq b_n - b_{k-1}$: it would follow that $z_j \geq (\sum_{t \in V_1' \cup \ldots \cup V_{k-1}'} \pi_t x_t + s_1 + \ldots + s_{k-1}) + \sum_{t \in V_k'} \pi_t x_t + s_k \geq b_{k-1} + \pi_i = b_n$; from $b_n > b_{h-1}$ and $z_j \leq b_{h-1}$, we reach a contradiction, thus it must holds $x_i = 0$ for $i \in V_k'$.

We have thus shown that that left hand side of constraint $h-1$ is equal to the objective function in every optimal solution, therefore $z_j \geq b_{h-1}$, and consequently $z_j = b_{h-1}$ and $\pi_j = b_n - b_{h-1}$.

Consider now the case, $a_j < b_n - b_{h-1}$; then $b_n - a_j > b_{h-1}$, so a feasible solution is $s_1 = b_n - a_j$ and $z_j \leq b_n - a_j$. To show that $z_j = b_n - a_j$, we note that points (a) and (b) of the previous case can be repeated in a similar way; we then apply induction for $k \geq h$, as described in (c):

c) Suppose $x_i = 1$ for $i \in V_k'$ with $a_i \geq b_n - b_{k-1}$, then for inductive hypothesis it holds that

$$z_j \geq (\sum_{t \in V_1' \cup \ldots \cup V_{k-1}'} \pi_t x_t + s_1 + \ldots + s_{k-1}) + \sum_{t \in V_k'} \pi_t x_t + s_k;$$

the right hand side of the above inequality is greater than

$$\begin{array}{ll} b_{h-1} + \pi_i = b_{h-1} + (b_n - b_{h-1}) = b_n & \text{if } k = h \\ (b_{k-1} - a_j) + \pi_i = (b_{k-1} - a_j) + (b_n - b_{k-1}) = b_n - a_j & \text{if } k > h, \end{array}$$

In both cases $z_j \geq b_n - a_j$, so there exists an optimal solution with $x_i = 0$ for $i \in V'_k$ with $a_i \geq b_n - b_{k-1}$.

Consequently, the objective function takes the same value of the left hand side of constraint n, and thus $z_j \geq b_n - a_j$. As we have shown a feasible solution with this value, $z_j = b_n - a_j$, and $\pi_j = a_j$. This complete the proof. □

We conclude this section with an example.

Example 4.1. Consider the following three knapsack constraints:

$$\begin{array}{lll} \sum_{i \in V_1} a_i x_i & +s_1 & \geq 47 \\ \sum_{i \in V_1} a_i x_i + \sum_{i \in V_2} a_i x_i & +s_1 + s_2 & \geq 182 \\ \sum_{i \in V_1} a_i x_i + \sum_{i \in V_2} a_i x_i + \sum_{i \in V_3} a_i x_i & +s_1 + s_2 + s_3 & \geq 1081, \end{array}$$

where the coefficients a_i belong to the set $\{7000,\ 10400,\ 11000\}$. Applying Theorem 4.1, we get the following valid inequality:

$$182 \sum_{i \in V_1} x_i + 135 \sum_{i \in V_2} x_i + 1081 \sum_{i \in V_3} x_i + s_1 + s_2 + s_3 \geq 182. \tag{9}$$

□

4.1. Complete description for instances with small right hand side values

Consider the following Nested Knapsack Inequalities:

$$\begin{array}{lllll} \sum_{j \in V_1} \pi_j^1 x_j & & & +s_1 & \geq b_1 \\ \sum_{j \in V_1} \pi_j^2 x_j + \sum_{j \in V_2} \pi_j^2 x_j & & & +s_1 + s_2 & \geq b_2 \\ \vdots & \ddots & & \vdots \quad \ddots & \vdots \\ \sum_{j \in V_1} \pi_j^n x_j + \sum_{j \in V_2} \pi_j^n x_j + & \cdots + \sum_{j \in V_n} \pi_j^n x_j & & +s_1 + s_2 + \cdots + s_n & \geq b_n, \\ x_j \in \{0,1\} \text{ for all } j \in V_h & s_h \geq 0 \text{ for all } h = 1, \ldots, n & & & \end{array} \tag{10}$$

where, for each $k = 1, \ldots, n$:

$$\pi_j^k = \min\{a_j, b_k - b_{h-1}\} \text{ for all } j \in V_h,\ h = 1, \ldots, k.$$

Inequalities (10) have been shown to be valid for (8) by Theorem 4.1. Below, we prove that under some hypotheses the Nested Knapsack Inequalities are sufficient to describe the convex hull of (8).

Theorem 4.2. *If* $\pi_j^k = b_k - b_{h-1}$ *for each* $j \in V_h$, $h = 1, \ldots, k$, *and* $k = 1, \ldots, n$, *then inequalities (10),* $x_j \in [0,1]$ *for all* j, *and* $s_k \geq 0$ *for all* k *define an integer polyhedron.*

Proof: We proof the theorem showing that there exists an optimal solution (x, s) to the relaxation of (10) with binary x for each objective function (c, f). We write again the formulation of the problem considering explicitly the hypothesis of the theorem:

$$\begin{array}{llllll}
\sum_{j\in V_1} b_1 x_j & & & +s_1 & & \geq b_1 \\
\sum_{j\in V_1} b_2 x_j + \sum_{j\in V_2} (b_2 - b_1) x_j & & & +s_1 + s_2 & & \geq b_2 \\
\vdots & \ddots & & \vdots & \ddots & \vdots \\
\sum_{j\in V_1} b_n x_j + \sum_{j\in V_2} (b_n - b_1) x_j & \cdots + \sum_{j\in V_n} (b_n - b_{n-1}) x_j & & +s_1 + s_2 & \cdots + s_n & \geq b_n, \\
x_j \in [0,1] \text{ for all } j \in V_h \quad s_h \geq 0 \text{ for all } h = 1, \ldots, n
\end{array} \tag{11}$$

To prove the theorem we deduce some simplifications and then we find a pair of primal and dual optimal solutions satisfying the complementary slackness conditions.

We assume $c > 0$ and $f \geq 0$, otherwise variables x_j with $c_j \leq 0$ could be fixed to 1, while, if $f_k < 0$, the problem is unbounded.

For each k let $\bar{c}_k = \min\{c_j \ : \ j \in V_k\}$ and $ind(k) = \operatorname{argmin}\{c_j \ : \ j \in V_k\}$. Then, for all $k = 1, \ldots, n$ and $j \in V_k$, $x_j = 0$ for each $j \neq ind(k)$. Moreover, $\sum_{j \in V_k} x_j \leq 1$, because a single variable satisfies all the knapsack constraints from k to n. Hence, we may assume that there is a single variable in each subset V_k that we denote with x_k, and, moreover, that $f_1 \geq f_2 \geq \ldots \geq f_n$; indeed, if $f_i < f_j$ with $i < j$, then $s_j = 0$ in the optimal solution, as s_i is contained in all those constraints that contain also s_j.

For ease of reference we rewrite the problem taking into account all simplifying hypotheses above discussed:

$$\begin{array}{lllllll}
\min & c_1 x_1 + c_2 x_2 + & \ldots + c_n x_n & +f_1 s_1 + f_2 s_2 + & \cdots + f_n s_n & \\
& b_1 x_1 & & +s_1 & & \geq b_1 \\
& b_2 x_1 + (b_2 - b_1) x_2 & & +s_1 + s_2 & & \geq b_2 \\
& \vdots & \ddots & \vdots & \ddots & \vdots \\
& b_n x_1 + (b_n - b_1) x_2 + & \cdots + (b_n - b_{n-1}) x_n & +s_1 + s_2 + & \cdots + s_n & \geq b_n, \\
& x_i \in [0,1] \quad s_i \geq 0 \text{ for all } i = 1, \ldots, n
\end{array} \tag{12}$$

If a variable $x_k = 1$, then all constraints from k to n are satisfied. Hence, in the optimal solution there cannot be two variables x_{k_1} and x_{k_2} with $x_{k_1} = x_{k_2} = 1$, (indeed, if $k_1 < k_2$ then the solution $\tilde{x}$ obtained by x modifying only $\tilde{x}_{k_2} = 0$ is feasible and with a smaller objective function value). Therefore, the optimal solutions of (12) have the following form:

$$x_i = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{if } i \neq k \end{cases} \quad s_i = \begin{cases} b_i - b_{i-1} & \text{for } i = 1, \ldots, k-1 \\ 0 & \text{for } i = k, \ldots, n \end{cases} \tag{13}$$

for an index $k \in \{1, \ldots, n\}$.

The cost of such solutions depends exclusively on index k. We denote with $g(k)$ the objective function value of the solution with $x_k = 1$.

$$g(k) = \sum_{i=1}^{k-1} f_i (b_i - b_{i-1}) + c_k \quad \text{for } k = 1, \ldots, n \tag{14}$$

Moreover, we denote with $g(n+1)$ the solution with $x_i = 0$ for all $i = 1, \ldots, n$.

$$g(n+1) = \sum_{i=1}^{n} f_i(b_i - b_{i-1}). \tag{15}$$

The optimal solution of (12) can be found from the minimum of $g(k)$, for $k = 1, \ldots, n+1$. Let $k^* = \operatorname{argmin}\{g(k) : k = 1, \ldots, n+1\}$. Now, we show that there is a dual feasible solution which satisfies the complementary slackness conditions with the primal solution (13) for $k = k^*$, and hence, they are optimal solutions.

Note that in the relaxation of (12) the bounds $x_i \leq 1$ are not needed as the objective function coefficients are positive, so its dual is the following problem (we write in parenthesis the corresponding primal variables):

$$\begin{array}{llllllll}
(x_1) & y_1 b_1 & + & y_2 b_2 & + \cdots + & y_n b_n & \leq c_1 & \\
(x_2) & & & y_2(b_2 - b_1) & + \cdots + & y_n(b_n - b_1) & \leq c_2 & \\
\vdots & & & & \ddots & & \vdots & \\
(x_n) & & & & & y_n(b_n - b_{n-1}) & \leq c_n & \\
(s_1) & y_1 & + & y_2 & + \cdots + & y_n & \leq f_1 & \quad (16)\\
(s_2) & & & y_2 & + \cdots + & y_n & \leq f_2 & \\
\vdots & & & & \ddots & & \vdots & \\
(s_n) & & & & & y_n & \leq f_n & \\
 & \multicolumn{6}{l}{y_i \geq 0 \text{ for all } i = 1, \ldots, n} &
\end{array}$$

The complementary slackness conditions with the primal solution (13) for $k = k^*$ are:

$$\begin{array}{llllllll}
(s_1) & y_1 & + & y_2 & + \cdots & + \cdots + & y_n & = f_1 \\
(s_2) & & & y_2 & + \cdots & + \cdots + & y_n & = f_2 \\
\vdots & & & & \ddots & & & \vdots \\
(s_{k-1}) & & & & y_{k-1} & + \cdots + & y_n & = f_{k-1}
\end{array}$$

$$(x_k) \qquad y_k(b_k - b_{k-1}) + \cdots + y_n(b_n - b_{k-1}) = c_k.$$

We define the following dual solution for decreasing indices $r = n, n-1, \ldots, k+1$ as

$$y_r = \min\left(f_r - \sum_{j=r+1}^{n} y_j, \min\left\{ \frac{c_i - \sum_{j=r+1}^{n} y_j(b_j - b_{i-1})}{b_r - b_{i-1}} : i = 1, \ldots, r \right\} \right). \tag{17}$$

For index k, $k-1$, and $i < k-1$, we have the following formulas:

$$\begin{aligned}
y_k &= \frac{c_k - \sum_{j=k+1}^{n} y_j(b_j - b_{k-1})}{b_k - b_{k-1}} \\
y_{k-1} &= f_{k-1} - \textstyle\sum_{j=k}^{n} y_j \\
y_i &= f_i - f_{i+1}
\end{aligned}$$

The solutions are defined to satisfy the complementary slackness conditions, so we have to test only the dual feasibility. We show that it is a feasible dual solution in three steps: first we prove non negativity, then that constraint associated with (s_k) is satisfied, and finally that the constraints associated with the primal variables x_i for $i = 1, \ldots, k-1$ are satisfied.

We show non negativity of the variables y_r for $r = k, \ldots, n$ by backward induction from n to k. Clearly $y_n \geq 0$. Let us suppose that for inductive hypothesis $y_{r+1} \geq 0$, then we show that so it is y_r. There are two possible cases:

a) $y_{r+1} = f_{r+1} - \sum_{j=r+2}^{n} y_j$;

b) $y_{r+1} = \dfrac{c_{\bar{k}} - \sum_{j=r+2}^{n} y_j(b_j - b_{\bar{k}-1})}{b_{r+1} - b_{\bar{k}-1}}$ where $\bar{k}$ is the index associated with the minimum index i in the second part of the formula (17).

If case (a) occurs, then if $y_r = f_r - \sum_{j=r+1}^{n} y_j = f_r - f_{r+1} \geq 0$. In a similar way if y_r is equal to second part of the minimum of (17), then $y_r \geq 0$ as for condition (a) for all $i = 1, \ldots, r$, $c_i \geq \sum_{j=r+1}^{n} y_j(b_j - b_{i-1})$. If case (b) occurs, then if $\bar{k} \leq r$ then $y_r = 0$. Otherwise $\bar{k} = r+1$, and

$$y_r = \frac{c_{\tilde{k}} - \sum_{j=r+1}^{n} y_j(b_j - b_{\tilde{k}-1})}{b_r - b_{\tilde{k}-1}}$$

where $\tilde{k} \leq r$ is the argument of the minimum in the formula (17) for y_r. Substituing the value of y_{r+1},

$$y_r = \frac{c_{\tilde{k}} - \sum_{j=r+2}^{n} y_j\left(b_j - b_{\tilde{k}-1}\right) - \dfrac{c_{\bar{k}} - \sum_{j=r+2}^{n} y_j\left(b_j - b_{\bar{k}-1}\right)}{b_{r+1} - b_{\bar{k}-1}}(b_{r+1} - b_{\tilde{k}-1})}{b_r - b_{\tilde{k}-1}}.$$

Hence $y_r \geq 0$, as if we multiply both sides for the positive quantity $\frac{b_r - b_{\tilde{k}-1}}{b_{r+1} - b_{\tilde{k}-1}}$, we obtain that the right hand side is equal to the following value

$$\frac{c_{\tilde{k}} - \sum_{j=r+2}^{n} y_j(b_j - b_{\tilde{k}-1})}{b_{r+1} - b_{\tilde{k}-1}} - \frac{c_{\bar{k}} - \sum_{j=r+2}^{n} y_j(b_j - b_{\bar{k}-1})}{b_{r+1} - b_{\bar{k}-1}} \geq 0$$

due to the definition of $\bar{k}$.

The constraints associated with s_i and x_i for $i = k+1, \ldots, n$ are satisfied for definition of the y variables. Assume that constraint (s_k) is not satisfied. Then, $y_k > f_k - \sum_{j=k+1}^{n} y_j$ and $c_k - \sum_{j=k+1}^{n} y_j(b_j - b_{k-1}) > f_k(b_k - b_{k-1}) - \sum_{j=k+1}^{n} y_j(b_k - b_{k-1})$ from which we obtain

$$c_k - f_k(b_k - b_{k-1}) > \sum_{j=k+1}^{n} y_j(b_j - b_k). \tag{18}$$

As $g(k) = \sum_{j=1}^{k-1} f_j(b_j - b_{j-1}) + c_k$, we derive c_k and substitute it in (18) obtaining

$$g(k) > \sum_{j=1}^{k} f_j(b_j - b_{j-1}) + \sum_{j=k+1}^{n} y_j(b_j - b_k).$$

Now, let $\bar{k} > k$ be the minimum index such that $y_{\bar{k}} = \dfrac{c_i - \sum_{j=\bar{k}+1}^{n} y_j(b_j - b_{i-1})}{b_{\bar{k}} - b_{i-1}} > 0$, where $i \in \{k, \ldots, \bar{k}\}$. Note that $y_i = y_{i+1} = \ldots = y_{\bar{k}-1} = 0$, while $y_j = f_j - f_{j+1}$ for $j = k+1, \ldots, i-2$

and $y_{i-1} = f_{i-1} - \sum_{j=\bar{k}}^{n} y_j$. Therefore,

$$g(k) > \sum_{j=1}^{k} f_j(b_j - b_{j-1}) + \sum_{j=k+1}^{i-2} (f_j - f_{j+1})(b_j - b_k) + (f_{i-1} - \sum_{j=\bar{k}}^{n} y_j)(b_{i-1} - b_k) + \sum_{j=\bar{k}}^{n} y_j(b_j - b_k) =$$

$$= \sum_{j=1}^{k} f_j(b_j - b_{j-1}) + \left[\sum_{j=k+1}^{i-2} f_j(b_j - b_k) - \sum_{j=k+2}^{i-1} f_j(b_{j-1} - b_k) \right] +$$

$$+(f_{i-1} - \sum_{j=\bar{k}}^{n} y_j)(b_{i-1} - b_k) + \sum_{j=\bar{k}}^{n} y_j(b_j - b_k) =$$

$$= \sum_{j=1}^{k} f_j(b_j - b_{j-1}) + f_{k+1}(b_{k+1} - b_k) + \sum_{j=k+2}^{i-2} f_j(b_j - b_{j-1}) - f_{i-1}(b_{i-2} - b_k) +$$

$$+f_{i-1}(b_{i-1} - b_k) - \sum_{j=\bar{k}}^{n} y_j(b_{i-1} - b_k) + \sum_{j=\bar{k}}^{n} y_j(b_j - b_k) =$$

$$= \sum_{j=1}^{i-1} f_j(b_j - b_{j-1}) + \sum_{j=\bar{k}}^{n} y_j(b_j - b_{i-1}) =$$

$$= \sum_{j=1}^{i-1} f_j(b_j - b_{j-1}) + \frac{c_i - \sum_{j=\bar{k}+1}^{n} y_j(b_j - b_{i-1})}{b_{\bar{k}} - b_{i-1}}(b_{\bar{k}} - b_{i-1}) + \sum_{j=\bar{k}+1}^{n} y_j(b_j - b_{i-1}) =$$

$$= \sum_{j=1}^{i-1} f_j(b_j - b_{j-1}) + c_i = g(i)$$

that is, $g(k)$ is not the minimum as we supposed. Therefore, constraint (s_k) is satisfied.

Finally, we show that for $i = 1, \ldots, k-1$ the constraints of the dual problem associated with primal variables x_i are satisfied. Consider constraint (x_i):

$$\sum_{j=i}^{n} y_j(b_j - b_{i-1}) \leq c_i.$$

By substituing the value of the variables y_j for $j = 1, \ldots, k$ in the left hand side we obtain

$$\sum_{j=i}^{k-2} (f_j - f_{j+1})(b_j - b_{i-1}) + (f_{k-1} - \sum_{j=k}^{n} y_j)(b_{k-1} - b_{i-1}) + y_k(b_k - b_{i-1}) + \sum_{j=k+1}^{n} y_j(b_j - b_{i-1}) =$$

$$= \sum_{j=i}^{k-2} f_j(b_j - b_{j-1}) - f_{k-1}(b_{k-2} - b_{i-1}) + f_{k-1}(b_{k-1} - b_{i-1}) + y_k(b_k - b_{k-1}) + \sum_{j=k+1}^{n} y_j(b_j - b_{k-1}) =$$

$$= \sum_{j=i}^{k-1} f_j(b_j - b_{j-1}) + \frac{c_k - \sum_{j=k+1}^{n} y_j(b_j - b_{k-1})}{b_k - b_{k-1}}(b_k - b_{k-1}) + \sum_{j=k+1}^{n} y_j(b_j - b_{k-1}) =$$

$$= \sum_{j=i}^{k-1} f_j(b_j - b_{j-1}) + c_k \leq c_i$$

Combining the above inequality with the definition of $g(k)$ we obtain

$$\sum_{j=i}^{k-1} f_j(b_j - b_{j-1}) + c_k = g(k) - \sum_{j=1}^{i-1} f_j(b_j - b_{j-1}) \leq c_i,$$

which can be rewritten as

$$g(k) \leq c_i + \sum_{j=1}^{i-1} f_j(b_j - b_{j-1}) = g(i).$$

We have thus shown that constraint (x_i) is satisfied if and only if $g(k) \leq g(i)$, which is obviously true as by definition $k = k^* = \text{argmin}\{g(i) \,:\, i = 1, \ldots, n+1\}$. □

4.2. Application of Nested Knapsack Inequalities

In Section 3.3 we derived knapsack constraints from the distribution problem in Supply Chain Management presented in Section 2. If we consider a subset of constraints (7) associated with different days for a given port p and a commodity k, we have a Nested Knapsack System at hand as defined in Section 4. In the following example we apply the inequalities of Example 4.1.

Example 4.2. Consider the following three inequalities of type (7):

$$\begin{array}{llll}
\sum_{(\sigma,e)\in V_1} q_{\sigma e} v_{\sigma e} & & +s_1 & \geq 47 \\
\sum_{(\sigma,e)\in V_1} q_{\sigma e} v_{\sigma e} + \sum_{(\sigma,e)\in V_2} q_{\sigma e} v_{\sigma e} & & +s_1 + s_2 & \geq 182 \\
\sum_{(\sigma,e)\in V_1} q_{\sigma e} v_{\sigma e} + \sum_{(\sigma,e)\in V_2} q_{\sigma e} v_{\sigma e} + \sum_{(\sigma,e)\in V_3} q_{\sigma e} v_{\sigma e} & & +s_1 + s_2 + s_3 & \geq 1081,
\end{array}$$

where $V_i = \{(\sigma, e) | \sigma \in \mathcal{S} \text{ and } e \in BS_\sigma^{pd'} \text{ for } i = 1, \ldots, d_i\}$, $d_i \in D$, $i = 1, 2, 3$.

As in Example 4.1, we suppose that the coefficients $q_{\sigma e} \in \{7000, 10400, 11000\}$, and, applying Theorem 4.1, we find an inequality equivalent to (9):

$$1081 \sum_{(\sigma,e)\in V_1} v_{\sigma e} + 1034 \sum_{(\sigma,e)\in V_2} v_{\sigma e} + 852 \sum_{(\sigma,e)\in V_3} v_{\sigma e} + s_1 + s_2 + s_3 \geq 1081. \tag{19}$$

□

It is straightforward that the same applies starting from knapsack constraints on a single commodity with the y variables (i.e., (6)). In Section 2.3 we presented an extension where the integrality constraints on the y variables were relaxed. In some real cases this extension may be applied as the transport means can be divided into segregations in several ways, and each fractional value of the y variables can be easily approximated with a real configuration of the segregations. In such a case we can again find Nested Knapsacks Inequalities on each commodity substituting the y variables with the v variables in (6), which remain integer even in this relaxed formulation.

As the following relations hold

$$\sum_{k\in K} y_{\sigma e}^k = v_{\sigma e} \text{ for all } \sigma \in \mathcal{S}$$

(see the Arc Formulation in sections 2.2 and 2.3), then from (6) we obtain a valid inequality substituting $q_{\sigma e}v_{\sigma e}$ with $q_{\sigma e}y^k_{\sigma e}$, and then we can apply the results of Section 4. Indeed, $q_{\sigma e}v_{\sigma e} \geq q_{\sigma e}y^k_{\sigma e}$, and (6) is a "$\geq$" inequality; so with this substitution we get a relaxation of (6).

Example 4.3. Suppose that in the case of Example 4.2 we have constraints on two commodities. For commodity $k = 0$ the total amounts to be shipped are 47, 182, and 670 tons for the three knapsack constraints, respectively. For commodity $k = 1$ the total amounts are 0 on the first and on the second constraint, and 411 tons on the third one. Making the above described substitution and considering the definition of Nested Knapsack Inequalities, we obtain the following cuts:

- for commodity 0:

$$670 \sum_{(\sigma,e)\in V_1} v_{\sigma e} + 623 \sum_{(\sigma,e)\in V_2} v_{\sigma e} + 488 \sum_{(\sigma,e)\in V_3} v_{\sigma e} + s_1^0 + s_2^0 + s_3^0 \geq 670; \qquad (20)$$

- for commodity 1:

$$411 \sum_{(\sigma,e)\in V_1} v_{\sigma e} + 411 \sum_{(\sigma,e)\in V_2} v_{\sigma e} + 411 \sum_{(\sigma,e)\in V_3} v_{\sigma e} + s_1^1 + s_2^1 + s_3^1 \geq 411. \qquad (21)$$

□

In Example 4.2 the violation variables s_i were associated with all the commodities, that is $s_i = \sum_{k\in K} s_i^k$. We see that inequality (19), obtained in Example 4.2, is the sum of the previous two inequalities. Hence, (19) is redundant if (20) and (21) are added to the problem.

We note that, when the coefficients of a Nested Knapsack Inequality on all the commodities and the coefficients of the corresponding Nested Knapsack Inequalities on each commodity are strictly less than the coefficients $q_{\sigma e}$, then the former inequality is the sum of the latter ones. In general, a cut on all the commodities is not always the sum of the cuts on the single commodities, as some coefficients of the Nested Knapsack Inequalities may be set to the original coefficients in the former case.

The substitution of the y variables with the v variables can also be done partially. Therefore, the coefficients in the cut of the remaining y variables are equal to the original coefficients q, while the coefficients of the v variables are equal to π as defined in Theorem 4.1. When adding a specific cut on a commodity, port, and day, the substitution of some of the y variables with the v variables is done in the following way:

a) let $y^k_{\sigma e}$ be a y variable with coefficient $q_{\sigma e}$;

b) let $v_{\sigma e}$ be the corresponding v variable with coefficient $\pi_{\sigma e}$ for the cut;

c) if, for the current fractional solution, $\pi_{\sigma e}v_{\sigma e} < q_{\sigma e}y^k_{\sigma e}$, then consider the term $\pi_{\sigma e}v_{\sigma e}$ in the cut, otherwise consider the term $q_{\sigma e}y^k_{\sigma e}$.

As for each pair (σ, e) we choose the minimum between $\pi_{\sigma e}v_{\sigma e}$ and $q_{\sigma e}y^k_{\sigma e}$, the cut obtained is the most violated among all Nested Knapsack Inequalities for the same commodity, port, and day.

Finally, we note that it is possible to define this class of cuts not only on all the commodities or on a single commodity, but also on a generic subset of commodities. The procedure to apply is similar to the one for a single commodity; it is sufficient to consider a sum of y variables on the subset of commodities instead of a single y variable; all other concepts and operations are still valid.

5. Cover Inequalities for Knapsack Constraints

In Section 3.3 we have derived knapsack constraints (6) and (7) from the structure of the distribution problem in Supply Chain Management. As a consequence, we can apply cuts on knapsack problems, and in particular the well-known lifted cover inequalities (see, e.g., the papers [2], [26], [44], [45], [14] or the textbooks [32], [46]). Other classes of valid inequalities for knapsack problems can be found in [34] ($(1, k)$ configurations) and in [43] (extended weight inequalities).

This section deals with the application of cover inequalities. Below we briefly recall the definition of cover inequalities for knapsack problems with general integer variables.

The following constraints

$$\begin{array}{ll} \sum_{i=1}^{n} a_i x_i \leq b & \\ 0 \leq x_i \leq u_i & \forall\, i = 1, \ldots, n \\ x_i \in \mathbb{Z} & \forall\, i = 1, \ldots, n, \end{array}$$

with a_i, u_i, and b non negative integers, describe the set of feasible points of the *Integer Knapsack Problem.*

Definition 5.1. *A cover for an integer knapsack problem is a subset of items $C \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in C} a_i u_i > b$. Given a cover C the following is defined in [8] as cover inequality:*

$$\sum_{i \in C} (u_i - x_i) \geq \alpha, \tag{22}$$

where $\alpha = \lceil \lambda / \bar{a} \rceil$, $\lambda = \sum_{i \in C} a_i u_i - b$, and $\bar{a} = \max\{a_i \mid i \in C\}$.

In this paper we are interested in the application of cover inequalities on mixed integer knapsack problems. For such cases techniques to lift a single continuos variable have been presented in [8].

In the remainder of this section we present a simple algorithm to efficiently compute the lifting coefficient defined in the lifting procedure of [8]. Moreover, we show how cover inequalities for knapsack problems derived in Section 3.3 can be strengthened taking into account certain compatibility conditions amongst travels carried out by the same transport mean.

5.1. Lifting procedure for continuous variables

Lifting on the continuous violation variables can be carried out using the formula introduced in [8], which is valid for the case of a single continuous variable. In the knapsack constraints derived from the problem of Section 2 there are more than one continuous variable; some of them have positive coefficients and some other negative coefficients, so we have to apply a transformation to reduce to the case of a single continuous variable. We consider knapsack constraints of the type "$\leq$" (the case "$\geq$" can be derived straightforwardly):

$$ax + \sum_i \hat{s}_i - \sum_i \check{s}_i \leq b. \tag{23}$$

The transformations needed to obtain constraints with a single continuous variable from (23) are:

Step 1: eliminate the continuous variables with positive sign ($\hat{s}_i$) in the left hand side (*lhs*), and move the continuous variables with negative sign ($\check{s}_i$) to the right hand side; as the original contraint has been relaxed in a new constraint by dropping the variables $\hat{s}_i$, valid inequalities for the latter are valid also for the former;

Step 2: let s be a new variable equal to the sum of the continuous variables still in the constraint ($s = \sum_i \check{s}_i$); then this becomes of the type $ax \leq b + s$;

Step 3: let $\pi x \leq \pi_0$ be the inequality to be lifted; find the lifting coefficient γ for s with the procedure described next, and obtain the valid inequality $\pi x \leq \pi_0 + \gamma s$;

Step 4: substitute s with the original continuous variables with negative sign in the *lhs* ($\check{s}_i$), and lift with zero coefficients the continuous variables with positive sign in the *lhs* ($\hat{s}_i$).

The lifting coefficient γ for s is equal to $1/\beta$ [8], where

$$\beta = \min_{s>0} \frac{s}{\eta(s) - \pi_0},$$

and

$$\eta(s) = \max\{\pi x \ : \ ax \leq b + s, \ x \in \{0,1\}^n\}.$$

The function $\eta(s)$ is defined on a continuous dominion. To simplify the computation of β we study the properties of the function $\beta(s)$ defined as follows:

$$\beta(s) = \frac{s}{\eta(s) - \pi_0},$$

and we propose a simple algorithm that carries out such a computation. The value of β will be the minimum of $\beta(s)$ over all $s > 0$. The function $\eta(s)$ is a non decreasing function with a staircase shape as the value of $\max\{\pi x \ : \ ax \leq b + s, \ x \in \{0,1\}^n\}$ is constant for an interval of values for s, and then goes up when s is increased to a value which allows a better solution for $\max\{\pi x \ : \ ax \leq b + s, \ x \in \{0,1\}^n\}$.

As the function $\eta(s)$ is a staircase function, then $\beta(s)$ is a piecewise growing function; that is, in each interval such that $\eta(s)$ is constant, $\beta(s)$ is a growing linear function with slope equal to $1/(\eta(s) - \pi_0)$ (see Figure 3).

On these premises, we can compute $\beta(s)$ only on its breakpoints (black dots in Figure 3). Therefore, we first consider all the possible values that πx can take in the feasible region of $\eta(s)$; they are in the set $\{\pi_0 + 1, \ \ldots, \ \pi e\}$, where e is the vector of all ones. Then, we compute the least value of $b+s$ needed for $\eta(s)$ to take a value $\tilde{\pi} \in \{\pi_0+1, \ \ldots, \ \pi e\}$, and finally compute $\beta(s)$ only for that value. We can summarize this result in the procedure described in the following proposition.

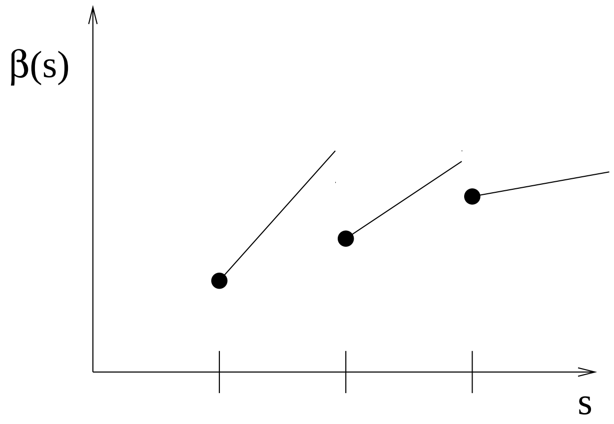

Figure 3: Graphic of function $\beta(s)$

Proposition 5.1. *The following algorithm correctly computes the value of* β*:*

Step 1. $\beta = \infty$; $\tilde{\pi}_0 = \pi_0 + 1$;
Step 2. while $(\tilde{\pi}_0 \leq \pi e)$ *do*

$$\xi(\tilde{\pi}_0) = \min\{ax \;:\; \pi x \geq \tilde{\pi}_0,\; x \in \{0,1\}^n\}$$
$$\beta = \min\left\{\beta, \frac{\xi(\tilde{\pi}_0) - b}{\tilde{\pi}_0 - \pi_0}\right\}$$
$$\tilde{\pi}_0 = \pi\tilde{x} + 1, \text{ where } \tilde{x} = \operatorname{argmin}\{ax \;:\; \pi x \geq \tilde{\pi}_0,\; x \in \{0,1\}^n\}$$

Thus we optimize the function $\beta(s)$, defined over a continuous dominion, by solving a number of binary knapsack problems. These knapsack problems can be easily solved as the coefficients in the constraints are those of the valid inequality to lift (i.e., π). In many cases, such as the lifted cover inequalities, most of the coefficients π are equal to zero or one, so the dynamic programming algorithm for knapsack can solve the problem very efficiently.

Another way to find β is by computing a value $\beta' \leq \beta$ that can be used to lift the inequality obtaining a weaker valid inequality. This can be easily done by the greedy algorithm that first solves the continuous relaxation of the knapsack problems in the algorithm of Proposition 5.1, and then rounds up the objective function value obtaining $\xi'(\tilde{\pi}_0) \leq \xi(\tilde{\pi}_0)$. This value is a good approximation due to the properties of the coefficients π: indeed, if π is a zero-one vector, the continuous relaxation has an integer solution and rounding is not needed. Moreover, to solve the problem with the next value of $\tilde{\pi}_0$ we can start from the previous solution; this way, provided that the n items are in descending order with respect to π_i/a_i and that the relaxations of the knapsack problems are used, the complexity of the algorithm in Proposition 5.1 is $O(n)$.

5.2. Compatibility Cover Inequalities

In the previous section we deduced binary knapsack constraints from the structure of the problem. The variables in these constraints represent travels and assignment of products to travels. The cuts from the literature (cover, $(1, k)$-configurations, extended weight inequalities, etc.) do not take into account compatibility conditions among travels, i.e., if a transport mean can carry out two travels respecting their initial and final times.

Using this additional feature, we can strengthen the already described cover inequalities. This is shown by the following example coming from contraints (7):

Example 5.1. Consider a consumption base p and a day d for which the total request of all products is 65,900 tons (including minimum stock and consumptions). There are two transport means that can carry products towards base p: mean 1 can make 14 travels and its capacity is 11,000 tons; mean 2 can make 15 travels and its capacity is 10,000 tons. The knapsack constraint of type (7) is the following (we dropped continuous variables):

$$\sum_{i=1}^{14} 11,000\, v_{1i} + \sum_{i=1}^{15} 10,000\, v_{2i} \geq 65,900$$

A lifted cover inequality that we may obtain starting from a cover containing all the variables of transport mean 2 is

$$\sum_{i=1}^{14} v_{1i} + \sum_{i=1}^{15} v_{2i} \geq 6 \qquad (24)$$

In Table 5.1 we report the list of the 14 loaded travels corresponding to the variables associated with transport mean 1. Moreover, to complete the data in the table, we add that the return travels from base 7 to base 3 last two days, from 7 to 5 last three days, and from 7 to 11 last two days. From the condition on initial and final times for each travel, we know, for instance, that tranport mean 1 cannot perform travels 1 and 3 together.

travel	*FROM*		*TO*	
no.	*port*	*on day*	*port*	*on day*
1	3	2	7	5
2	3	3	7	6
3	3	7	7	10
4	3	8	7	11
5	3	9	7	12
6	3	12	7	15
7	11	12	7	16
8	5	16	7	18
9	11	17	7	21
10	3	18	7	21
11	3	23	7	27
12	11	23	7	27
13	5	24	7	27
14	3	24	7	27

Table 1: List of selected travels associated with transport mean 1

The relations of incompatibility among travels can be represented with a graph for each transport mean, where nodes are the travels and an arc between two travels denotes that they cannot be performed together by the corresponding mean in a monthly plan. In Figure 5.1 we show the graph associated with transport mean 1.

The travels that can be in a schedule of a transport mean correspond to a stable set in the previous graph. We can see that transport mean 1 can carry out at most 5 travels (see for instance grey nodes in Figure 5.1) covering only 55,000 tons; thus, the residual 10,900 tons

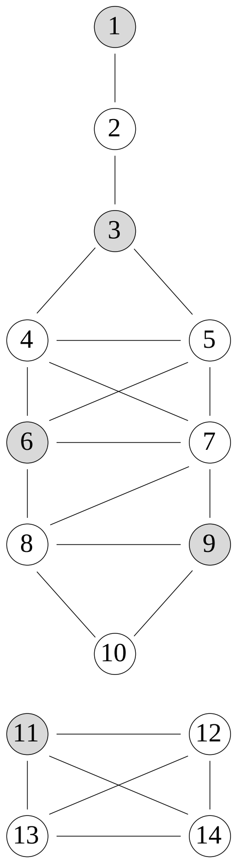

Figure 4: Incompatibility graph associated with transport mean 1

must be covered by two travels of transport mean 2. Therefore, 6 travels are not sufficient and, consequently, the following inequality is valid for the whole problem:

$$\sum_{i=1}^{14} v_{1i} + \sum_{i=1}^{15} v_{2i} \geq 7 \tag{25}$$

□

We call the cuts as the one derived in the Example 5.1 *Compatibility Cover Inequalities.* Now we describe a general procedure to obtain this type of cuts considering general integer knapsack constraints. First, we discuss how to get this type of constraints from problem (2).

We start from constraints (7), but we could start also from a constraint of the type (6) as well, and then introduce new variables z_σ equal to the sum of a subset $\tilde{E}_\sigma$ of variables associated with ship σ. As in real applications the coefficients $q_{\sigma e}$ seldom depend on e, we assume $q_{\sigma e} = q_\sigma$ such that $e \in \tilde{E}_\sigma$ and then use the following relation to substitute variables in $\tilde{E}_\sigma$ with the single integer variable z_σ:

$$\sum_{e \in \tilde{E}_\sigma} q_{\sigma e} v_{\sigma e} = q_\sigma z_\sigma .$$

We have obtained a knapsack constraint on variables z_σ on which we have to define upper bounds. These are given by the maximum number of travels that ship σ can make satisfying compatibility constraints between travels. For each ship this value is found by solving a maximum stable set problem on the incompatibility graph defined as in Example 5.1 for ship 1. This problem can be solved in linear time, as it can be reduced to a maximum path problem on the acyclic graph $SG(\sigma)$ introduced in Section 2, where we set weights on arcs as follows:

- 1 for all arcs $e \in \tilde{E}_\sigma$;
- 0 otherwise.

The maximum path on this graph results in the largest subset of compatible travels in $\tilde{E}_\sigma$. The process is described below continuing Example 5.1.

Example 5.2. *(cont.)* From Example 5.1 we can define two variables z_1 and z_2, such that $z_1 = \sum_{i=1}^{14} v_{1i}$ and $z_2 = \sum_{i=1}^{15} v_{2i}$. The bound for z_1 is 5 (e.g., travels 1, 3, 6, 9, and 11 in Figure 5.1). Similarly we obtain that the one for z_2 is 7. The knapsack problem is then the following

$$\begin{array}{rlcrll} 11,000 & z_1 & + & 10,000 & z_2 & \geq 65,900 \\ & z_1 & & & & \leq 5 \\ & & & & z_2 & \leq 7 \\ & z_1, z_2 \in \mathbb{Z}_+. & & & & \end{array}$$

If we complement the variables by setting $\bar{z}_i = u_i - z_i$, we obtain an equivalent knapsack problem of type "$\leq$". Therefore, using Definition 5.1 we derive the following inequality:

$$(7 - \bar{z}_2) \geq 2.$$

Now lifting $\bar{z}_1$ with coefficient 1 we obtain

$$(5 - \bar{z}_1) + (7 - \bar{z}_2) \geq 7.$$

Complementing back the variables we then have:

$$z_1 + z_2 \geq 7,$$

that is equivalent to (25), and stronger than (24). □

Hence, the class of Compatibility Cover Inequalities is defined by:

a) reformulating the knapsack constraints with the aggregating variables z_σ;

b) computing tight bounds on the z_σ by solving maximum path problems in acyclic graphs;

c) finding lifted cover inequalities on the resulting knapsack problem with general integer variables.

Then, the resulting cover inequalities may be lifted on the variables not considered with the sets $\tilde{E}_\sigma$ using standard lifting procedures (see [33] or the textbook [32]), which involve the solution of a series of knapsack problems.

6. An Application to Ship Scheduling in Oil Industries

We describe an application of the model and of the solution algorithm for the Tactical Distribution Problem in Oil Industry Supply Chain Management. First, we describe the problem to be solved. Then, we discuss the branching rules that have been implemented. Finally, we present the computational results obtained on test and real cases provided by two of the major oil production and distribution companies: the italian AgipPetroli and the spanish CLH (Compañia Logistica de Hidrocarburos).

Distribution of oil products from refineries to depots for supplying large geographical areas is performed by means of medium and small oil tankers ranging from 5,000 to 25,000 tons of deadweight. A set of different commodities (e.g., petrol, Diesel oil, etc.) is to be shipped amongst a set of ports (associated with refineries or depots). A given number of ships, with known characteristics, can be used. All ports have constraints on stock level and daily production or consumption rates for several commodities. The problem is to move the commodities between the ports so that the stock level constraints are respected in a given planning period (e.g., one month), and the total cost of transportation is minimized. The data available to describe the problem are summarized below.

For each port we know the distances from the other ports; furthermore, for each commodity we are given:

- the initial stock level;
- the minimum and the maximum stock levels;
- the amount produced or consumed for each day of the planning period.

Moreover, we have information and constraints associated with the ships:

- the commercial speed;
- the number of segregations and their load (a segregation is a compartment that can be filled with only one product);
- the constraints on ports (some large ships cannot harbour in little ports or they can only if not fully loaded).

The ships are divided into two categories with a different cost structure:

- *time charter ships*: rented on a yearly or monthly basis;
- *spot ships*: rented just for specific travels.

The objective function aims to minimize the use of spots ships, which are more expensive, optimizing the use of time charter ships, for which fixed costs prevail. This is done in the following way. First, the set of ships used in the model is composed of time charter ships only, and violation variables are assigned large costs in the objective function. The violation variables with positive value in the optimal solution associated with this problem are used to make decisions on how many spot ships and of which type are to be rented. Then, the algorithm is executed once again considering also a number of spot ships which are capable to take care of the violations.

Branching Rules

Here we describe some branching rules that can be applied to the Ship Scheduling application, and generalized to all the cases where the tactical model of Section 2 applies. These rules must have the property of being compatible with the column generation routine, therefore we consider only rules that can be represented by changing the costs associated with nodes and arcs of the space-time graphs. A first example of this kind of rules is the following:

- select a ship σ, a port p and a day d in the planning period;
- on the first branch, force the ship σ to pass through port p on day d;
- on the second branch, instead, the ship σ is not allowed to go through port p on day d.

As a pair of port and day is a node of the ship-graph $SG(\sigma)$, this rule amounts to considering only paths passing through that node in the first case, and only paths not going through that node in the second one.

A slight modification of this rule is the following one:

- select a ship σ, a port p, a day d and a sense: *outwards* or *inwards*;
- if the outwards sense is chosen, on the first branch, we impose the constraint that the sum of the arcs corresponding to loaded travels of the ship σ leaving from the selected node (p, d) must be equal to one, and
- on the second branch, we impose that the above sum is fixed to zero.

It is straightforward to define the case of the inwards sense.

Computational Results

We have solved some instances of the Ship Scheduling problem with the algorithm discussed in Section 3. The model we have adopted for these tests is the one described in Section 2 considering the extension with multiple loads on the same transport mean without fixed segregations described in Section 2.3.

We report on four test cases. We used the classes of cuts derived in sections 4 and 5, and the second of the above branching rules. The first two test cases, CLH1 and CLH2, have been proposed by CLH (Compañia Logistica de Hidrocarburos of the Repsol group), the main company involved in the distribution of oil products in Spain. The other two test cases, AP1 and AP2, have been studied by Agip Petroli (ENI group), the leading Italian company both for the production and distribution of oil derivatives.

In Table 2 we report for each test case the number of commodities, the number of days in the planning period, the number of ports, the number of ships, the lower bounds without and with the use of cutting planes, the optimal solution, the number of nodes in the branching tree, and the total computational times.

All the above test cases come from real planning problems of the two companies. However in these cases it is possible to produce solutions which satisfy all stock constraints without using violation variables.

We conclude reporting on the results on a real instance corresponding to the complete distribution plan of AgipPetroli in one month. In this instance there are 12 commodities, 30 days, 19 ports, and 10 ships. The lower bound obtained without cuts is 52,231, the one obtained with the

	Lower Bounds								
	K	D	P	S	LP	Cuts	Optimum	Nodes	Time
CLH1	4	6	2	4	0.437	2.000	2	1	2
CLH2	4	30	5	4	6.935	8.467	10	167	36
AP1	5	30	8	6	6.266	13.000	13	3	52
AP2	5	30	8	6	50.000	50.000	50	57	74

Table 2: Computational Results on Test Cases

cuts is 96,500, and, finally, we reach a feasible solution with value 104,107. Therefore, in this case, where violations variables have been included, the gap is reduced from 50% to 7%. The feasible solution determined by the algorithm has been positively evaluated by the potential final users (AgipPetroli), and it exhibited significant savings with respect to the solution obtained with the currently used method.

References

[1] Y. Agarwal, K. Mathur, and H. Salkin, "A Set-Partitioning-Based Exact Algorithm for the Vehicle Routing Problem," *Networks*, vol. 19, pp. 731–749, 1989.

[2] E. Balas, "Facets of the Knapsack Polytope," *Mathematical Programming*, vol. 8, pp. 146–164, 1975.

[3] M. L. Balinski and R. E. Quandt, "On an Integer Program for a Delivery Problem," *Operations Research*, vol. 12, pp. 300–304, 1964.

[4] C. Barnhart, L. Hatay, and E. L. Johnson, "Deadhead Selection for the Long-Haul Crew Pairing Problem," *Operations Research*, vol. 43/3, pp. 491–499, 1995.

[5] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance, "Branch-and-Price: Column Generation for Solving Huge Integer Programs," *Operations Research*, vol. 46, pp. 316–329, 1998.

[6] G. G. Brown, G. W. Graves, and D. Ronen, "Scheduling ocean transportation of crude oil," *Management Science*, vol. 33, pp. 335–346, 1987.

[7] A. Caprara and M. Fischetti, "Branch & Cut Algorithms," in *Annotated Bibliographies in Combinatorial Optimization* (M. Dell'Amico, F. Maffioli, and S. Martello, eds.), ch. 4, Wiley, 1997.

[8] S. Ceria, C. Cordier, H. Marchand, and L. A. Wolsey, "Cutting Planes for Integer Programs with General Integer Variables," *Mathematical Programming*, vol. 81, pp. 201–214, 1998.

[9] M. Christiansen and B. Nygreen, "A method for solving ship routing problems with inventory constraints," *Annals of Operations Research*, vol. 81, pp. 357–378, 1998.

[10] M. A. Cohen and P. R. Kleindorfer, "Creating value through operations: The legacy of Elwood S. Buffa," in *Perspective in Operations Management (Essays in Honor of Elwood S. Buffa)* (R. Sarin, ed.), pp. 3–21, Kluwer Academic Publishers, Boston, MA, 1993.

[11] M. A. Cohen and H. L. Lee, "Manufacturing strategy: Concepts and methods," in *The Management of Productivity and Technology in Manufacturing* (P. Kleindorfer, ed.), pp. 153–188, Plenum, New York, 1985.

[12] M. A. Cohen and H. L. Lee, "Strategic analysis of integrated production-distribution systems: Models and methods," *Operations Research*, vol. 36/2, pp. 216–228, 1988.

[13] M. A. Cohen and S. Moon, "An integrated plant loading model with economies of scale and scope," *European Journal of Operations Research*, vol. 50, pp. 266–279, 1991.

[14] H. Crowder, E. L. Johnson, and M. W. Padberg, "Solving Large-Scale Zero-One Linear Programming Problems," *Operations Research*, vol. 31, pp. 803–834, 1983.

[15] G. B. Dantzig and D. R. Fulkerson, "Minimizing the number of tankers to meet a fixed schedule," *Naval Research Logistics Quarterly*, vol. 1, pp. 217–222, 1954.

[16] M. Desrochers and F. Soumis, "A Column Generation Approach to the Urban Transit Crew Scheduling Problem," *Transportation Science*, vol. 23, pp. 1–13, 1989.

[17] L. F. Escudero, E. Galindo, G. García, E. Gómez, and V. Sabau, "Schumann, a modelling framework for supply chain management under uncertainty," *European Journal of Operations Research*, vol. 119, pp. 14–34, 1999.

[18] L. F. Escudero, F. J. Quintana, and J. Salmerón, "CORO, a modeling and an algorithmic framework for oil supply, transformation and distribution optimization under uncertainty," *European Journal of Operations Research*, vol. 114, pp. 638–656, 1999.

[19] M. L. Fisher and M. B. Rosenwein, "An interactive optimization system for bulk-cargo ship scheduling," *Naval Research Logistics*, vol. 36, pp. 27–42, 1989.

[20] A. M. Geoffrion and G. W. Graves, "Multicommodity distribution system designed by Benders decomposition," *Management Science*, vol. 20/5, pp. 822–844, 1974.

[21] A. M. Geoffrion, G. W. Graves, and S. J. Lee, "Strategic distribution system planning: A status report," in *Studies in Operations Management* (A. Hax, ed.), pp. 179–204, North-Holland, Amsterdam, 1978.

[22] A. M. Geoffrion, G. W. Graves, and S. J. Lee, "A management support system for distribution planning," *INFOR*, vol. 20/4, pp. 287–314, 1982.

[23] M. Goetschalckx, M. H. Cole, R. Wei, and K. Dogan, "A generic model for the strategic design of production-distribution systems," *Georgia Institute of Technology*, 1995.

[24] M. Goetschalckx, G. L. Nemhauser, M. H. Cole, R. Wei, K. Dogan, and X. Zang, "Computer aided design of industrial logistic systems," in *Proceedings of the Third Triennal Symposium on Transportation Analysis (TRISTAN III)*, (Capri, Italy), pp. 151–178, 1994.

[25] E. Hadjiconstantinou, N. Christofides, and A. Mingozzi, "A New Exact Algorithm for the Vehicle Routing Problem based on q-Paths and k-Shortest Paths Relaxations," *Annals of Operations Research*, vol. 61, pp. 21–43, 1995.

[26] P. L. Hammer, E. L. Johnson, and U. N. Peled, "Facets of Regular 0-1 Polytopes," *Mathematical Programming*, vol. 8, pp. 179–206, 1975.

[27] J. E. Hodder and M. C. Dincer, "A multifactor model for international plant location and financing under uncertainty," *Computer & Operations Research*, vol. 13/5, pp. 601–609, 1986.

[28] J. Laderman, L. Gleiberman, and J. F. Egan, "Vessel allocation by linear programming," *Naval Research Logistics Quarterly*, pp. 315–320, 1966.

[29] S. Lavoie, M. Minoux, and E. Odier, "A New Approach for Crew Pairing Problems by Column Generation with an Application to Air Transportation," *European Journal of Operations Research*, vol. 35, pp. 45–58, 1988.

[30] M. D. McKay and H. O. Hartley, "Computerized scheduling of seagoing tankers," *Naval Research Logistics Quarterly*, vol. 21, pp. 255–264, 1974.

[31] D. M. Miller, "An interactive computer-aided ship scheduling system," *European Journal of Operations Research*, vol. 32, pp. 363–379, 1987.

[32] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Programming.* Wiley, New York, 1988.

[33] M. W. Padberg, "On the Facial Structure of Set Packing Polyhedra," *Mathematical Programming*, vol. 5, pp. 199–215, 1973.

[34] M. W. Padberg, "(1,k)-Configurations and Facets for Packing Problems," *Mathematical Programming*, vol. 18, pp. 94–99, 1980.

[35] M. W. Padberg and G. Rinaldi, "A Branch and Cut Algorithm for Resolution of Large Scale Symmetric Salesman Problems," *SIAM Review*, vol. 33, pp. 60–100, 1991.

[36] Y. Pochet and L. A. Wolsey, "Algorithm and Reformulations for Lot-Sizing Problems," in *Combinatorial Optimization* (W. Cook, L. Lovasz, and P. Seymour, eds.), vol. 20 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pp. 245–294, AMS, 1995.

[37] D. Ronen, "Cargo ships routing and scheduling: Survey of models and problems," *European Journal of Operations Research*, vol. 12, pp. 119–126, 1983.

[38] D. Ronen, "Ship scheduling: The last decade," *European Journal of Operations Research*, vol. 71, pp. 325–333, 1993.

[39] Ş. Selçuk Erengüç, N. C. Simpson, and A. Vakharia, "Integrated production/distribution planning in supply chains: An invited review," *European Journal of Operations Research*, vol. 115, pp. 219–236, 1999.

[40] D. J. Thomas and P. M. Griffin, "Coordinated supply chain management," *European Journal of Operations Research*, vol. 94, pp. 1–15, 1996.

[41] C. J. Vidal and M. Goetschalckx, "The role and limitations of quantitative techniques in the strategic design of global logistics systems," *Georgia Institute of Technology*, 1996.

[42] C. J. Vidal and M. Goetschalckx, "Strategic production-distribution models: A critical review with emphasis on global supply chain models," *European Journal of Operations Research*, vol. 98, pp. 1–18, 1997.

[43] R. Weismantel, "On the 0/1 knapsack polytope," *Mathematical Programming*, vol. 77, pp. 49–68, 1997.

[44] L. A. Wolsey, "Faces for a Linear Inequality in 0-1 Variables," *Mathematical Programming*, vol. 8, pp. 165–178, 1975.

[45] L. A. Wolsey, "Facets and Strong Valid Inequalities for Integer Programs," *Operations Research*, vol. 24, pp. 367–372, 1976.

[46] L. A. Wolsey, *Integer Programming.* Wiley, New York, 1998.