AN OIL PIPELINE DESIGN PROBLEM

JACK BRIMBERG

Department of Business Administration, Royal Military College of Canada, P.O. Box 17000, Station Forces, Kingston, Ontario, Canada K7K 7B4, and GERAD, jack.brimberg@rmc.ca

PIERRE HANSEN

GERAD, and École des Hautes Études Commerciales, 3000, chemin de la Côte-Sainte-Catherine, Montréal, Québec, Canada H3T 2A7, pierreh@crt.umontreal.ca

KEH-WEI LIH

AT&T Labs, 200 South Laurel Avenue, Middletown, New Jersey 07748, lih@att.com

NENAD MLADENOVIĆ

Mathematical Institute, Serbian Academy of Sciences, Belgrade, Yugoslavia, and GERAD, nenad@mi.sanu.ac.yu

MICHÈLE BRETON

GERAD, and École des Hautes Études Commerciales, 3000, chemin de la Côte-Sainte-Catherine, Montréal, Québec, Canada H3T 2A7, michele.breton@hec.ca

We consider a given set of offshore platforms and onshore wells producing known (or estimated) amounts of oil to be connected to a port. Connections may take place directly between platforms, well sites, and the port, or may go through connection points at given locations. The configuration of the network and sizes of pipes used must be chosen to minimize construction costs. This problem is expressed as a mixed-integer program, and solved both heuristically by Tabu Search and Variable Neighborhood Search methods and exactly by a branch-and-bound method. Two new types of valid inequalities are introduced. Tests are made with data from the South Gabon oil field and randomly generated problems.

1. INTRODUCTION

In this paper we explore how to tailor methods of integer programming to a particular and difficult real-world problem: the optimal design of an oil pipeline network for the South Gabon oil field. This problem involves multiple-choice constraints as well as multiple-fixed charges. These make it very hard to solve the problem exactly. Problems of that type for which an optimal solution may be proved are still relatively small, of the order of, say, 30 nodes. Oil production from southern Gabon is expanding. A number of offshore platforms are currently built and under operation; extensive explorations have been made and proved reserves located in various places to be exploited under favorable circumstances. Moreover, there are other possible reserves, i.e., locations where traces of oil have been found. The infrastructure existing at the time of the study requires the oil to be brought by shuttle to the port of Cap Lopez (north of the map shown in Figure 1), from where it is shipped to its final destination. Because this solution is very expensive, the Direction Générale des Hydrocarbures du Gabon, in charge of management of oil production, decided to tender for the construction of an oil pipeline along the coast,

with connections to offshore platforms and onshore wells. The objective of the pipeline system would be to significantly reduce oil transportation costs to the port of Gamba (which would then be used for export of this region's oil instead of Cap Lopez), and thus, allow expansion of production, due to increased profitability. Several international oil companies would share use of the pipeline.

Since the start of the study, an expansion of the pipeline system has been underway. This study was conducted to assist in the planning phase. Due to confidentiality reasons, we present a particular scenario with simulated production and cost data; however, the network in Figure 1 depicts the locations of platforms, wells, and possible intermediate connections in the actual problem. (See §5 for further details.)

The paper is organized as follows. Previous work on pipeline design problems and the related one-terminal telpak problem is surveyed in the next section. Our model is stated mathematically in §3; we consider both optimal design of a new pipeline network and optimal expansion of an existing one. A new feature of the model formulation is that it ensures a tree structure (or arborescence)

Figure 1. South Gabon oil field problem with heuristic solution.
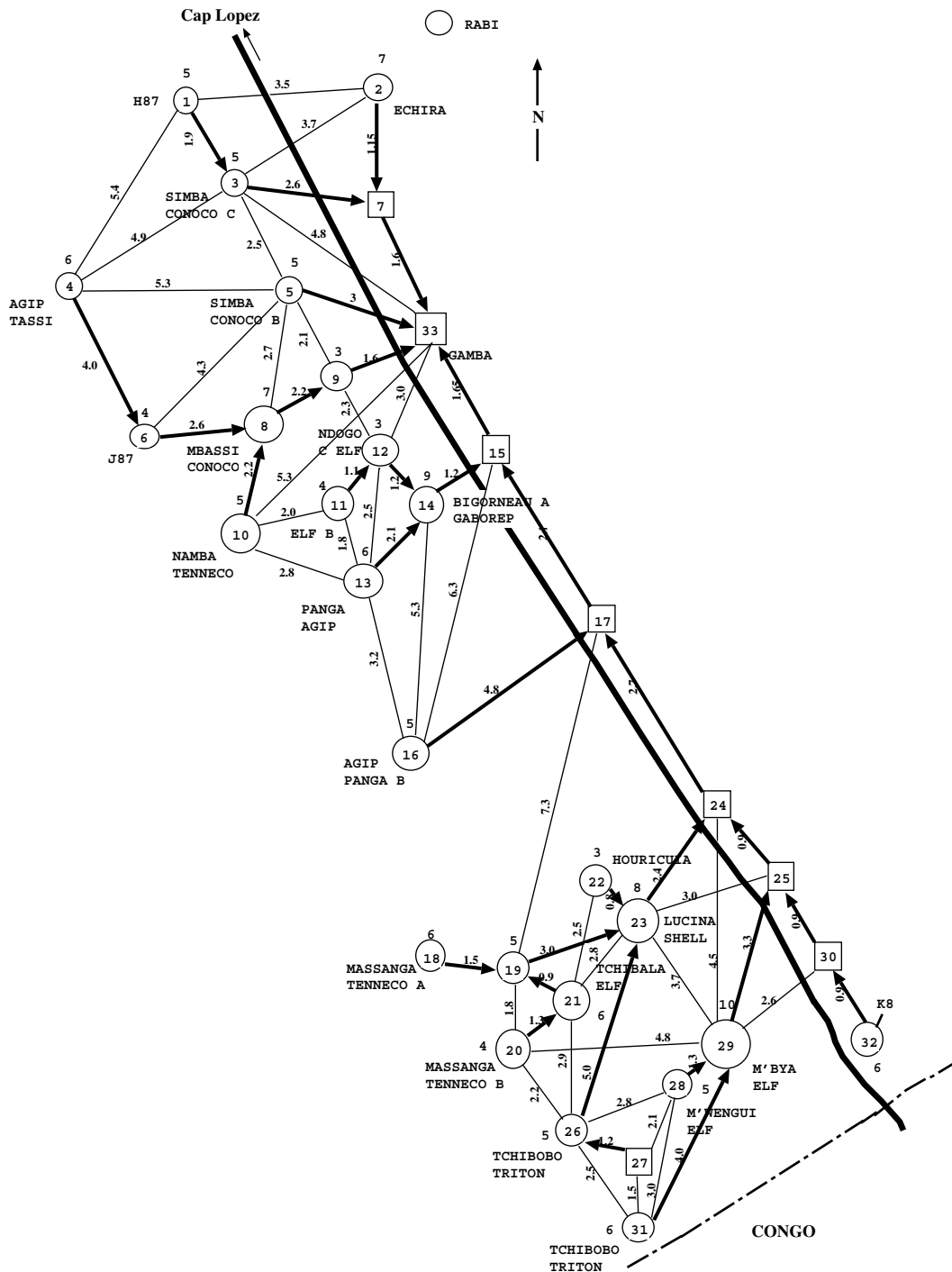
rooted at the port. Two heuristic algorithms based on Tabu Search and Variable Neighborhood Search methodologies are described in §4, as well as an exact solution method supplemented with two new types of valid inequalities. Detailed solution of a large example, simulating the offshore pipeline design problem in southern Gabon, is discussed in the last section, as well as computational results of the heuristics on a series of randomly generated problems.

2. RELATED WORK ON PIPELINE DESIGN

A seminal paper by Rothfarb et al. (1970) studies the optimal design of offshore natural gas pipeline systems. Three problems are investigated:

(a) selection of optimal diameters in a given pipeline network;

(b) design of an optimal pipeline system given gas-field locations and delivery requirements; and

(c) optimal expansion of an existing pipeline network.

The network is assumed to have a tree (or arborescence) structure with known production rates at specified nodes. Flow of gas in a pipe is governed by nonlinear pressure-drop constraints, with maximum and minimum pressures applying. The costs of pipes and compressors are to be minimized. The cost of a compressor depends on the path from the delivery node to the point of greatest pressure. Rothfarb et al. (1970) propose a heuristic solution method. In a first phase an initial tree is generated according to rules garnered from trial and error experience. Optimal diameters of the individual pipes are then computed. In the second phase, local transformations called Δ-changes are applied as follows. A circuit is formed by adding an arc connecting a node to the closest nonadjacent one. Then each other arc from this circuit is deleted in turn and the corresponding flows, diameters, pressures, and costs computed. If an improved feasible solution is found, one moves to it and iterates the procedure until no further improvements can be made with Δ-changes. This type of heuristic search scheme is close to a greedy algorithm in that improving moves are made at each iteration, but it differs in that only a subset of all possible moves is considered at each iteration before selecting the best one among them. Rosenthal (1982) studies the solution of the pipe diameter problem by nonserial dynamic programming (see Bertelè and Brioschi 1972). More recent advances in nonsequential dynamic programming applied to pipeline design are described in Carter (1998). However, the methodology assumes that the network layout is already given, and furthermore, in order to work efficiently, the network must have an "almost" serial structure. A different approach to the pipeline network design problem, due to Dolan et al. (1989), combines the approach of Rothfarb et al. with simulated annealing (Kirkpatrick et al. 1983, Aarts and van Laarhoven 1985).

The work of Rothfarb et al. is also discussed in several surveys on optimization in oil and gas pipeline engineering (Mah and Shacham 1978, Huang and Seireg 1985) and on network optimization (Golden et al. 1981). A more general treatment of the network expansion problem than that given by Rothfarb et al. is due to Olorunniwo and Jensen (1982a, 1982b): Future expansion times, sizes, and locations are jointly considered in a two-level hierarchical method. The related problem of determining the optimum number, sizes, and locations of offshore platforms has been studied by Yoo and Tcha (1986) and Hansen et al. (1992).

The graphsack problem is a simplified version of the pipeline network design problem, in which a single dimension for pipes is considered together with linear constraints for pressure drop. Solution of the continuous relaxation of the graphsack problem and of its dual is studied by Tinhofer (1984) and later Schreck and Tinhofer (1985). This last paper contains an $O(n^2)$ algorithm for solving the dual, where n is the number of nodes in the network. An even more simplified version arises if the pressure-drop constraints are deleted but integrality constraints are kept. One then obtains, up to a reversal of orientations, the minimum weighted arborescence problem, for which $O(n^2)$ algorithms have been proposed by Chu and Liu (1965), Edmonds (1967), and Karp (1971). Guignard and Rosenwein (1990) consider a constrained version of the minimum weighted arborescence (or directed tree with a single sink) problem where linear constraints on the sum of arcs entering each node apply. The problem is NP-hard, but fairly large instances ($n \leqslant 50$) are solved using Lagrangean decomposition.

When the substance being transported in the pipeline is in the form of a liquid such as oil instead of a gas, the use of costly compressors or pumps and the resulting constraints on pressure may be secondary considerations in the model formulation. Under the assumption that pressure constraints may be omitted, which is taken in this paper, the pipeline design problem simplifies considerably. The problem then closely resembles another basic one found in telecommunications: the one-terminal telpak problem (see Rothfarb and Goldstein 1971), which arises when long-distance telephone facilities are leased from common carriers. Lessees have certain voice and data communication needs, usually expressed as numbers of voice-equivalent channels required between given pairs of locations. Circuits to implement these requirements can be leased individually or in bulk units of, say, 60 or 240 voice-equivalent channels called TELPAKS. The lessee must submit for billing purposes a configuration of individual lines and TELPAKS which satisfies his requirements (usually the way they will be met in practice by the common carrier is different). The one-terminal TELPAK problem arises when all communications are to a common destination. It can then be expressed as a single-commodity network flow problem with nonlinear and nonconvex cost functions. The same structure applies to communications between a central computer and remote terminals, or to many locations requiring channels to a switching center only. Rothfarb and Goldstein (1971) propose a heuristic based on linear programming. Their work is discussed in the surveys of Frisch (1975) and Chang (1976). It is also studied with multicommodity flow extensions in Minoux (1975, 1976).

There are several similar network design models considered in the literature, but these do not take into account multiple-choice constraints and/or do not assume an arborescence structure of the solution. For example, Bousba and Wolsey (1991) give a model with fixed capacities on both nodes and arcs on the tree; Toth and Vigo (1995) also consider capacity- and degree-constrained models. Bienstock et al. (1998) formulate a *minimum cost capacity installation* (MCCI) problem as multicommodity network flow on a directed graph $G(V, A)$. Each pair (i, j) corresponds to a commodity to be sent from source node i to destination node j using arcs in A. The demand for commodity k may be fractional. Capacities are installed on the arcs in integral units. The objective is to obtain a minimum cost installation of capacities to ensure that all commodities can be shipped simultaneously. Closely related problems to MCCI, including *multicommodity survivable network design*, have been studied in Barahona (1996), Bienstock and Gunluk (1996), and Nemhauser and Wolsey (1988).

3. THE MODEL

In this section basic definitions are given, as well as a mathematical statement of the pipeline design problem which we consider. Flow of oil in the pipeline system is modeled by a network (N, A) with node-set N and arc-set A. The node-set $N = \{i | i = 1, 2, \ldots, n\}$ corresponds to the wells, i.e., both those on offshore platforms and at onshore sites (aggregated by platform or site), to potential connection points between pipeline segments (not all of which need to be used), and to the port. We specify the index of the port to be n. The amount of oil produced in the reference period at node i is denoted by p_i. Thus, production of all wells at a given platform is summed up and the same holds for wells at a given onshore site. These values are assumed to be known. We also assume that there is no production; i.e., $p_i = 0$, at connection points and at the port.

The arc-set A corresponds to potential layouts of pipeline segments between offshore platforms, onshore production sites, connection points, and the port. These are assumed to be given, on the basis of proximity between nodes and a study of soil and sea conditions—i.e., rocks, depth, and currents. Arcs are assumed to be oriented, but in some of the arcs flow may be sent either from i to j or from j to i; that is, there are two potential directions for these pipes, only one of which will be chosen in the optimal solution. A set of pipe diameters is associated with each arc (i, j) of A. Together with physical restrictions, such as pressure at the ends, we assume that the pipe capacity is fixed once the diameter is fixed. To write the flow conservation conditions, the set of predecessors and successors of node i will be denoted by $P(i)$ and $S(i)$, respectively, for $i = 1, 2, \ldots, n$; i.e., $P(i) = \{j \in N \setminus \{n\} | (j, i) \in A\}$ and $S(i) = \{j \in N | (i, j) \in A\}$. The capacities of pipes are to be chosen among a given set which may vary with the pair of nodes i and j. These capacities are denoted by C_{ij}^k for $k = 1, 2, \ldots, \bar{k}_{ij}$. Usually not more than five or six capacities will be considered. The fact that potential connection points need not necessarily be used is modeled by a special capacity C_{ij}^1 equal to 0; i.e., no flow can go through this pipe and its cost E_{ij}^1 is, of course, also 0. Costs associated with each arc and each capacity are denoted by E_{ij}^k. They correspond to all expenses needed to set up the corresponding pipes and to the cost of these pipes themselves as well as to the cost of pumps necessary to ensure the desired flow direction. Pipe costs are assumed to increase with diameter at a decreasing rate; i.e., there are economies of scale.

Moreover, this leads to the reasonable assumption that there is a single outgoing flow from any platform or well site, at most one outgoing flow from each connection point, and none from the port (from where ships are used to transport the oil). This assumption plays an important role in both the heuristic and the exact solution methods; indeed, it ensures that the optimal design of the pipeline network is an arborescence rooted at the port.

Flow in arc (i, j) will be denoted by $f_{ij} \geqslant 0$, for all $(i, j) \in A$. Let $y_{ij}^k = 1$ if a pipe with the kth capacity is laid out between nodes i and j, and 0, otherwise, $\forall (i, j)$ and k.

Note that due to the assumption on the tree structure of the pipeline network, once the y_{ij}^k variables are fixed, they induce an arborescence, and the flows are fixed also by conservation, assuming the capacity constraints are satisfied. Moreover, it is also true that once the layout of the pipeline is known, i.e., the arborescence corresponding to it is given, the decision variables are readily fixed at their optimal values. First, flows can be computed from the various sources to the sink by conservation equations and, second, smallest capacities sufficient to accommodate these flows can be chosen. This important property will be exploited by the heuristics proposed in §4, and will allow the obtainment of Type 2 valid inequalities explained later in that section.

The mathematical model of the pipeline design problem we consider is the following:

$$\text{minimize} \sum_{\forall (i,j) \in A} \sum_{k=1}^{\bar{k}_{ij}} E_{ij}^k y_{ij}^k \tag{1}$$

subject to

$$\sum_{j \in S(i)} \sum_{k=1}^{\bar{k}_{ij}} y_{ij}^k = 1 \quad \forall i \in N \setminus \{n\}, \tag{2}$$

$$\sum_{j \in P(i)} f_{ji} + p_i = \sum_{j \in S(i)} f_{ij} \quad \forall i \in N \setminus \{n\}, \tag{3}$$

$$f_{ij} \leqslant \sum_{k=1}^{\bar{k}_{ij}} C_{ij}^k y_{ij}^k \quad \forall (i, j) \in A, \tag{4}$$

$$f_{ij} \geqslant 0 \quad \forall (i, j) \in A, \tag{5}$$

$$y_{ij}^k \in \{0, 1\} \quad \forall (i, j) \in A, \quad k = 1, 2, \ldots, \bar{k}_{ij}. \tag{6}$$

The objective function (1) is simply the sum of costs for all pipes. The multiple-choice constraints (2) express that a single pipe that can accommodate the flow leaves node i in accordance with the assumption on the arborescence structure of the pipeline network and flow conservation at each node except the port. This is a unique feature of our model which differentiates it from other network design problems such as the multicommodity flow models referred to in the previous section. Equations (3) express conservation of flow: The total flow entering node i plus the flow due to production there equals the leaving flow. Constraints (4) express limitations on flow due to pipe capacity. Constraints (5) express that flows are nonnegative and constraints (6) that pipes are set up entirely or not at all.

Model (1)–(6) can be easily adapted to the problem of optimal expansion of an existing pipeline network. Still, assuming that it has an arborescence structure implies that all arcs in the existing pipeline network are oriented towards the port. (Otherwise there would be outgoing flow from a node to two different ones, a contradiction.) The expanded network will have an arborescence structure that must make use of all the pipes in place in the existing one. Let the variable $y_{ij}^e = 1$ if the capacity of the existing pipe between i and j is sufficient in the expanded network,

and 0, otherwise. Then those of the constraints (2) and (4) which correspond to nodes and arcs of the existing network should be replaced by the following ones:

$$y_{ij}^e + \sum_{k=1}^{\bar{k}_{ij}} y_{ij}^k = 1 \quad \forall i \in N' \setminus \{n\}, \quad j \in S(i), \tag{2'}$$

$$f_{ij} \leqslant C_{ij}^e + \sum_{k=1}^{\bar{k}_{ij}} C_{ij}^k y_{ij}^k \quad \forall (i, j) \in A', \tag{4'}$$

where N' denotes the set of nodes and A' the set of arcs of the existing network. Note that A' is an arborescence on N' rooted at the port. For each node $i \in N' \setminus \{n\}$, the successor set of i consists of only one node. Constraints $(2')$ and $(4')$ express that:

(i) if there is an arc between i and j in the existing network, there will be a pipe between these nodes in the expanded network;

(ii) an additional pipe between i and j may be constructed in parallel to the existing one.

Constraints $(4')$ express that the capacity of the pipe(s) between nodes i and j is equal to that of the existing pipe, i.e., C_{ij}^e, plus possibly that of a pipe built in parallel. Constraints (2) and (4) appear as usual for nodes in $N \setminus N'$ and arcs not in the existing network. Moreover, arcs joining pairs of nonadjacent nodes of the existing network or outgoing arcs from nodes of the existing network to nodes outside it need not be considered. Again, this follows from the assumption that the expanded network has an arborescence structure that embeds the existing structure.

Problem (1)–(6) is close to several well-known combinatorial optimization problems. If linear costs for the flows in all arcs are added to (1), it becomes the one-terminal telpak problem (see Rothfarb and Goldstein 1971, Frisch 1975, and Chang 1976 for some early references). When there are such linear costs for flows but only one capacity for each arc, the problem reduces to the single-source fixed-charge network flow problem (Hochbaum and Segev 1989). If only one capacity is considered for each arc and linear costs for the flows are omitted, the problem becomes a pure fixed-charge one (McKeown and Ragsdale 1990 and Ragsdale and McKeown 1991). All these problems are NP-hard. The first reduction of the fixed-charge problem to Steiner's problem in graphs is due to Karp (1972). A slight variation on this proof shows problem (1)–(6) to be NP-hard.

4. SOLUTION METHODS

In this section we propose two heuristics and an exact algorithm to solve problem (1)–(6). We first discuss a general graph-theoretic approach to the problem. Tabu Search and Variable Neighborhood Search heuristics are developed in §4.1. These heuristics may also be used to obtain an upper bound for the branch-and-bound procedure explained in §4.4. Lower bounds on the optimal solution of problem (1)–(6) are obtained by linear relaxation and improved by using two new types of valid inequalities, described in §4.2.

We also discuss possible decomposition of the problem according to geographical considerations in §4.3.

As mentioned in the previous section, the optimum pipeline network has an arborescence structure. This suggests a way to solve small instances of problem (1)–(6) by complete enumeration of all the arborescences rooted at the terminal port. We first assume that there is only one port in the oil pipeline system; thus, all the oil flows to the port. The problem can be separated into two cases.

1. No intermediary connection points in the system; i.e., pipes connect only offshore platforms, onshore well sites, and the port.

2. Intermediary connection points given onshore or offshore together with offshore platforms, onshore well sites, and the port.

First, assume that there are no intermediary connection points. The solution to this problem can then be found by complete enumeration of spanning trees rooted at the port. For each spanning tree, there is a corresponding total expense to set up the pipes for transporting oil from offshore platforms and onshore production sites to the port. This cost is determined once the spanning tree is fixed because there is exactly one path from any platform or well site to the port; i.e., the production at each location has exactly one way to be sent to the port. Thus, the flow in each arc of the tree is determined and the diameter of the pipe for that arc is that of the least expensive pipe that can accommodate the flow. Note that the direction of the flow is also determined as it is directed toward the root (port). Thus, the spanning tree (undirected) structure is in fact a spanning arborescence (directed) structure. The cost of the pipeline system corresponding to this spanning arborescence is the sum of the costs on each arc of the spanning arborescence. The minimum cost of the oil pipeline system is the cost corresponding to the least expensive spanning arborescence.

We next discuss the case where there are intermediary connection points included in the network. The problem is then similar to Steiner's problem on graphs (e.g., see Winter 1987): A fixed set of nodes corresponding to the offshore platforms, the onshore production sites, and the port must be included in a subgraph, but not all of the intermediary connection points need be included in that subgraph. This extension may be reduced to the previous case of enumeration of spanning trees. As discussed in the modelling section, we may treat the connection points as platforms with zero production. When a connection point is a leaf in the spanning tree, oil transported through the pipe from that connection point is zero and the pipe chosen for that arc has a zero capacity and costs nothing. This generalizes to the case where there are several connection points in sequence with no flow. Thus, oil sent out of any point is the sum of oil flow into that point and the production at that point. The pipe used in any arc is the least expensive one that can accommodate the flow on that arc. Note, however, that by adding potential connection points the problem size is enlarged.

In case there is more than one port, we may add a "super sink" connected to all the ports and consider these ports as intermediary connection points without production. Oil that is sent to any of the ports will be transported to the super sink without cost. Note that not every spanning tree corresponds to a solution in this case. Only spanning trees that include all the arcs between the super sink and the original ports correspond to valid solutions.

The resolution method of complete enumeration of spanning trees just discussed may allow the solution of problems with up to 15 to 20 nodes, depending on the density of the associated network. More efficient general mixed-integer programming codes such as CPLEX may alternatively be used. Larger problems are difficult to solve in either of these ways as the number of spanning trees or branching nodes increases rapidly. This suggests the following general approaches:

(i) explore in a heuristic way the space of all spanning trees, quickly getting a near-optimal or unproved optimal one as an upper bound; and

(ii) proceed to the exact solution of Problem (1)–(6) by general mixed-integer programming techniques such as branch and bound, using cuts in the form of valid inequalities embedded in this solution procedure to reduce the duality gap.

We combine both approaches in our solution method. Tabu Search and (or) Variable Neighborhood Search heuristics are used to obtain an upper bound, and linear relaxation together with two types of valid inequalities to obtain a lower bound for the branch-and-bound procedure.

4.1. Heuristics

Many frameworks have recently been proposed for designing heuristic algorithms which, contrarily to more traditional ones, are not blocked as soon as a local optimum is found. In this section, we adapt the well-known Tabu search (Glover 1989, 1990; Hansen and Jaumard 1990; and Glover and Laguna 1997) and the more recent variable neighborhood search (Mladenović 1995, Brimberg and Mladenović 1996, Mladenović and Hansen 1997, and Hansen and Mladenović 2001) heuristics to our pipeline network model.

Our methods are based on elementary tree transformations that define the neighborhood structure. The set of all neighbors of a given solution for the pipeline design problem is naturally defined as all those spanning arborescences with distance 1 to the given solution, i.e., the set of spanning arborescences which differ from the current solution by adding one new (or free) arc while deleting one existing arc. The deleted arc must belong to the cycle closed by the added arc in order to maintain the tree structure. Note that the neighboring spanning arborescences may require the direction of some arcs in the formed cycle to be reversed. A local search is performed by examining all points in the 1-neighborhood of the current solution. If the best neighborhood point provides a better solution, a move is made to that point. The process is repeated until a local minimum

is reached. Since the number of neighbors may be large, a complete enumeration to find the direction of steepest descent (or mildest ascent) tends to be costly in computing time. It may therefore be more efficient to choose a few entering arcs and for each of these to examine several outgoing arcs.

Tabu Search. After obtaining an initial solution (e.g., one corresponding to the minimum length tree of the given network), performing a local search, and initializing the Tabu list ($= \emptyset$), the following steps are taken:

Step 1. Consider the set of all arcs outside the current arborescence T and not in the Tabu list, or alternatively, generate at random p (a parameter) of them.

Step 2. For each such arc, determine the unique cycle C that it closes upon being added to T; delete in turn each arc of C (except the newly added one); update the flows in the remaining arcs of C; and compute the resulting difference in pipe costs.

Step 3. From all the elementary tree transformations above, keep the pair of entering and leaving arcs for which the cost decreases the most or increases the least.

Step 4. Perform the corresponding exchange to obtain the new current solution.

Step 5. Set a Tabu restriction on the reverse move of the leaving arc (update Tabu list); if an improved solution has been found, it becomes the new incumbent.

Step 6. Repeat the preceding steps until a stopping criterion is reached.

An alternate strategy, instead of examining the unique cycle obtained by adding the free arc (u, v), simply deletes arc (u, w). This is possible so long as C is not a directed cycle starting and ending at u, in which case we ignore the attempt. A much smaller number of points in the neighborhood is examined with this strategy.

Variable Neighborhood Search. The basic idea of the recent metaheuristic known as Variable Neighborhood Search (VNS) is to allow a change of neighborhoods in the search for a better solution. To construct different neighborhood structures and to perform a systematic search, one needs to have a way of finding the distance between any two solutions; i.e., one needs to supply the solution space with some metric (or quasi metric), and then induce neighborhoods from it. In our case, the distance between two spanning arborescences T_1 and T_2 is defined as the symmetric difference between their sets of edges. It is well known that this distance is a metric function, and thus, a set of spanning trees represents a metric space.

In our basic version of the VNS, a single parameter k_{max} is specified that gives the maximum radius (or distance) that will be allowed from the current solution. After an initial solution is found and a local search performed, the following steps are repeated until a stopping criterion is reached.

Step 1. (First neighborhood) Set distance $k = 1$.

Step 2. If $k > k_{max}$, return to Step 1.

(a) (Shaking) Draw an arborescence T' at a distance k from the current solution T; i.e., add k times one free arc at random and delete each time the best existing arc from the closed cycle.

(b) (Local search) Perform a local search starting from T' to obtain a local minimum T''.

Step 3. (Move or not) If $f'' < f_{\text{opt}}$ (value of $T'' <$ value of incumbent, T), $T \leftarrow T''$, $f_{\text{opt}} \leftarrow f''$, and return to Step 1; otherwise, $k \leftarrow k + 1$, and go to Step 2.

4.2. Valid Inequalities

Solving problem (1)–(6) exactly may be made difficult by a large duality gap. The gap is attributed to the several sizes of pipes available that make the solution of the continuous relaxation far removed from the integer one. Typically, for any node i with positive production there will be two variables y_{ij}^k with positive (fractional) values: One of the values will correspond to a portion of the largest-capacity pipe that is exactly enough to allow the outgoing flow to traverse the pipe, while the other corresponds to the portion of the least-cost pipe from that node required to satisfy constraint (2) on that node. For intermediary points, the least-cost pipe will correspond to one of the zero-capacity/zero-cost pipes. Few results on cutting-plane methods for fixed-charge problems appear in the literature. To the best of our knowledge, the first derivation of cuts for multiple fixed-charge problems is found in Lih (1993) for the particular problem under study; also see Glover et al. (1997) for more general models. Cuts based upon considering a subproblem have also been proposed in other types of network design problems in, for example, Stoer and Dahl (1994), Bienstock et al. (1998), and Bienstock and Muratore (2000). However, a careful search of the literature did not find the equivalent cuts of ours which are derived on the objective function. Two types of specialized valid inequalities for our pipeline design problem, which are capable of drastically reducing the duality gap, are given below.

Valid Inequalities of Type 1. Let W be a subset of the set of $N \setminus \{n\}$. The total production that must be transported across the boundary of the bipartition $(W, N \setminus W)$ is at least $\sum_{i \in W} p_i$. Moreover, for each node i in W there will be at most one arc directed to a node j in $N \setminus W$. A valid inequality will then be obtained by finding a lower bound on the cost of the arcs belonging to the cut $(W, N \setminus W)$. This is done by solving a multiple-choice knapsack problem formulated as follows:

$$\text{minimize} \sum_{i \in W} \sum_{j \notin W} \sum_{k=1}^{\bar{k}_{ij}} E_{ij}^k y_{ij}^k$$

subject to

$$\sum_{i \in W} \sum_{j \notin W} \sum_{k=1}^{\bar{k}_{ij}} C_{ij}^k y_{ij}^k \geqslant \sum_{i \in W} p_i$$

$$\sum_{j \notin W} \sum_{k=1}^{\bar{k}_{ij}} y_{ij}^k \leqslant 1 \quad \forall i \in W,$$

$$y_{ij}^k \in \{0, 1\} \quad \forall i \in W, \ j \in N \setminus W, \ k = 1, \dots, \bar{k}_{ij}.$$

Assume that the minimum objective value for this problem is V_W. A valid inequality of Type 1 is then defined by

$$\sum_{i \in W} \sum_{j \notin W} \sum_{k=1}^{\bar{k}_{ij}} E_{ij}^k y_{ij}^k \geqslant V_W.$$

Even for fairly large pipeline design problems, the multiple-choice knapsack problems needed to find valid inequalities of Type 1 may be sufficiently small to allow solution by hand. Should this not be the case, a specialized algorithm can be used (e.g., Bulfin and Liu 1985 or Chern and Jan 1986).

Valid Inequalities of Type 2. As mentioned at the beginning of this section, small pipeline design problems can be solved by complete enumeration of all trees. This suggests a second type of valid inequality, using a bound on the cost of pipes in all arcs of a given subset instead of in a cut only, as in Type 1 inequalities. Consider again a bipartition $(W, N \setminus W)$ of the nodes; then merge all nodes of $N \setminus W$ into a node n' and keep for each node $i \in W$ and each possible capacity k the arc with least cost E_{ij}^k between node i and some node j of $N \setminus W$. This gives a smaller pipeline design problem on $|W| + 1$ nodes. Solve this last problem by enumeration of trees and let V_{st} be the optimal value found. Then a valid inequality of Type 2 is given by

$$\sum_{i \in W} \sum_{j \in N} \sum_{k=1}^{\bar{k}_{ij}} E_{ij}^k y_{ij}^k \geqslant V_{st}.$$

4.3. Decomposition

Pipeline design problems have a particular geometric structure which may sometimes be exploited to simplify their solution. For instance, reservoirs may be geographically dispersed, which induces some natural decomposition. If the network (N, A) has a cut vertex, i.e., a vertex j the suppression of which disconnects the network, the problem can be solved for the subnetwork(s) so obtained and not containing the port, considering vertex j as a port. Then a smaller problem is obtained by deleting these subnetworks except node j and adding their production to that at j. A less powerful decomposition scheme may be used in the case where (N, A) has a small disconnecting set of nodes, say $\{i, j\}$; then the subproblem corresponding to the different distribution of flow at i and j must be considered.

4.4. Branch-and-Bound Procedure

The proposed algorithm, while exact (i.e., leading to a guaranteed optimal solution), relies on the following heuristic strategy: The optimal solution (or at least a solution very close to the optimal one and coinciding with it on

many arcs) should be determined in a first phase by a (sophisticated) heuristic; then the optimality of this solution should be proved in a second phase using cutting planes and branch and bound. This implies that the choice of cutting planes should be guided by the differences between the heuristic solution and the solution of the current relaxation; they should be aimed at reducing, and eventually eliminating, such differences.

As mentioned above, problems with multiple fixed charges are very difficult to solve as the duality gap is usually very large. Therefore, we propose to follow an interactive procedure and to exploit the geographic structure of the problem (which is easily done visually, but difficult to automate).

The steps of the interactive branch-and-bound procedure are as follows.

Step 1. Initialization: Read the input data of the problem to be solved. Let LIST (the list of subproblems to be solved) be empty.

Step 2. Decomposition: Determine, using depth-first search (Tarjan 1972), the articulation points of the network, if any. Assuming there are some, decompose the problem by considering each articulation point as the root of a subproblem in turn, beginning by those the farthest from the root of the network.

Step 3. Heuristic: Apply the heuristic procedure TABU_PIPE or VNS_PIPE to the decomposed problem. Let T_{opt} denote the incumbent solution so found and f_{opt} its value.

Step 4. Initial bound: Solve the continuous relaxation of the decomposed problem. Note the value of the initial estimate of the duality gap; i.e., the difference between the value $\underline{f}$ of the optimal solution of the relaxation and f_{opt}, the solution obtained by the heuristic procedure in Step 3.

Step 5. Valid inequalities from dense zones: Consider the geographical description of offshore platforms and onshore wells. Find clusters of them geographically isolated from others, and of a size not exceeding 15 such nodes. (This limit is chosen so as not to use up excessive computing time when generating valid inequalities.) Compute Type 2 valid inequalities for all clusters so determined. Add these valid inequalities to the continuous relaxation and solve the resulting program, from the previous solution on, using the dual simplex method. (It may be of interest to introduce valid inequalities one at a time and note their effect on the estimate of the duality gap; valid inequalities which do not lead to any improvement may be deleted.)

Step 6. Valid inequalities from comparison of solutions: Compare the solutions of the current relaxation and the current incumbent. List all arcs in which these solutions do not agree (this includes two cases: arcs out of a node joining different endpoints in both solutions, and arcs joining the same nodes in both solutions, but such that several parallel arcs are used fractionally in the continuous relaxation). Introduce additional Type 1 or Type 2 valid inequalities which cut at least one arc in the list. Continue this step interactively as long as the estimate of the duality gap decreases (or, in other words, the lower bound increases).

Step 7. Optimality test: If $\underline{f} > f_{opt}$, discard the current subproblem and go to 10.

Step 8. Solution test: If $\underline{f} < f_{opt}$ and the solution of the continuous relaxation is all integers, replace f_{opt} by $\underline{f}$, T_{opt} by the current tree, and go to 10.

Step 9. Branching: Consider a pair of nodes joined by an arc in which the solutions of the continuous relaxation and T_{opt} differ. Apply multiple branching with all feasible diameters of the pipe. Update the solutions of these subproblems with the revised dual simplex method; store all such solutions, i.e., subproblems, in LIST (including all specifications of fixed ones), except that one for which the optimal value of the relaxation is smallest. Consider this last subproblem as the current one and go to 6.

Step 10. Subproblem: If LIST is empty, this branch-and-bound procedure ends; otherwise, choose as the current subproblem the one with the lowest value of $\underline{f}$, and go to 6.

Clusters may be generated automatically in Step 5 using, for example, a *sequential clustering* algorithm with *minimum radius* and/or *maximum split* criteria (see Hansen et al. 1994). Depending on the number of offshore platforms and onshore wells, there could be many potential clusters obtained in such a manner. However, only a few carefully selected clusters should be examined to avoid excessive computing time. Furthermore, cluster selection may depend on previous ones selected for maximum effect. We argue that this process may be best accomplished through a visual examination of the geographical layout of the network.

5. COMPUTATIONAL RESULTS

In this section, we investigate the exact solution of the pipeline design problem for the South Gabon oil field. An important step in the solution procedure is to obtain a high quality heuristic solution that provides a tight upper bound. Thus, the South Gabon oil field problem is followed by a comparative study of the Tabu Search and Variable Neighborhood Search heuristics on a series of randomly generated pipeline problems.

South Gabon Oil Field. The South Gabon oil field consists of 33 nodes. These represent offshore platforms, onshore wells (both represented by circles in Figure 1), seven connection points (represented by squares in Figure 1), and one port (Gamba). The number inside each circle and square identifies the node. The numbers adjacent to the circles are the production rates at those sites. All the oil production in this region is transported to Gamba, where it is exported abroad by sea. There are 129 possible arcs with associated "generalized" distances shown in Figure 1. The total cost of a section of pipe is obtained by multiplying the arc distance by the unit price for each pipe capacity (monetary units 10, 15, 25, 40, and 65, for capacity sizes 5, 10, 25, 50, and 100, respectively).

The branch-and-bound procedure is applied to the South Gabon oil field problem shown in Figure 1. A heuristic solution of value 1,423 was obtained by both TABU_PIPE and VNS_PIPE; this solution is also shown in Figure 1.

Computing time for both heuristics was less than one second on a SUN SPARC 10 station. Observe that the network naturally decomposes into a northern and a southern part with node 17 as an articulation point. The first (southern) problem has nodes 18 to 32 with node 17 as the root (a port). This problem was solved by complete enumeration of trees, which is equivalent to computing a Type 2 inequality with the same node set. (The subproblem was also solved in 33.7 seconds with a branching tree of 18,119 nodes and 83 cuts applied, using Version 7.0 of the commercial code CPLEX.) An optimal value of 672 was obtained for the first subproblem, which coincides with the heuristic solution.

The second (northern) problem has nodes 1 to 17 with node 33 as the port. Connection point 17, being the articulation point of the northern and southern subnetworks, is now given a total production equal to the sum of all productions in the southern subproblem. The continuous relaxation of the northern problem obtained a value of 668.78 compared with a value of 751 from the heuristic solution of this subproblem. Three Type 2 inequalities were eventually added in order to reduce this (quite large) gap. The first cut corresponded to the southern part of the subproblem, i.e., nodes 11 to 17; the effect of this inequality was to raise the lower bound $\underline{f}$ to 687.81. A second inequality corresponded to the northern part, i.e., nodes 1 to 10. Adding this inequality alone gave a lower bound of 691.82. Adding both inequalities gave a lower bound $\underline{f}$ of 740; i.e., 87% of the estimated duality gap was closed. The third inequality was aimed to remove the fractional capacity existing between node 10 and node 8 after adding the first two inequalities. This inequality corresponds to a subset of the nodes in the second inequality, i.e., nodes 1 to 9. Adding all three inequalities resulted in a lower bound $\underline{f}$ of 749, or 99.73% of the estimated duality gap was closed.

With the lower bound of 749 and the estimated optimal value of 751 for the northern problem, we proceeded to branching. Multiple branching, by fixing the size of the pipe, was used on arc (10,8), which had a flow in the heuristic solution but not in the current solution. Among the five resulting subproblems, one obtained the same value as the heuristic solution, while the others had a larger value or were infeasible. This allowed us to conclude that the heuristic solution must be optimal. The northern problem was also solved directly with CPLEX 7.0 in 4.2 seconds using a branching tree of 904 nodes and 89 cuts applied.

In conclusion, despite the fact that the South Gabon oil field problem is quite large by comparison with other problems involving multiple fixed charges, it could be solved with decomposition, only three inequalities, and one multiple branching. On the other hand, when we tried the original problem directly on CPLEX 7.0, the optimal solution was obtained after 4,867 seconds with a branching tree of 1,298,242 nodes. During the solution process, 167 general cuts were added, raising the lower bound from an initial value of 698.0675 to 1,191.3. Meanwhile, by adding only three of our inequalities to the original model the initial duality gap was reduced by 83% (by increasing the LP relaxation value to 1,300.375).

Random Test Problems. The South Gabon oil field example has a number of nodes, $n = 33$, and arc density, dens = 0.131. In order to test our two heuristics on a wide range of problem sizes, we randomly generated graphs with $n = 15, 20, 25, 50, 75$, and 100 vertices and dens $= 0.1, 0.2, \ldots, 0.8$.

Random test problems were generated for each parameter pair (n, dens) on a $[0, 100] \times [0, 100]$ square. Node n (the port) always had coordinates $(100, 100)$, while the remaining $(n - 1)$ nodes were distributed uniformly in the square. A random number RND was generated for each pair of points (i, j). If RND < dens, an undirected arc (i, j) was added and distance $d(i, j)$ calculated as the Euclidean distance between i and j; otherwise, arc (i, j) was deleted. Oil production p_i was obtained from a uniform distribution over the interval $(0, 1)$, $i = 1, \ldots, n - 1$. The number of possible pipe diameters was set at 10 for all arcs. The associated capacities were given by cap $(k) = nk/10$ with associated cost per unit distance, cost $(k) = 10(\text{cap}(k))^{0.75}$, $k = 1, \ldots, 10$.

Table 1 compares the optimal solution obtained by CPLEX 7.0 for $n = 15, 20$, and 25 with the values obtained by various heuristics. One random test instance was generated for each parameter pair (n, dens) listed. In the fourth column of Table 1, the Minimum Spanning Tree (MST) solution (as % deviation from optimal) is given as a first

Table 1. Comparison of heuristics with optimal solution.

		f_{opt}	% Deviation from the Optimal Value				
n	dens	CPLEX	MST	RA	LS	TS	VNS
15	0.10	1221.00	25.27	22.90	0.00	0.00	0.00
15	0.20	1170.36	9.23	3.65	0.03	0.00	0.03
15	0.30	1532.96	6.08	3.84	2.71	0.62	0.00
15	0.40	1572.39	0.69	0.69	0.00	0.00	0.00
15	0.50	1221.00	25.27	22.90	0.00	0.00	0.00
15	0.60	1142.17	0.00	0.00	0.00	0.00	0.00
15	0.70	1221.00	25.27	22.90	0.00	0.00	0.00
15	0.80	1683.09	4.45	4.45	0.00	0.00	0.00
	Average		12.03	10.17	0.34	0.08	0.00
20	0.10	2281.74	17.56	14.69	2.36	0.00	1.84
20	0.20	2028.93	27.83	27.83	0.84	0.00	0.84
20	0.30	2622.42	10.73	2.70	0.91	0.00	0.91
20	0.40	1488.30	2.81	2.00	0.00	0.00	0.00
20	0.50	1722.18	4.20	1.39	0.00	0.00	0.00
20	0.60	2281.74	17.56	14.69	2.36	0.00	0.00
20	0.70	1845.33	56.12	46.70	0.00	0.00	0.00
20	0.80	1967.44	3.38	3.38	0.00	0.00	0.00
	Average		17.52	14.17	0.81	0.00	0.45
25	0.10	2738.40	7.25	7.25	2.88	0.00	2.88
25	0.20	3283.76	7.40	6.15	0.00	0.00	0.00
25	0.30	3274.47	4.69	4.69	0.19	0.00	0.00
25	0.40	2087.49	7.90	2.09	1.22	0.00	0.00
25	0.50	2970.78	6.91	6.91	1.05	0.00	0.00
25	0.60	3083.81	18.60	13.72	1.36	1.33	0.00
25	0.70	2331.56	11.69	7.91	1.89	1.68	1.89
25	0.80	3451.02	16.72	6.05	0.62	0.62	0.62
	Average		10.15	6.85	1.15	0.45	0.67

Table 2. Average results on 10 random instances.

n	dens	Best Value	% Deviation from the Best Value					t_{max} (sec.)
			MST	RA	LS	TS	VNS	
50	0.10	5601.14	6.871	3.737	0.433	0.000	0.056	1.8
50	0.20	4123.37	22.252	11.644	0.716	0.000	0.185	6.5
50	0.30	3266.24	15.105	12.370	0.404	0.000	0.015	11.8
50	0.40	2866.18	15.928	12.318	0.810	0.569	0.000	15.9
50	0.50	2745.04	16.922	7.118	0.079	0.000	0.032	34.3
50	0.60	2626.65	12.013	7.318	0.923	0.635	0.000	25.6
50	0.70	2432.90	11.918	6.884	0.935	0.935	0.000	30.1
50	0.80	2393.97	11.900	9.209	0.922	0.868	0.000	42.5
	Average		14.114	8.825	0.653	0.376	0.036	
75	0.10	9912.14	11.471	5.346	0.263	0.000	0.200	24.0
75	0.20	6453.67	9.285	3.427	0.810	0.172	0.000	63.7
75	0.30	5415.07	8.679	6.710	1.018	0.000	0.382	96.9
75	0.40	4828.67	13.150	7.457	0.404	0.000	0.190	236.5
75	0.50	4323.93	20.328	3.915	0.323	0.125	0.000	237.8
75	0.60	3995.21	5.904	4.664	0.337	0.256	0.000	184.7
75	0.70	3832.21	3.218	2.888	1.265	0.000	0.000	177.6
75	0.80	3645.71	4.050	3.508	0.200	0.105	0.000	356.8
	Average		9.511	4.739	0.577	0.082	0.097	
100	0.10	11204.55	9.499	5.860	0.163	0.000	0.035	69.1
100	0.20	7808.18	6.924	5.919	0.740	0.654	0.000	118.9
100	0.30	6541.03	8.254	6.197	0.950	0.000	0.212	337.0
100	0.40	5800.70	9.924	7.307	0.174	0.000	0.070	806.4
100	0.50	5446.56	6.258	4.087	0.455	0.392	0.000	558.4
100	0.60	5293.90	7.956	4.279	0.339	0.184	0.000	826.0
100	0.70	5054.36	8.474	5.231	0.301	0.278	0.000	1287.2
100	0.80	4866.67	7.541	5.733	0.280	0.177	0.000	1554.0
	Average		8.104	5.577	0.425	0.211	0.040	

attempt, since the cost of a pipe section is assumed to be proportional to its length. To try to improve on the MST, we then generated 1,000 random arborescences in a "controlled" vicinity using a variation of Prim's algorithm (see Mladenović et al. 1993). The best result from the Random Arborescence (RA) procedure is shown in column 5. This solution was used as the initial point for Local Search (LS); see column 6. The LS solution was then used to launch the basic versions of Tabu Search (TS) and Variable Neighborhood search (VNS). The results for TS and VNS are shown in columns 7 and 8, respectively. For both heuristics, the stopping criterion was based on a maximum execution time, $t_{max} = 10 \times$ execution time of LS. In all cases t_{max} was 0.1 sec. or less. Compare this with the execution times of CPLEX, which were of the order of 1 to 10 seconds for the networks with 15 nodes and 1,000 seconds for those with 25. Furthermore, when t_{max} was increased by a factor of 3, both TS and VNS obtained the optimal solution in all cases.

Table 2 gives the results of the various heuristics on larger problem instances that were impractical to solve by CPLEX. In this case, 10 random test instances were generated for each parameter pair (n, dens), and the average result is reported in the table as a % deviation from the best average result (shown in column 3). The stopping criterion for TS and VNS was again set at $t_{max} = 10 \times$ execution time of *LS*.

The length of the Tabu list was set in all runs at 7, while the VNS parameter k_{max} was set at 5. In VNS we used an "intensified" shaking by choosing the entering arc at random and then finding the best one to be removed. Parameter p (percent of free edges considered in the neighborhood of the current solution in both TS and VNS) was fixed at 100%, although sometimes, much better results were obtained with $p = 50\%$ in the given time limit t_{max}. For example, this neighborhood reduction was seen to be very useful in the 33-node Gabon real-life problem.

From Tables 1 and 2 we may conclude that the MST solution can be quite poor; and, more importantly, both TS and VNS are able to improve the already good MST+RA+LS solution. Furthermore, VNS appears to perform better when the density is larger, while TS performs better on average for small densities, at least for the suggested parameter values.

6. CONCLUSIONS

This paper formulates a pipeline design problem as a mixed-binary-integer linear program. Unique features of the model that make it useful in certain settings such as oil pipeline design include multiple choice on the capacities of the pipes, and the construction of tree structures in the network rooted at a single node (the port). The model is also adapted to the problem of expansion of an existing pipeline system. The deletion of pressure constraints improves the

tractability of the model, but may limit the usefulness of the model to a preliminary stage of the design.

An interactive branch-and-bound algorithm is proposed which incorporates two new specialized types of valid inequalities. The procedure is tested successfully on a real problem of considerable size from the South Gabon oil field. The duality gap is seen to be drastically reduced by the introduction of a few of these constraints. Two heuristics, one based on Tabu Search and the other on Variable Neighborhood Search, are able to find the optimal solution in less than one second.

Future work will examine the effectiveness of the valid inequalities on larger problem sizes and apply the solution concepts in other forms of network design.

ACKNOWLEDGMENTS

REFERENCES

Aarts, E. H. L., P. J. M. van Laarhoven. 1985. Statistical cooling: A general approach to combinatorial optimization problems. *Philips J. Res.* **40** 193.

Barahona, F. 1996. Network design using cut inequalities. *SIAM J. Optim.* **6** 823–837.

Bertelè U., F. Brioschi. 1972. *Nonserial Dynamic Programming.* Academic Press, New York.

Bienstock, D., O. Gunluk. 1996. Capacitated network design–Polyhedral structure and computation. *INFORMS J. Comput.* **8** 243–260.

——, G. Muratore. 2000. Strong inequalities for capacitated survivable network design problems. *Math. Programming, Ser. A* **89** 127–147.

——, S. Chopra, O. Gunluk, C-Y. Tsai. 1998. Minimum cost capacity installation for multicommodity network flows. *Math. Programming* **81** 177–199.

Bousba, C., L. Wolsey. 1991. Finding minimum cost directed trees with demands and capacities. *Ann. Oper. Res.* **33** 285–303.

Brimberg, J., N. Mladenović. 1996. A variable neighborhood algorithm for solving the continuous location-allocation problem. *Stud. Locational Anal.* **10** 1–12.

Bulfin, R. L., C-Y. Liu. 1985. Optimal allocation of redundant components for large systems. *IEEE Trans. Reliability* **R–34**(3) 241–247.

Carter, R. G. 1998. Pipeline optimization: Dynamic programming after 30 years. Stoner Associates Inc., Houston, TX.

Chang, S-K. 1976. A model for distributed computer system design. *IEEE Trans. Systems, Man, Cybernetics* **SMC–5**(6) 344–359.

Chern, M-S., R-H. Jan. 1986. Reliability optimization problems with multiple constraints. *IEEE Trans. Reliability* **R–35**(4) 431–436.

Chu, Y., T. Liu. 1965. On the shortest arborescence of a directed graph. *Sciencia Sinica* **4** 1396–1400.

Dolan, W. B., P. T. Cummings, M. D. LeVan. 1989. Process optimization via simulated annealing: Application to network design. *AIChE J.* **35**(5) 725–736.

Edmonds, J. 1967. Optimum branching. *J. Res. National Bureau Standards* **71B** 233–240.

Frisch, I. T. 1975. Technical problems in nationwide networking and interconnection. *IEEE Trans. Comm.* **COM–23**(1) 78–88.

Glover, F. 1989. Tabu search part I. *ORSA J. Comput.* **1**(3) 190–206.

——. 1990. Tabu search part II. *ORSA J. Comput.* **2**(1) 4–32.

——, M. Laguna. 1997. *Tabu Search.* Kluwer Academic Publishers, Boston, MA.

——, H. Sherali, Y. Lee. 1997. Generating cuts from surrogate constraint analysis for zero-one and multiple choice programming. *Comput. Optim. Appl.* **8**(2) 151–172.

Golden, B., M. Ball, L. Bodin. 1981. Current and future research directions in network optimization. *Comput. Oper. Res.* **8**(2) 71–81.

Guignard, M., M. B. Rosenwein. 1990. An application of Lagrangean decomposition to the resource-constrained minimum weighted arborescence problem. *Networks* **20**(3) 1087–1093.

Hansen, P., B. Jaumard. 1990. Algorithms for the maximum satisfiability problem. *Computing* **44**(4) 279–303.

——, N. Mladenović. 2001. Variable neighborhood search: Principles and applications. *Eur. J. Oper. Res.* **130** 449–467.

——, B. Jaumard, N. Mladenović. 1994. How to choose *k* entities among *n*. I. Cox, P. Hansen, B. Julesz, eds. *Partitioning Data Sets.* DIMACS, Rutgers, NJ, 105–116.

——, E. Pedrosa Filho, C. Ribeiro.1992. Location and sizing of offshore platforms for oil exploration. *Eur. J. Oper. Res.* **58** 202–214.

Hochbaum, D. S., A. Segev. 1989. Analysis of a flow problem with fixed charges. *Networks* **19**(3) 291–312.

Huang, Z., A. Seireg. 1985. Optimization in oil and gas pipeline engineering. *J. Energy Resources Tech.* **107**(2) 264–267.

Karp, R. M. 1971. A simple derivation of Edmond's algorithm for optimum branching. *Networks* **1** 265–272.

——. 1972. Reducibility among combinatorial problems. R. E. Miller, J. W. Thatcher, eds. *Complexity of Computer Computations.* Plenum Press, New York, 85–103.

Kirkpatrick, S., C. D. Gelatt, M. P. Vecchi. 1983. Optimization by simulated annealing. *Sciences* **220** 671–680.

Lih, K-W. 1993. Applications of zero-one linear programming with multiple-choice constraints. Ph.D. thesis, Rutgers University, Rutgers, NJ.

Mah, R. S., M. Shacham. 1978. Pipeline network design and synthesis. *Adv. Chemical Engrg.* **10** 125–209.

McKeown, P. G., C. T. Ragsdale. 1990. A computational study of using preprocessing and stronger formulations to solve large general fixed charge problems. *Comput. Oper. Res.* **17**(1) 9–16.

Minoux, M. 1975. Multicommodity dynamic flows of minimal updated cost. *Annales des Télécommunications* **30**(12) 51–58.

——. 1976. Minimum cost multicommodity flows with concave cost functions. *Annales des Télécommunications* **31**(34) 77–92.

Mladenović, N. 1995. Variable neighborhood algorithm—A new metaheuristic for combinatorial optimization. *Optim. Days*. Montreal, Quebec, Canada, 112.

——, P. Hansen. 1997. Variable neighborhood search. *Comput. Oper. Res.* **24** 1097–1100.

——, J. Petric, R. Gledhill. 1993. Spanning tree and arborescence at random. B. Papathanasiu, ed. *Proc. Second Balkan Conf. Oper. Res.* Thessaloniki, Greece, 611–620.

Nemhauser, G. L., L. A. Wolsey. 1988. *Integer and Combinatorial Optimization*. Wiley, New York.

Olorunniwo, F. O., P. A. Jensen. 1982a. Dynamic sizing and locationing of facilities on natural gas transmission networks. *Engrg. Optim.* **6** 95–109.

——, ——. 1982b. Optimal capacity expansion policy for natural gas transmission network: A decomposition approach. *Engrg. Optim.* **6** 13–30.

Ragsdale, C. T., P. G. McKeown. 1991. An algorithm for solving fixed-charge problems using surrogate constraints. *Comput. Oper. Res.* **18**(1) 87–96.

Rosenthal, A. 1982. Dynamic programming is optimal for nonserial optimization problems. *SIAM J. Comput.* **11**(1) 47–59.

Rothfarb, B., M. Goldstein. 1971. The one-terminal telpak problem. *Oper. Res.* **19** 156–169.

——, H. Frank, D. M. Rosenbaum, K. Steiglitz, D. J. Kleitman. 1970. Optimal design of offshore natural-gas pipeline systems. *Oper. Res.* **18**(6) 992–1020.

Schreck, H., G. Tinhofer. 1985. An $O(n^2)$ simplex algorithm for a class of linear programs with tree structure. *Eur. J. Oper. Res.* **21**(2) 225–232.

Stoer, M., G. Dahl. 1994. A polyhedral approach to multicommodity survivable network design. *Numerische Mathematik* **68** 149–167.

Tarjan, R. E. 1972. Depth-first search and linear graph algorithms. *SIAM J. Comput.* **1** 146–160.

Tinhofer, G. 1984. Rational solutions of the graphsack problem. *Math. Programming* **28**(3) 337–348.

Toth, P., D. Vigo. 1995. An exact algorithm for the capacitated shortest spanning arborescence. *Ann. Oper. Res.* **61** 121–141.

Winter, P. 1987. Steiner problem in networks: A survey. *Networks* **17** 129–167.

Yoo, C., D. Tcha. 1986. A cross decomposition procedure for the facility location problem with a choice of facility type. *Comput. Indus. Engrg.* **10** 283–290.