REPORTS
IN
INFORMATICS

ISSN 0333-3590

Analysis and Computational Study of Flow-based Formulations for Minimum-Energy Multicasting in Wireless Ad Hoc Networks

Joanna Bauer, Dag Haugland and Di Yuan

REPORT NO 305 **August 2005**

Department of Informatics

UNIVERSITY OF BERGEN

Bergen, Norway

This report has URL http://www.ii.uib.no/publikasjoner/texrap/ps/2005-305.ps

Reports in Informatics from Department of Informatics, University of Bergen, Norway, is available at http://www.ii.uib.no/publikasjoner/texrap/.

Requests for paper copies of this report can be sent to:

Department of Informatics, University of Bergen, Høyteknologisenteret,
P.O. Box 7800, N-5020 Bergen, Norway

Analysis and Computational Study of Flow-based Formulations for Minimum-Energy Multicasting in Wireless Ad Hoc Networks *

Joanna Bauer[†], Dag Haugland[‡], Di Yuan[§]

8th August 2005

Abstract

Multicast sessions in wireless ad hoc networks require messages to be routed from a source node to a set of recipient nodes. Message forwarding from any intermediate node in the route is a power-consuming operation, and since batteries are the only available energy source, this resource is scarce. In this paper we present and analyze integer programming models for routing multicast or broadcast messages in wireless ad hoc networks, such that the total energy consumption is minimized. A well-known single commodity flow model, is strengthened by introducing multicommodity flow variables. Further strengthening is accomplished by lifting the capacity constraints of the multicommodity flow model. We also develop equally strong cut formulations, and by extensions of the network we show how the problem can be formulated as Minimum Steiner Arborescence models. The strengths of the suggested models are analyzed and compared, and numerical experiments are made on a selection of the models.

Keywords: ad hoc networks; broadcasting; multicasting; integer programming

1 Introduction

Ad hoc wireless networks are distinguished from cellular wireless networks and wired networks in that no fixed backbone infrastructure is installed. A typical configuration for signal transmission in such networks is that each node uses omnidirectional antennas, and in order to establish communication, a transmission power must be assigned to each antenna. Practical applications of ad hoc networks arise for instance in emergency disaster relief and in battlefields.

Since the only energy source of the antennas in many cases is a battery, it is relevant to minimize the total energy required to accomplish a communication session. Transmission of messages is based on the principle of broadcasting. This is a direct implication of the fact that the signal transmitted from any antenna reaches all other nodes in a certain range, which is given by the transmission power. Therefore the power required at any transmitting node is the maximum, rather than

*Supported by The Research Council of Norway under contract 160233/V30

†Email: `Joanna.Bauer@ii.uib.no`, Department of Informatics, University of Bergen, PB. 7800, N-5020 Bergen, Norway

‡Email: `Dag.Haugland@ii.uib.no`, Department of Informatics, University of Bergen, PB. 7800, N-5020 Bergen, Norway

§Email: `diyua@itn.liu.se`, Department of Science and Technology, Linköping University, SE-601 74 Norrköping, Sweden

the sum, of the powers needed to reach all its recipients. This property is commonly referred to as the wireless multicast advantage.

A multicast session in an ad hoc network is a message routing from a designated node termed the source to a set of destination nodes. When all nodes but the source itself are destinations, we refer to it as a broadcast session. In either case, the message to any destination can be transmitted directly, or relayed through intermediate nodes. All relay nodes, and also the source, have to be assigned high enough power to pass the signal on to their successors in the relay.

The total energy consumed by a multicast session is given by the sum of such power assignments. Minimizing this energy subject to the constraint that the message can be transferred from the source to all destination nodes is referred to as the Minimum Ennergy Multicast Problem (MEMP). Note the resemblence of this problem with the minimum Steiner tree problem (minimum spanning tree problem in the broadcast case), but also the important distinction implied by the wireless multicast advantage. Recently, Čagalj et al. [3] proved that MEMP is NP-hard.

Wieselthier et al. [19, 20] suggested the frequently cited Broadcast Incremental Power (BIP) heuristic for the broadcast version of MEMP. They tested it experimentally, and proved good performance when compared to techniques designed for wired networks, such as minimum spanning tree algorithms. Theoretical analysis of the heuristic were conducted by Wan et al. [17] and Klasing et al. [12], concluding that the approximation ratio of MIP is in the range from 3.25 to 12.15. An extension of the heuristic was given in [12], and it was proved that the minimum spanning tree solution is no better than the solution suggested by this method. Heuristics without performance guarantee, based on refinements of BIP, are given by e.g. Das et al. [7, 8, 9, 15].

For the multicast version, Wieselthier et al. also formulated the Multicast Incremental Power (MIP) heuristic, which is nothing but BIP followed by pruning of all nodes that are not found on the path from the source to any destination. In [18] it was shown that this algorithm does not have a constant approximation ratio, and neither has the straightforward method of pruning the minimum spanning tree. However, there exist several approximation algorithms with constant approximation ratio for the minimum Steiner tree problem. Any of these can be applied also to MEMP, and in [18] it is shown that the corresponding approximation ratio for MEMP is constant. Other heuristics for MEMP with no performance guarantee are given by e.g. Cartigny et al. [4], Chu and Nikolaidis [5], Li and Nikolaidis [13], Lindsey and Raghavendra [14] and Montemanni et al. [16].

Approaches to solve MEMP to optimality do not appear extensively in the literature. Altinkemer et al. [1, 2] formulate the problem as an integer programming model of set-covering type. Das et al. [6] suggest three different mixed integer programming models for MEMP. One of these is a network flow model with binary design variables, which also is briefly discussed in [12]. In [22], Yuan strengthens it to a multi-commodity flow model, and uses this to evaluate the performance of the BIP heuristic.

Mixed integer programming formulations are useful for exact solution of MEMP instances of moderate size. Their relaxations can also be of great value for larger instances, since they enable lower bounding of the minimum power assignment. Whatever be the goal of a (mixed) integer model, it is relevant to aim for a strongest possible formulation, that is a model for which the continuous relaxation provides a largest possible lower bound.

In this paper we give a host of mixed integer programming models for MEMP. This study is accompanied with a careful analysis of the model strengths. Most of the formulations are based on network flow considerations, and the exceptions to this rule are closely related cut models. We also demonstrate that by certain extensions of the underlying network, the flow models can be developed into a

Minimum Steiner Arborescence problem. Hence our problem formulation invites the solution approaches designed for this problem, exact or approximate, to be applied to MEMP.

The rest of the paper is organized as follows: In the next section we introduce the basic concepts and mathematical notation used throughout. Sections 3-4 are devoted to definitions of the flow and cut models, as well as an analysis and a comparison of their strength. A main result here is that the strongest versions in each category are equally strong. In Section 5 we suggest equivalent flow and cut formulations based on incremental design variables. The Steiner models are given in Section 6, where we show that the strongest formulations in Sections 5-6 are just as strong as the strongest in Sections 3-4. Finally, Section 7 gives the results of some numerical experiments obtained from a selection of the models.

2 Preliminaries

Throughout the paper the following notation is used: We let $\mathcal{G} = (V_\mathcal{G}, A_\mathcal{G})$ denote the digraph with vertex and arc sets $V_\mathcal{G}$ and $A_\mathcal{G}$, respectively. If $S \subseteq V_\mathcal{G}$, we let $\bar{S} = V_\mathcal{G} \setminus S$. For any finite set S and any vector $u \in \Re^S$ we let u_s denote the component of u corresponding to $s \in S$. For vectors $u \in \Re^{A_\mathcal{G}}$, we use either u_a or u_{ij}, whichever is more convenient, to denote the component of u corresponding to the arc $a = (i, j)$. Furthermore, we let $A^+_\mathcal{G}(i) \subseteq A_\mathcal{G}$ and $A^-_\mathcal{G}(i) \subseteq A_\mathcal{G}$ denote the sets of arcs starting and ending, respectively, at node i. For any $b \in \Re^{V_\mathcal{G}}_+$, we let $\mathcal{F}(\mathcal{G}, b)$ denote the set of flow vectors $f \in \Re^{A_\mathcal{G}}_+$ satisfying $\sum_{a \in A^+_\mathcal{G}(i)} f_a - \sum_{a \in A^-_\mathcal{G}(i)} f_a = -b_i \ \forall i \in V_\mathcal{G}$, and for any $u \in \Re^{A_\mathcal{G}}_+$, we define the flow polytope $\mathcal{F}(\mathcal{G}, b, u) = \{f \in \mathcal{F}(\mathcal{G}, b) : f_a \leq u_a \forall a \in A_\mathcal{G}\}$.

Consider the digraph G with node set $V_G = \{i_0, i_1, \ldots, i_n\}$ and arc set $A_G = V_G \times V_G$ with arc weights (power requirements) $c \in \Re^{A_G}_+$, and a set of *destinations* $D \subseteq V_G \setminus \{i_0\}$. The problem is defined as follows:

PROBLEM 1. *Find a* power assignment $p \in \Re^{V_G}_+$ *minimizing* $\sum_{i \in V_G} p_i$ *such that for all* $d \in D$ *there is a path* $P_d = (V_{P_d}, A_{P_d})$ *from* i_0 *to* d *where* $p_i \geq c_{ij}$ *for all* $(i, j) \in A_{P_d}$.

In Problem 1, we let ζ^* denote the optimal value of $\sum_{i \in V_G} p_i$. Let $N = \{1, \ldots, n\}$, and for all $i \in V_G$ let $\pi_i : N \mapsto V_G \setminus \{i\}$ be a bijection such that $\left(c_{i\pi_i(1)}, \ldots, c_{i\pi_i(n)}\right)$ is monotonously non-decreasing, and let π'_i be its inverse. Hence $\pi_i(k)$ is the k-closest neighbor of i, where distance is measured by weights on out-going arcs. Also $\pi'_i(j)$ is interpreted as the position of $j \in V_G$ when all neighbors of i are ordered by non-decreasing distance from i. We also define $\pi'_i(S) = \min\{\pi'_i(j) : j \in S\}$ for any set $S \subseteq V_G \setminus \{i\}$, that is the position of the closest neighbor in the set S.

For any vector $u \in \Re^{A_G}$ and $k \in N$ we use u_{ik} as a short hand notation for $u_{i,\pi_i(k)}$. For any matrix $f \in \Re^{A_G \times D}$ we let $f^d \in \Re^{A_G}$ denote the column indexed by $d \in D$. Let $c_{i0} = 0$ and $\Delta c_{ik} = c_{ik} - c_{i,k-1}$ for $k \in N$. Define $b^d \in \Re^{V_G}$ $(d \in D)$ such that $b^d_{i_0} = -1$, $b^d_d = 1$ and $b^d_i = 0$ for $i \in V_G \setminus \{i_0, d\}$.

Denote the continuous relaxation of any (mixed) integer programming problem $\mathcal{I}$ by $\mathcal{I}^{\mathbf{LP}}$, and let $\zeta(\mathcal{I})$ denote the optimal objective function value of $\mathcal{I}^{\mathbf{LP}}$.

3 Network Flow Formulations

Feasible solutions of Problem 1 can be characterized by network flow between the source and the set of destinations. In this section we discuss several network flow formulations for the problem. Among them, the following single-commodity flow formulation [6] is probably the most straightforward one. This formulation, which we denote by **F0**, uses three sets of variables: flow variables $\{f_{ij}, (i,j) \in A_G\}$, power variables $\{p_i, i \in V_G\}$, and binary variables $\{z_{ij}, (i,j) \in A_G\}$. In **F0**, the demand vector b is defined as $b_{i_0} = -|D|$, $b_d = 1$ for all $d \in D$, and $b_i = 0$ for $i \in V_G \setminus (D \cup \{i_0\})$.

$$
\begin{aligned}
[\mathbf{F0}] \quad & \min \sum_{i \in V_G} p_i \\
\text{s. t.} \quad & f_{ij} \le |D| z_{ij} \quad \forall (i,j) \in A_G, && (1)\\
& z_{ij} \le c_{ij} p_i \quad \forall (i,j) \in A_G, && (2)\\
& f \in \mathcal{F}(G,b), && (3)\\
& z \in \{0,1\}^{A_G}.
\end{aligned}
$$

Due to (3), every destination $d \in D$ receives exactly one unit of flow from the source in any feasible solution of **F0**. Constraints (1) state that variable z_{ij} is one if arc (i,j) carries a positive amount of flow. The z-variables are linked to node power in (2).

From a bounding point of view, the continuous relaxation of formulation **F0** is very poor; the gap between $\zeta(\mathbf{F0})$ and ζ^* is huge. As is well known, a way to (significantly) improve the continuous relaxation is to use a multicommodity flow representation. A second weak point of **F0** lies in (2). Note that in any integer feasible solution, the power of node i is either zero, or equals one of the values in the discrete set $\{c_{i1}, c_{i2}, \ldots, c_{in}\}$. We can therefore associate binary variables to these power values, and represent the power of i as a scalar product.

Following the above discussion, we define the design variables $\{y_{ik}, i \in V_G, k \in N\}$ and flow variables $\{f^d_{ij}, (i,j) \in A_G, d \in D\}$, where

$$
y_{ik} = \begin{cases} 1, & \text{node } i \text{ is assigned power } c_{ik}, \\ 0, & \text{otherwise.} \end{cases}
$$

Using these two sets of variables, we arrive at our first multicommodity flow formulation for Problem 1:

$$
\begin{aligned}
[\mathbf{F1}] \quad & \min \sum_{i \in V_G} \sum_{k=1}^{n} c_{ik} y_{ik} \\
\text{s. t.} \quad & f^d_{ik} \le \sum_{\ell=k}^{n} y_{i\ell} \quad \forall i \in V_G, k \in N, d \in D, && (4)\\
& f^d \in \mathcal{F}(G, b^d), && (5)\\
& y \in \{0,1\}^{A_G}.
\end{aligned}
$$

Constraints (5) ensure that d is connected to the source, while constraints (4) say that flow can be assigned to the arc from i to $\pi_i(k)$ only if i is assigned a power at least as large as c_{ik}. Obviously, we have $\zeta(\mathbf{F0}) \le \zeta(\mathbf{F1})$. In practice, the former is often less than few percent of the latter.

We can strengthen formulation **F1** further. Note that, because at most one unit of flow will leave any node to any destination, we can include all flow variables of destination d for which the corresponding transmission power is greater than or equal to c_{ik} in the left-hand sides of (4). Doing so gives us the following strengthened multicommodity flow formulation, denoted by **F2**.

$$
\begin{aligned}
[\mathbf{F2}] \quad & \min \sum_{i \in V_G} \sum_{k=1}^{n} c_{ik} y_{ik} \\
\text{s. t.} \quad & \sum_{\ell=k}^{n} f_{i\ell}^{d} \leq \sum_{\ell=k}^{n} y_{i\ell} \quad \forall i \in V_G, k \in N, d \in D, && (6) \\
& f^d \in \mathcal{F}(G, b^d), && (7) \\
& y \in \{0,1\}^{A_G}.
\end{aligned}
$$

Clearly, the inequality $\zeta(\mathbf{F1}) \leq \zeta(\mathbf{F2})$ holds. Our computational experiments show that the difference between them can be quite significant.

Before finishing the presentation of the two multicommodity flow formulations, it is worth mentioning a set of valid inequalities. These inequalities are derived from the observation that in **F1** and **F2** the power assignment of each node is unique.

$$
\sum_{k=1}^{n} y_{ik} \leq 1 \quad \forall i \in V_G. \tag{8}
$$

Although adding (8) to **F1** and **F2** reduces the sets of feasible solutions, these inequalities are redundant for defining any optimal integer solution of the two formulations. Moreover, as stated by the proposition below, these inequalities do not improve the continuous relaxation of **F1** or **F2**. However, inequalities (8) may be useful when applying a Lagrangian relaxation technique [22].

PROPOSITION 1. $\mathbf{F1^{LP}}$ *and* $\mathbf{F2^{LP}}$ *have an optimal solution* (f, y) *satisfying (8).*

Proof. It is obvious that $\mathbf{F1^{LP}}$ ($\mathbf{F2^{LP}}$) has optimal solutions. Let $(\bar{f}, \bar{y})$ be one such solution, and f a reduction of $\bar{f}$ such that for all $d \in D$ and all directed cycles C in G, $f_a^d = 0$ for at least one $a \in A_C$. Since $f^d \in \mathcal{F}(G, b^d)$, it follows from the definition of b^d that $\sum_{a \in A_G^+(i)} f_a^d \leq 1$. By defining $y_{ik} = \max\{0, \bar{y}_{ik} - \max\{0, \sum_{\ell=k}^{n} \bar{y}_{ik} - 1\}\}$ ($i \in V_G$, $k \in N$), which implies $\sum_{\ell=k}^{n} y_{i\ell} = \min\{\sum_{\ell=k}^{n} \bar{y}_{i\ell}, 1\}, \forall k \in N$, the solution (f, y) is clearly feasible in $\mathbf{F1^{LP}}$ ($\mathbf{F2^{LP}}$). In addition, $\sum_{k=1}^{n} y_{ik} \leq 1$ for all $i \in V_G$, and $\sum_{a \in A_G} c_a y_a \leq \sum_{a \in A_G} c_a \bar{y}_a$. Hence the conclusion of the proposition. □

4 Cut Formulations

Dualism between flow and cut formulations is paramount in network problems. As an alternative to flow conservation constrains, connectivity between the source i_0 and nodes in D can be based on the a consideration of cuts separating i_0 from some destination $d \in D$. In this section we give two cut formulations of Problem 1, and discuss the strengths of their continuous relaxations in relation to their flow-based counterparts.

For any binary design vector y, let $G(y)$ denote the subgraph induced by y. That is, we let $V_{G(y)} = \{i \in V_G : \sum_{k=1}^{n} y_{ik} = 1\}$ and $A_{G(y)} = \left\{(i,j) \in A_G : \sum_{k=\pi_i'(j)}^{n} y_{ik} = 1\right\}$. For all $d \in D$, constraints (4) and (5) in **F1** ((6) and (7) in **F2**) ensure that $G(y)$ has a path from i_0 to d. Alternatively, connectivity can be obtained by forcing $G(y)$ to contain at least one arc in every cut separating i_0 from d. That is, in all sets $S \subset V_G$ containing i_0 but not d, there must be some node $i \in S$ whose power is strong enough to reach some $j \in \bar{S}$. This condition is both sufficient and necessary to characterize a feasible solution to Problem 1.

PROPOSITION 2. *A binary design vector y is feasible in Problem 1 if and only if for any $S \subset V_G$ such that $i_0 \in S$ and $D \cap \bar{S} \neq \emptyset$, the power of the nodes in S enables at least one arc in the cut defined by S and $\bar{S}$.*

The proof of the proposition is straightforward and is therefore omitted. In mathematical terms, the above condition for node $j \in \bar{S}$ can be stated as $\sum_{k=\pi'_i(j)}^{n} y_{ik} \geq 1$. This yields our first cut formulation for Problem 1:

$$[\mathbf{C1}] \quad \min \sum_{i \in V_G} \sum_{k=1}^{n} c_{ik} y_{ik}$$

$$\text{s. t.} \quad \sum_{i \in S} \sum_{j \in \bar{S}} \sum_{k=\pi'_i(j)}^{n} y_{ik} \geq 1 \quad \forall S \subset V_G : i_0 \in S, \bar{S} \cap D \neq \emptyset, \tag{9}$$

$$y \in \{0,1\}^{A_G}.$$

In **C1**, the same variable y_{ij} often appears multiple times (i.e., having a coefficient greater than one) in a constraint of (9). This set of constraints is immediately strengthened by removing multiple occurrences of y-variables in the same constraint. This leads to the second cut formulation. Following the notation defined in Section 2, for the sorted power vector of node i, $\pi'_i(\bar{S})$ is the position of the minimum power required for i to reach at least one node in $\bar{S}$. Our second cut formulation is thus:

$$[\mathbf{C2}] \quad \min \sum_{i \in V_G} \sum_{k=1}^{n} c_{ik} y_{ik}$$

$$\text{s. t.} \quad \sum_{i \in S} \sum_{k=\pi'_i(\bar{S})}^{n} y_{ik} \geq 1 \quad \forall S \subset V_G : i_0 \in S, \bar{S} \cap D \neq \emptyset, \tag{10}$$

$$y \in \{0,1\}^{A_G}.$$

For the two cut formulations, $\zeta(\mathbf{C1}) \leq \zeta(\mathbf{C2})$. Also, note that the inequalities (8) are valid in both **C1** and **C2**. We make observations similar to those for the two multicommodity flow formulations regarding (8): These inequalities are not necessary to define the integer optimum, nor do they improve the continuous relaxations of **C1** or **C2**.

PROPOSITION 3. $\mathbf{C1^{LP}}$ *and* $\mathbf{C2^{LP}}$ *have an optimal solution y satisfying (8).*

Proof. The proof is similar to that of Proposition 1. Given an optimal solution $\bar{y}$ to $\mathbf{C1^{LP}}$ ($\mathbf{C2^{LP}}$), we obtain y feasible in $\mathbf{C1^{LP}}$ ($\mathbf{C2^{LP}}$) by setting $y_{ik} = \max\{0, \bar{y}_{ik} - \max\{0, \sum_{l=k}^{n} \bar{y}_{ik} - 1\}\}, i \in V_G, k \in N$. Solution y satisfies $\sum_{k=1}^{n} y_{ik} \leq 1$ for all $i \in V_G$ and $\sum_{a \in A_G} c_a y_a \leq \sum_{a \in A_G} c_a \bar{y}_a$. □

The two cut formulations, **C1** and **C2**, are tightly connected to the two flow formulations **F1** and **F2**. We end this section by proving that the continuous relaxations of **C1** and **F1** have the same strength.

PROPOSITION 4. $\zeta(\mathbf{F1}) = \zeta(\mathbf{C1})$.

Proof. Consider an arbitrary $y \in [0,1]^{A_G}$, and assume that arc $(i,j) \in A_G$ is assigned the capacity $\sum_{k=\pi'_i(j)}^{n} y_{ik}$. Thus y satisfies (9) if and only if all cuts separating i_0 from some destination d have at least unit capacity, which in its turn is true if and only if the maximum flow from i_0 to all $d \in D$ is at least one. The latter condition is equivalent to the constraint sets (4) and (5), and the result follows by observing that **F1** and **C1** have identical objective functions. □

Having observed $\zeta(\mathbf{F1}) = \zeta(\mathbf{C1})$, a natural question to ask is whether such an equivalence exists for the two strengthened formulations **F2** and **C2**. This is indeed the case. We will formalize and prove this result in our discussion of Steiner arborescence models in Section 6.

5 Formulations using Incremental Power

The flow and cut formulations presented in previous sections can be alternatively stated using incremental node power. We introduce a new set of binary variables:

$$x_{ik} = \begin{cases} 1, & \text{the power of node } i \text{ is at least } c_{ik}, \\ 0, & \text{otherwise.} \end{cases}$$

The above definition can also be given in terms of the design variables used in formulations **F1** and **F2**. For a design vector y satisfying (8), we can derive the corresponding values of the x-variables as

$$x_{ik} = \sum_{\ell=k}^{n} y_{i\ell} \quad i \in V_G, k \in N. \tag{11}$$

Expressing the total power in the x-variables, a flow formulation using incremental power for Problem 1 is as follows.

$$\begin{aligned} [\mathbf{\Delta F1}] \quad & \min \sum_{i \in V_G} \sum_{k=1}^{n} \Delta c_{ik} x_{ik} \\ \text{s. t.} \quad & x_{i,k-1} \geq x_{ik} \quad \forall i \in V_G, k = 2, \ldots, n, \qquad (12) \\ & f^d \in \mathcal{F}(G, b^d, x) \quad \forall d \in D, \\ & x \in \{0,1\}^{A_G}. \end{aligned}$$

The total flow of any commodity d leaving node i is at most one. Moreover, x_{ik} must be one if any flow of d is sent on any arc $(i, \pi_i(\ell))$ where $\ell \geq k$. We thus obtain the following strengthened version of $\mathbf{\Delta F1}$:

$$\begin{aligned} [\mathbf{\Delta F2}] \quad & \min \sum_{i \in V_G} \sum_{k=1}^{n} \Delta c_{ik} x_{ik} \\ \text{s. t.} \quad & \sum_{\ell=k}^{n} f_{i\ell}^d \leq x_{ik} \quad \forall i \in V_G, k \in N, d \in D, \qquad (13) \\ & f^d \in \mathcal{F}(G, b^d) \quad \forall d \in D, \qquad (14) \\ & x \in \{0,1\}^{A_G}. \end{aligned}$$

Constraints (12) are probably worth a remark. These constraints are not present in $\mathbf{\Delta F2}$, as they are implied by (13) and (14), a result that holds also for the continuous relaxation of $\mathbf{\Delta F2}$.

PROPOSITION 5. $\mathbf{\Delta F2^{LP}}$ *has an optimal solution* (f, x) *satisfying (12).*

Proof. Let $(\bar{f}, \bar{x})$ be an optimal solution of $\mathbf{\Delta F2^{LP}}$. Assume that $\bar{x}_{i,k-1} < \bar{x}_{ik}$ for some i and k. Using (13), we have $\sum_{l=k}^{n} \bar{f}_{il}^d + \bar{f}_{i,k-1}^d = \sum_{l=k-1}^{n} \bar{f}_{il}^d \leq \bar{x}_{i,k-1} < \bar{x}_{ik}, \forall d \in D$. Thus reducing the value of x_{ik} to $\bar{x}_{i,k-1}$ still satisfies (13). Doing so repeatedly, if necessary, we obtain a feasible solution to $\mathbf{\Delta F2^{LP}}$ with the same or better objective function value than that of $(\bar{f}, \bar{x})$. Hence the conclusion. □

The following proposition states that $\mathbf{\Delta F1}$ and $\mathbf{\Delta F2}$ are essentially reformulations of **F1** and **F2**, respectively.

PROPOSITION 6. $\zeta(\mathbf{F1}) = \zeta(\mathbf{\Delta F1})$, *and* $\zeta(\mathbf{F2}) = \zeta(\mathbf{\Delta F2})$,

Proof. To show $\zeta(\mathbf{F2}) = \zeta(\mathbf{\Delta F2})$, consider an (f, y) that is optimal in $\mathbf{F2^{LP}}$ and satisfies (8) (As shown in Proposition 1, such a solution always exists.) We obtain (f, x) feasible in $\mathbf{\Delta F2^{LP}}$ by choosing x to satisfy (11). Conversely, if (f, x) is optimal in $\mathbf{\Delta F2^{LP}}$, we can by Proposition 5 also assume that it satisfies (12). We solve the equation system defined by (11) to obtain a feasible solution (f, y) of $\mathbf{F2^{LP}}$. Note that the equation system always has, due to its structure, a unique solution, which by (12) is non-negative. In addition, (11) implies that the objective function values in the two formulations are equal, therefore $\zeta(\mathbf{F2}) = \zeta(\mathbf{\Delta F2})$. A similar, but slightly simpler argument can be used to prove $\zeta(\mathbf{F1}) = \zeta(\mathbf{\Delta F1})$. □

We can use the x-variables to state constraints (9) in **C1** and (10) in **C2**. This gives us two cut formulations using incremental power. We denote these two formulations by $\mathbf{\Delta C1}$ and $\mathbf{\Delta C2}$, respectively.

$$
\begin{aligned}
[\mathbf{\Delta C1}] \quad & \min \sum_{i \in V_G} \sum_{k=1}^{n} \Delta c_{ik} x_{ik} \\
\text{s. t.} \quad & x_{i,k-1} \geq x_{ik} \quad \forall i \in V_G, k = 2, \ldots, n, \\
& \sum_{i \in S} \sum_{j \in \bar{S}} x_{i\pi'_i(j)} \geq 1 \quad \forall S \subseteq V_G : i_0 \in S, \bar{S} \cap D \neq \emptyset, \qquad (15) \\
& x \in \{0,1\}^{A_G}.
\end{aligned}
$$

$$
\begin{aligned}
[\mathbf{\Delta C2}] \quad & \min \sum_{i \in V_G} \sum_{k=1}^{n} \Delta c_{ik} x_{ik} \\
\text{s. t.} \quad & x_{i,k-1} \geq x_{ik} \quad \forall i \in V_G, k = 2, \ldots, n, \qquad (16) \\
& \sum_{i \in S} x_{i,\pi'_i(\bar{S})} \geq 1 \quad \forall S \subseteq V_G : i_0 \in S, \bar{S} \cap D \neq \emptyset \\
& x \in \{0,1\}^{A_G}.
\end{aligned}
$$

The strengths of the continuous relaxations of **C1** and **C2** are the same as their respective counterparts based on incremental power. We formalize this result as a proposition.

PROPOSITION 7. $\zeta(\mathbf{C1}) = \zeta(\mathbf{\Delta C1})$, *and* $\zeta(\mathbf{C2}) = \zeta(\mathbf{\Delta C2})$,

Proof. Among the solutions satisying (8), let y denote an optimal solution in $\mathbf{C1^{LP}}$ (see Proposition 3). We use (11) to obtain a solution in the x-variables. It is easy to verify that this solution satisfies constraints (15). Conversely, for any x feasible in $\mathbf{\Delta C1^{LP}}$, (11) gives a unique solution in y, where $y_{in} = x_{in}, \forall i$, and $y_{ik} = x_{ik} - x_{i,k+1}, \forall i, k = 1, \ldots, n-1$. This y-solution clearly satisfies (9). Moreover, each of the two solution pairs gives the same objective function value. Therefore $\zeta(\mathbf{C1}) = \zeta(\mathbf{\Delta C1})$. The second part of the proposition can be proved analogously. □

Summarizing the results for the formulations discussed so far, we have $\zeta(\mathbf{F1}) = \zeta(\mathbf{\Delta F1}) = \zeta(\mathbf{C1}) = \zeta(\mathbf{\Delta C1})$. We have also shown $\zeta(\mathbf{F2}) = \zeta(\mathbf{\Delta F2})$, and $\zeta(\mathbf{C2}) = \zeta(\mathbf{\Delta C2})$. In the next section we introduce Steiner arborescence models, which we use to prove $\zeta(\mathbf{F2}) = \zeta(\mathbf{C2})$.

6 Steiner Arborescence Models

The *Steiner arborescence problem* defined on a digraph $\mathcal{G}$ with arc weights $w \in \Re_+^{A_\mathcal{G}}$, terminal nodes $\mathcal{T} \subset V_\mathcal{G}$, and root $r \in V_\mathcal{G} \setminus \mathcal{T}$ [21], is to find an r-arborescence with minimum arc weight that spans $\mathcal{T}$. It can be formulated as

$$\min_{x \in \{0,1\}^{A_\mathcal{G}}} \left\{ \sum_{(i,j) \in A_\mathcal{G}} w_{ij} x_{ij} : \mathcal{F}\left(\mathcal{G}, \delta^t, x\right) \neq \emptyset \quad \forall t \in \mathcal{T} \right\}$$

where for all $t \in \mathcal{T}$ we have $\delta_r^t = -1$, $\delta_t^t = 1$ and $\delta_i^t = 0$ for all $i \in V_\mathcal{G} \setminus \{r, t\}$.

By extensions of G, problems **F2** and **ΔF2** can be transformed to Steiner arborescence problems.

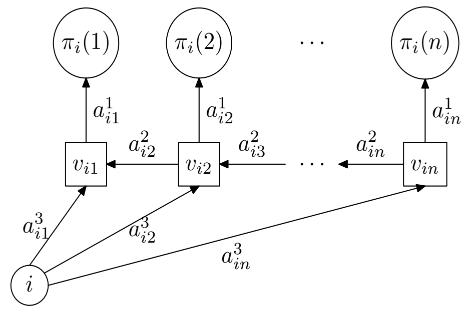

Figure 1: Arcs incident to $v_{i1}, \ldots, v_{in}$ in H

Define the digraph H such that $V_H = V_H' \cup V_G$ and $A_H = A_H^1 \cup A_H^2 \cup A_H^3$. The set of *square nodes* (see Figure 1) is defined as $V_H' = \{v_{ik} : i \in V_G, k \in N\}$, the set of *vertical arcs* is defined as $A_H^1 = \left\{a_{ik}^1 : i \in V_G, k \in N\right\}$, the set of *horizontal arcs* is $A_H^2 = \left\{a_{ik}^2 : i \in V_G, k = 2, \ldots, n\right\}$, and the set of *diagonal arcs* is $A_H^3 = \left\{a_{ik}^3 : i \in V_G, k \in N\right\}$, where $a_{ik}^1 = (v_{ik}, \pi_i(k))$, $a_{ik}^2 = (v_{ik}, v_{i,k-1})$, and $a_{ik}^3 = (i, v_{ik})$. The component of any vector $u \in \Re^{A_H}$ corresponding to a_{ik}^q is denoted u_{ik}^q. Consider the Steiner arborescence problem where $B \in \Re^{V_H}$ is an extension of b^d such that $B_v = 0$ for all $v \in V_H'$.

$$[\mathbf{S}] \quad \min \sum_{i \in V_G} \sum_{k=1}^{n} c_{ik} Y_{ik}^3 \tag{17}$$

$$\text{s. t.} \quad f^d \in \mathcal{F}(H, B^d, Y) \quad \forall d \in D, \tag{18}$$

$$Y \in \{0,1\}^{A_H}. \tag{19}$$

The following result is useful when proving that **S** is exactly as trong as **F2**:

LEMMA 1. *Let* $d \in D$, $f \in \mathcal{F}(G, b^d)$, $Y \in [0,1]^{A_H}$. *If* $Y_a = 1$ $\forall a \in A_H^1 \cup A_H^2$ *and* $\sum_{\ell=k}^n f_{i\ell} \leq \sum_{\ell=k}^n Y_{i\ell}^3 \leq 1$ $\forall (i,k) \in V_G \times N$, *then there exists some* $h \in \mathcal{F}(H, B^d, Y)$ *such that* $h_{ik}^1 = f_{ik}$ $\forall (i,k) \in V_G \times N$.

Proof. Consider an arbitrary $i \in V_G$. We define the flow h by maximizing flow on the arcs $a_{in}^3, \ldots, a_{i1}^3$, in the given order, while respecting the specified conditions. Flow conservation can be satisfied only if the toal flow along these arcs is $\sum_{\ell=1}^n h_{i\ell}^1$, and only if the flow entering $\{v_{ik}, \ldots, v_{in}\}$ equals the leaving flow. Therefore we define:

$$\begin{aligned} h_{ik}^1 &= f_{ik} && \forall i \in V_G, k = 1, \ldots, n \\ h_{ik}^3 &= \min\left\{Y_{ik}^3, \textstyle\sum_{\ell=1}^n h_{i\ell}^1 - \sum_{\ell=k+1}^n h_{i\ell}^3\right\} && \forall i \in V_G, k = 1, \ldots, n \\ h_{ik}^2 &= \textstyle\sum_{\ell=k}^n h_{i\ell}^3 - \sum_{\ell=k}^n h_{i\ell}^1 && \forall i \in V_G, k = 2, \ldots, n \end{aligned} \tag{20}$$

We must prove that this implies $h \in \mathcal{F}(H, B^d, Y)$.

Clearly, $h_a \in [0, Y_a]\ \forall a \in A^1_H$. By (20) we have $h^3_{ik} \le Y^3_{ik}$ and $\sum_{\ell=1}^n h^1_{i\ell} \ge \sum_{\ell=k}^n h^3_{i\ell}\ \forall k \in N$, and hence also $h^3_{ik} \ge 0$ since it is the minimum of two non-negative numbers. Furthermore, $\sum_{\ell=k}^n h^3_{i\ell} = \sum_{\ell=1}^n h^1_{i\ell}$ if the minimum in (20) for some $k' = k, \ldots, n$ is attained at $\sum_{\ell=1}^n h^1_{i\ell} - \sum_{\ell=k'+1}^n h^3_{i\ell}$, and $\sum_{\ell=k}^n h^3_{i\ell} = \sum_{\ell=k}^n Y^3_{i\ell}$ otherwise. Therefore, either $h^2_{ik} = \sum_{\ell=1}^n h^1_{i\ell} - \sum_{\ell=k}^n h^1_{i\ell} = \sum_{\ell=1}^{k-1} f_{i\ell} \in [0,1]$ or $h^2_{ik} = \sum_{\ell=k}^n Y^3_{i\ell} - \sum_{\ell=k}^n h^1_{i\ell} = \sum_{\ell=k}^n Y^3_{i\ell} - \sum_{\ell=k}^n f_{i\ell} \in [0,1]$.

To prove conservation of flow, consider first the total flow leaving node i. Since $\sum_{\ell=1}^n h^3_{il} \in \left\{\sum_{\ell=1}^n h^1_{i\ell}, \sum_{\ell=1}^n Y^3_{i\ell}\right\}$ and $\sum_{\ell=1}^n h^1_{i\ell} = \sum_{\ell=1}^n f_{i\ell} \le \sum_{\ell=1}^n Y^3_{i\ell}$, it follows from the minimum operation in (20) that the total flow leaving node i is $\sum_{\ell=1}^n h^3_{il} = \sum_{\ell=1}^n f_{i\ell}$. The total flow entering i is $\sum_{j \in V_G \setminus \{i\}} h^1_{j\pi'_j(i)} = \sum_{j \in V_G \setminus \{i\}} f_{j\pi'_j(i)}$. Thus h satisfies conservation of flow at i since f does. For $v_{ik} \in V'_H$, the flow balance is $h^2_{ik} + h^1_{ik} - h^3_{ik} - h^2_{i,k+1} = 0$ (let $h^2_{i,n+1} = 0$). Hence $h \in \mathcal{F}\left(H, B^d, Y\right)$. □

PROPOSITION 8. *For any $d \in D$, $y \in [0,1]^{A_G}$ and $Y \in [0,1]^{A_H}$ satisfying $\sum_{\ell=1}^n y_{i\ell} \le 1\ \forall i \in V_G$, $Y_a = 1\ \forall a \in A^1_H \cup A^2_H$ and $Y^3_{ik} = y_{ik}\ \forall (i,k) \in V_G \times N$, the projection of $\mathcal{F}(H, B^d, Y)$ onto A^1_H is exactly the set of flow vectors in $\mathcal{F}(G, b^d)$ satisfying $\sum_{\ell=k}^n f^d_{i\ell} \le \sum_{\ell=k}^n y_{i\ell}\ \forall i \in V_G, k \in N$.*

Proof.

Assume $h \in \mathcal{F}(H, B^d, Y)$. We show that putting $f_{ik} = h^1_{ik}\ \forall i \in V_G,\ k \in N$, yields $f \in \mathcal{F}(G, b^d)$ and $\sum_{\ell=k}^n f_{i\ell} \le \sum_{\ell=1}^n y_{i\ell}$.

Since all $v \in V'_H$ have zero external flow, the total flow entering $\{v_{ik}\}_{k \in N}$ must equal the total flow leaving this node set. Hence $\sum_{k \in N} h^3_{ik} = \sum_{k \in N} h^1_{ik}$. Since h satisfies conservation of flow at i in H, we have $\sum_{a \in A^-_G(i)} f_a = \sum_{a \in A^-_H(i)} h_a = \sum_{k \in N} h^3_{ik} + b^d_i = \sum_{k \in N} h^1_{ik} + b^d_i = \sum_{k \in N} f_{ik} + b^d_i = \sum_{a \in A^+_G(i)} f_a + b^d_i$. Thus $f \in \mathcal{F}(G, b^d)$.

Also, conservation of flow at $\{v_{i\ell}\}_{\ell=k}^n$ implies $\sum_{\ell=k}^n f_{i\ell} = \sum_{\ell=k}^n h^1_{i\ell} = \sum_{\ell=k}^n h^3_{i\ell} - h^2_{ik} \le \sum_{\ell=k}^n h^3_{i\ell} \le \sum_{\ell=k}^n Y^3_{i\ell} = \sum_{\ell=k}^n y_{i\ell}$.

Next assume that $f \in \mathcal{F}(G, b^d)$ and $\sum_{\ell=k}^n f_{i\ell} \le \sum_{\ell=k}^n y_{i\ell}\ \forall k \in N$. Then f and Y satisfy the sufficient conditions of Lemma 1, and hence there exists some $h \in \mathcal{F}(H, B^d, Y)$ of which f is the projection on $\mathcal{F}(G, b^d)$. □

COROLLARY 1. **S** *is a valid formulation for Problem 1, and* $\zeta(\mathbf{S}) = \zeta(\mathbf{F2})$.

Proof. Follows directly from Propositions 1 and 8. □

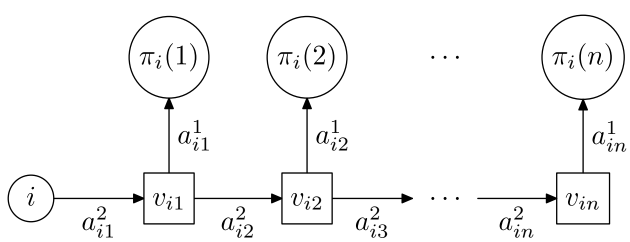

Figure 2: Arcs incident to $v_{i1}, \ldots, v_{in}$ in ΔH

Starting from $\mathbf{\Delta F2}$, we arrive at a different digraph, ΔH, as the basis of a Steiner arborescence formulation. Let $V_{\Delta H} = V_H$ and $A^1_{\Delta H} = A^1_H$, but let the horizontal arc set (see Figure 2) be defined as $A^2_{\Delta H} = \left\{a^2_{ik} = (v_{i,k-1}, v_{ik}) : i \in V_G, k \in N\right\}$, where v_{i0} signifies i. To complete the definition of ΔH, let $A_{\Delta H} = A^1_{\Delta H} \cup A^2_{\Delta H}$. he component of any vector $u \in \Re^{A_{\Delta H}}$ corresponding to a^q_{ik} is denoted u^q_{ik}. Define [$\mathbf{\Delta S}$] as the Steiner arborescence problem

$$[\mathbf{\Delta S}] \quad \min \sum_{i \in V_G} \sum_{k=1}^{n} \Delta c_{ik} X_{ik}^2$$
$$\text{s. t.} \quad f^d \in \mathcal{F}(\Delta H, B^d, X) \quad \forall d \in D,$$
$$X \in \{0,1\}^{A_{\Delta H}}.$$

PROPOSITION 9. $\mathbf{\Delta S}$ *is a valid formulation for Problem 1, and* $\zeta(\mathbf{\Delta S}) = \zeta(\mathbf{\Delta C2})$.

Proof. To prove $\zeta(\mathbf{\Delta S}) \geq \zeta(\mathbf{\Delta C2})$, assume (X, f) is optimal in $\mathbf{\Delta S}$. Consider any subset $S \subset V_G$ where $i_0 \in S$ and $D \cap \bar{S} \neq \emptyset$. If $d \in D \cap \bar{S}$, then $\left\{a_{i\pi_i'(\bar{S})}^2\right\}_{i \in S}$ is a cut separating i_0 and d. Hence $\sum_{i \in S} X_{i\pi_i'(\bar{S})}^2 \geq 1$. Since $f_{i,k-1}^2 \geq f_{ik}^2$, it can also be assumed that $X_{i,k-1}^2 \geq X_{ik}^2$. Putting $x_{ik} = X_{ik}^2$ for all $i \in V_G$ and all $k \in N$ thus makes x feasible in $\mathbf{\Delta C2}$, and the result follows since the objective function values coincide.

To prove $\zeta(\mathbf{\Delta S}) \leq \zeta(\mathbf{\Delta C2})$, assume x is feasible in $\mathbf{\Delta C2^{LP}}$. Consider any $S \subset V_G$ where $i_0 \in S$ and $D \cap \bar{S} \neq \emptyset$, and note that there is a set of $|S|$ paths $\{P_i\}_{i \in S}$ in ΔH from S to $\bar{S}$ where $A_{P_i} = \left\{a_{i1}^2, a_{i2}^2, \ldots, a_{i\pi_i'(\bar{S})}^2, a_{i\pi_i'(\bar{S})}^1\right\}$. Assume that a_{ik}^1 and a_{ik}^2 are assigned capacities $X_{ik}^1 = 1$ and $X_{ik}^2 = x_{ik}$, respectively. It follows from (16) that the flow capacity of P_i is $x_{i\pi_i'(\bar{S})}$. Since the arc sets of the paths are disjoint, any cut separating S and $\bar{S}$ has capacity no smaller than $\sum_{i \in S} x_{i\pi_i'(\bar{S})} \geq 1$. Hence there is some $f^d \in \mathcal{F}(\Delta H, B^d, X)$, and the result follows. □

PROPOSITION 10. *For any feasible design* Y *in* $\mathbf{S^{LP}}$ *satisfying* $\sum_{\ell=1}^{n} Y_{i\ell}^3 \leq 1$ $\forall i \in V_G$ *there exists a feasible design* X *in* $\mathbf{\Delta S^{LP}}$ *with identical objective function value, and for any feasible design* X *in* $\mathbf{\Delta S^{LP}}$ *satisfying* $X_{i,k+1}^2 \leq X_{ik}^2$ $\forall i \in V_G, k = 1, \ldots, n-1$ *there exists a feasible design* Y *in* $\mathbf{S^{LP}}$ *with identical objective function value.*

Proof. Let Y be a feasible design in $\mathbf{S^{LP}}$ satisfying $\sum_{\ell=1}^{n} Y_{i\ell}^3 \leq 1$. Similarly to (11), let $X_{ik}^2 = \sum_{\ell=k}^{n} Y_{i\ell}^3$, implying equal objective function values. Furthermore, let $X_{ik}^1 = 1$ $\forall (i,k) \in V_G \times N$.

For all $d \in D$ we have to show that $\mathcal{F}\left(\Delta H, B^d, X\right)$ is nonempty. Choose $d \in D$ and $f \in \mathcal{F}\left(H, B^d, Y\right)$ arbitrarily. A corresponding flow $g \in \Re^{A_{\Delta H}}$ is for all $(i,k) \in V_G \times N$ given through $g_{ik}^1 = f_{ik}^1 \geq 0$ and $g_{ik}^2 = \sum_{\ell=k}^{n} f_{i\ell}^1 \geq 0$.

Consider an arbitrary $(i,k) \in V_G \times N$. As already observed in the proof of Proposition 8, all nodes in V_H' have zero external flow, and thus $\sum_{\ell=k}^{n} f_{i\ell}^3 = f_{ik}^2 + \sum_{\ell=k}^{n} f_{i\ell}^1$, where $f_{i1}^2 = 0$. Therefore $g_{ik}^2 = \sum_{\ell=k}^{n} f_{i\ell}^3 - f_{ik}^2 \leq \sum_{\ell=k}^{n} f_{i\ell}^3 \leq \sum_{\ell=k}^{n} Y_{i\ell}^3 = X_{ik}^2 \leq 1$, and $g_{ik}^1 \leq 1 = X_{ik}^1$. The flow balance defined by g at i is $g_{i1}^2 - \sum_{j \in V_G \setminus \{i\}} g_{j\pi_j'(i)}^1 = \sum_{\ell=1}^{n} f_{i\ell}^3 - \sum_{j \in V_G \setminus \{i\}} f_{j\pi_j'(i)}^1$, which is the flow balance defined by f at i. Hence g staisfies flow conservation at i since f does. The flow conservation constraint at v_{ik} is satisfied since $g_{ik}^2 - g_{i,k+1}^2 - g_{ik}^1 = f_{ik}^1 - f_{ik}^1 = 0$ (let $f_{i,n+1}^2 = 0$). Hence $g \in \mathcal{F}\left(\Delta H, B^d, X\right)$.

Conversely, let X be a feasible design in $\mathbf{\Delta S^{LP}}$ satisfying $X_{i,k+1}^2 \leq X_{ik}^2$ $\forall i \in V_G, k = 1, \ldots, n-1$. Choose $Y \in [0,1]^{A_H}$ such that $Y_a = 1$ $\forall a \in A_H^1 \cup A_H^2$ and such that its remaining components are given by solving $X_{ik}^2 = \sum_{\ell=k}^{n} Y_{i\ell}^3$ $\forall (i,k) \in V_G \times N$. This results again in equal objective function values, and also note that $\sum_{\ell=1}^{n} Y_{i\ell}^3 \leq 1$.

Choose $d \in D$ and $g \in \mathcal{F}\left(\Delta H, B^d, X\right)$ arbitrarily, and define $f \in \Re^{A_G}$ such that $f_{ik} = g_{ik}^1$ $\forall (i,k) \in V_G \times N$. Thus $\sum_{\ell=k}^{n} f_{i\ell} \leq g_{ik}^2 \leq X_{ik}^2 = \sum_{\ell=k}^{n} Y_{i\ell}^3 \leq 1$ $\forall (i,k) \in V_G \times N$. It is also easily seen that $f \in \mathcal{F}\left(G, b^d\right)$, and by Lemma 1 there hence exists some $h \in \mathcal{F}\left(H, B^d, Y\right)$. □

COROLLARY 2. $\zeta(\mathbf{S}) = \zeta(\mathbf{\Delta S})$.

Proof. Follows directly from Propositions 1, 5 and 10. □

From the results in the previous and the current section we conclude that $\zeta(\mathbf{F2}) = \zeta(\mathbf{C2}) = \zeta(\mathbf{S}) = \zeta(\mathbf{\Delta S}) = \zeta(\mathbf{\Delta C2}) = \zeta(\mathbf{\Delta F2})$. Hence the strength of the mixed integer programming formulations studied in this paper define an equivalence relation with three equivalence classes. The single commodity flow formulation suggested in [6] is the sole member of the first class, the second class consists of all formulations referred to as 'weak' in the current paper (**F1**, **C1**, **ΔF1**, **ΔC1**), while the third class consists of the 'strong' ones (**F2**, **C2**, **S**, **ΔF2**, **ΔC2**, **ΔS**).

In the next section we study experimentally the strength difference between these classes, and we also compare the computational capabilities of formulations belonging to the same equivalence class.

7 Numerical experiments

In this section we compare the computational performance of the different formulations. We use networks of 10, 20, 50, and 100 nodes. The number of destination nodes varies from as few as 2 (a small multicast group) to n (broadcast). Every combination of $|V_G|$ and $|D|$ defines an instance set, and within each such set 100 networks are generated using the instance generation procedure by Wieselthier et al. [19, 20]. The parameter α is set to 2. Those instances are formulated with our models and then handed over to an integer programming solver [11]. The program is run on a 2.4 GHz-Processor with 2 GB RAM.

To be of practical interest, an optimal solution should be found in a reasonable amount of time. Therefore, the time given to the solver for obtaining the optimal solution of the linear relaxation of one problem is limited to one hour. If this is accomplished, an additional hour is given to find the optimal integer solution.

First, the flow formulations of Section 3 are compared. For each such formulation $\mathcal{I}$, Table 1 shows the value of the integrality gap $\zeta^* - \zeta(\mathcal{I})$, where ζ^* is the optimal total power consumption. Each gap is computed as a percentage of ζ^*, and the numbers reported in the table are obtained by averaging this over all instances in each set.

Not only do the experiments confirm that **F2** and **ΔF2** are stronger formulations than **F1** and **ΔF1**, which in their turn are stronger than **F0**. The average integrality gap is reduced to less than one tenth when moving from **F1**/**ΔF1** to **F2**/**ΔF2**, and the latter formulations improve the **F0**-bound by a factor as large as 960. The benefit of choosing the stronger formulations is largest when the destinations are few. Unlike the other formulations, **F1** and **ΔF1** seem to have their lower bounds improved when the number of destinations increases.

Table 1: Average integrality gap given by LP-relaxations

$\lvert V_G\rvert$	$\lvert D\rvert$	average gap in %		
		Das et al.	**F1** / **ΔF1**	**F2** / **ΔF2**
10	2	261.7	39.1	0.1
	5	565.9	25.4	0.1
	9	864.6	20.1	0.2
20	5	852.7	50.9	0.4
	10	1335.4	37.3	1.2
	19	1920.1	23.8	2.0

In particular, the relaxations of **F2** and **ΔF2** are strong enough to provide integer solutions to many instances. Table 2 gives the percentage of each group of instances for which $\zeta(\mathbf{F2}) = \zeta(\mathbf{\Delta F2}) = \zeta^*$. The corresponding percentages of the other formulations are not given, since they all turned out to be 0.

Table 2: Proportion (%) of cases where $\zeta(\mathbf{F2}) = \zeta(\mathbf{\Delta F2}) = \zeta^*$

$\|V_G\|$	10			20		
$\|D\|$	2	5	9	5	10	19
Proportion	98	95	89	87	75	51

Further experiments are carried out only with the strongest formulations **F2**, **ΔF2**, **S** and **ΔS**. For these formulations we compare the execution time required to compute the optimum, and the number of cases for which this is feasible within the given time limit. Formulations **C2** and **ΔC2** need $2^{|V_G|} - 2^{|V_G|-|D|}$ constraints. Due to this exponential model size, these formulations are excluded from the experiments.

The solver uses branch-and-bound to find an optimal solution. In every step of this algorithm, a linear relaxation of the problem is solved. Formulations **F2** and **ΔF2** use $|D|(|V_G|+1)^2$ constraints, whereas **S** and **ΔS** use $|D|(4|V_G|^2 + 4|V_G| - 1)$ constraints. Following a hypothesis that the number of pivots required to solve a single LP-problem increases with the number of constraints, it is therefore reasonable to believe that **S** and **ΔS** are slower than **F2** and **ΔF2**.

In the solution to the relaxation, a binary variable with fractional value is picked, and two subproblems where this variable is fixed to 0 and 1, respectively, are generated. Fixing y_{ik} to 1 in **F2** determines $y_{i\ell} = 0 \,\forall\, \ell \neq k$ because of constraints (8), thus reducing the number of unfixed variables by $|V_G|$. On the other hand, fixing y_{ik} to 0 gives no clue about the $y_{i\ell}$-variables with $\ell \neq k$, hence reducing the number of unfixed variables only by 1. Contrarily, fixing x_{ik} to 1 in **ΔF2** determines $x_{i\ell} = 1 \,\forall\, \ell < k$, thus reducing the number of unfixed variables by k, whereas fixing $x_{ik} = 0$ determines $x_{i\ell} = 0 \,\forall\, \ell > k$, hence reducing the number of unfixed variables by $|V_G| - k$. The same argumentation holds for **S** and **ΔS**. Therefore the **Δ**-formulations are expected to lead to a better balanced branch-and-bound tree and hence to find an optimal solution faster.

Table 3 lists the performance of every formulation for all combinations of numbers of nodes and destinations. To describe the performance, we first state how many instances could be solved to optimality (column solved). For these instances, the distribution of the running time (CPU-seconds) is given in terms of the mean, minimum, maximum, median and standard deviation. These values were calculated taking into account only those instances which were solved to optimality by all models, because this gives comparable values. This includes all instances with 10 and 20 nodes as well as all instances with 50 nodes and 5 destinations. For $|V_G| = 50$, the number of instances solved by all formulations was 95 and 23 for $|D| = 10$ and $|D| = 25$, respectively.

For 50 nodes and 49 destinations, formulation **ΔS** could not solve any of the instances to optimality. Formulations **F2**, **ΔF2** and **S** solved 7, 2 and 3 instances, respectively. One instance was solved by all three formulations, and the solution time was 717, 4003 and 2530 seconds, respectively. The relaxed solutions to both instances solved to optimality by formulation **ΔF2** had an objective function value equal to ζ^*, and the same is true for 4 of the 7 instances solved by **F2**.

For $|V_G| = 100$, 86 instances were solved by all formulations in the case of $|D| = 5$. For $|D| = 10$, formulation **S** could not solve any of the instances to optimality. Formulations **F2**, **ΔF2** and **S** solved 54, 61 and 2 instances respectively. Again, the solutions to the linear relaxation of these two instances were ζ^* all formulations. In the case of 50 destinations, none of the formulations could solve any of the instances.

Table 3: Results of the experiments

$\|V_G\|$	$\|D\|$	formulation	solved	average	min	max	median	σ
10	2	**F2**	100	0.03	0.0	1.0	0.0	0.24
		ΔF2	100	0.04	0.0	1.0	0.0	0.28
		S	100	0.05	0.0	1.0	0.0	0.31
		ΔS	100	0.02	0.0	1.0	0.0	0.20
	5	**F2**	100	0.05	0.0	1.0	0.0	0.31
		ΔF2	100	0.06	0.0	1.0	0.0	0.34
		S	100	0.09	0.0	1.0	0.0	0.41
		ΔS	100	0.07	0.0	1.0	0.0	0.37
	9	**F2**	100	0.08	0.0	1.0	0.0	0.39
		ΔF2	100	0.05	0.0	1.0	0.0	0.31
		S	100	0.16	0.0	1.0	0.0	0.54
		ΔS	100	0.12	0.0	1.0	0.0	0.47
20	5	**F2**	100	0.27	0.0	1.0	0.0	0.68
		ΔF2	100	0.25	0.0	3.0	0.0	0.71
		S	100	0.93	0.0	3.0	1.0	1.07
		ΔS	100	0.53	0.0	1.0	1.0	0.88
	10	**F2**	100	0.95	0.0	4.0	1.0	1.28
		ΔF2	100	0.88	0.0	6.0	1.0	1.46
		S	100	3.83	2.0	21.0	3.0	3.35
		ΔS	100	1.75	0.0	8.0	1.0	1.94
	19	**F2**	100	3.16	1.0	17.0	2.0	3.68
		ΔF2	100	3.78	0.0	24.0	2.0	4.76
		S	100	15.82	4.0	78.0	11.0	14.28
		ΔS	100	7.38	2.0	23.0	5.0	6.10
50	5	**F2**	100	11.85	4.0	227.0	6.0	23.53
		ΔF2	100	10.28	2.0	108.0	5.0	15.81
		S	100	136.24	49.0	966.0	116.0	108.11
		ΔS	100	35.55	12.0	341.0	25.0	37.23
	10	**F2**	100	82.27	12.0	927.0	43.0	135.37
		ΔF2	100	70.98	8.0	486.0	49.0	75.12
		S	95	667.65	181.0	3041.0	485.0	492.44
		ΔS	99	287.45	45.0	2955.0	212.0	330.73
	25	**F2**	75	337.17	96.0	712.0	315.0	186.09
		ΔF2	70	827.44	114.0	2182.0	665.0	596.73
		S	28	3035.04	1305.0	4737.0	2933.0	959.13
		ΔS	36	3226.00	819.0	5856.0	3011.0	1304.72
	49	**F2**	7	717.0				
		ΔF2	2	4003.0				
		S	3	2530.0				
100	5	**F2**	98	135.09	40.0	602.0	97.0	97.68
		ΔF2	100	129.12	19.0	525.0	117.0	89.98
		S	87	3715.53	665.0	5884.0	3793.0	1014.56
		ΔS	94	812.86	171.0	1865.0	698.0	415.23
100	10	**F2**	54	194.00	188.0	200.0	188.0	15.17
		ΔF2	61	188.00	164.0	212.0	164.0	27.64
		ΔS	2	2108.50	1423.0	2794.0	1423.0	687.04

As starting from 50 nodes and 25 destinations, the number of instances solved by all formulations gets rather small, Table 4 shows the statistical values for every formulation calculated from all instances it could solve.

Table 4: Results of the experiments, taking into account all solved instances

$\|V_G\|$	$\|D\|$	formulation	solved	average	min	max	median	σ
50	25	**F2**	75	1040.89	96.0	3591.0	812.0	878.84
		ΔF2	70	1462.60	114.0	3597.0	1183.0	898.05
		S	28	3227.71	1305.0	4737.0	2933.0	1009.33
		ΔS	36	3501.42	819.0	5856.0	3612.0	1225.76
	49	**F2**	7	1911.00	717.0	3315.0	2258.0	776.22
		ΔF2	2	4749.00	4003.0	5495.0	4003.0	749.18
		S	3	4888.67	2530.0	6139.0	6139.0	1670.30
100	5	**F2**	98	200.85	40.0	2233.0	107.0	277.79
		ΔF2	100	228.39	19.0	2159.0	135.0	349.88
		S	87	3734.43	665.0	5884.0	3827.0	1023.84
		ΔS	94	908.77	171.0	3564.0	820.0	542.583
100	10	**F2**	54	1366.31	188.0	3761.0	1198.0	846.574
		ΔF2	61	1300.16	164.0	3497.0	1079.0	746.066
		ΔS	2	2108.50	1423.0	2794.0	1423.0	687.036

The hypothesis that the **Δ**-formulations are faster is confirmed for the **S**-formulations. With $|V_G| \leq 20$, the average running time of **ΔS** is between 40.0% ($|V_G| = 10$, $|D| = 2$) and 77.8% ($|V_G| = 10$, $|D| = 9$) of the average running time of **S**. When $|V_G| \geq 50$, the corresponding range is from 21.9% ($|V_G| = 100$, $|D| = 25$) to 106.3% ($|V_G| = 50$, $|D| = 25$), where the latter set of instances is the only one for which **S** is faster than **ΔS**.

Comparing **F2** and **ΔF2** is more difficult. For seven data sets, **ΔF2** is faster than **F2**, and needs between 62.5% ($|V_G| = 10$, $|D| = 9$) and 97.1% ($|V_G| = 100$, $|D| = 10$) of the running time required by **F2**. For the remaining data sets, running **ΔF2** takes between 119.6% ($|V_G| = 20$, $|D| = 19$) and as much as 558.3% ($|V_G| = 50$, $|D| = 49$, only two instances solved) of the time it takes to run **F2**.

In the case of 50 nodes, **F2** solved 5 instances more than did **ΔF2** for 25 and 49 destinations, whereas for 100 nodes, **ΔF2** solved 2 instances more than did **F2** for 5 destinations, and 7 more for 10 destinations.

The experiments do not support the hypothesis that **ΔF2** gives a more efficient search than does **F2**. However, a large proportion of the instances were solved by the continuous relaxations, and hence the properties of the search algorithm do not affect the running time.

Table 3 shows that formulations **F2** and **ΔF2** are consistently faster than **S** and **ΔS**, and that the difference grows with problem size. The only exception to this is the set with the smallest instances ($|V_G| = 10$, $|D| = 2$). For $|V_G| = 20$, choosing the best Steiner formulation implies at least twice the running time of the best flow formulation, and the factor increases to more than 3 for $|V_G| = 50$, and more than 10 for $|V_G| = 100$.

The overall conclusion from the experiment is that the strength of the relaxation considerably affects the computational effort. Furthermore, the size, in terms of number of constraints, of the Steiner formulations makes them less efficient than the flow models, and this effect grows with the problem size. To decide whether or not an incremental formulation should be chosen, the experiments presented in this work are rather inconclusive. They indicate that the incremental flow formulation is preferable for destination-sparse cases ($|D|/|V_G|$ is small), whereas the non-incremental flow formulation is superior for more dense cases, especially for large networks. This conjecture is however difficult to test, since the effect is first observed for some of the largest instances we can solve.

The flow models **F2** and **ΔF2** seem to be superior to the set-covering model suggested in [1]. In [2], computational experiments with the latter model are reported, and the authors were able to solve instances with up to 30 nodes. No instance with as much as 50 nodes was reported solved. For instances of 50 nodes, CPLEX 9.0 was still left with a gap ranging from 15% to 40% after two CPU-hours on a 2.4 GHz processor. This indicates that the set-covering formulation is not as strong as our strong formulations.

8 Conclusions

We have demonstrated how to solve the minimum-energy multicast problem by use of various network models. The problem in question is to minimize the total power assignment to the nodes in a wireless ad hoc network, such that all destination nodes are reachable, either directly or via relay nodes, from the source node. Different mixed integer programming approaches have been considered, including network flow models, minimum cut formulations and Steiner arborescence models. The models have been analyzed and compared theoretically, and the flow and Steiner models have also been evaluated experimentally.

All suggested integer programming models have been categorized by means of the strength of their continuous relaxation. We have proved that they are all stronger than a previously suggested single-commodity network flow model, and the numerical experiments indicated that the strength difference is large. The strength of the models defines an equivalence relation consisting of three equivalence classes, where the single-commodity flow model is the sole member of the very weakest class. Then there is one class consisting of weak flow and cut formulations, and one consisting of strong flow and cut formulations as well as two Steiner arborescence models.

Our experiments have confirmed that moving from weak to strong formulations very much affects the instance size that can be solved, and the time needed to solve them. For all sets of instances tested, the average optimality gap of the relaxed solutions of the weak formulations was more than hundred times the corresponding gap provided by strong formulations. By using the strongest flow formulations, it was possible to solve up to 61 out of 100 instances with 100 nodes out of which 10 were destinations, and we solved 75 out of 100 instances with 50 nodes and 25 destinations. This problem size is however about the largest that can be solved within the time bound on a modern computer.

The study has demonstrated that integer programming models with strong linear relaxations are useful when exact solutions to small or medium sized instances of the minimum-energy multicast problem are to be found. Also when the goal is a tight lower bound on the optimal objective function value, such models will be useful. For large instances, however, fast and effective heuristics or approximation algorithms will still be the preferred approach. It is our belief that some of the models proposed and analyzed in this paper can also serve as a basis for developing such methods.

References

[1] K. Altinkemer, F. S. Salman, P. Bellur, Solving the minimum energy broadcasting problem in ad hoc wireless networks by integer programming, in: Proceedings of the second workshop on modeling and optimization in mobile, ad hoc, and wireless Networks (WiOpt '04), April 2004, pp. 48–54.

[2] K. Altinkemer, F. S. Salman, P. Bellur, Solving the minimum energy broadcasting problem in ad hoc wireless networks by integer programming, in: Proceedings of the International Network Optimization Conference (INOP), pp. 635–642, 2005.

[3] M. Čagalj, J.-P. Hubaux, C. Enz, Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues, in: Proceedings of ACM MobiCom '02, September 2002, pp. 172–182.

[4] J. Cartigny, D. Simplot, I. Stojmenovic, Localized minimum-energy broadcasting in ad-hoc networks, in: Proceedings of IEEE INFOCOM '03, April 2003, pp. 2210–2217.

[5] T. Chu, I. Nikoladis, Energy efficient broadcast in mobile ad hoc networks, in: Proceedings of the First International Conference on Ad-Hoc Networks and Wireless (ADHOC-NOW), September 2002, pp. 177–190.

[6] A. K. Das, R. J. Marks, M. El-Sharkawi, P. Arabshahi, A. Gray, Minimum power broadcast trees for wireless networks: integer programming formulations, Proceedings of IEEE INFOCOM '03, April 2003, pp. 1001–1110.

[7] A. K. Das, R. J. Marks, M. El-Sharkawi, P. Arabshahi, A. Gray, A cluster-merge algorithm for solving the minimum power broadcast problem in large scale wireless networks, in: Proceedings of IEEE MILCOM '03, October 2003, pp. 416–421.

[8] A. K. Das, R. J. Marks, M. El-Sharkawi, P. Arabshahi, A. Gray, r-Shrink: A heuristic for improving minimum power broadcast trees in wireless networks, in: Proceedings of IEEE GLOBECOM '03, December 2003, pp. 523–527.

[9] A. K. Das, R. J. Marks, M. El-Sharkawi, P. Arabshahi, A. Gray, e-Merge: A Heuristic for improving minimum power broadcast trees in wireless networks, Technical Report, Department of Electrical Engineering, University of Washington, WA, 2003.

[10] Ömer Eğecioğlu, T. F. Gonzalez, Minimum-energy broadcast in simple graphs with limited node power, Proceedings of IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS) '01, August 2001, pp. 334–338.

[11] ILOG, Ilog CPLEX 9.1, user's manual, April 2005.

[12] R. Klasing, A. Navarra, A. Papadopoulos, S. Pérennes, Adaptive Broadcast Consumption (ABC), a new heuristic and new bounds for the minimum energy broadcast routing problem, in: N. Mitrou et al. (eds.): Networking 2004, Lecture Notes in Computer Science 3042, 2004, pp. 866–877.

[13] F. Li, I. Nikolaidis, On minimum-energy broadcasting in all-wireless networks, in: Proceedings of the 26th Annual IEEE Conference on Local Computer Networks (LCN '01), November 2001, pp. 193–202.

[14] S. Lindsey, C. S. Raghavendra, Energy efficient all-to-all broadcasting for situation awareness in wireless ad hoc networks, Journal of Parallel and Distributed Computing 63 (2003) 15–21.

[15] R. J. Marks, A. K. Das, M. El-Sharkawi, P. Arabshahi, A. Gray, Minimum power broadcast trees for wireless networks: optimizing using the viability lemma, Proceedings of the NASA Earth Science Technology Conference, June 2002, pp. 273–276.

[16] R. Montemanni, L. M. Gambardella, A. K. Das, The minimum power broadcast problem in wireless networks: a simulated annealing approach, Technical Report, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, Manno, Switzerland, 2004.

[17] P.-J. Wan, G. Călinescu, X.-Y. Li, O. Frieder, Minimum-energy broadcast routing in static ad hoc wireless networks, Wireless Networks 8 (2002) 607–617.

[18] P.-J. Wan, G. Călinescu, X.-Y. Li, O. Frieder, Minimum-power multicast routing in static ad hoc wireless networks, IEEE/ACM Transactions on Networking, 12 (2004), 3, pp. 507-514.

[19] J. E. Wieselthier, G. D. Nguyen, A. Ephremides, On the construction of energy-efficient broadcast and multicast trees in wireless networks, in: Proceedings of IEEE INFOCOM '00, March 2000, pp. 585–594.

[20] J. E. Wieselthier, G. D. Nguyen, A. Ephremides, Algorithms for energy-efficient multicasting in static ad hoc wireless networks, ACM Mobile Networks and Applications (MONET) 6 (2001) 251–263.

[21] P. Winter, Steiner problem in networks: A survey. *Networks* 17 (1987) 129–167.

[22] D. Yuan, An Integer Programming Approach for the Minimum-Energy Broadcast Problem in Wireless Networks, Proceedings of the International Network Optimization Conference (INOC), pp. 643-650, 2005.