

TECHNICAL RESEARCH REPORT

A Dual Ascent Approach to the Fixed-Charge Capacitated Network Design Problem

*by J.W. Herrmann, G. Ioannou,
I. Minis, and J.M. Proth*

T.R. 94-29



*Sponsored by
the National Science Foundation
Engineering Research Center Program,
the University of Maryland,
Harvard University,
and Industry*

A dual ascent approach to the fixed-charge capacitated network design problem

J.W. Herrmann* G. Ioannou† I. Minis† J.M. Proth‡

Abstract

In this paper we consider the problem of constructing a network over which a number of commodities are to be transported. Fixed costs are associated to the construction of network arcs and variable costs are associated to routing of commodities. In addition, one capacity constraint is related to each arc. The problem is to determine a network design that minimizes the total cost; i.e. it balances the construction and operating costs. A dual ascent procedure for finding improved lower bounds and near-optimal solutions for the fixed-charge capacitated network design problem is proposed. The method is shown to generate tighter lower bounds than the linear programming relaxation of the problem.

Keywords: Network programming, Heuristics, Manufacturing

*Institute for Systems Research, University of Maryland, College Park, MD 20742.

†Department of Mechanical Engineering and Institute for Systems Research, University of Maryland, College Park, MD 20742.

‡INRIA-Lorraine, 4 Rue Marconi, Metz 2000, 57070 Metz, France.

1 Introduction

Network design problems occur in many applications, including transportation, distribution, communication, computer, and energy systems. In such problems, there exists communication of some intensity between selected nodes of an underlying graph, and this traffic travels along arcs which may have finite capacities. A central question to be decided is which arcs of the underlying graph should be included in the network design. In the fixed-charge network design problem, a tradeoff between operating costs and network construction costs is targeted. A larger network may cost more to build but may reduce operating costs by including more attractive origin-destination paths. Conversely, a smaller network may increase the operating costs. This problem comprises strategic, tactical, and operational decision-making and includes a number of problems as special cases, such as shortest path, minimum spanning tree, plant location, Steiner tree, and traveling salesman problems (Magnanti and Wong [7]). Although other approaches can be more useful for evaluating a small number of choices or for modeling networks with uncertainty, integer programming formulations allow one to pick a quantitatively good solution from a large feasible solution space.

This particular network design problem, like many others, is *NP-complete* (Johnson *et al.* [4]). A number of techniques have been used for fixed-charge network design problems, including Benders decomposition, branch-and-bound, Lagrangean relaxation, dual ascent, and linear programming relaxations (see the survey by Magnanti and Wong [7]). In particular, Balakrishnan *et al.* [1] achieve good results with a dual ascent procedure for large-scale problems. However, the capacitated problem has received little attention. Capacitated minimum spanning tree problems have been solved using branch-and-bound, by Lagrangean relaxation of the integer programming problem, and by linear programming relaxations (see the survey by Minoux [8]). A fixed-charge capacitated network design problem with only one source is studied by Khang and Fujiwara [5], who use Lagrangean relaxation and subgradient optimization to get a lower bound and a scaling heuristic to obtain an upper bound.

In this paper we present an integer programming formulation of the fixed-charge capacitated network design problem and extend the dual ascent method of Balakrishnan *et al.* [1] to this problem. While as Balakrishnan *et al.* [1] predict, the dual ascent method by itself

cannot find good lower bounds for the capacitated problem, it is clear that increasing the variable cost of bottleneck arcs (the capacity of which is exceeded in the initial primal solution) will force the algorithm to include additional paths in the network. The proposed algorithm iteratively updates the variable cost bottleneck arcs, thus increasing the lower bound. We show that the algorithm generates a sequence of lower bounds and terminates with a primal feasible solution. Experimental results on problems with 20 to 60 arcs show that the lower bounds obtained are better than those achieved by linear programming relaxations.

The motivation for this work stems from the material handling network design problem of manufacturing facilities. Given a steady demand pattern in a job shop environment, the objective is to form a material handling network that minimizes the cost of moving parts between selected nodes (resource input and output stations) and the cost of constructing the links in the network (material flow paths). Capacity constraints on each arc limit the total volume of parts which can use the arc for transport in order to prevent traffic congestion. The general problem has not been previously studied, despite its applicability beyond material handling systems design. For related work in the material handling system area see Chhajed *et al.* [2] and Kim and Tanchoco [6].

The primary contribution of this paper is the proposed solution technique, which yields good lower and upper bounds and can be applied to large-scale problems. We emphasize the derivation of lower bounds, since they can be employed to evaluate heuristic solution algorithms and can be embedded in branch-and-bound implicit enumeration procedures to obtain optimal solutions.

The paper is organized as follows: Section 2 presents the integer programming formulation of the fixed-charge capacitated network design problem, as well as the dual problem of the linear programming relaxation. In Section 3 the dual ascent procedure is described, some associated definitions and properties are presented and the labeling method used by the dual ascent scheme is discussed. In Section 4 the proposed iterative procedure for finding a good lower bound and a primal feasible solution is presented. Section 4 also includes the proof that a lower bound is generated at each step of the procedure. Section 5 presents a small example to demonstrate the steps of the proposed algorithm. In Section 6 our computational results for a number of problem sets are discussed. The conclusions of this study are provided in

2 Mathematical Formulations

Consider an undirected graph $G = (N, \bar{A})$, the vertices of which represent possible origin and destination points for the transportation of discrete physical goods. The arcs of G are alternative route segments for the transportation system. The design model targets the optimal selection of arcs to be included in the network, which depends upon the tradeoff between fixed construction costs and variable operating costs.

In the model presented below, there exist multiple commodities representing distinct physical goods with different nodes of origin and destination. Let K denote the set of these commodities. Several units of commodity k are to be transported from the point of origin, $O(k)$, to the point of destination, $D(k)$.

The model contains two types of variables. The first type models discrete design choices and the second models continuous flow decisions (Balakrishnan *et al.* [1]). Let y_{ij} be a binary variable defined as follows:

$$y_{ij} = \begin{cases} 1 & \text{if arc } \{i, j\} \in \bar{A} \text{ is included in the network design} \\ 0 & \text{otherwise} \end{cases}$$

Also, let x_{ij}^k denote the fraction of the flow of commodity k that travels on the directed arc (i, j) . Although the network is undirected, the flow is directed and is characterized by an origin and a destination. To capture this *orientation* requirement, let A be the set of directed arcs that correspond to \bar{A} ; i.e. for each undirected arc $\{i, j\} \in \bar{A}$, $(i, j) \in A$ and $(j, i) \in A$ denote the corresponding directed arcs with opposite orientations. Then, if (x, y) is the vector of design and flow variables, with $x = (x_{ij}^k)$ and $y = (y_{ij})$, the network design problem can be formulated as follows.

Problem CFP(G) (Fixed Charge Capacitated Network Design Problem)

$$\text{minimize} \quad Z = \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k x_{ij}^k + \sum_{\{i,j\} \in \bar{A}} F_{ij} y_{ij} \quad (1)$$

subject to :

$$\sum_{j \in N} x_{ji}^k - \sum_{l \in N} x_{il}^k = \begin{cases} -1 & \text{if } i = O(k) \\ 1 & \text{if } i = D(k) \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in N, k \in K \quad (2)$$

$$\sum_{k \in K} f_k(x_{ij}^k + x_{ji}^k) \leq B_{ij} \quad \forall \{i, j\} \in \bar{A} \quad (3)$$

$$x_{ij}^k, x_{ji}^k \leq y_{ij} \quad \forall \{i, j\} \in \bar{A}, k \in K \quad (4)$$

$$x_{ij}^k \geq 0 \quad \forall (i, j) \in A, k \in K \quad (5)$$

$$y_{ij} \in \{0, 1\} \quad \forall \{i, j\} \in \bar{A} \quad (6)$$

In this formulation, F_{ij} is the non-negative fixed-charge attributed to arc $\{i, j\}$. The cost, c_{ij}^k , of routing flow from commodity k on the directed arc (i, j) is scaled to reflect the unit arc cost and the total flow for this commodity. In other words, if f_k units of commodity k must be transported, and \bar{c}_{ij}^k is the per unit arc routing cost for this commodity, then

$$c_{ij}^k = f_k \cdot \bar{c}_{ij}^k$$

The objective function Z of (1) reflects the basic tradeoff between routing cost savings and fixed costs for using network arcs. Constraints (2) are the standard flow conservation equations, imposed on each commodity k . They ensure the continuity of the transportation path between each origin-destination pair. Inequalities (3) provide the arc capacity constraints. The constant B_{ij} is the capacity of arc $\{i, j\} \in \bar{A}$ and the lefthand side of (3) represents the total amount of flow of all commodities through each edge $\{i, j\} \in \bar{A}$. The *forcing* constraints (4) prohibit flow through inactive arcs, i.e. arcs $\{i, j\} \in \bar{A} : y_{ij} = 0$. Constraints (5) ensure the non-negativity of the continuous variables x_{ij}^k , while constraints (6) force the discrete variables y_{ij} to assume binary values.

Relationships (1)-(2) and (4)-(6) define the fixed-charge uncapacitated network design problem on G , hereafter designated as **FP**(G). Its formulation is due to Balakrishnan *et al.* [1].

It is emphasized that the capacitated version of the network design problem, **CFP**(G), is substantially more complicated than **FP**(G), and that the linear programming relaxation does not provide a good approximation to the binary integer program (Balakrishnan *et*

al. [1]). However, this model offers a more accurate mathematical representation of most network design problems, since capacity constraints are often encountered in practice (e.g. communication networks, material flow paths).

Let us consider the linear program formed by relaxing constraints (6) in problem $\mathbf{FP}(G)$. The dual of this linear programming problem can be formulated as follows:

Problem $\mathbf{DP}(G)$ (Dual of the linear relaxation of $\mathbf{FP}(G)$)

$$\text{maximize} \quad Z_D = \sum_{k \in K} u_{D(k)}^k \quad (7)$$

subject to :

$$u_j^k - u_i^k \leq c_{ij}^k + w_{ij}^k \quad \forall k \in K, (i, j) \in A \quad (8)$$

$$\sum_{k \in K} (w_{ij}^k + w_{ji}^k) \leq F_{ij} \quad \forall \{i, j\} \in \bar{A} \quad (9)$$

$$w_{ij}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (10)$$

The dual variables $u_{D(k)}^k, \forall k \in K$, and $w_{ij}^k, \forall k \in K, (i, j) \in A$, correspond to the flow conservation equations (2) and the forcing constraints (4), respectively. Note that, since one of the flow conservation equations (2) for each commodity $k \in K$ is redundant (Balakrishnan *et al.* [1]), one of the corresponding dual variables can be set equal to zero. To simplify the formulation, we have set $u_{O(k)}^k = 0, \forall k \in K$.

An important observation for the dual ascent method discussed in the following sections is that for any given vector $w = (w_{ij}^k)$ that satisfies the dual constraints (9), the above problem decomposes by commodity. Each subproblem is the dual of the shortest path problem from origin $O(k)$ to destination $D(k)$ with respect to the *modified arc lengths*, $\hat{c}_{ij}^k = c_{ij}^k + w_{ij}^k$.

3 The dual ascent framework

In general dual ascent approaches explore the set of feasible solutions to the dual of the linear programming relaxation of an integer program. In particular, the method we consider computes approximate solutions to the uncapacitated dual problem $\mathbf{DP}(G)$. In doing so, it provides a lower bound to the capacitated problem $\mathbf{CFP}(G)$, which is further improved by appropriate variable cost adjustments.

3.1 Preliminaries

Given the formulations of Section 2, we proceed with some definitions of sets and variables that will be employed in the discussion of the dual ascent approach.

Definition 3.1 Q_G is the convex polyhedron formed by equalities and inequalities (2)-(3), i.e., the set of feasible solutions to $\mathbf{CFP}(G)$.

Due to the particular structure of constraints (2)-(4), problem $\mathbf{FP}(G)$ can be decomposed to shortest path subproblems, given a choice of binary variables y_{ij} . This results to the integrality property below.

Property 3.1 (Balakrishnan *et al.* [1]) *There always exists an integral optimal solution to the uncapacitated network design problem, $\mathbf{FP}(G)$.*

Definition 3.2 For any integral feasible solution, (x, y) , of the uncapacitated problem, let $\bar{H} = \{\{i, j\} \in \bar{A} : y_{ij} = 1\}$ and $H = \{(i, j) \in A : \{i, j\} \in \bar{H}\}$.

Definition 3.3 For any integral feasible solution, (x, y) , of $\mathbf{FP}(G)$, let $\forall k \in K$, $H_k = \{(i, j) \in A : x_{ij}^k = 1\}$ and $\bar{H}_k = \{\{i, j\} \in \bar{A} : (i, j) \in H_k \vee (j, i) \in H_k\}$.

Definition 3.4 For any feasible solution to the dual problem, $\mathbf{DP}(G)$, let s_{ij} be the slack variable that corresponds to constraint (9) for $\{i, j\} \in \bar{A}$.

Since there exists an one-to-one correspondence between undirected arcs and slack variables, we will employ the term *(non)zero-slack arc*, instead of *(non)zero-slack variable* of constraint (9) that corresponds to an arc. From the above definition, it follows directly that:

Property 3.2 For any feasible solution of $\mathbf{DP}(G)$, $F_{ij} = s_{ij} + \sum_{k \in K} (w_{ij}^k + w_{ji}^k)$, $\forall \{i, j\} \in \bar{A}$.

The following lemma provides an expression for the primal objective function (1) of any integral primal feasible solution of $\mathbf{FP}(G)$, in terms of the dual and slack variable values, w_{ij}^k and s_{ij} , respectively, of any feasible solution of the dual problem, $\mathbf{DP}(G)$.

Lemma 3.1 *The objective function value, Z^h , of any integral primal feasible solution of $\mathbf{FP}(G)$, (x, y) , is*

$$Z^h = \sum_{k \in K} \sum_{(i,j) \in H^k} (c_{ij}^k + w_{ij}^k) + \sum_{\{i,j\} \in \bar{H}} s_{ij} + \sum_{k \in K} \sum_{(i,j) \in H \setminus H_k} w_{ij}^k$$

where w_{ij}^k and s_{ij} are the dual and slack variables of any dual feasible solution of \mathbf{DP} .

Proof For any integral primal feasible solution (x, y) , define the sets H , \bar{H} , H_k and \bar{H}_k $\forall k \in K$ as in (3.2) and (3.3). Then, the objective (1) for (x, y) can be written as

$$Z^h = \sum_{k \in K} \sum_{(i,j) \in H_k} c_{ij}^k + \sum_{\{i,j\} \in \bar{H}} F_{ij} \quad (11)$$

since all other terms in (1) are zero.

From Property 3.2, $F_{ij} = s_{ij} + \sum_{k \in K} (w_{ij}^k + w_{ji}^k)$. Substituting in (11) and transforming the second summation from undirected to directed arcs, we obtain

$$Z^h = \sum_{k \in K} \sum_{(i,j) \in H^k} c_{ij}^k + \sum_{\{i,j\} \in \bar{H}} s_{ij} + \sum_{k \in K} \sum_{(i,j) \in H} w_{ij}^k \quad (12)$$

By grouping together the terms that correspond to each directed arc and each commodity, (12) can be written as

$$Z^h = \sum_{k \in K} \sum_{(i,j) \in H^k} (c_{ij}^k + w_{ij}^k) + \sum_{\{i,j\} \in \bar{H}} s_{ij} + \sum_{k \in K} \sum_{(i,j) \in H \setminus H_k} w_{ij}^k$$

Q.E.D.

3.2 The labeling method

The labeling method of Balakrishnan *et al.* [1] is a dual ascent procedure motivated by the decomposition of the linear dual problem $\mathbf{DP}(G)$ to duals of shortest path subproblems (see section 2). Given a feasible dual variable vector $w = (w_{ij}^k)$, each dual variable $u_{D(k)}^k$ can be set equal to the shortest origin-destination path with respect to the modified arc lengths $\hat{c}_{ij}^k = c_{ij}^k + w_{ij}^k$. In fact, this is the optimal solution to the primal shortest path subproblem. Therefore, as the values of the w_{ij}^k variables increase (but remain dual feasible), the corresponding shortest paths also increase. In this context, the labeling method increases the dual objective function by increasing the w_{ij}^k values and, thus, absorbing fixed-charges

into the dual objective and simultaneously reducing the slack of some arcs. The maximum increase occurs when the slack of some arcs is reduced to zero. Appendix A provides a brief outline of the method and proper definitions for the sets $N_1(k)$ and $N_2(k)$ that are employed below.

In the remainder of this section, two important properties of the labeling method are presented and an analytical expression for the dual objective is derived.

Property 3.3 (zero-slack path, Balakrishnan et al. [1]) *For every commodity $k \in K$, all nodes in $N_2(k)$ are connected to the destination $D(k)$ via shortest paths containing only zero-slack arcs.*

Property 3.4 (shortest path, Balakrishnan et al. [1]) *At every step of the labeling algorithm, $u_i^k, \forall i \in N, k \in K$ represents the shortest path distance from origin $O(k)$ to node i , using the modified arc lengths $\hat{c}_{ij}^k = c_{ij}^k + w_{ij}^k$.*

Given the above property, the following corollary can be derived in a straight forward manner.

Corollary 3.1 *At the termination of the labeling method, i.e., when $O(k) \in N_2(k) \forall k \in K$, the value of the dual objective function (7) is*

$$Z_D^o = \sum_{k \in K} \sum_{(i,j) \in A_o^k} (c_{ij}^k + w_{ij}^k)$$

where A_o^k is the set of zero-slack arcs that constitute a shortest origin-destination path for commodity $k \in K$.

Proof Since the solution derived by the labeling method is dual feasible (Balakrishnan et al.), the dual objective (7) is given by

$$Z_D^o = \sum_{k \in K} u_{D(k)}^k \quad (13)$$

Consider a commodity $k \in K$. From Property 3.3 and since $O(k) \in N_2(k)$, it follows that there exists an $O(k) - D(k)$ shortest path containing only zero-slack arcs. Let A_o^k be the set of directed arcs in this path. Then, from Property 3.4, each dual variable $u_{D(k)}^k$ can be expressed

as the shortest path with respect to the modified costs \hat{c}_{ij}^k between an origin-destination pair, i.e.,

$$u_{D(k)}^k = \sum_{(i,j) \in A_o^k} \hat{c}_{ij}^k \quad (14)$$

Equation (14) holds for every commodity $k \in K$. From (13) and the definition of the modified costs we conclude that

$$Z_D^o = \sum_{k \in K} \sum_{(i,j) \in A_o^k} (c_{ij}^k + w_{ij}^k)$$

Q.E.D.

4 Solution approach

In this section we present the proposed dual ascent algorithm, which extends the labeling method of Balakrishnan *et al.* [1] to the case of capacitated networks. It is noted that the sharpness of the lower bound derived by the algorithm directly depends upon the degree to which capacity constraints affect the optimal solution of the network design problem. In Section 4.2 we show that the algorithm generates a lower bound at each iteration.

4.1 The Iterative Dual Ascent (IDA) algorithm

Consider the dual feasible solution to $\mathbf{DP}(G)$ derived by applying the labeling method to G . Let $\bar{A}_o \subseteq \bar{A}$ be the set of zero-slack arcs, i.e., $\bar{A}_o = \{\{i, j\} \in \bar{A} : s_{ij} = 0\}$, and A_o the corresponding set of directed arcs, i.e., $A_o = \{(i, j) \in A : \{i, j\} \in \bar{A}_o\}$.

If $G' = (N, \bar{A}_o)$, then there exist two mutually exclusive alternatives for the convex polyhedron $Q_{G'}$:

1. $Q_{G'} = \emptyset$; i.e. there exists no feasible solution to the capacitated problem $\mathbf{CFP}(G')$.
2. $Q_{G'} \neq \emptyset$; i.e. there exists at least one feasible solution to the capacitated problem $\mathbf{CFP}(G')$.

If problem $\mathbf{CFP}(G')$ is feasible, then no additional labeling steps are required to obtain a network design that satisfies the capacity constraints. Furthermore, the lower bound

cannot be improved by the dual ascent strategy. On the other hand, if problem $\mathbf{CFP}(G')$ is infeasible, then we reduce the slack of non-zero slack arcs $(\{i, j\} \in \bar{A} \setminus \bar{A}_o)$ and include one or more of them in \bar{A}_o in order to obtain a new subgraph G' of G such that there exists a feasible solution to $\mathbf{CFP}(G')$. The basic idea stems from the shortest path property of the dual solution derived by the labeling method (Property 3.4). Since the arcs of \bar{A}_o which form the shortest paths with respect to the modified costs, \hat{c}_{ij}^k , cannot accommodate all of the commodities, flow must be directed to new arcs. This can be achieved by increasing the cost of the current shortest paths in order to create new origin-destination paths (consisting of zero-slack arcs) that satisfy the shortest path Property 3.4.

The underlying mechanism for the required flow diversion is straightforward. Bottleneck arcs, $\{i, j\} \in \bar{A}_o$, are identified and their variable costs c_{ij}^k are increased for some commodity $k \in K$ in an attempt to alter the origin-destination shortest paths. This variable cost increase can be thought of as a penalty for violated capacity constraints. Subsequently, by re-applying the labeling method, the slacks of additional arcs are reduced to zero. Consequently, by iteratively implementing the labeling method and increasing the cost of bottleneck arcs, we eventually obtain a set A_o such that $\mathbf{CFP}(G')$ has a feasible solution.

The following important issue should be emphasized. Our method seeks to obtain a lower bound that is better than the one obtained by the linear programming relaxation. It is clear that by altering the variable costs the value of the dual objective function obtained by the labeling method is increased. However, it remains to be shown that the bounds obtained by this iterative algorithm remain always below the optimum. In the following section, this important property is proved.

Based on the definitions of the previous sections, we directly state the following algorithm (Iterative Dual Ascent), which provides both a lower bound and a feasible design for the fixed-charged capacitated network design problem. It is noted that *Steps 2-4* are employed by the labeling method discussed in Appendix A, while *Steps 5-6* implement the flow diversion procedure discussed above.

Algorithm IDA

Step 1 : initialization of dual variables and slacks

$$q \leftarrow 1$$

$$w_{ij}^k \leftarrow 0 \quad \forall (i, j) \in A, k \in K$$

$$s_{ij} \leftarrow F_{ij} \quad \forall \{i, j\} \in \bar{A}$$

$$u_i^k \leftarrow \text{shortest path from } O(k) \text{ to node } i, \forall i \in N, k \in K$$

$$Z_D^o \leftarrow \sum_{k \in K} u_{D(k)}^k$$

Step 2 : initialization of labeled/unlabeled arc sets

$$N_2(k) \leftarrow \{D(k)\}, \forall k \in K$$

$$N_1(k) \leftarrow N \setminus \{D(k)\}, \forall k \in K$$

$$\text{Set } CND = \{k \in K : O(k) \in N_1(k)\}$$

Step 3 : evaluation of δ -increase

Select $k \in CND$

$$\text{Set } A(k) = \{(i, j) \in A : i \in N_1(k), j \in N_2(k)\}$$

$$\text{Set } A'(k) = \{(i, j) \in A(k) : c_{ij}^k + w_{ij}^k - (u_j^k - u_i^k) = 0\}$$

$$\text{Calculate } \delta_1 = \min\{s_{ij} : (i, j) \in A'(k)\}$$

$$\text{Calculate } \delta_2 = \min\{c_{ij}^k + w_{ij}^k - (u_j^k - u_i^k) : (i, j) \in A(k) \setminus A'(k)\}$$

$$\text{Set } \delta \leftarrow \min\{\delta_1, \delta_2\}$$

Step 4 : dual variable update and node labeling

$$w_{ij}^k \leftarrow (w_{ij}^k + \delta), s_{ij} \leftarrow (s_{ij} - \delta), \forall (i, j) \in A'(k)$$

$$u_i^k \leftarrow (u_i^k + \delta), \forall i \in N_2(k) \text{ and } Z_D^o \leftarrow (Z_D^o + \delta)$$

Update sets $N_1(k)$ and $N_2(k)$ by labeling (at most) one node:

If $\delta = \delta_1$, $s_{ij} = 0$ for some $(i, j) \in A'(k)$ set $N_1(k) \leftarrow N_1(k) \setminus \{i\}$ and $N_2(k) \leftarrow N_2(k) \cup \{i\}$

Set $CND \leftarrow CND \setminus \{k\}$; if $CND \neq \emptyset$ go to *Step 3*

If $O(k) \in N_2(k)$, $\forall k \in K$, set $Z_D^q = Z_D^o$ and go to *Step 5*

Otherwise set $CND = \{k \in K : O(k) \in N_1(k)\}$ and go to *Step 3*

Step 5 : feasibility check on zero slack arcs

$$\text{Set } \bar{A}_o^q = \{\{i, j\} \in \bar{A} : s_{ij} = 0\}$$

$$\text{Set } G'_q = (N, A_o^q)$$

if $Q_{G'_q} \neq \emptyset$ and $q \neq 1$ set $Z_{lb}^q = Z_{lb}^{q-1}$ and go to *Step 7*

if $Q_{G'_q} \neq \emptyset$ and $q = 1$ set $Z_{lb}^q = Z_D^o$ and go to *Step 7*

if $Q_{G'_q} = \emptyset$ identify $\tilde{A} = \{\{i, j\} \in \bar{A}_o^q \text{ that violate constraint (3)}\}$

Step 6 : variable cost update

Set $\phi_q = \min\{s_{ij} : \{i, j\} \in \bar{A} \setminus \bar{A}_o^q\}$

Select $k_q \in K$ and $\{i_q, j_q\} \in \tilde{A}$

Set $c_{i_q j_q}^{k_q} \leftarrow (c_{i_q j_q}^{k_q} + \phi_q)$

Set $Z_{lb}^q = (Z_D^o + \phi_q)$

$q = q + 1$

Go to *Step 2*

Step 7 : termination

Output lower bound Z_{lb}^q

Solve the linear relaxation of $\mathbf{CFP}(G'_q)$ to obtain $x = (x_{ij}^k)$

Set $y_{ij} = 1, \forall \{i, j\} \in \bar{A}_o^q : x_{ij}^k > 0$

Output primal feasible solution (x, y)

The arcs of \tilde{A} which violate capacity constraint (3) and the excess flow on them are identified by solving in *Step 5* the linear programming relaxation of the problem of Eqs.(1)-(6) on an augmented graph, \bar{G} . This graph contains the arcs of \bar{A}_o^q as well as a copy of each arc in \bar{A}_o^q . If A'_q is the set of these duplicate arcs, very large variable costs and zero fixed costs are assigned to $\{i, j\} \in A'_q$. In the optimal solution of the linear programming relaxation of $\mathbf{CFP}(\bar{G})$, if some flow is routed to a duplicate arc, i.e. $\exists \{i, j\} \in A'_q : x_{ij}^k > 0$, then problem $\mathbf{CFP}(G)$ is infeasible. This follows from the fact that such flows cannot exist in the optimal solution due to the large variable cost of arcs $\{i, j\} \in A'_q$. The formulation of the capacitated network design problem on this augmented graph and the method for identifying arcs that violate capacity constraints are presented in Appendix B.

In *Step 6* the algorithm selects the combination of arc $\{i_q, j_q\} \in \tilde{A}$ and commodity $k_q \in K$ such that the excess flow is maximum. This arc and flow pair $(\{i_q, j_q\}, k_q)$ impacts the capacity constraints most severely and the associated variable cost is updated; i.e.

$$\hat{f}_{i_q j_q}^{k_q} = \max\{\hat{f}_{ij}^k : \{i, j\} \in \tilde{A}, k \in K\}$$

where \hat{f}_{ij}^k represents the excess flow for commodity k on arc $\{i, j\}$ (see Appendix B). The following remarks clarify the way the IDA algorithm improves the lower bound and results in a feasible design.

Remark 4.1 *Once the slack of an arc becomes zero, it remains so in every subsequent step of the algorithm.*

This is a direct consequence of the non-negativity of the slack variables and the fact that the labeling method only decreases the slacks. If the slack of $\{i, j\}$ becomes zero at any iteration q of IDA, this arc always belongs to some shortest path(s) of this iteration; it may also be included in additional origin-destination shortest paths. Consequently, $A_o^q \subseteq A_o^{q+1}$.

Remark 4.2 *At each iteration the IDA algorithm increases the value of the dual objective Z_D^o by increasing the variable cost $c_{i_q j_q}^{k_q}$.*

This is clear from *Step 4*, since at each iteration only non-negative constants (δ) are added to Z_D^o . Furthermore, since the dual objective comprises shortest path lengths, increasing the variable cost $c_{i_q j_q}^{k_q}$ in *Step 6* results in increased Z_D^o .

Remark 4.3 *Algorithm IDA converges to a primal feasible solution of $\mathbf{FPG}(G)$ if such a solution exists.*

According to Remark 4.1, $A_o^q \subseteq A_o^{q+1}$. In addition, the stopping criterion guarantees that the solution comprises a set of arcs that accommodate the flow requirements and respect the capacity constraints. Thus, the only case that the algorithm IDA may not converge is when the cardinality of A_o is not increased in consecutive iterations. However, this can only occur if there exists a single path between some origin-destination pair that is not capable to accommodate the flow of the corresponding commodity, which leads to a contradiction.

Remark 4.4 *The minimum non-zero slack (ϕ_q) is selected in Step 6 for updating the variable cost in order to guarantee the lower bound property of IDA.*

This is required in order to obtain a sequence of lower bounds (see Section 4.2). If ϕ_q is set equal to any other non-zero slack, the bound obtained may be higher than the objective function value of the optimal primal solution of $\mathbf{CFP}(G)$.

4.2 The lower bound property of IDA for CFP

In this section we initially prove that the dual objective Z_D^o , which is evaluated by the first application of the labeling method is lower than the optimal primal objective of problem $\mathbf{CFP}(G)$ by at least ϕ (where $\phi = \min\{s_{ij} : \{i, j\} \in \bar{A} \setminus \bar{A}_o\}$). This is true given that the set of zero-slack arcs does not provide a feasible solution to the capacitated network design problem, i.e. $Q_{G'} = \emptyset$. Then, we extend this result to all iterations of the IDA procedure.

Assuming that $\mathbf{CFP}(G)$ has a feasible solution, let Z_c^* be the objective function value of the optimal solution of $\mathbf{CFP}(G)$, (x^*, y^*) .

Theorem 1 *If $Q_{G'} = \emptyset$, then $Z_c^* \geq Z_D^o + \phi$.*

Proof Given an optimal solution (x^*, y^*) to $\mathbf{CFP}(G)$, the optimal value of the objective function is

$$Z_c^* = \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k x_{ij}^{k*} + \sum_{\{i,j\} \in \bar{A}} F_{ij} y_{ij}^* \quad (15)$$

If $\bar{A}_c^* = \{\{i, j\} \in \bar{A} : y_{ij}^* = 1\}$ and $A_c^* = \{(i, j) \in A : \{i, j\} \in \bar{A}_c^*\}$, (15) can be written as

$$Z_c^* = \sum_{k \in K} \sum_{(i,j) \in A_c^*} c_{ij}^k x_{ij}^{k*} + \sum_{\{i,j\} \in \bar{A}_c^*} F_{ij} \quad (16)$$

since all other terms in (15) are zero.

Consider now the graph $G_c^* = (N, A_c^*)$. Note that $\mathbf{CFP}(G_c^*)$ has the same optimal solution as $\mathbf{CFP}(G)$. An integral feasible solution to problem $\mathbf{FP}(G_c^*)$ can be constructed by:

1. solving a shortest path problem for each commodity $k \in K$ on G_c^* with respect to the arc lengths c_{ij}^k
2. setting $x_{ij}^k = 1$ for all arcs in the shortest origin-destination path for commodity k , and $y_{ij} = 1$ for all arcs that belong to at least one of these shortest paths

The value of the objective function for this integral solution (x, y) is

$$Z_c = \sum_{k \in K} \sum_{(i,j) \in A_c^*} c_{ij}^k x_{ij}^k + \sum_{\{i,j\} \in \bar{A}_c^*} F_{ij} y_{ij} \quad (17)$$

From definitions (3.2) and (3.3) construct the sets H , \bar{H} , H_k and \bar{H}_k for (x, y) . Note that $H \subseteq A_c^*$ and $\bar{H} \subseteq \bar{A}_c^*$. Then (17) becomes

$$Z_c = \sum_{k \in K} \sum_{(i,j) \in H_k} c_{ij}^k + \sum_{\{i,j\} \in \bar{H}} F_{ij} \quad (18)$$

Since $\bar{H} \subseteq \bar{A}_c^*$, $\sum_{\{i,j\} \in \bar{H}} F_{ij} \leq \sum_{\{i,j\} \in \bar{A}_c^*} F_{ij}$. Also, since H_k contains only the shortest path arcs for commodity k , $\sum_{(i,j) \in H_k} c_{ij}^k \leq \sum_{(i,j) \in A_c^*} c_{ij}^k x_{ij}^{k*}$. Summing the latter inequality over the set of commodities and adding it to the former, we conclude that

$$Z_c^* \geq Z_c \quad (19)$$

From Lemma 3.1 Z_c can be expressed as

$$Z_c = \sum_{k \in K} \sum_{(i,j) \in H_k} (c_{ij}^k + w_{ij}^k) + \sum_{\{i,j\} \in \bar{H}} s_{ij} + \sum_{k \in K} \sum_{(i,j) \in H \setminus H_k} w_{ij}^k \quad (20)$$

where (u, w) is the dual feasible solution derived by the labeling method when applied to $G = (N, A)$, and s_{ij} the associated slacks. The second and third terms in (20) are always greater than or equal to zero, since the slack and dual variables are non-negative (by definition and feasibility, respectively). Thus,

$$Z_c \geq \sum_{k \in K} \sum_{(i,j) \in H_k} (c_{ij}^k + w_{ij}^k) \quad (21)$$

Note that H_k contains arcs that form an origin-destination path for k , the length of which, in terms of \hat{c}_{ij}^k , is greater than the one derived by the labeling method (see Property 3.4). Let A_k^o be the set of zero-slack arcs that constitute a shortest origin-destination path (with respect to the modified costs \hat{c}_{ij}^k) for commodity k derived by the labeling method. Consequently,

$$\sum_{(i,j) \in H_k} (c_{ij}^k + w_{ij}^k) \geq \sum_{(i,j) \in A_k^o} (c_{ij}^k + w_{ij}^k) \quad (22)$$

Summing both sides of (22) over all commodities $k \in K$, we obtain

$$\sum_{k \in K} \sum_{(i,j) \in H_k} (c_{ij}^k + w_{ij}^k) \geq \sum_{k \in K} \sum_{(i,j) \in A_k^o} (c_{ij}^k + w_{ij}^k) \quad (23)$$

From Corollary 3.1 $Z_D^o = \sum_{k \in K} \sum_{(i,j) \in A_k^o} (c_{ij}^k + w_{ij}^k)$. Thus, comparing (19) and (23) we conclude that

$$Z_c \geq Z_D^o \quad (24)$$

By hypothesis, $Q_{G'} = \emptyset$. Since $G' = (N, A_o)$, there exists at least one arc in the optimal solution that does not belong to \bar{A}_o . Otherwise, G' would contain the optimal solution (contradiction). Therefore,

$$\exists \{i_1, j_1\} \in \bar{A}_c^* \setminus \bar{A}_o \quad (25)$$

The slack, $s_{i_1 j_1}$ of $\{i_1, j_1\}$ is non-zero by definition of \bar{A}_o , i.e., $s_{i_1 j_1}^o > 0$.

Reconsider now H , i.e., the set of arcs that belong to the shortest path, with respect to the original costs c_{ij}^k , for at least one commodity $k \in K$, in problem $\mathbf{FP}(G_c^*)$. H may or may not be a subset of A_o .

Case 1 $H \subseteq A_o$

In this case, H is a subset of the set of zero-slack arcs produced by the labeling method. Note that $H \subseteq A_o$ implies that $\bar{H} \subseteq \bar{A}_o$ and from (25) we have $\{i_1, j_1\} \notin \bar{H}$. In addition, since $\{i_1, j_1\} \in \bar{A}_c^*$ and $\bar{H} \subseteq \bar{A}_c^*$ by definition, we obtain

$$\sum_{\{i,j\} \in \bar{A}_c^*} F_{ij} \geq F_{i_1, j_1} + \sum_{\{i,j\} \in H} F_{ij} \quad (26)$$

Since H_k is the shortest path (with respect to the original arc costs, c_{ij}^k) for commodity $k \in K$,

$$\sum_{(i,j) \in A_c^*} c_{ij}^k x_{ij}^{k*} \geq \sum_{(i,j) \in H_k} c_{ij}^k \quad \forall k \in K \quad (27)$$

Summing both sides of inequality (27) over the set of commodities K , adding the result to (26) and taking into account the definitions of Z_c^* and Z_c , (16) and (18) respectively, we obtain

$$Z_c^* \geq F_{i_1, j_1} + Z_c \quad (28)$$

By definition, $\phi \leq s_{i_1 j_1}^o \leq F_{i_1 j_1}$, since each slack is less than or equal to the associated fixed cost and ϕ is the minimum non-zero slack. Considering that $Z_c \geq Z_D^o$ (from 24), inequality (28) results in

$$Z_c^* \geq Z_c + \phi \geq Z_D^o + \phi \quad (29)$$

Case 2 $H \not\subseteq A_o$

In this case, the integral solution (x, y) defined at the beginning of this proof comprises arcs with non-zero slack in the dual solution (u, w) . Note that, $H \not\subseteq A_o$ implies $\bar{H} \not\subseteq \bar{A}_o$.

Thus, there exists a non-zero slack arc in $\bar{H} \setminus \bar{A}_o$; i.e.

$$\exists \{i_2, j_2\} \in \bar{H} \setminus \bar{A}_o : s_{i_2 j_2}^o > 0 \quad (30)$$

Adding the term $\sum_{\{i,j\} \in \bar{H}} s_{ij}$ in both sides of inequality (23) we obtain

$$\sum_{k \in K} \sum_{(i,j) \in H_k} (c_{ij}^k + w_{ij}^k) + \sum_{\{i,j\} \in \bar{H}} s_{ij} \geq \sum_{k \in K} \sum_{(i,j) \in A_k^o} (c_{ij}^k + w_{ij}^k) + \sum_{\{i,j\} \in \bar{H}} s_{ij} \quad (31)$$

From the definition of Z_c , (20), Corollary 3.1 and the non-negativity of the dual variables, w , (31) results in

$$Z_c^h \geq Z_D^o + \sum_{\{i,j\} \in \bar{H}} s_{ij}^o \quad (32)$$

since $\{i_2, j_2\} \in \bar{H} \setminus \bar{A}_o$, $s_{i_2 j_2} \leq \sum_{\{i,j\} \in \bar{H}} s_{ij}$ and $\phi \leq s_{i_2 j_2}$. Then, inequality (32) can be written as

$$Z_c^h \geq Z_D^o + s_{i_2 j_2}^o \geq Z_D^o + \phi \quad (33)$$

Finally, from (33) and (24) we conclude that

$$Z_c^* \geq Z_D^o + \phi$$

Q.E.D.

Theorem 2 below, shows that the bound Z_{lb}^q obtained from each iteration q of algorithm IDA is a lower bound to the optimal solution of **CFP**(G).

Theorem 2 *At each iteration q of algorithm IDA, $Z_{lb}^q \leq Z_c^*$.*

Proof This result will be shown by induction. Consider the first iteration, i.e. $q = 1$. In *Step 5* of IDA, there are two possibilities. Either $Q_{G'_1} \neq \emptyset$ and the algorithm terminates or $Q_{G'_1} = \emptyset$ and the algorithm proceeds to identify capacity constraint violations. In the first case, $Z_{lb}^1 = Z_D^o$, and Z_{lb}^1 is a lower bound to the capacitated problem, since Z_D^o is a lower bound to the uncapacitated problem. In the second case, Z_{lb}^1 is updated in *Step 6*: $Z_{lb}^1 = Z_D^o + \phi_1$. According to Theorem 1, $Z_D^o + \phi_1$ is a lower bound to the optimal solution of the capacitated problem **CFP**(G). Thus, for $q = 1$, $Z_{lb}^q \leq Z_c^*$.

Assume that $Z_{lb}^q \leq Z_c^*$ holds for iteration $q = l$. It will be shown that this inequality also holds for iteration $q = l + 1$.

If $Q_{G'_{l+1}} \neq \emptyset$, the algorithm terminates with $Z_b^{l+1} = Z_D^l + \phi_l$, i.e. the lower bound of iteration l (*Step 5*). From the induction hypothesis, $Z_b^{l+1} \leq Z_C^*$.

If $Q_{G'_{l+1}} = \emptyset$, then it is left to show that $Z_D^{l+1} + \phi_{l+1} \leq Z_C^*$, where Z_D^{l+1} is the dual objective function value after *Step 4* of iteration $l+1$. Note that after iteration l only one of the variable costs was altered (see *Step 6*). Consequently, during iteration $l+1$ the labeling method will provide a dual objective, Z_D^{l+1} , that may differ from the one derived in iteration l by at most ϕ_l , i.e.

$$Z_D^{l+1} \leq Z_D^l + \phi_l \quad (34)$$

Since $Q_{G'_{l+1}} = \emptyset$, there exists at least one non-zero slack arc that is active in the optimal solution of the capacitated problem but is not included in the dual-ascent solution since its slack is positive. Let $\{i', j'\} \in \bar{A}_c^* \setminus \bar{A}_o^{l+1}$ be such an arc. Since $A_o^l \subseteq A_o^{l+1}$ (by construction), then $(i', j') \notin A_o^l$.

Consider now the commodity k_l for which the cost of arc $\{i_l, j_l\}$ has been modified in *Step 6*. Obviously the shortest paths with respect to the modified costs in iteration l do not contain arc $\{i', j'\}$. Let H' be the set of arcs of the shortest origin-destination path, with respect to \hat{c}_{ij}^k , for commodity k_l that includes arc $\{i', j'\}$ (see Fig.1). Then, if

$$\sum_{(i,j) \in H'} (c_{ij}^{k_l} + w_{ij}^{k_l}) + \sum_{\{i,j\} \in \bar{H}'} s_{ij} - \sum_{(i,j) \in A_o^{k_l}} (c_{ij}^{k_l} + w_{ij}^{k_l}) \leq \phi_l$$

the dual ascent iterations in *Step 4* would reduce the slack of arcs $\{i, j\} \in \bar{H}'$ to zero and, thus would include arc $\{i', j'\}$ in \bar{A}_o^{l+1} . This contradicts our conclusion above, and consequently

$$\sum_{(i,j) \in H'} (c_{ij}^{k_l} + w_{ij}^{k_l}) + \sum_{\{i,j\} \in \bar{H}'} s_{ij} - \sum_{(i,j) \in A_o^{k_l}} (c_{ij}^{k_l} + w_{ij}^{k_l}) > \phi_l \quad (35)$$

Let us now consider the set of zero-slack arcs after *Step 6* of iteration $l+1$. There are two alternatives: i) $A_o^{k^{l+1}} \subseteq A_c^*$; i.e., the optimal solution contains all arcs with zero-slack at iteration $l+1$, and ii) $A_o^{k^{l+1}} \not\subseteq A_c^*$; i.e., the optimal solution does not contain all zero-slack arcs in $A_o^{k^{l+1}}$.

In the first case, the difference between Z_D^{l+1} at iteration $l+1$ and the optimal objective function value Z_C^* includes at least the unabsorbed fixed cost of $\{i', j'\} \in \bar{A}_c^* \setminus \bar{A}_o^{l+1}$. This is because the fixed cost of the optimal solution contains the fixed cost of all arcs in the path

$A_o^{k^l}$ plus the cost $F_{ij'}$. Since $F_{ij'} \geq s_{ij'} \geq \phi_{l+1}$,

$$Z_D^{l+1} + \phi_{l+1} \leq Z_c^* \quad (36)$$

In the second case, flow should be diverted from path \bar{A}_o^{l+1} to path H' , or some segments of path \bar{A}_o^{l+1} should be eliminated and path H' should be included. From inequality (35), we know that H' has length at least ϕ_l greater than $A_o^{k^l}$. Since the increase of $c_{ij_l}^{k_l}$ by ϕ_l has not transformed H' to the shortest modified cost path, the difference between the left and right hand side of inequality (35) has not been eliminated. Consequently, we need to further increase the length of the current path by at least the minimum slack of the arcs in H' . Since ϕ_{l+1} is less than or equal to this slack, we obtain

$$Z_D^{l+1} + \phi^{l+1} \leq Z_c^* \quad (37)$$

Furthermore, considering that $Z_b^{l+1} = Z_D^{l+1} + \phi^{l+1}$ and inequalities (36) and (37) we have $Z_b^{l+1} \leq Z_c^*$, i.e. the inequality $Z_b^q \leq Z_c^*$ holds for $q = l + 1$. Consequently, it holds $\forall q \in \mathcal{N}_+$.

Q.E.D.

5 An illustrative example

The sample network shown in Figure 2, will be employed to illustrate the application of algorithm IDA. The graph consists of eight nodes, $N = \{0, 1, \dots, 7\}$ and nine arcs, $\bar{A} = \{\{0, 2\}, \dots, \{5, 7\}\}$. Two commodities are to be transported :

- five units of commodity 0 ($f_0 = 5$) from node $O(0) = 0$ to node $D(0) = 6$
- three units of commodity 1 ($f_1 = 3$) from node $O(1) = 1$ to node $D(1) = 7$

All unit variable costs ($\bar{c}_{ij}^{k_i}$) are set equal to one. Thus, $c_{ij}^{k_i} = f_k \forall (i, j) \in A$, and the fixed charge is the same for each arc ($F_{ij} = 10$). Also, all arc capacities are assumed to be equal to seven units of flow ($B_{ij} = 7$).

Note that the solution of the linear programming relaxation of **CFP**(G) on this graph yields $Z_{LP} = 79$ and the optimal objective function value is $Z_c^* = 95$.

At the beginning of IDA, the dual variables u_6^0 and u_7^1 are set equal to the shortest origin-destination distances for flow 0 and 1, i.e. 15 and 9 respectively. After the first termination of the labeling method, i.e. after *Step 4* of the first iteration ($q = 1$) of IDA, the non-zero dual and slack variables assume the following values:

- $u_0^1 = 10, u_1^0 = 10, u_2^0 = 15, u_2^1 = 13, u_3^0 = 12, u_3^1 = 16, u_4^0 = 12, u_4^1 = 16, u_5^0 = 27, u_5^1 = 19, u_6^0 = 42, u_6^1 = 9, u_7^0 = 15, u_7^1 = 32$
- $w_{02}^0 = 10, w_{12}^1 = 10, w_{25}^0 = 7, w_{25}^1 = 3, w_{35}^0 = 10, w_{45}^0 = 10, w_{56}^0 = 10, w_{57}^1 = 10$
- $s_{23} = 10, s_{24} = 10$

The set of zero-slack arcs is $A_o^1 = \{\{0, 2\}, \{1, 2\}, \{2, 5\}, \{3, 5\}, \{4, 5\}, \{5, 6\}, \{5, 7\}\}$ and $G'_1 = (N, A_o^1)$. Since $Q_{G'_1} = \emptyset$, IDA uses the augmented graph to identify the arcs for which the capacity constraints are violated (see Appendix B). In this case, only the capacity of arc $\{2, 5\}$ is violated. Thus, the cost c_{25}^0 is increased by the minimum non-zero slack ($\phi_1 = 10$), and the procedure returns to *Step 2*. Note that commodity $k = 0$ is chosen for variable cost update since $f_0 > f_1$. The dual objective is $Z_D^1 = u_6^0 + u_7^1 = 74$ and the lower bound is $Z_{lb}^1 = Z_D^1 + \phi_1 = 84$.

After the second iteration of the dual ascent procedure (*Step 4* of IDA) the non-zero dual and slack variables assume the following values:

- $u_0^1 = 10, u_1^0 = 10, u_2^0 = 15, u_2^1 = 13, u_3^0 = 22, u_3^1 = 16, u_4^0 = 22, u_4^1 = 16, u_5^0 = 37, u_5^1 = 19, u_6^0 = 52, u_6^1 = 9, u_7^0 = 15, u_7^1 = 32$
- $w_{02}^0 = 10, w_{12}^1 = 10, w_{23}^0 = 2, w_{24}^0 = 2, w_{25}^0 = 7, w_{25}^1 = 3, w_{35}^0 = 10, w_{45}^0 = 10, w_{56}^0 = 10, w_{57}^1 = 10$
- $s_{23} = 8, s_{24} = 8$

The set of zero-slack arcs is $A_o^2 = A_o^1$ and $G'_2 = (N, A_o^2)$. Since $Q_{G'_2} = \emptyset$ and $\phi_2 = 8$, the algorithm sets $c_{25}^0 \leftarrow (c_{25}^0 + 8)$ and returns to *Step 2*. The dual objective is $Z_D^2 = u_6^0 + u_7^1 = 84$ and the lower bound is $Z_{lb}^2 = Z_D^2 + \phi_2 = 92$.

After the third iteration of the dual ascent, $A_o^3 = A$ and $G'_3 = (N, A_o^3)$. Now, $Q_{G'_3} \neq \emptyset$ and the algorithm terminates with $Z_{ib}^3 = Z_{ib}^2 = 92$. Note that

$$Z_{LP} = 79 < Z_{ib}^3 = 92 < Z_c^* = 95$$

as expected. Finally, the cost of the feasible network design on G'_3 is $Z_c = 95$.

6 Computational results

Algorithm IDA was implemented in C on a Sun Spark IPX Workstation. Numerous computational tests were performed. In this section we present the results of randomly generated case problems with number of arcs varying from 20 to 60. These are mainly problems on grid graphs (i.e. the degree of each node is at most 4). For each case, 50 different problems were solved. The parameters employed to generate the case problems were uniformly distributed on the ranges shown in Table 1.

Table 1: Ranges of randomly selected parameters in case problems

Parameter	Range
Fixed charge	5 - 40
Number of commodities	10 - 35
Commodity flow	4 - 12
Arc capacity	30 - 250

We emphasize the comparison of the lower bounds obtained by the linear programming relaxation and the dual ascent algorithm. However, we also note that the designs generated by the dual ascent algorithm can be employed as initial feasible solutions for local improvement heuristics (e.g. add-drop heuristics or random search techniques).

Table 2 shows the number of test problems for which the dual ascent method (IDA) provides a sharper lower bound than the linear programming relaxation (out of 50 problems per case).

Analysis of these results showed that in the few cases for which $Z_{LP} < Z_{ib}$, the networks consisting of the arcs that belonged to the shortest path (with respect to the variable costs c_{ij}^k)

Table 2: Lower bound comparison (50 problems per case)

Number of arcs	Number of times $Z_{lb} \geq Z_{LP}$
20	43
30	45
40	41
50	42
60	46

for at least one commodity were feasible for **CFP**. In other words, routing every commodity on a shortest origin-destination path violated no capacity constraints. In addition, for these cases only one step of dual ascent procedure was executed during the application of algorithm IDA. Consequently, no variable cost updates (*Step 6*) were performed and the lower bound obtained by IDA was equal to the dual objective function evaluated by the first application of the labeling method. Since the latter is an approximate solution to the dual of the linear relaxation of the uncapacitated problem, it is clear that $Z_{LP} > Z_{lb}$ was expected in such cases.

Table 3: Comparison of IDA solution vs. optimum and lower bounds (average % deviation)

Number of arcs	Optimum	Dual ascent	Linear relaxation
	$\frac{Z_{IDA}-Z_c^*}{Z_c^*} \times 100\%$	$\frac{Z_{IDA}-Z_{lb}}{Z_{lb}} \times 100\%$	$\frac{Z_{IDA}-Z_{LP}}{Z_{LP}} \times 100\%$
20	3.7	6.5	10.3
30	4.5	7.2	10.8
40	-	7.4	11.9
50	-	8.1	12.2
60	-	8.3	12.6

Columns 3 and 4 of Table 3 present the average deviation between the value of the objective function of the feasible network design obtained by IDA (Z_{IDA}) and the two lower bounds, Z_{LP} and Z_{lb} , respectively. The results show that the lower bounds derived by the dual ascent approach are much tighter, on the average, than those of the LP relaxation.

Note that the examples in which capacity constraints are not active are included in these results. Furthermore, column 2 of Table 3 presents the average deviation of the IDA solution (Z_{IDA}) from the optimum (Z_c^*) for the 20 and 30 arc examples. The optimum solution was evaluated by a branch-and-bound scheme. It is worth noting that for most of these cases the deviation from the optimum was not very large (less than 12% in the worst case).

Figure 3 relates the quality of the lower bounds to the ratio of fixed to average scaled variable cost. Recall that the fixed cost F_{ij} , and the variable cost c_{ij}^k for commodity $k \in K$, were the same for all $\{i, j\} \in \bar{A}$. The following terms are defined for this comparison:

- $r = \frac{F_{ij}}{\sum_{k \in K} c_{ij}^k / |K|}$, where $|K|$ is the cardinality of the set of commodities
- $D(x)$ = average percentage difference between dual ascent and LP-relaxation lower bounds for all those problems with $r \in (x - 0.5, x]$. For example, $D(r)$ for $r = 1.5$ in Figure 3 includes all test problems for which the ratio r is between 1.0 and 1.5.

Table 4 shows the number of test problems in each range.

Table 4: Distribution of r for example problems

Range of r	Number of problems
0.0-0.5	28
0.5-1.0	31
1.0-1.5	29
1.5-2.0	32
2.0-2.5	32
2.5-3.0	30
3.0-3.5	33
3.5-4.0	35

The trend displayed in Figure 3 is not surprising. The dual ascent method proceeds by including unabsorbed fixed-charges to the dual objective; when it terminates, the fixed-charges of the arcs that have zero slack have been included in the lower bound. On the other hand, the linear programming relaxation may include only fractions of arc fixed-charges.

Consequently, it is expected that higher values of the ratio r will result in improved dual ascent lower bounds.

Finally, Figure 4 shows the difference between the dual ascent and LP-relaxation lower bounds, as a function of the overflow (flow on arc copies) after the first iteration of algorithm IDA. The horizontal axis of the graph represents the percentage (s) of overflow, with respect to the total network flow; i.e.

$$s = \frac{\sum_{k \in K} \sum_{(i,j) \in A} \hat{f}_{ij}^k}{\sum_{k \in K} f_k}$$

The vertical axis shows the percentage difference between the lower bounds as defined above.

As shown in Figure 4, the larger the overflow after the first iteration, the better the dual ascent lower bound. This was also expected, since a large flow on arc copies means that the solution derived by the labeling method after the first iteration of IDA cannot accommodate large amounts of the overall material flow. Consequently, the linear programming relaxation does not provide a close approximation of the mixed integer program. The number of test problems in each region of s is shown in Table 5.

Table 5: Distribution of s for example problems

Range of s	Number of problems
0-5	89
5-10	73
10-15	52
15-20	24
20-25	12

7 Conclusions

In this paper the multi-commodity fixed-charge capacitated network design problem has been studied. This work is motivated by the design of material handling flow paths in a manufacturing facility. Applications of the problem can be also found in othe areas, including transportation, communication, and distribution networks.

An iterative dual-ascent procedure that determines a lower bound and a feasible solution to the problem has been developed. The proposed algorithm includes additional arcs to the network iteratively by increasing selected arc variable costs. This method is similar to the Lagrangean concept of increasing the multiplier values of violated constraints to direct the solution towards primal feasibility. Extensive experimental results have shown that the resulting lower bounds are sharper than those obtained by the linear programming relaxation. In addition, the procedure yields good primal feasible solutions.

The lower bound evaluation procedure can be embedded in a branch-and-bound scheme to derive optimal solutions to the mixed integer program and may also serve as a basis for comparison of heuristics. The application of this procedure to the facility design problem can be found in Ioannou *et al.* [3].

Acknowledgments

This work was supported by the Institute for Systems Research of the University of Maryland under Grant # NSFD CD 8803012.

Appendix A

This appendix provides a brief description of the labeling method used in the proposed algorithm. This method generalizes Dijkstra's algorithm for shortest path and the algorithm of Wong for Steiner tree problems.

The labeling method, as the name implies, successively labels nodes, starting from each commodity's destination and proceeding towards its origin. Two sets are required $\forall k \in K$: i) $N_2(k)$, the set of labeled nodes for commodity $k \in K$, and ii) $N_1(k) = N \setminus N_2(k)$, the set of unlabeled nodes. At each step, at most one arc (i, j) is included in the origin-destination shortest path for one commodity by reducing its slack to zero. For this arc, node j was already labeled and node i now enters $N_2(k)$. The dual ascent strategy of the labeling method can be summarized as follows.

1. Initialize the dual variables: $w_{ij}^k = 0$, $u_{D(k)}^k = \text{shortest } O(k) - D(k) \text{ path with respect to the arc lengths } c_{ij}^k$, $N_2(k) = \{D(k)\} \forall k \in K$.
2. Select $k \in K$ and (i, j) between $N_1(k)$ and $N_2(k)$. Increase w_{ij}^k and reduce the corresponding slack s_{ij} .
3. If s_{ij} equals zero, include node i in $N_2(k)$. Update the shortest path lengths, $u_j^k \forall j \in N_2(k)$. This increases the value of the dual objective.
4. Repeat steps 2 and 3 until $O(k) \in N_2(k)$, $\forall k \in K$.

For an extended description of the labeling method, see Balakrishnan *et al.* [1].

Appendix B

In this appendix the generation of the augmented graph used to identify arcs that violate capacity constraints is presented.

Consider the set of zero slack arcs, \bar{A}_o^q , produced by the dual ascent algorithm at a certain iteration q . For each arc $\{i, j\} \in \bar{A}_o^q$, we introduce an artificial (duplicate) arc $\{i', j'\}$ and assign to it a very large variable cost, $M \sim \infty$, for each commodity $k \in K$, a zero fixed-charge and an unlimited capacity. Let \bar{A}'_q be the set of duplicate arcs and A'_q the corresponding set of directed arcs. Also, let \hat{x}_{ij}^k be the fraction of overflow for commodity k on arc $\{i, j\} \in A'_q$.

The fixed-charge capacitated network design problem is formulated on the augmented graph $G = (N, \bar{A}'_q)$ as follows:

$$\text{minimize} \quad Z = \sum_{k \in K} \left(\sum_{(i,j) \in \bar{A}_o^q} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in A'_q} M \hat{x}_{ij}^k \right) + \sum_{\{i,j\} \in \bar{A}_o^q} F_{ij} y_{ij}$$

subject to :

$$\sum_{j \in N} (x_{ji}^k + \hat{x}_{ji}^k) - \sum_{l \in N} (x_{il}^k + \hat{x}_{il}^k) = \begin{cases} -1 & \text{if } i = O(k) \\ 1 & \text{if } i = D(k) \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in N, k \in K$$

$$\sum_{k \in K} f_k(x_{ij}^k + x_{ji}^k) \leq B_{ij} \quad \forall \{i, j\} \in \bar{A}_o^q$$

$$\begin{aligned}
x_{ij}^k, x_{ji}^k &\leq y_{ij} & \forall \{i, j\} \in \bar{A}_o^q, k \in K \\
x_{ij}^k, \hat{x}_{ij}^k &\geq 0 & \forall (i, j) \in A_o^q \cup A_q', k \in K \\
y_{ij} &\in \{0, 1\} & \forall \{i, j\} \in \bar{A}_o^q
\end{aligned}$$

Let us consider the linear programming relaxation of the above problem, i.e. the formulation that results by relaxing the last set of constraints ($y_{ij} \in \{0, 1\}$). If any variable \hat{x}_{ij}^k in the optimal solution of this mixed integer program relaxation assumes positive value, it is clear that the problem $\mathbf{FP}(G_q = (N, \bar{A}_o^q))$ is infeasible. Note that if there existed a feasible solution to $\mathbf{FP}(G_q)$, then $\hat{x}_{ij}^k = 0 \ \forall \{i, j\} \in \bar{A}_q'$, since the very large variable cost of arcs $\{i, j\} \in \bar{A}_q'$ would prohibit flow across them in the optimal solution. Thus, arcs that violate capacity constraints can be identified by checking the values of the variables \hat{x}_{ij}^k . The amount of commodity overflow on each arc is given by:

$$\hat{f}_{ij}^k = f_k \cdot \hat{x}_{ij}^k$$

where f_k is the total commodity flow.

References

- [1] Balakrishnan, A., Magnanti, T.L. and Wong, R.T., 1989. "A dual-ascent procedure for large-scale uncapacitated network design," *Operations Research*, **37**(5), 716-740.
- [2] Chhajed, D., Montreuil, B. and Lowe, T.J., 1992. "Flow network design for manufacturing systems layout," *European Journal of Operational Research*, **57**(2), 145-161.
- [3] Ioannou, G., Minis, I., Nagi, R. and Proth, J.M., 1994. "Design of material flow paths," *Proceedings of the 1994 NSF Design and Manufacturing Grantees Conference*, Cambridge MA, January 5-7.
- [4] Johnson, D.S., Lenstra, J.K. and Rinnooy, H.G., 1978. "The complexity of the network design problem," *Networks*, **8**, 279-285.
- [5] Khang, D.B. and Fujiwara, O., 1991. "Approximate solutions of capacitated fixed-charge minimum cost network flow problems," *Networks*, **21**, 689-704.

- [6] Kim, H.K. and Tanchoco, J.M.A., 1993. "Economical design of material flow paths," *International Journal of Production Research*, **31**(6), 1387-1407.
- [7] Magnanti, T.L. and Wong, R.T., 1984. "Network design and transportation planning: models and algorithms," *Transportation Science*, **18**(1), 1-55.
- [8] Minoux, M., 1989. "Network synthesis and optimum network design problems: models, solution methods and applications," *Networks*, **19**, 313-360.

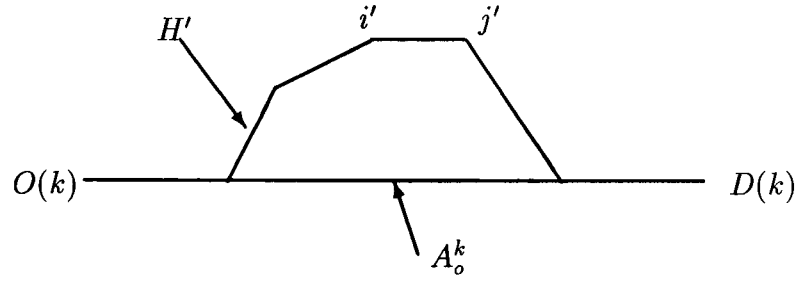


Figure 1: Origin-destination paths for commodity k

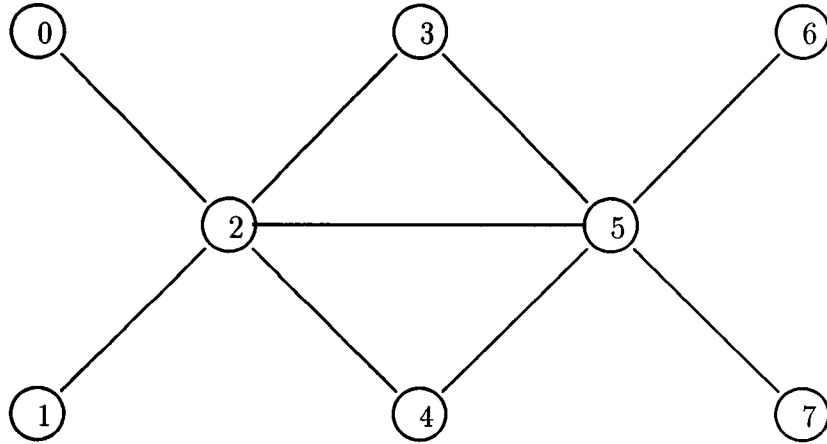


Figure 2: Example network

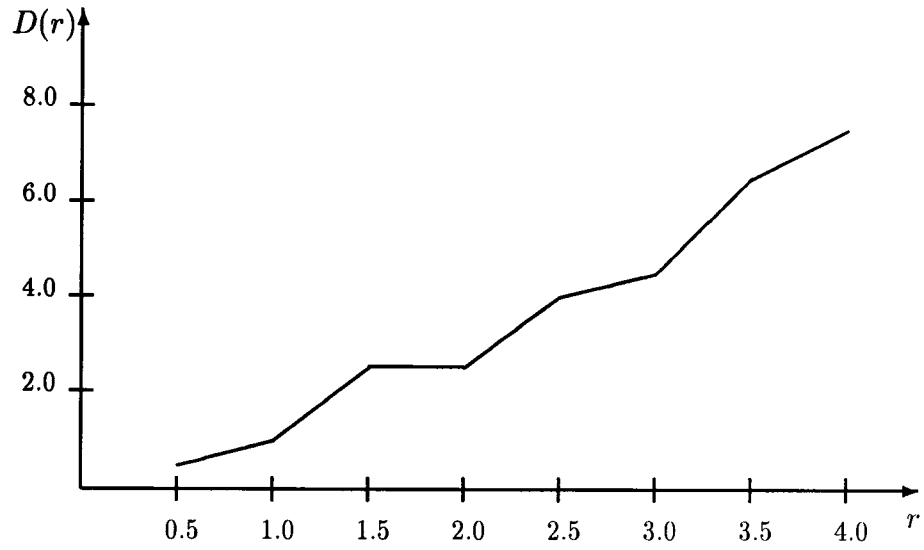


Figure 3: Average percentage difference between lower bounds (D vs. r)

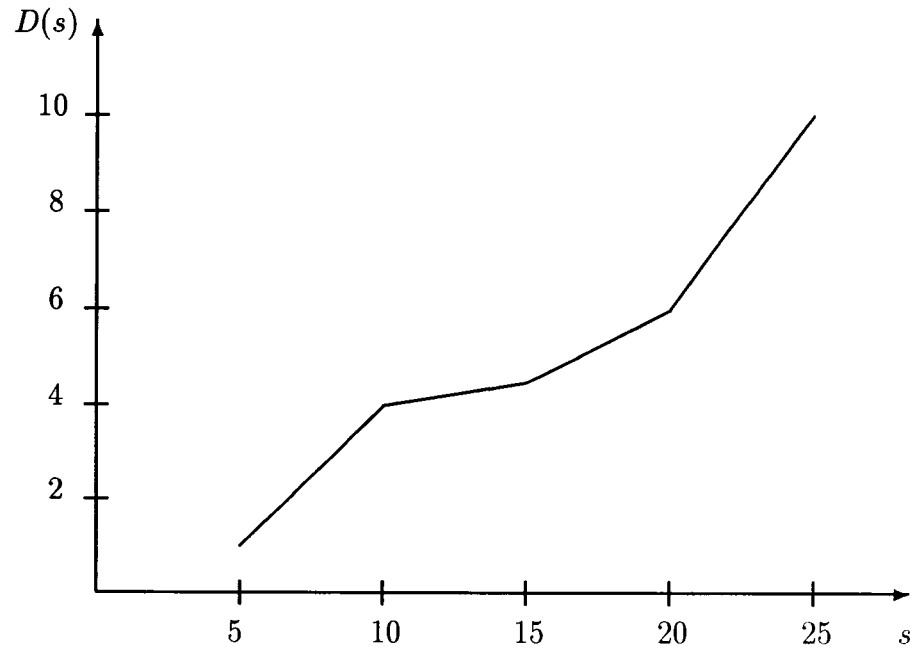


Figure 4: Average percentage difference between lower bounds (D vs. s)