NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

EFFICIENTLY INTERDICTING A TIME-EXPANDED TRANSSHIPMENT NETWORK

by

H. Dan Derbes

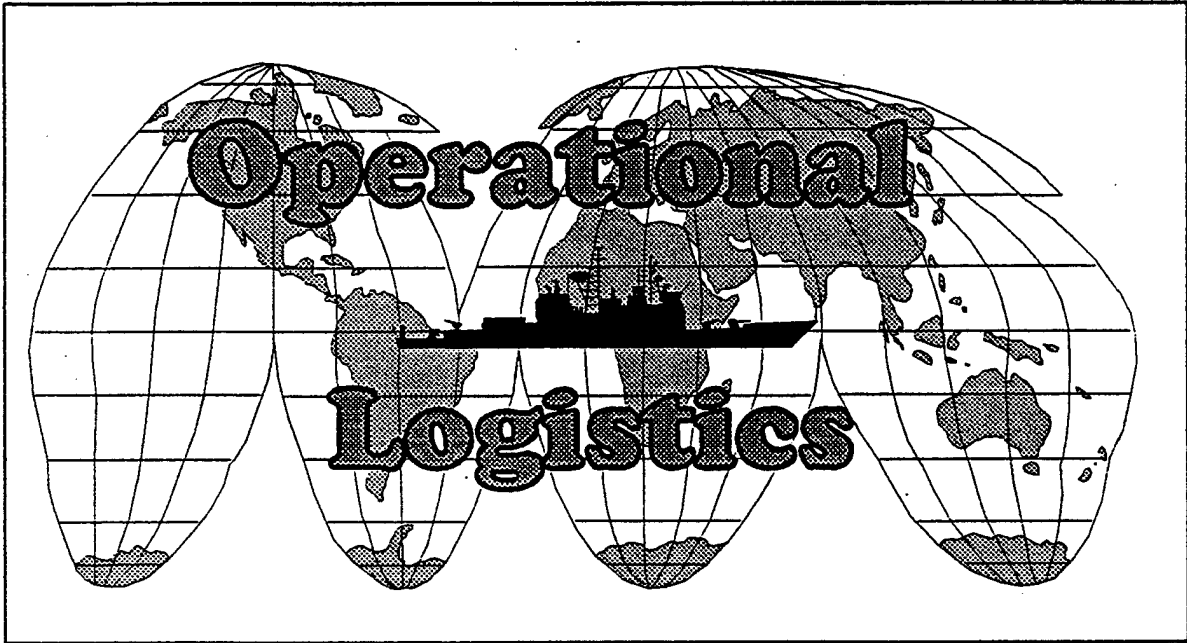September, 1997

Thesis Advisor: R. Kevin Wood
Second Reader: Paul S. Bloch

Approved for public release; distribution is unlimited.

Operational Logistics

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 1997	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE
Efficiently Interdicting A Time-Expanded Transshipment Network

5. FUNDING NUMBERS

6. AUTHOR(S)
Derbes, Henry D.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

A "network interdictor" has a limited supply of resource with which to disrupt a "network user's" flow of supplies in a capacitated transshipment network. The interdictor's problem of minimizing the maximum flow through the network is a difficult-to-solve integer programming problem but we show that a heuristic based on Lagrangian relaxation is very effective in approximately solving the problem.

We implement algorithms in C to approximately solve both the static (without considering time) and dynamic network interdiction problems. Static test networks range in size from 25 nodes and 64 arcs to 400 nodes and 1519 arcs. Using an IBM RS/6000 Model 590 workstation, we find optimal solutions for seven of 12 test networks and solve the largest problem in only 31.0 seconds. We model a dynamic network in time-expanded form in order to assign time weights of 0 or 1 to flow, include repair time of interdicted arcs, and provide a schedule to the network interdictor that identifies arcs and time periods for interdictions. Dynamic networks range in size from 525 nodes and 1,344 arcs to 40,400 nodes and 153,419 arcs (in time-expanded form). We find near-optimal solutions in 13 of 24 test networks and solve the largest network in 1729.5 seconds.

14. SUBJECT TERMS
Network Interdiction, Mathematical Modeling, Lagrangian Relaxation

15. NUMBER OF PAGES
85

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

EFFICIENTLY INTERDICTING A TIME-EXPANDED TRANSSHIPMENT
NETWORK

H. Dan Derbes
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1985

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the
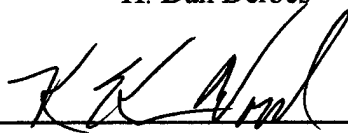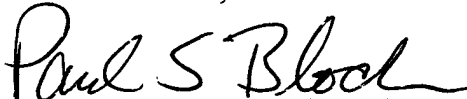
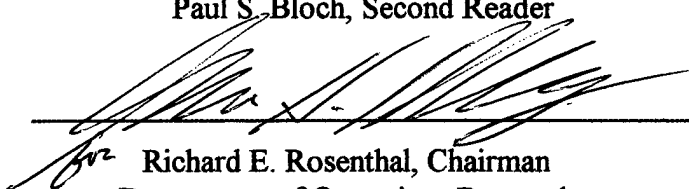**NAVAL POSTGRADUATE SCHOOL**
September 1997

Author: _____

H. Dan Derbes

Approved by: _____

R. Kevin Wood, Thesis Advisor

_____

Paul S. Bloch, Second Reader

_____

Richard E. Rosenthal, Chairman
Department of Operations Research

iii

ABSTRACT

A "network interdictor" has a limited supply of resource with which to disrupt a "network user's" flow of supplies in a capacitated transshipment network. The interdictor's problem of minimizing the maximum flow through the network is a difficult-to-solve integer programming problem but we show that a heuristic based on Lagrangian relaxation is very effective in approximately solving the problem.

We implement algorithms in C to approximately solve both the static (without considering time) and dynamic network interdiction problems. Static test networks range in size from 25 nodes and 64 arcs to 400 nodes and 1519 arcs. Using an IBM RS/6000 Model 590 workstation, we find optimal solutions for seven of 12 test networks and solve the largest problem in only 31.0 seconds. We model a dynamic network in time-expanded form in order to assign time weights of 0 or 1 to flow, include repair time of interdicted arcs, and provide a schedule to the network interdictor that identifies arcs and time periods for interdictions. Dynamic networks range in size from 525 nodes and 1,344 arcs to 40,400 nodes and 153,419 arcs (in time-expanded form). We find near-optimal solutions in 13 of 24 test networks and solve the largest network in 1729.5 seconds.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

x

EXECUTIVE SUMMARY

Opponents face each other in a field of battle. The environment and terrain affect both sides equally. To remain ready for battle, both sides need a constant supply of food, fuel, and repair parts; disruptions of those flows cause an immediate loss of combat power. In this scenario, one side uses a transportation network to provide supplies, troops, and ammunition to his forces. The other side, far from home, has the ability to identify his opponent's supply points and to temporarily stop movement of supplies over segments of that opponent's transportation network.

This thesis develops mathematical programming methods for the effective employment of limited interdiction assets to reduce the flow of a single commodity through a capacitated transportation (transshipment) network. The network user strives to maximize flow of a commodity through the network, while an interdictor, with limited assets, attempts to interdict (destroy) arcs or links in the network to minimize the maximum flow. We develop a dynamic model that allows the interdictor to assign time weights of 0 or 1 to the arrival of war material at battlefield destinations. The interdictor allocates his resources appropriately, keeping in mind that interdicted arcs can be repaired over time.

While this thesis is motivated by the possibility of weakening the military force of the network user before engagement in battle, other uses may include disrupting the escape routes of a fugitive or reducing the flow of illegal drugs and precursor chemicals moving through a network of rivers and roads. This problem has been studied before, during the Vietnam War and, more recently, in support of the war on illegal drugs.

Previous studies have not modeled the time aspect of moving logistics through a network. By representing the network in time-expanded form, we can define a specific time period of interest, include attributes such as a time-weighted value for flows arriving at sinks, the repair of arcs after interdiction, and a schedule for the employment of interdiction assets.

We assign time weights of 0 or 1 to flow arriving at a sink. We give value of 1 to flow arriving at a sink before the end of a "cutoff" time and 0 otherwise. A more general version of the dynamic network model might include arbitrary non-negative time weights but is beyond the scope of this thesis. We model repair of arcs by allowing interdictions to be effective for limited periods of time. We assume complete interdiction of an arc until repair, i.e., an interdicted arc has zero capacity until it is repaired. A repaired arc is restored to its nominal capacity.

The network interdiction problem is difficult to solve due to the interdiction budget constraint for the network interdictor. We relax this constraint using a Lagrangian relaxation that allows the interdictor to violate the constraint while paying a penalty. For a fixed value of a penalty parameter, the relaxation is an easy-to-solve maximum flow problem with a solution that provides a lower bound on the optimal solution to the network interdiction problem. The solution may or may not be feasible. We maximize the lower bound using binary search on the value of the penalty parameter, solving a maximum flow problem at each step. The best feasible solution obtained is the heuristic solution to the problem. The maximized lower bound and the objective value of the solution to the best feasible solution are compared to judge solution quality.

We implement algorithms to approximately solve both the static (without considering time) and dynamic network interdiction problems in C using an IBM RS/6000 Model 590 workstation. The static test networks range in size from 25 nodes and 64 arcs to 400 nodes and 1519 arcs. We find optimal solutions for seven of 12 test networks solving the largest problem in only 31.0 seconds. Dynamic networks range in size from 525 nodes and 1,344 arcs to 40,400 nodes and 153,419 arcs (in time-expanded form). We find near-optimal solutions in 13 of 24 test networks and solve the largest network in 1729.5 seconds.

ACKNOWLEDGMENT

I. INTRODUCTION

Opponents face each other in a field of battle. The environment and terrain affect both sides equally. To remain ready for battle, both sides need a constant supply of food, fuel, and repair parts. Disruptions of those flows cause an immediate loss of combat power. In this battle, one side uses a transportation network to provide supplies, troops, and ammunition to his forces. The other side, far from home, has the ability to identify his opponent's supply points and to temporarily stop movement of supplies over segments of the opponent's transportation network.

This thesis develops new mathematical programming methods for the effective employment of limited interdiction assets to reduce the time-weighted flow (weights are 0 or 1) of a single commodity through a capacitated transshipment network. The network user strives to maximize flow of a commodity through the network, while an interdictor, with limited assets, attempts to interdict (destroy) arcs or links in the network to minimize the maximum flow. The interdictor knows that his adversary values the commodity differently depending upon its time of arrival. The interdictor therefore gives weighted values to the arrival of war material at battlefield destinations and wishes to allocate his resources appropriately keeping in mind that broken arcs can be repaired over time. While this thesis is motivated by possibility of weakening the military force of the network user before engagement in battle, other uses may include disrupting the escape routes of a fugitive or reducing the flow of illegal drugs and precursor chemicals moving through a network of rivers and roads.

We introduce the reader to the network interdiction problem and its notation in this chapter. We heuristically solve a static model in Chapter II using a Lagrangian relaxation heuristic that relaxes the network interdictor's resource budget constraint. Although the network interdiction problem is difficult to solve, later, we see that this method often produces an optimal solution. In Chapter III we introduce the reader to the dynamic model and the time-expanded form of a network. Through the dynamic models presented in Chapter III, we can consider time-weighted flow and the repair of interdicted

1

arcs. Our solutions tell the interdictor not only where to strike but also when to strike. A Lagrangian heuristic provides reliable solutions for the dynamic network, also.

A.   BACKGROUND

Using limited resources, an interdictor attempts to restrict an enemy's use of a capacitated transshipment network. For a military interdictor, the immediate objective could be to decrease the fighting effectiveness of the enemy by minimizing the amount of supplies such as shipments of fuel, repair parts, ammunition, available to the enemy commander who is the "network user." Another objective might be to limit the enemy commander's ability to maneuver for a specified time period by destroying the bridges and roads around him. We model the logistic network as a single-commodity dynamic transshipment network with capacities and transit times on its arcs. The network has several sources, the supply points, and several sinks, the military units in the field. Arc capacities restrict flow rates while transit times determine how long each unit of flow spends traversing the network.

The goal of the network user is to move an appropriate amount of flow, the war material, out of each source and into each sink. The interdictor decides which arcs to interdict, the "interdiction set," while recognizing the time-weighted value of the logistics. In this thesis, the values of the time weights are 0 or 1 for reasons explained later. Each interdiction consumes an amount of a limited resource that may depend on the particular arc. There is a fixed amount of total resource available to the interdictor. Our models do not allow partial interdiction; interdicted arcs have zero capacity until repaired. We assume that the network user repairs each interdicted arc, restoring the arc to full capacity within a specified number of time units.

In this chapter, we introduce the notation and models used in solving a simple network interdiction problem. We describe the problem from the network user's point of view, the maximum flow model. Building on the maximum flow model, we develop the

network interdiction model. This model, an integer program, finds an interdiction set that minimizes the maximum network flow subject to limited assets.

In Chapter II, we solve the static network interdiction problem. The integer program for this model is hard to solve in practice, so we use a Lagrangian relaxation heuristic to solve the problem approximately. By adding small random amounts to arc capacities, the heuristic often finds an optimal solution.

We address the time-expanded network and repair of arcs in Chapter III. We use a single commodity dynamic transshipment network in time-expanded form. This form of the network allows us introduce a time-weighted aspect to network flow and to model the repair and possible re-attack of interdicted arcs during the time periods under consideration. We assume complete repair of arcs at the end of a repair interval. Our solution identifies a set of arcs to interdict and the time period in which to attack or re-attack these arcs.

B.   NETWORKS AND INTERDICTION

We first address interdiction in a static network, that is, one without time attributes. We present the maximum flow, minimum cut, and a simple network interdiction model to help the reader understand the basic problem. Most definitions follow Cormican (1995).

1.   Description of a Static Network

We define a network with respect to a directed graph $G = (N, A)$ where N is a set of n nodes and A is a set of m directed "edges" or "arcs." In a transshipment network, an arc (i, j) can be thought of as a length of roadway, river segment, etc., that provides a path for the flow of a commodity from i to j. A node, i, can be thought of as a road junction or the endpoint of a road segment. A commodity flowing through the network originates at a "source node" $a \in N$ in the network and flows to "sink node" $b \in N$. If there is more than one source or sink, we create a "super-source" and/or a "super-sink" and

artificial arcs with appropriately large capacities linking them to the sources and sinks, respectively. This idea is graphically presented in Figure 1. We expand the set of arcs A to include the artificial arcs and another artificial "return arc," (b, a), from the sink to the source or (b', a') from the supersink b' to the supersource a'.

(a) Original network   (b) Network with artificial arcs
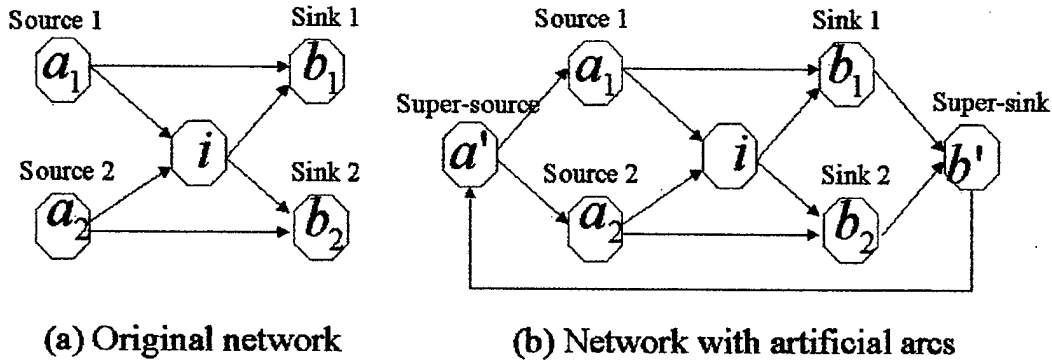
Figure 1. Network with and without artificial arcs.

Each arc (i, j) has an associated set of parameters that describes its characteristics. The finite nominal capacity or maximum allowable flow on an arc is denoted u_{ij}, where $u_{ij} \geq 0$. The capacity of the artificial return arc is large, such as $\sum_{(i,j) \in A} u_{ij} + 1$. The cost, in units of resource, to interdict an arc (i, j) is designated r_{ij}, and is typically assumed to be a small integer. It may occur that an arc cannot be interdicted at any cost for political, tactical, or other reasons and therefore the interdiction cost is large, $r_{ij} = \infty$. The interdictor has a total of R units of resource available for interdiction. In this thesis, we assume a single type of interdiction resource. A natural extension would include multiple types of interdiction resources available to the attacker and arcs that require specific types of resource to interdict them.

2.   Network Maximum Flow Models

The task of the network user is to move supplies from the sources to the sinks. The standard maximum flow linear programming model (e.g., Ahuja, Magnanti, and Orlin

4

1993, p. 168) determines the maximum quantity of a single commodity that can be moved through a capacitated network from source node a to sink node b. This maximum flow model, denoted MF, is:

MF

Indices:

$i, j \in N$ Nodes of $G = (N, A)$, includes two special nodes: a, the source or super-source and b, the sink or super-sink

$(i, j) \in A$ Arcs of $G = (N, A)$

Data:

u_{ij} Nominal capacity of arc (i, j)

Decision variable:

x_{ij} Amount of flow on arc (i, j)

The Formulation:

$$\max \ x_{ba} \qquad\qquad\qquad : \text{dual variables}$$

$$\text{s.t.} \ \sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = 0 \quad \forall \ i \in N \qquad : \alpha_i \tag{1}$$

$$0 \le x_{ij} \le u_{ij} \quad \forall \ (i,j) \in A \qquad : \theta_{ij} \tag{2}$$

The quantity x_{ij} is the flow of the commodity from node i to node j on directed arc $(i, j) \in A$, and x_{ba} is the flow from sink node b to source node a, on artificial arc (b,a), the return arc. The flow on arc (b,a) is the sum of all the flows from the source to the sink. Maximizing flow x_{ba} is equivalent to maximizing flow through the network. The flow balance constraints (1) require that the flow arriving at a node equal the flow leaving the node. The capacity constraints (2) require a non-negative flow on an arc that is not greater than the capacity of the arc.

The dual of the maximum flow problem is the minimum cut problem. The dual variables of the maximum flow model, α_i and θ_{ij} are shown in MF. When we find an optimal solution to a maximum flow problem, we also find an optimal solution to the minimum cut problem.

A "cut" is a partition of the node set N into two sets N_a and N_b, with $a \in N_a$ and $b \in N_b$. Each cut defines a set of arcs that have one endpoint in N_a and the other endpoint in N_b. With respect to the cut, an arc (i, j) is a "forward" arc if it is directed from a node $i \in N_a$ to a node $j \in N_b$. The capacity of the cut is the sum of the capacities of all the forward arcs associated with the cut. A minimum cut, then, is a cut of minimum capacity among all possible cuts in the network. (The above definitions follow Ahuja et al. (1993).) By the maximum flow-minimum cut theorem (Ford and Fulkerson, 1956), the maximum flow equals the capacity of a minimum cut. A minimum cut can be found directly by solving the dual of the maximum flow problem MFD (e.g., Wood, 1993):

MFD

Indices:

$i, j \in N$ Nodes of $G = (N, A)$, includes two special nodes: a, the source or super-source and b, the sink or super-sink

$(i, j) \in A$ Arcs of $G = (N, A)$

Data:

u_{ij} Nominal capacity of arc (i, j)

Decision variables:

α_i $\alpha_i = 1$ if $i \in N_b$ else $\alpha_i = 0$ if $i \in N_a$,

θ_{ij} $\theta_{ij} = 1$ if (i, j) is a forward arc of the minimum capacity cut, else $\theta_{ij} = 0$

The Formulation:

$$\min_{\alpha,\theta} \quad \sum_{(i,j)\in A} u_{ij}\theta_{ij}$$

$$\text{s.t.} \quad \alpha_i - \alpha_j + \theta_{ij} \geq 0 \quad \forall\, (i,j) \in A - (b,a) \tag{3}$$

$$\alpha_b - \alpha_a + \theta_{ba} \geq 1 \tag{4}$$

$$\theta_{ij} \geq 0 \quad \forall\, (i,j) \in A$$

$$\theta_{ba} = 0$$

MFD is totally unimodular and if we arbitrarily set $\alpha_a = 0$, then all variables will be 0 or 1 in an optimal extreme point solution. The variables in the model have the following physical interpretation: $\alpha_i = 1$ indicates $i \in N_b$, $\alpha_i = 0$ indicates $i \in N_a$, $\theta_{ij} = 1$ indicates arc (i,j) $i \in N_a$, $j \in N_b$, and $\theta_{ij} = 0$, otherwise.

3. The Network Interdiction Model

The network interdiction problem can be formalized in a min-max flow-based model. The network user attempts to maximize the flow across the network, while the interdictor simultaneously strives to minimize this maximum flow. The network interdictor's activities are limited by a budget constraint. The standard network interdiction model is:

S-NIM
Indices:

$i, j \in N$ Nodes of $G = (N, A)$, includes two special nodes: a, the source or super-source and b, the sink or super-sink

$(i,j) \in A$ Arcs of $G = (N, A)$

Data:

u_{ij} Nominal capacity of arc (i,j)

r_{ij} Amount of resource required to interdict arc (i,j)

7

R Total amount of resource for interdiction available to the network interdictor

Network user decision variables:

x_{ij} Amount of flow on arc (i,j)

Network interdictor decision variables:

γ_{ij} $\gamma_{ij} = 1$ indicates arc (i,j) is interdicted; else $\gamma_{ij} = 0$

The Formulation:

$$z^* = \min_{\gamma \in \Gamma} \; \max_{\mathbf{x}} \; x_{ba} \tag{5}$$

$$\text{s.t} \sum_{(i,j) \in A} x_{ij} - \sum_{(i,j) \in A} x_{ji} = 0 \quad \forall i \in N \tag{6}$$

$$0 \le x_{ij} \le u_{ij}(1 - \gamma_{ij}) \quad \forall \, (i,j) \in A \tag{7}$$

$$\gamma_{ba} \equiv 0$$

$$\text{where } \Gamma = \left\{ \gamma \mid \sum_{(i,j) \in A} r_{ij} \gamma_{ij} \le R, \; \gamma_{ij} \in \{0,1\} \; \forall (i,j) \in A \right\} \tag{8}$$

The objective function (5) seeks to minimize the maximum flow. Constraints (6) are just the flow balance constraints from MF. The arc capacity constraints (7) restrict the amount of flow on the arc to either the nominal capacity if the arc is not interdicted or zero if the arc is interdicted. In this model, $\gamma_{ij} = 1$ if arc (i,j) is interdicted and $\gamma_{ij} = 0$ if the arc is not interdicted. Interdiction resource constraint (8) limits the interdiction decisions. Each interdiction consumes an amount of interdiction resource, r_{ij}. The total resource consumed by the interdiction set must be less than the amount of resource available R.

Wood (1993) shows that the inner maximum flow model can be converted to its dual and the minimum cut model and the resulting nonlinear integer program linearized. This model is a simple minimizing integer program that will be addressed in Chapter II together with a solution method using Lagrangian relaxation.

C. LITERATURE SEARCH

There was strong interest in the network interdiction problem during the Vietnam War. Works during this time period were either very general, such as Wollmer (1964), to the very specific also by Wollmer (1970). More recently, the war on illegal drugs generated new interest in network interdiction, Phillips (1992). There are other contributors to the topic but almost all these works share the characteristic of being specific to the application and not easily generalizable. More recent works by Steinrauf (1991), Wood (1993) and Cormican (1995) overcome this limitation by adopting a mathematical programming approach. The advantage of this approach is that it readily generalizes and is easily adaptable to a variety of network interdiction problems. These mathematical models, however, are integer and mixed-integer programs that are difficult to solve. Wood (1993) shows that the basic network interdiction problem is NP-complete, even when restricted to planar graphs where interdictions require varying amounts of resource, or to non-planar graphs requiring only one unit of resource to interdict any arc.

We have found no sources that address the network interdiction problem in time-expanded form. Only Wollmer (1970) addresses repair time of arcs. Wollmer allows partial interdiction of arcs and finds the best single arc to break, repeating the process for multiple interdictions. His algorithm selects an arc for interdiction that maximizes the sum of the repair cost plus the product of the repair time and the cost increase of a minimum-cost circulation flow. Wollmer's methodology determines an approximately optimal interdiction set in exponential time.

Ratliff, Sicillia, and Lubore (1975) solve the network interdiction problem by finding a set of n arcs whose simultaneous removal from a connected single commodity network results in the greatest decrease in the throughput capacity of the remaining system between the source and the sink. The method applies to planar and non-planar networks but addresses neither a dynamic network nor repair of arcs.

Steinrauf (1991) develops a mathematical programming approach for the network interdiction problem using integer programming. It solves small problems but is not very useful for large integer problems. A relaxation or decomposition is needed.

Cormican (1995) develops a deterministic model using Benders decomposition with an original "flow-dispersion heuristic." The flow-dispersion heuristic achieves a maximum flow while keeping flows on individual arcs as small as possible. This serves to decrease solution time by reducing the number of iterations that Benders decomposition requires for convergence. Cormican extends the method to include stochastic arc capacities.

Phillips (1992) describes pseudo-polynomial time algorithms that provide the interdictor with a strategy that optimally uses exactly the amount of resources he is willing to commit to the attack. Phillips proves that for a fixed cut, a greedy attack strategy is optimal. A greedy attack strategy is one that removes as much of the cut's fixed capacity as possible, expending exactly the entire interdiction budget. A simple algorithm for the network interdictor's problem would enumerate all cuts, (an exponential number are possible), compute the result of a greedy attack on each, and pick the best one. Phillips' method does not solve an integer problem and instead assigns some benefit to partial interdictions. Recognizing the need for faster algorithms, Phillips proposes pseudo-polynomial-time algorithms for planar graphs and shows how to convert them into fully polynomial-time approximation schemes.

Wood (1993) develops integer programming models for the network interdiction problem and its variations. He develops a simple minimization model that is derived from the formal min-max network interdiction model. The problem is shown to be NP complete. He does not consider the time-expanded dynamic network.

This thesis continues the work of others to develop mathematical programming models for the network interdiction problem. We seek to include aspects associated with time in our models by using a time-expanded form. This form allows us to model time weights on flow of 0 or 1 and arcs that are repaired a certain amount of time after

interdiction. We have already introduced the maximum flow model with its dual and the standard network interdiction model. In Chapter II, we develop the network interdiction problem in its simpler, static form, without time attributes, and present a Lagrangian relaxation heuristic to find a good feasible set of arcs to interdict.

II. SOLVING THE BASIC MODEL

In Chapter I, we presented the standard network interdiction problem S-NIM. Because S-NIM involves static flows, i.e., does not involve flows over time, we call it a "static version" of the network interdiction problem to differentiate it from the "dynamic version" covered in Chapter III. Wood (1993) shows that the static network interdiction problem is NP-complete, even when restricted to planar graphs where interdictions require varying amounts of resource. In this chapter, we seek faster methods to solve the static problem (at least approximately) using Lagrangian relaxation. Although not guaranteed to find an optimal solution, optimal solutions are consistently obtained by the Lagrangian relaxation method described in this chapter when $r_{ij} = 1$ for all arcs (i, j).

A. THE INTEGER PROBLEM

Recall from Chapter I that the standard network interdiction model, S-NIM, is stated as a min-max problem where the network user attempts to maximize flow across the network while the interdictor is simultaneously striving to minimize that flow subject to the interdiction budget constraint.

For a fixed interdiction decision, note that the inner maximization of S-NIM is just a maximum flow problem. We can take the dual of the inner maximum flow model and linearize the resulting model to obtain the simple network interdiction model (Wood, 1993):

NIM

Indices:

$i, j \in N$ Nodes of $G = (N, A)$, includes two special nodes, a the source or super-source, and b, the sink or super-sink

$(i, j) \in A$ Arcs of $G = (N, A)$

Data:

u_{ij} Nominal capacity of arc (i, j)

r_{ij} Amount of resource required to interdict arc (i, j), $r_{ij} \geq 0$ and integer

R Total resource for interdiction available to the network interdictor, $R \geq 0$ and integer

Decision variables:

α_i $\alpha_i = 1$ if $i \in N_b$, $\alpha_i = 0$ if $i \in N_a$,

β_{ij} $\beta_{ij} = 1$ if arc (i, j) is in the cut and not interdicted, otherwise $\beta_{ij} = 0$

γ_{ij} $\gamma_{ij} = 1$ indicates arc (i, j) is interdicted; otherwise $\gamma_{ij} = 0$

The Formulation:

$$z^* = \min_{\alpha, \gamma, \beta} \sum_{(i,j) \in A} u_{ij} \beta_{ij}$$

$$\text{s.t.} \quad \alpha_i - \alpha_j + \gamma_{ij} + \beta_{ij} \geq 0 \quad \forall (i, j) \in A - (b, a)$$

$$\alpha_b - \alpha_a + \gamma_{ba} + \beta_{ba} \geq 1$$

$$\alpha_i \in \{0, 1\} \ \forall i \in N$$

$$\beta_{ij} \in \{0, 1\} \ \forall (i, j) \in A$$

$$\beta_{ba} \equiv 0$$

$$\sum_{(i,j) \in A} r_{ij} \gamma_{ij} \leq R \tag{9}$$

$$\gamma_{ij} \in \{0, 1\} \ \forall (i, j) \in A$$

$$\gamma_{ba} \equiv 0$$

Note that this model resembles the dual of the maximum flow model, MFD, from Chapter I, but with θ_{ij} replaced by $\gamma_{ij} + \beta_{ij}$. NIM identifies a cut where the variables α_i have the same meaning as in MFD. γ_{ij} and β_{ij} represent interdiction decisions and have the following interpretation: For forward arcs (i, j) in the cut, $\alpha_i - \alpha_j = -1$ so $\gamma_{ij} + \beta_{ij} = 1$ is required. So, either $\gamma_{ij} = 1$, indicating that this arc is interdicted, or $\beta_{ij} = 1$, indicating that this arc forms part of the minimum capacity cut after interdiction.

$\gamma_{ij} = \beta_{ij} = 0$ outside the cut indicating that these arcs are neither interdicted nor do they form part of the minimum capacity cut after interdiction.

B.   LAGRANGIAN RELAXATION FOR THE SIMPLE INTEGER PROBLEM

The basic integer program formulation NIM is known to be hard to solve. Without the complicating interdiction budget constraint (9), the problem is an easy-to-solve model like MFD. While constraint (9) cannot be ignored, it can be relaxed: We use a Lagrangian relaxation of the interdiction resource constraint. The relaxation allows us to approximately solve the problem by moving the resource constraint into the objective function. The resulting problem is almost as easy to solve as if constraint (9) were ignored. NIM, with the Lagrangian relaxation, is:

LR(λ)

$$z(\lambda) = \min_{\alpha,\gamma,\beta} \sum_{(i,j)\in A}(u_{ij}\beta_{ij} + \lambda r_{ij}\gamma_{ij}) - \lambda R \qquad (10)$$

$$\text{s.t.} \quad \alpha_i - \alpha_j + \gamma_{ij} + \beta_{ij} \geq 0 \quad \forall\ (i,j) \in A - (b,a) \qquad (11)$$

$$\alpha_b - \alpha_a + \gamma_{ba} + \beta_{ba} \geq 1$$

$$\alpha_i \in \{0,1\} \quad \forall i \in N$$

$$\alpha_a \equiv 0, \alpha_b \equiv 1$$

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

$$\beta_{ba} \equiv 0$$

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

$$\gamma_{ba} \equiv 0$$

The objective function (10) incorporates the resource constraint and a Lagrangian multiplier λ. The function $z(\lambda)$ is a concave function in λ so that we can find the best value for $z(\lambda)$ by iteratively adjusting λ using a binary search. The objective function is

15

derived from $\min\limits_{\alpha,\gamma,\beta} \sum\limits_{(i,j)\in A} u_{ij}\beta_{ij} + \lambda(\sum\limits_{(i,j)\in A} r_{ij}\gamma_{ij} - R)$ where the reader can more clearly observe

the interdiction resource constraint (9) in the objective function with multiplier λ.

Because of the relaxation, $z(\lambda) \le z*$ but as we adjust λ we may find a multiplier such that $z(\lambda) = z*$. In fact when we have found a Lagrangian multiplier such that $\sum\limits_{(i,j)\in A} r_{ij}\gamma_{ij} = R$, then

$$z(\lambda) = \min\limits_{\alpha,\gamma,\beta} \sum\limits_{(i,j)\in A} u_{ij}\beta_{ij} + \lambda(\sum\limits_{(i,j)\in A} r_{ij}\gamma_{ij} - R) = \min\limits_{\alpha,\gamma,\beta} \sum\limits_{(i,j)\in A} u_{ij}\beta_{ij} = z*. \tag{12}$$

However, we cannot be assured of an optimal solution if some portion of the interdiction resource, R, remains unused. In this case, $\lambda(\sum\limits_{(i,j)\in A} r_{ij}\gamma_{ij} - R) < 0$ and there may be some benefit in expending more interdiction resource. Therefore, a solution with $\sum\limits_{(i,j)\in A} r_{ij}\gamma_{ij} < R$ may not be optimal.

With the interdiction budget constraint in the objective function, LR(λ) is unimodular and can be solved as a linear program. Since the extreme point solutions to the related dual of the maximum flow problem are binary, we can remove the upper bound constraints on α_i and β_{ij} and γ_{ij} in the linear relaxation without changing the solution. The relaxation of LR(λ) is:

LRR(λ)

$$z(\lambda) = \min_{\alpha,\gamma,\beta} \sum_{(i,j)\in A} (u_{ij}\beta_{ij} + \lambda r_{ij}\gamma_{ij}) - \lambda R$$

s.t. $\quad \alpha_i - \alpha_j + \gamma_{ij} + \beta_{ij} \geq 0 \quad \forall\, (i,j) \in A - (b,a)$

$$\alpha_b - a_a + \gamma_{ba} + \beta_{ba} \geq 1$$

$$\alpha_i \geq 0 \quad \forall i \in N$$

$$\alpha_a \equiv 0, \quad \alpha_b \equiv 1$$

$$\beta_{ij} \geq 0 \quad \forall (i,j) \in A$$

$$\beta_{ba} \equiv 0$$

$$\gamma_{ij} \geq 0 \quad \forall (i,j) \in A$$

$$\gamma_{ba} \equiv 0$$

LRR(λ) looks similar to the dual to the maximum flow problem MFD which also has integer extreme points. As a result, we may solve LR(λ) by solving LRR(λ), or more importantly, through its dual. For a fixed λ, the λR term is a constant and remains in the objective function for the dual. The dual of the relaxed model of the Lagrangian relaxation is:

LRR-D1(λ)

$$z(\lambda) = \max\ x_{ba} - \lambda R$$

s.t. $\quad \displaystyle\sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = 0 \quad \forall\, i \in N$

$$0 \leq x_{ij} \leq u_{ij} \quad \forall\, (i,j) \in A$$

$$0 \leq x_{ij} \leq \lambda r_{ij} \quad \forall\, (i,j) \in A$$

Since the flow must meet both capacity constraints, the capacity constraints can be restated as $0 \leq x_{ij} \leq \min\{u_{ij}, \lambda r_{ij}\}$. Model LRR-D2($\lambda$) is equivalent to LRR-D1(λ) with the restated capacity constraint:

LRR-D2(λ)

$$z(\lambda) = \max \ x_{ba} - \lambda R$$

$$\text{s.t.} \quad \sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = 0 \quad \forall \ i \in N$$

$$0 \leq x_{ij} \leq \min\{u_{ij}, \lambda r_{ij}\} \quad \forall \ (i,j) \in A$$

This is very similar to the maximum flow model, MF, with capacities modified by the dual cost of interdicting the arc. Any solution to LRR-D2(λ) finds a minimum capacity cut that corresponds to a feasible or infeasible solution to the original problem NIM as follows:

a. $\alpha_i = 1 \ \forall i \in N_b$, $\alpha_i = 0 \ \forall i \in N_a$.

b. $\beta_{ij} = 1$ if $i \in N_a, j \in N_b$ and $x_{ij} = u_{ij}$, i.e., if (i, j) is a forward arc of the minimum cut and $u_{ij} < \lambda r_{ij}$, then (i, j) is not interdicted.

c. $\gamma_{ij} = 1$ if $i \in N_a, j \in N_b$ and $x_{ij} = \lambda r_{ij}$, i.e., if (i, j) is a forward arc of the minimum cut and $u_{ij} > \lambda r_{ij}$, then (i, j) is interdicted.

d. $\gamma_{ij} = \beta_{ij} = 0 \ \forall (i, j)$ that are not forward arcs in the cut.

Note that λ can always be perturbed so that $u_{ij} = \lambda r_{ij}$ does not occur. The solution is feasible if the interdiction budget constraint (9) holds.

C.   SOLVING THE RELAXED NETWORK INTERDICTION PROBLEM

Given a fixed λ, we can find a minimum capacity cut by solving LR(λ) using an easy-to-implement polynomial-time maximum flow algorithm for LRR-D2(λ). Arcs in the minimum capacity cut, A_C, are examined to find the corresponding interdiction set, $A_I \subseteq A_C$, that may or may not be feasible. If A_I is feasible for Γ, the corresponding maximum flow value, $\bar{z}$ is an upper bound for z^*, the solution to the network interdiction model. We continue to adjust the Lagrangian multiplier, re-solving LRR-D2(λ) and trying

to maximize $z(\lambda)$, the lower bound until the "best" upper bound, the smallest value of $\bar{z}$ found, and lower bound are equal, or the difference is as small as possible.

We iteratively solve LR(λ) using binary search on λ to find a good feasible solution. We decrease λ if $\sum_{(i,j)\in A} r_{ij}\gamma_{ij} < R$ encouraging use of more interdiction resource.

We increase λ if $\sum_{(i,j)\in A} r_{ij}\gamma_{ij} > R$, raising the penalty for exceeding the interdiction budget.

We have found an optimal solution to NIM if $\sum_{(i,j)\in A} r_{ij}\gamma_{ij} = R$ since for this solution the equality (12) holds. In words, the lower bound $z(\lambda)$ equals the upper bound z^* and we have found an optimal interdiction set.

The method may fail to find an optimal solution to NIM when it cannot identify a set of arcs to interdict that consumes the entire interdiction resource budget. One such failure occurs if the optimal cut found in LRR-D2(λ) is composed of a number of arcs with equal capacity. For example, the network in Figure 2 consists of five arcs in parallel between the source and sink with $r_{ij} = 1$ and $u_{ij} = u \; \forall (i,j)$. We have three units of interdiction resource, i.e., $R = 3$. There is only one cut. For any λ, the corresponding solution to NIM interdicts all arcs (infeasible) or no arcs rather than the $R = 3$ arcs that are optimal. Thus, Lagrangian relaxation will never find an optimal solution for this problem.
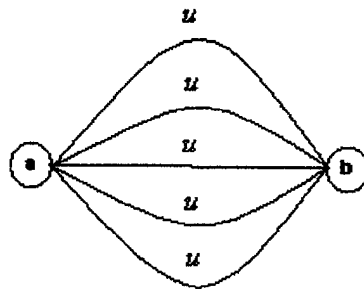
Figure 2. A network without an easy interdiction set

But, suppose that we add a very small random amount of capacity to each arc. The algorithm is able to differentiate between arcs that would otherwise appear identical

and the amount added is small enough not to affect the solution substantially. In practice, we add a random 1/100,000 to 100/100,000 of capacity to each arc. We accomplish this by multiplying all capacities by a scaling factor of 100,000 and then add a random amount that varies uniformly from 1 to 100. This small amount causes only negligible changes in the maximum flow in the network. This method has the advantage of only increasing solution time by extra log (100,000) iterations while enabling the algorithm to solve a difficult problem.

We use a shortest augmenting path algorithm (Edmonds and Karp, 1972) to solve LRR-D2(λ). (See Appendix A.) We modify the algorithm to find the "shortest augmenting path," the augmenting path with as few arcs as possible, with maximum capacity. We implement the Lagrangian relaxation and solve the maximum flow problem to identify the minimum capacity cut A_C and the set of arcs to interdict $A_I \subseteq A_C$.

We solve this problem for several test cases with the results discussed in Chapter IV. The shortest augmenting path algorithm used in this thesis has a worst case complexity of $O(nm^2)$ (Ahuja et al., 1993, p. 213). The methods used to discriminate between arcs, the random amount and the scaling, add a factor of log(100,000) to the work of the algorithm. The process of finding the best Lagrangian multiplier using binary search requires $O(\log U)$ solutions of the maximum flow problem where U is the maximum capacity of an arc in the network. The complexity of the relaxed network interdiction problem is polynomial:

$O(nm^2(\log U + \log(100,000))) = O(nm^2 \log U)$.

We have tested the relaxed network interdiction problem with many different networks. With $r_{ij} = 1 \ \forall (i,j) \in A$, we often find an optimal solution. We find good solutions that are sometimes optimal for the more general problem with general integer r_{ij}.

Building on the material presented in this chapter, we next explore the dynamic network in time-expanded form.

III. THE TIME-EXPANDED PROBLEM

We have introduced the reader to the static network interdiction problem and will now develop a dynamic version of that problem. We propose models in which the value of the flow is time-weighted and interdicted arcs are repaired. In this thesis the time weights are 0 or 1 depending on the time of arrival of flow at sinks. The solution identifies arcs and time periods of interdiction.

We introduce the reader to the dynamic model in Section A by developing the time-expanded maximum flow model. In Section B, we develop the time-expanded network interdiction model. In Section C, we develop a constrained minimum cut model from the network interdiction model. In Section D, we use a Lagrangian relaxation to find a feasible solution of the time-expanded network interdiction model. In Section E, we present a method that implements the models of Section D to find a feasible interdiction set for the time-expanded network interdiction problem.

A.   DYNAMIC MAXIMUM FLOW MODELS

Dynamic network problems can be solved as traditional network problems on exponentially (pseudo-polynomially) large, time-expanded networks. Our motivation for using a dynamic network model is a desire to include additional attributes in the models such as time-weighted flow, repair time for interdicted arcs, and a schedule for the network interdictor to use in allocating assets. For a given network $G = (N, A)$, we form a time-expanded network $G^T = (N^T, A^T)$ as follows: The quantity T represents the integer-valued time horizon for the dynamic problem consisting of $T + 1$ time periods; we make $T + 1$ copies $i_0, i_1, i_2, ..., i_T$ of each node i. Node i_t in the time-expanded network represents node i in the original network at time t. τ_{ijt} is defined as the time required for a commodity to traverse (i, j) from i to j at time t. Normally, $\tau_{ij} = \tau_{ijt}$ for all $(i, j) \in A$ and all t. We include arc $(i, j)_t \equiv (i_t, j_{t+\tau_{ij}})$ with capacity u_{ijt} in G^T whenever $(i, j) \in A$ and $0 \le t \le T - \tau_{ij}$. This allows flow to leave i at time t and arrive at j at time $t + \tau_{ij}$.

21

The quantity x_{ijt} denotes the amount of flow on $(i,j)_t$. In a feasible dynamic flow, at most u_{ijt} units of flow can be sent on arc $(i,j)_t$ although we assume $u_{ijt} = u_{ij}$ for all $(i,j) \in A$ and all t.

The time-expanded problem requires us to adjust our definition of source and sink. For each source and sink in the static network, there are $T + 1$ sources and sinks in the time-expanded network. We add artificial arcs from super-source a' to each source $a_t \in N^T$, and artificial arcs from the sinks, $b_t \in N^T$ to the super-sink b'. Each of these artificial arcs is included in A^T. We also include an artificial arc "return arc" (b',a') from the super-sink at time T to the super-source at time 0 to complete the circulation of flow from the sinks to the sources. The capacity of this arc is large and unconstraining, such as $\sum_{(i,j)_t \in A^T} u_{ij} + 1$. We define the super-source to exist only at time period 0 and the super-sink only at the last time period T. Artificial arcs connecting the super-source to the regular source nodes, (a'_0, a_t), have $\tau_{a'a_t} = t$. Artificial arcs connecting the sinks to the super-sink, b'_T, have $\tau_{b_t b'} = T - t$. The return arc has $\tau_{b'a't} = -T$ as a notational convenience.

In the dynamic maximum flow problem, the task of the network user is to find the maximum amount of logistics that can move from the sources to the sinks, subject to arc capacity constraints. The time-expanded maximum flow model, TE-MF, represents this problem. Note that the objective function value $x_{b'a'T}$ is the return arc representing the total flow through the time-expanded network. The arrival of the flow at the sinks is not time-weighted in this model.

TE-MF

Indices:

$i, j \in N$ Nodes of $G = (N, A)$, including two special nodes, a' the super-source, and b' the super-sink

t, t' Time periods: $t, t' = 0, 1, 2, \ldots, T$

$(i,j)_t$ Arc of $G = (N, A)$ at time period t

Data:

u_{ij} Nominal capacity of arc $(i,j)_t$

τ_{ij} Traverse time of arc $(i,j)_t$; flow leaves i at time t and arrives at j at time $t +$

       τ_{ij}

Network user decision variable:

x_{ijt} Flow on arc $(i,j)_t$

The Formulation:

$$\max_{\mathbf{x}} \; x_{b'a'T} \qquad\qquad\qquad : \text{dual variables}$$

$$\text{s.t.} \sum_{(i,j)_t \in A^T} x_{ijt} - \sum_{(j,i)_{t'} \in A^T | t'=t-t_{ij}} x_{jit'} = 0 \quad \forall \, i_t \in N^T \qquad : \alpha_{it} \qquad (12)$$

$$0 \le x_{ijt} \le u_{ij} \quad \forall \, (i,j)_t \in A^T \qquad : \theta_{ijt} \qquad (13)$$

The time-weighted maximum flow problem attaches different values, $w_{ib't}$, to the arc flows depending on their time of arrival at the sink. This attribute recognizes that the network user does not place equal value to the same shipment of supplies when it arrives before a battle as when it arrives after the battle. The time-weighted maximum flow model is:

TW-MF

$$\max_{\mathbf{x}} \sum_{t=0}^{T} \sum_{(i,b')_t \in A^T} w_{ib't} x_{ib't}$$

$$\text{s.t.} \sum_{(i,j)_t \in A^T} x_{ijt} - \sum_{(j,i)_{t'} \in A^T | t'=t-t_{ij}} x_{jit'} = 0 \quad \forall \, i_t \in N^T$$

$$0 \le x_{ijt} \le u_{ij} \quad \forall \, (i,j)_t \in A^T$$

A completely general dynamic network problem would allow each $w_{ib't}$ to be an arbitrary nonnegative value. This would significantly increase the difficulty of the problem, however, and methods required to solve it are beyond the scope of this thesis. (Benders decomposition may work; see Cormican (1995).) We use binary weights only: We assign weights so that flow arriving at sink b, on or before the latest allowable time period T_b has a value of 1. Flow arriving after this cutoff time T_b has a value of 0. If we simply remove the artificial arcs from A^T that have a time-weighted value of 0, TW-MF is equivalent to TE-MF.

The dual of the maximum flow problem is the minimum cut problem. The dual variables of the dynamic maximum flow model, α_{it} and θ_{ijt} are shown in TE-MF. When we find an optimal solution to a maximum flow problem, we also find an optimal solution to the minimum cut problem.

We adjust our definitions pertaining to cuts to fit the time-expanded problem. A "cut" is a partition of the node set N^T into two sets $N_{a'}^T$ and $N_{b'}^T$, with $a' \in N_{a'}^T$ and $b' \in N_{b'}^T$. Each cut defines a set of arcs that have one endpoint in $N_{a'}^T$ and the other endpoint in $N_{b'}^T$. The capacity of the cut is the sum of the capacities of the forward arcs in the cut. We find a minimum capacity cut by solving TE-MFD:

TE-MFD

Indices:

$i, j \in N$ Nodes of $G = (N, A)$, including two special nodes, a' the super-source, and b the super-sink

t, t Time periods: $t, t = 0, 1, 2, \ldots, T$

$(i, j)_t$ Arc of $G = (N, A)$ at time period t

Data:

u_{ij} Nominal capacity of arc $(i, j)_t$

τ_{ij} Traverse time of arc $(i,j)_t$; flow leaves i at time t and arrives at j at time $t +$

τ_{ij}

Decision variables:

α_{it} $\alpha_{it} = 1$ if $i_t \in N_{b'}^T$, else $\alpha_{it} = 0$ if $i_t \in N_{a'}^T$,

θ_{ijt} $\theta_{ijt} = 1$ if $(i,j)_t$ is a forward arc of the minimum capacity cut, else $\theta_{ijt} = 0$

The Formulation:

$$\min_{\alpha,\theta} \quad \sum_{(i,j)_t \in A^T} u_{ij} \theta_{ijt}$$

$$\text{s.t.} \quad \alpha_{it} - \alpha_{jt} + \theta_{ijt} \geq 0 \quad \forall \ (i,j)_t \in A - (b',a')_T \tag{14}$$

$$\alpha_{b'T} - \alpha_{a'0} + \theta_{b'a'T} \geq 1 \tag{15}$$

$$\theta_{ijt} \geq 0 \quad \forall \ (i,j)_t \in A$$

$$\theta_{b'a'T} = 0$$

TE-MFD is totally unimodular and if we arbitrarily set $\alpha_{a'0} = 0$, then all variables will be 0 or 1 in an optimal extreme point solution.

B.   THE TIME-EXPANDED INTERDICTION MODEL

Recall in Chapters I and II, that the network interdiction problem is stated as a min-max problem. In the time-expanded network, the network user attempts to maximize the time-weighted flow across the network over a specified time interval $[0, T]$. The network interdictor attempts to minimize the time-weighted maximum flow by selecting an appropriate interdiction set subject to the interdiction budget constraint.

We state the time-expanded model in a fundamentally different manner than we state S-NIM. The approach that will be taken can best be illustrated with the static model. Recall that S-NIM is a min-max model in which interdictions reduce maximum arc capacities to zero. An alternative formulation (Cormican, Morton and Wood, 1996) subtracts interdicted flow in the objective function. This formulation is:

S-NIM1

$$z^* = \min_{\gamma \in \Gamma} \max_{\mathbf{x}} \; x_{ba} - \sum_{(i,j) \in A} \gamma_{ij} x_{ij} \qquad (5')$$

$$\text{s.t} \; \sum_{(i,j) \in A} x_{ij} - \sum_{(i,j) \in A} x_{ji} = 0 \quad \forall i \in N \qquad (6)$$

$$0 \le x_{ij} \le u_{ij} \quad \forall \, (i,j) \in A \qquad (7')$$

$$\gamma_{ba} \equiv 0$$

$$\text{where } \Gamma = \left\{ \gamma \,\middle|\, \sum_{(i,j) \in A} r_{ij} \gamma_{ij} \le R, \; \gamma_{ij} \in \{0,1\} \; \forall (i,j) \in A \right\} \qquad (8)$$

We extend this model to a time-expanded dynamic network interdiction model. The dynamic network interdiction problem allows each arc to have an additional attribute, q_{ij}, an integer number of time periods required to repair arc (i,j) to full capacity starting the period after the interdiction. Interdicting (i,j) at time t' means $x_{ijt} \equiv 0$ for $t = t', t'+1, t'+2, \ldots, t'+q_{ij}$. Nominal capacity u_{ij} is restored for arc $(i,j)_t$ at time $t = t'+q_{ij} + 1$.

The time-expanded, network interdiction, min-max model is:

TE-NIM

Indices:

$i, j \in N$ Nodes of $G = (N, A)$, including two special nodes, a' the super-source, and b' the super-sink

t, t' Time periods: $t, t' = 0, 1, 2, \ldots, T$

$(i,j)_t$ Arc of $G = (N, A)$ at time period t

Data:

u_{ij} Nominal capacity of arc $(i,j)_t$

τ_{ij} Traverse time of arc $(i,j)_t$; flow leaves i at time t, arrives at j at time $t + \tau_{ij}$

26

r_{ij} Amount of resource required to interdict arc $(i,j)_t$

q_{ij} Repair time interval for arc $(i,j)_t$ (begins the period after interdiction)

R Resource for interdiction available to the network interdictor

Network user decision variables:

x_{ijt} Flow on arc $(i,j)_t$

Interdictor decision variables:

γ_{ijt} $\gamma_{ijt} = 1$ indicates arc $(i,j)_t$ is interdicted at time t; otherwise $\gamma_{ijt} = 0$

The Formulation:

$$z^{TE*} = \min_{\gamma \in \Gamma} \max_{x} x_{b'a'T} - \sum_{(i,j)_t \in A^T} \left(\sum_{t'=t-q_{ij}}^{t} \gamma_{ijt'} \right) x_{ijt} \tag{16}$$

$$\text{s.t.} \sum_{(i,j)_t \in A^T} x_{ijt} - \sum_{(j,i)_t \in A^T | t'=t-\tau_{ij}} x_{jit'} = 0 \quad \forall i_t \in N^T \tag{17}$$

$$0 \le x_{ijt} \le u_{ij} \quad \forall (i,j)_t \in A^T \tag{18}$$

$$\text{where } \Gamma = \left\{ \gamma_{ijt} \mid \sum_{(i,j)_t \in A^T} r_{ij} \gamma_{ijt} \le R \right\} \tag{19}$$

The form of the objective function (16) reflects the struggle between the network user who seeks to maximize flow through the network and the network interdictor who, using interdiction resources, seeks to minimize that maximum flow. Flow through the network is represented by the flow across the return arc $(b',a')_T$. The network interdictor removes flow from the network (effectively) by subtracting flow across interdicted arcs. Constraints (17) are the flow balance constraints. (A model variant would allow storage at node i_t by creating "inventory" arcs from i_t to i_{t+1}.) The arc capacity constraints (18) limit the maximum amount of flow entering an arc at time period t to the nominal capacity.

Interdiction resource constraint (19) limits the interdiction effort by the amount of resource available R.

C. A CONSTRAINED MINIMUM CUT MODEL

We convert the inner maximum flow problem of TE-NIM to its dual, the minimum cut model to obtain:

TE-NIM1

Indices:

$i, j \in N$ Nodes of $G = (N, A)$, including two special nodes, a' the super-source, and

             b', the super-sink

t, t' Time periods: $t, t' = 0, 1, 2, \ldots, T$

$(i, j)_t$ Arc of $G = (N, A)$ at time period t

Data:

u_{ij} Nominal capacity of arc $(i, j)_t$

τ_{ij} Traverse time of arc $(i, j)_t$; flow leaving i at time t, arrives at j at time $t + \tau_{ij}$

r_{ij} Amount of resource required to interdict arc $(i, j)_t$

q_{ij} Repair time interval for arc $(i, j)_t$ (begins the period after interdiction) .

R Resource for interdiction available to the network interdictor

Decision variables:

α_{it} $\alpha_{it} = 1$ if $i_t \in N_{b'}^T$, else $\alpha_{it} = 0$

γ_{ijt} $\gamma_{ijt} = 1$ indicates arc $(i, j)_t$ is interdicted at time t; otherwise $\gamma_{ijt} = 0$

The Formulation:

$$z^{TE*} = \min_{\alpha, \gamma, \beta} \sum_{(i,j)_t \in A^T} u_{ij} \beta_{ijt}$$

$$\text{s.t.} \quad \alpha_{it} - \alpha_{jt} + \sum_{t'=t-q_{ij}}^{t} \gamma_{ijt'} + b_{ijt} \geq 0 \; \forall \; (i,j)_t \in A^T - (b',a')_T \quad \text{(20a)}$$

$$\alpha_{b't} - \alpha_{a't} + \gamma_{b'a'T} + \beta_{b'a'T} \geq 1 \quad \text{(20b)}$$

$$\alpha_{it} \in \{0,1\} \; \forall i_t \in N^T$$

$$\beta_{ijt} \in \{0,1\} \; \forall (i,j)_t \in A^T$$

$$\beta_{b'a'T} \equiv 0$$

$$\sum_{(i,j)_t \in A^T} r_{ij} \gamma_{ijt} \leq R$$

$$\gamma_{ijt} \in \{0,1\} \; \forall (i,j)_t \in A^T$$

$$\gamma_{b'a'T} \equiv 0$$

Because there is no advantage gained by the interdictor interdicting more often than necessary, we may assume that $\sum_{t'=t-q_{ij}}^{t} \gamma_{ijt'}$ is 0 or 1 in any optimal solution to TE-NIM1. Thus, we may interpret a solution of this model as in the static model: The solution of this model identifies a cut defined by $\alpha_{it} = 1$ for all $i_t \in N_b^T$ and $\alpha_{it} = 0$ for all $i_t \in N_a^T$. For arcs $(i,j)_t$ not in the cut, $\gamma_{ijt} = \beta_{ijt} = 0$. For arc $(i,j)_t$ in the cut, either $\beta_{ijt} = 1$, indicating that this arc forms part of the minimum capacity cut after interdiction or the arc is part of the *interdicted set* A_I^T where $A_I^T = \{(i,j)_t \in A^T \mid \gamma_{ijt'} = 1 \text{ for } t - q_{ij} \leq t' \leq t\}$. So, the interdicted set identifies arcs and time periods in which those arcs are interdicted or are under repair (and thus out of commission).

Note the distinction between "interdiction set" and "interdicted set" for the dynamic problem: The interdiction set $A_S^T \subseteq A_I^T$ comprises those arcs, $(i,j)_t$ for which $\gamma_{ijt} = 1$ indicating a "strike" or interdiction resource expenditure occurs. Arcs in the

interdicted set with $\gamma_{ijt} = 0$ have previously been interdicted and are still under repair during time period t; they have no capacity, but require no expenditure of interdiction resource.

TE-NIM is clearly an NP-complete problem, and even small problems can be difficult to solve. However, if we could relax the interdiction budget constraint, TE-NIM1 would become easy.

D.   LAGRANGIAN RELAXATION FOR TE-NIM1

We approach Lagrangian relaxation for TE-NIM as we did for S-NIM: We move the budget constraint, with a Lagrangian multiplier, into the objective function. The Lagrangian relaxation model provides a lower bound to z^{TE*} and when we relax the integrality constraints, is a concave function in λ. In the process of finding lower bounds we find feasible and infeasible solutions to TE-NIM1. (A feasible solution is a binary solution for which constraint (19) holds.) Any feasible solution yields an upper bound on z^{TE*}.

Since $z^{TE}(\lambda)$ is a concave function, we look for the best λ using binary search to find the greatest lower bound. The Lagrangian multiplier acts as a penalty adding cost to the minimization. Large values of λ encourage less use of interdiction resources while small values encourage use of more resource to minimize the overall cost. As we adjust λ, we hope to find a solution for which the difference between the upper and lower bounds is small.

Because of the relaxation, $z^{TE}(\lambda) \le z^{TE*}$, but as we adjust λ we may find a Lagrangian multiplier such that $\sum\limits_{(i,j)_t \in A^T} r_{ij}\gamma_{ijt} = R$. Then,

$$z^{TE}(\lambda) = \min_{\alpha,\gamma,\beta} \sum_{(i,j)_t \in A^T} u_{ij}\beta_{ijt} + \lambda(\sum_{(i,j)_t \in A^T} r_{ij}\gamma_{ijt} - R) = \min_{\alpha,\gamma,\beta} \sum_{(i,j)_t \in A^T} u_{ij}\beta_{ijt} = z^{TE*}, \tag{21}$$

30

and the corresponding set of interdiction decisions is both feasible and optimal. In economic terms, at optimality, the value of the Lagrangian multiplier λ is the value of an additional unit of interdiction resource.

The Lagrangian relaxation of TE-NIM1 is:

TE-LR(λ)

$$z^{TE}(\lambda) = \min_{\alpha,\gamma,\beta} \sum_{(i,j)_t \in A^T} (u_{ij}\beta_{ijt} + \lambda r_{ij}\gamma_{ijt}) - \lambda R \tag{22}$$

$$\text{s.t } \alpha_{it} - \alpha_{jt} + \sum_{t'=t-q_{ij}}^{t} \gamma_{ijt'} + \beta_{ijt} \geq 0 \quad \forall (i,j)_t \in A^T - (b',a')_T \tag{23}$$

$$\alpha_{b'T} - \alpha_{a'0} + \gamma_{b'a'T} + \beta_{b'a'T} \geq 1$$

$$\alpha_{it} \text{ free } \forall i_t \in N^T$$

$$\beta_{ijt} \geq 0 \quad \forall (i,j)_t \in A^T$$

$$\beta_{b'a'T} \equiv 0$$

$$\gamma_{ijt} \in \{0,1\} \quad \forall (i,j)_t \in A^T$$

$$\gamma_{b'a'T} \equiv 0$$

If we restrict γ_{ijt} to binary values, TE-LR(λ) will naturally have binary extreme point solutions. We can therefore relax the binary constraints on α_{it} and β_{ijt}. We then relax the binary constraint on γ_{ijt} to $0 \leq \gamma_{ijt} \leq 1$. The relaxation of TE-LR(λ) is:

31

TE-LRR(λ)

$$z^{TE}(\lambda) = \min_{\alpha,\gamma,\beta} \sum_{(i,j)_t \in A^T} (u_{ij}\beta_{ijt} + \lambda r_{ij}\gamma_{ijt}) - \lambda R$$

s.t. $\quad \alpha_{it} - \alpha_{jt} + \sum_{t'=t-q_{ij}}^{t} \gamma_{ijt'} + \beta_{ijt} \geq 0 \quad \forall (i,j)_t \in A^T - (b',a')_T$

$$\alpha_{b't} - \alpha_{a't} + \gamma_{b'a'T} + \beta_{b'a'T} \geq 0$$

$$\alpha_{it} \geq 0 \quad \forall i_t \in N^T$$

$$\alpha_{a'0} \equiv 0, \ \alpha_{b'T} \equiv 1$$

$$\beta_{ijt} \geq 0 \quad \forall (i,j)_t \in A^T$$

$$\beta_{b'a'T} \equiv 0$$

$$0 \leq \gamma_{ijt} \leq 1 \quad \forall (i,j)_t \in A^T$$

$$\gamma_{b'a'T} \equiv 0$$

For a fixed λ:

$$z^{TE}(\lambda) \equiv z'^{TE}(\lambda) - \lambda R \text{ where } z'^{TE}(\lambda) = \min_{\alpha,\gamma,\beta} \sum_{(i,j)_t \in A^T} (u_{ij}\beta_{ijt} + \lambda r_{ij}\gamma_{ijt}).$$

Note that $z^{TE}(\lambda)$ will still yield a valid lower bound on z^{TE*}. Instead of solving TE-LRR(λ) directly, we can solve its dual:

TE-LRR-D1(λ)

$$z^{TE}(\lambda) = \max_{\mathbf{x}} x_{b'a'T} - \lambda R$$

s.t. $\quad \sum_{(i,j)_t \in A^T} x_{ijt} - \sum_{(j,i)_{t'} \in A^T | t'=t-\tau_{ji}} x_{jit'} = 0 \quad \forall i_t \in N^T \quad\quad (24)$

$$0 \leq x_{ijt} \leq u_{ij} \quad \forall (i,j)_t \in A^T \quad\quad (25)$$

$$0 \leq \sum_{t'=t}^{t+q_{ij}} x_{ijt'} \leq \lambda r_{ij} \quad \forall (i,j)_t \in A^T \quad\quad (26)$$

TE-LRR-D1(λ) would be a simple maximum flow problem without constraints (26). We replace these constraints with restrictions of the constraints, specifically

$0 \le x_{ijt} \le \frac{\lambda r_{ij}}{q_{ij}+1}$. The restriction is reasonable since the flow on $(i,j)_t$ falls into one of three categories:

1. $x_{ijt} = 0$ because there is no "useful" path from a source node to a sink node that includes $(i,j)_t$; therefore constraint (26) is slack (a useful path is a path from a source to a sink with flow arriving at a sink before time period $T+1$),

2. arc capacities do not change over time, and thus

3. the same amount of flow will be pushed along in every time period until any path containing (i,j) is no longer useful, and x_{ijt} goes to zero.

Note that a restriction of this model must still lead to a valid lower bound on z^{TE*}.

As in the static problem, the flow on each arc must meet both of the capacity constraints, (25) and (26). We restate model TE-LRR-D1(λ) with the restricted capacity constraints (26) as:

TE-LRR-D2(λ)

$$z^{TE}(\lambda) = \max_{\mathbf{x}} x_{b'a'T} - \lambda R$$

$$\text{s.t.} \sum_{(i,j)_t \in A^T} x_{ijt} - \sum_{(j,i)_{t'} \in A^T | t'=t-\tau_{ji}} x_{jit'} = 0 \quad \forall i_t \in N^T$$

$$0 \le x_{ijt} \le \min\{u_{ij}, \frac{\lambda r_{ij}}{q_{ij}+1}\} \quad \forall (i,j)_t \in A^T \qquad (27)$$

This model is much like the maximum flow model TE-MF with capacities modified by the value of an interdiction. Any solution to TE-LRR-D2(λ) finds a minimum capacity cut $\{N_{a'}^T, N_{b'}^T\}$ that corresponds to a feasible or infeasible solution to the original problem TE-NIM1 as follows:

1. $\alpha_{it} = 1 \ \forall i_t \in N_{b'}^T, \ \alpha_{it} = 1 \ \forall i_t \in N_{a'}^T,$

2. $\gamma_{ijt} = \beta_{ijt} = 0 \ \forall$ arcs $(i,j)_t$ that are not forward arcs in the cut,

3. for $t = 1, \ldots, T$, $i_t \in N_{a'}^T, j_t \in N_{b'}^T$ and $x_{ijt} = \frac{\lambda r_{ij}}{q_{ij}+1}$, for each arc in the interdicted set, γ_{ijt} is defined recursively as $\gamma_{ijt} = 1 - \max\limits_{\max\{0, t-q_{ij}\} \le t' \le t-1} \gamma_{ijt'}$,

4. $\gamma_{ijt} = 0 \ \forall i_t \in N_{a'}^T, j_t \in N_{b'}^T$ and $x_{ijt} < \frac{\lambda r_{ij}}{q_{ij}+1}$, and

5. $\beta_{ijt} = 1$ if $i_t \in N_{a'}^T, j_t \in N_{b'}^T$ and $x_{ijt} = u_{ij}$, i.e., if $(i,j)_t$ is a forward arc of the minimum cut and $u_{ij} < \frac{\lambda r_{ij}}{q_{ij}+1}$, then $(i,j)_t$ is not in the interdicted set.

Note that λ can always be perturbed so that $u_{ij} = \frac{\lambda r_{ij}}{q_{ij}+1}$ does not occur. The solution is feasible if the interdiction budget constraint (19) holds.

E. A METHOD OF SOLVING THE TIME-EXPANDED NIM

As in the static problem, we use a polynomial-time maximum flow algorithm to solve TE-LRR-D2(λ). (See Appendix B.) The algorithm actually runs in pseudo-polynomial-time since the network is represented in time-expanded form. As in the static network interdiction model, arcs in the minimum cut are examined to find the corresponding interdicted set A_I^T that may or may not be feasible. If A_I^T is feasible for Γ, the maximum flow for the time-expanded network after interdiction is an upper bound on the solution to the network interdiction model.

The objective function for model TE-LRR(λ) is a concave function in λ. We find the maximum value of $z^{TE}(\lambda)$ using binary search on λ. Given a starting value for λ, and given an interdiction vector Γ, we decrease λ if $\sum\limits_{(i,j)_t \in A^T} r_{ij} \gamma_{ijt} < R$ encouraging use of more interdiction resource. We increase λ if $\sum\limits_{(i,j)_t \in A^T} r_{ij} \gamma_{ijt} > R$, raising the penalty for exceeding the interdiction budget. In equation (21) for NIM, we know that have found an optimal solution the static problem if $\sum\limits_{(i,j) \in A} r_{ij} \gamma_{ij} = R$ and we stop our search for the best value of λ, λ^*. For the dynamic problem, we cannot make the same claim nor should we stop our

search for $\lambda*$. Because we use a restricted form of constraint (26) in TE-LRR-D2(λ), we may find two interdiction sets, both consuming R units of interdiction resource and one set may be better than the other. For example, suppose solving TE-LRR-D2(λ) for a value of λ identifies interdiction set A, which consumes R units of interdiction resource and contains $(i,j)_t$ and $(i,j)_{t+1}$ with $q_{ij} = 1$. We re-solve TE-LRR-D2(λ) for a value of $\lambda + \varepsilon$ ($\varepsilon > 0$ and small) finding interdiction set B, also consuming R units of interdiction resource and containing $(i,j)_{t+1}$ (and not $(i,j)_t$). If $x_{ij(t+2)} < \frac{\lambda r_{ij}}{q_{ij}+1}$, then interdiction set B is not optimal. It may be that, if we do not restict constraint (26), then any solution that uses exactly R units of interdiction resource, is optimal. We are not able to test this theory in this thesis.

In Chapter II, we introduced a method for discriminating between arcs with equal capacity. This method is particularly useful in the time-expanded network since each arc is represented in multiple time periods. We again add a very small random amount of capacity to each arc. We add a random $(T+1)/1,000,000$ to $2,500(T+1)/1,000,000$ of capacity to each arc. We accomplish this by multiplying all capacities by a scaling factor of 1,000,000 and then add a random amount that varies uniformly from 1 to 2,500 multiplied by the number of time periods plus one.

Additionally, the algorithm may need to discriminate between the same arc in different time periods. After scaling and randomization, we add one unit of capacity for each time period so that each copy of arc (i,j) has an increasing capacity over time. As a result, $u_{ijt} < u_{ij(t+1)}$ and if $(i,j)_t$ is in the interdicted set, then $\frac{\lambda r_{ij}}{q_{ij}+1} < u_{ijt} < u_{ij(t+1)}$ so that $(i,j)_{t+1}$ is also in the interdicted set. However, since $u_{ij(t-1)} < u_{ijt}$, then we may have $u_{ij(t-1)} < \frac{\lambda r_{ij}}{q_{ij}+1} < u_{ijt}$ and arc $(i,j)_{t-1}$ would not be in the interdicted set.

Because of these small amounts of additional capacity for each arc, the algorithm is able to differentiate between arcs that would otherwise appear identical. These perturbations cause only negligible changes in the maximum flow in the network.

35

We use the same modified shortest augmenting-path algorithm (Edmonds and Karp, 1972) to solve LRR-D2(λ) as we use for solving TE- LRR-D2(λ). We implement the Lagrangian relaxation and solve a maximum flow problem to identify a minimum cut A_C^T and a set of interdicted arcs $A_I^T \subseteq A_C^T$. An interdicted set may contain the same arc in multiple time periods. We therefore screen the interdicted set to find the interdiction set by selecting arcs that would not be under repair from an interdiction in a previous time period. For example, suppose arc $(i,j)_t$ with a repair time of two time periods is interdicted at time $t = 21$. If arcs $(i,j)_{22}$, $(i,j)_{23}$ and $(i,j)_{24}$ also appear in the interdicted set, then arcs $(i,j)_{22}$ and $(i,j)_{23}$ are not part of the interdiction set since they are under repair. Arcs $(i,j)_{21}$ and $(i,j)_{24}$ are in the interdiction set consuming $r_{ij} + r_{ij}$ units of interdiction resource. After finding the interdiction set, the algorithm adds the amount of interdiction resource required for the current interdiction set and adjusts the Lagrangian multiplier accordingly. Uninterdicted flow is the sum of the flow on arcs in the cut that are not in the interdicted set.

IV. COMPUTATIONAL RESULTS

Chapters II and III show how to use Lagrangian relaxation to approximately solve static and dynamic network interdiction problems. The algorithms that implement the methods are described in Appendices A and B. In this chapter, we provide computational results for these algorithms. We describe the networks tested in Section A, present the results for the static problem in Section B and the results for the dynamic problem in Section C.

A. TEST NETWORK DESIGN

We do not attempt to model an actual transportation network in this thesis since our purpose is simply to test the proposed formulations and methods. We generate several grid networks of various sizes with random integer arc data r_{ij} and u_{ij} for the static networks and, r_{ij}, u_{ij}, τ_{ij} and q_{ij} for the dynamic networks. We use single- and multiple-source networks and single- and multiple-sink networks. We include arcs that are uninterdictable and some arcs with $\tau_{ij} = 0$ in the dynamic networks. In general, arcs immediately adjacent to the source (or super-source) and sink (or super-sink) are given a large capacity and large r_{ij} so that a trivial interdiction set adjacent to these nodes is not optimal.

As stated in the appendices, we employ a maximum flow algorithm that uses a breadth-first-search labeling method to find a shortest augmenting path from a source to a sink. The test programs implement the Static Network Interdiction Heuristic of Appendix A which solves LRR-D2(λ) (the dual of the Lagrangian relaxation of NIM-1), and implement the Dynamic Network Interdiction Heuristic of Appendix B which solves TE-LRR-D2(λ) (the dual of the Lagrangian relaxation of TE-NIM1).

We study six test networks, called SNET25, DNET25, SNET100, DNET100, SNET400, and DNET400. A general description of each test network follows: SNET25

has 25 nodes and 64 arcs with artificial arcs from a super-source to five sources and artificial arcs from four sinks to a super-sink. We include 11 arcs that are uninterdictable. DNET25 is a dynamic network that shares the same arc attributes as the static test network, SNET25, plus the time attributes. DNET25, in time expanded form, has 525 nodes with 1,344 arcs for $T = 20$, and 1,275 nodes with 3,264 arcs for $T = 50$. It also has artificial arcs from a super-source to five sources and artificial arcs from four sinks to a super-sink for each time period.

SNET100 is based on a 10-node by 10-node grid and has 100 nodes and 359 arcs. A super-source is connected to five sources in each network and a super-sink is connected to five sinks in each network. Only these artificial arcs are uninterdictable. DNET100 has the same structure as SNET100, but in time-expanded form. It has 5,100 nodes with 17,901 arcs for $T = 50$, and 8,100 nodes with 28,431 arcs for $T = 80$

SNET400 is based on a 20-node by 20-node grid and has 400 nodes and 1519 arcs. A super-source is connected to eleven sources in each network and a super-sink is connected to ten sinks in each network. These artificial arcs are uninterdictable, as are several arcs that we arbitrarily selected in the center of the grid; perhaps these arcs are uninterdictable for political reasons. DNET400 has the same structure as SNET400, but in time-expanded form. It has 32,400 nodes with 123,039 arcs for $T = 80$, and 40,400 nodes with 153,419 arcs for $T = 100$

B. STATIC NETWORKS

The results of testing static networks SNET25, SNET100 and SNET400 are listed in Tables 1 and 2. Testing indicates that the Static Network Interdiction Heuristic (Appendix A) often finds an optimal or near-optimal solution, but sometimes fails dramatically. Data sets with $r_{ij} = 1$ are usually solved optimally.

The best upper bound, $\bar{z}\,^*$, on a solution to the network interdiction problem is the maximum flow in the network after the best (feasible) interdiction set is applied. By "best" we mean the smallest observed maximum flow over all feasible solutions obtained

by the algorithm. The best lower bound, $z(\lambda^*)$, is the value of $z(\lambda)$, computed from LRR-D2(λ), maximized over all values of λ. (Actually, the lower bound is slightly pessimistic because we stop the algorithm when the interval of uncertainty on λ^* is less than or equal to 1.) The optimality gap, both absolute and relative measures the quality of the best solution found. The absolute optimality gap is computed as $\bar{z}^* - z(\lambda^*)$ and the

relative optimality gap is $\dfrac{100\%(\bar{z}^* - z(\lambda^*))}{z(\lambda^*)}$. The network interdictor would probably be

more interested in knowing the quality of the solution in terms of interdicted flow. The percentage of maximally interdicted flow achieved is a percentage of the "worst case"

interdiction divided by the "best case" interdiction $\dfrac{100\%(z_0^* - \bar{z}^*)}{(z_0^* - z(\lambda^*))}$ where z_0^* is the value

of the maximum uninterdicted network flow. The "worst case" is the best known feasible solution that interdicts $(z_0^* - \bar{z}^*)$ units of flow. The "best case" is an unknown solution that may interdict as much as $(z_0^* - z(\lambda^*))$ units of flow.

It is interesting to see how solution quality varies as a function of the arc parameters and the amount of interdiction resource applied. Table 2 gives the results of sensitivity testing using SNET400 as the test network and varying R and the parameters r_{ij} and u_{ij}.

For results A, we vary r_{ij} and R such that R is six times the midpoint of the r_{ij} interval. Solution quality in terms of either the optimality gap or the percentage of maximally interdicted flow shows no relationship with increasing the interval width on r_{ij}. While the algorithm seeks the best interdiction set for a given amount of interdiction resource, we find that in some cases, generally where the quality of the solution is poor, decreasing R slightly and re-solving the problem provides a result that interdicts the same amount of flow. For example, comparing SNET400-1 (the first result for SNET400) in Table 1, with result A-1 in Table 2, we see that the upper bounds are the same. Result A-1 is just as effective at interdicting the network flows with $R = 6$ as the Table 1 result

fewer units of interdiction resource. When the relative optimality gap and the percent of maximally interdicted flow indicate a poor quality solution, we recommend that the network interdictor vary the amount of interdiction resource available and re-solve the problem.

For results B, we vary the interval width on the uniformly distributed capacities u_{ij}. Results B-1 and B-2 have the same network structure and interval width. The capacities are scalar multiples of each other and, as expected, the bounds and network flows are also scalar multiples. Results B-1 through B-6 are optimal for SNET400 with six units of interdiction resource allowed. We use the same random number seed to generate the random arc capacities for each test network and, as a result, find the same interdiction set for each solution.

Network		u_{ij}	r_{ij}	R	Flow w/ Inter.	$z(\lambda^*)$	Abs. Gap	Rel. Gap	% Max Inter.	z_0^*	Run Time
SNET25	1	[4,18]	[1,1]	3	59	59	0	0.0	100.0	89	0.05
	2	[4,18]	[1,8]	3	66	66	0	0.0	100.0	89	0.03
	3	[4,18]	[1,1]	8	13	13	0	0.0	100.0	89	0.03
	4	[4,18]	[1,8]	8	48	44	4	9.1	91.1	89	0.06
SNET100	1	[0,96]	[1,1]	6	81	81	0	0.0	100.0	333	0.18
	2	[0,96]	[1,10]	6	243	209	34	16.3	72.6	333	1.24
	3	[0,96]	[1,1]	10	15	15	0	0.0	100.0	333	0.20
	4	[0,96]	[1,10]	10	175	148	27	17.6	85.4	333	1.20
SNET400	1	[0,96]	[1,1]	8	586	323	263	81.4	42.9	785	31.00
	2	[0,96]	[1,10]	8	621	479	142	29.5	53.6	785	25.13
	3	[0,96]	[1,1]	15	78	78	0	0.0	100.0	791	5.63
	4	[0,96]	[1,10]	15	352	352	0	0.0	100.0	791	9.63

Legend:

[a,b]: indicates the parameter is a random integer, distributed uniformly from a to b

u_{ij}: arc capacities

r_{ij}: resource required to interdict an arc

R: interdiction budget

Flow w/ Inter.: (or $\bar{z}^*$) maximum flow through the test network after the best interdiction set is applied,

$z(\lambda^*)$: lower bound of the network interdiction problem

Abs. Gap: $\bar{z}^* - z(\lambda^*)$

Rel. Gap: $100\% (\bar{z}^* - z(\lambda^*))/z(\lambda^*)$

%Max Inter.: $100\% (z_0^* - \bar{z}^*)/(z_0^* - z(\lambda^*))$

z_0^*: maximum flow through the test network before interdiction

Time: time to solve the test network in CPU seconds on an IBM RS/6000 Model 590

Table 1. Static Network Test Results

Network: SNET400		u_{ij}	r_{ij}	R	Flow w/ Inter.	$z(\lambda^*)$	Abs. Gap	Rel. Gap	% Max Inter.	z_0^*	Run Time
A	1	[0,96]	[1,1]	6	587	428	159	36.9	65.5	785	31.5
	2	[0,96]	[1,2]	9	330	330	0	0.0	100.0	785	5.0
	3	[0,96]	[1,3]	12	339	318	21	6.8	95.3	785	30.0
	4	[0,96]	[1,7]	24	206	206	0	0.0	100.0	785	8.2
	5	[0,96]	[1,15]	48	161	161	0	0.0	100.0	785	7.1
B	1	[1,1]	[1,1]	6	33	33	0	0.0	100.0	39	44.8
	2	[50,50]	[1,1]	6	1650	1650	0	0.0	100.0	1950	44.3
	3	[40,60]	[1,1]	6	1464	1464	0	0.0	100.0	1805	19.5
	4	[20,80]	[1,1]	6	1107	1107	0	0.0	100.0	1524	12.0
	5	[10,90]	[1,1]	6	926	926	0	0.0	100.0	1368	15.8
	6	[1,99]	[1,1]	6	754	754	0	0.0	100.0	1227	15.4

Legend:
See Table 1.

Table 2. Sensitivity Testing for a Static Network

C.   DYNAMIC NETWORKS

As described in Section A, the dynamic test networks have the same u_{ij} and r_{ij} as the corresponding static test networks in Section B with the additional arc attributes of τ_{ij} and q_{ij}. The results of testing the dynamic networks are listed in Tables 3 and 4. Testing indicates that the Dynamic Network Interdiction Hueristic (Appendix B) finds results with a relative optimality gap of less than 15% in 13 of 24 test networks but sometimes fails to find an answer with a relative optimality gap of less than 100%.

The long run times associated with the dynamic networks are due to the exponential (pseudo-polynomial) increase in size of the network and a similar increase in the number of paths to the sinks. The results show that run times of the time-expanded problems depend strongly on the time horizon, T. We are unable to test a network with 10,000 arcs because of excessive run time.

The best upper bound, $\bar{z}^{TE*}$, on a solution to the time-expanded network interdiction problem is the maximum flow in the network after the best (feasible) interdiction set is applied. As in the static problem, by "best" we mean the smallest observed maximum flow over all feasible solutions obtained by the algorithm. The best lower bound, $z^{TE}(\lambda*)$, is the value of $z^{TE}(\lambda)$, computed from TE-LRR-D2(λ), maximized over all values of λ. We again use optimality gap, both absolute and relative, to measure the quality of the best solution found.

The results for the dynamic networks show a general decline in solution quality for the larger networks. While static networks with $r_{ij} = 1$ are often solved optimally, dynamic networks with $r_{ij} = 1$ are not usually solved optimally and consistently yield results that are worse than networks with r_{ij} distributed uniformly on [1,10]. While we are unable to find the reason for the declining solution quality, sensitivity testing, described next, gives us some insight into possible causes.

Network		u_{ij}	r_{ij}	τ_{ij}	q_{ij}	R	T	$\bar{z}^{TE*}$	$z^{TE}(\lambda^*)$	Abs. Gap	Rel. Gap	%Max Inter.	z_0^*	Run Time
DNET25	1	[4,18]	[1,1]	[1,10]	[1,4]	8	20	977	923	54	5.7	83.5	1251	1.5
	2	[4,18]	[1,8]	[1,10]	[1,4]	8	20	996	963	33	3.4	88.5	1251	1.5
	3	[4,18]	[1,1]	[1,10]	[1,4]	15	20	835	667	168	25.0	71.2	1251	1.5
	4	[4,18]	[1,8]	[1,10]	[1,4]	15	20	934	870	63	7.2	83.2	1251	1.3
	5	[4,18]	[1,1]	[1,10]	[1,4]	8	50	3706	3677	29	0.8	91.0	3999	4.2
	6	[4,18]	[1,8]	[1,10]	[1,4]	8	50	3685	3685	0	0.0	100.0	3999	3.8
	7	[4,18]	[1,1]	[1,10]	[1,4]	15	50	3468	3408	60	1.7	89.8	3999	4.3
	8	[4,18]	[1,8]	[1,10]	[1,4]	15	50	3449	3419	30	0.8	94.8	3999	3.9

Legend:

$[a,b]$: indicates the parameter is a random integer, distributed uniformly from a to b

u_{ij}: arc capacities

r_{ij}: resource required to interdict an arc

τ_{ij}: arc traversal time

q_{ij}: arc repair time

R: interdiction budget

T: time horizon

$\bar{z}^{TE*}$: upper bound of the network interdiction problem

$z^{TE}(\lambda^*)$: lower bound of the network interdiction problem

Abs. Gap: $\bar{z}^{TE*} - z^{TE}(\lambda^*)$

Rel. Gap: $100\%\,(\bar{z}^{TE*} - z^{TE}(\lambda^*))/z^{TE}(\lambda^*)$

%Max Inter: $100\%\,(z_0^* - \bar{z}^{TE*})/(z_0^* - z^{TE}(\lambda^*))$

z_0^*: maximum flow through the test network before interdiction

Time: time to solve the test network in CPU seconds on an IBM RS/6000 Model 590.

Table 3a. Dynamic Network Test Results for DNET25

44

Network:		u_{ij}	r_{ij}	τ_{ij}	q_{ij}	R	T	$\bar{z}^{TE*}$	$z^{TE}(\lambda*)$	Abs. Gap	Rel. Gap	%Max Inter.	z_0*	Run Time
DNET100	1	[0,96]	[1,1]	[1,10]	[1,10]	8	50	2635	1682	953	56.7	56.3	3862	41.5
	2	[0,96]	[1,10]	[1,10]	[1,10]	8	50	3248	3016	232	7.7	72.6	3862	42.5
	3	[0,96]	[1,1]	[1,10]	[1,10]	15	50	1755	698	1057	151.4	66.6	3862	47.7
	4	[0,96]	[1,10]	[1,10]	[1,10]	15	50	2991	2488	502	20.1	63.4	3862	45.4
	5	[0,96]	[1,1]	[1,10]	[1,10]	8	80	10842	10006	836	8.4	75.5	13414	99.1
	6	[0,96]	[1,10]	[1,10]	[1,10]	8	80	13026	12424	602	4.8	39.2	13414	104.5
	7	[0,96]	[1,1]	[1,10]	[1,10]	15	80	9277	7867	1410	17.9	74.6	13414	111.5
	8	[0,96]	[1,10]	[1,10]	[1,10]	15	80	12184	11687	497	4.3	71.2	13414	104.0
DNET400	1	[0,96]	[1,1]	[1,10]	[1,10]	5	80	2132	771	1361	176.5	25.2	2591	574.5
	2	[0,96]	[1,10]	[1,10]	[1,10]	5	80	2273	1953	320	16.4	49.8	2591	523.8
	3	[0,96]	[1,1]	[1,10]	[1,10]	8	80	1814	626	1188	189.8	39.5	2591	518.8
	4	[0,96]	[1,10]	[1,10]	[1,10]	8	80	2273	1676	597	35.6	34.7	2591	520.6
	5	[0,96]	[1,1]	[1,10]	[1,10]	5	100	9440	7911	1529	19.3	34.3	10239	1784.5
	6	[0,96]	[1,10]	[1,10]	[1,10]	5	100	9385	8899	485	5.5	63.7	10239	1767.3
	7	[0,96]	[1,1]	[1,10]	[1,10]	8	100	8390	6848	1542	22.5	54.5	10239	1729.5
	8	[0,96]	[1,10]	[1,10]	[1,10]	8	100	9176	8382	693	8.3	62.7	10317	1748.4

Legend:
   See Legend in Table 3a.

Table 3b. Dynamic Network Test Results for DNET100 and DNET400

Table 4 gives the results of sensitivity testing using DNET400 as the test network and varying the parameters u_{ij}, r_{ij}, τ_{ij} and q_{ij}. The arc attributes tested are uniformly distributed on the interval indicated and the random number seed is the same for each network tested. The results show that solution time and the quality of the solution are insensitive to u_{ij} and τ_{ij}.

For results C, we vary r_{ij} and R such that R is six times the midpoint of the r_{ij} interval. The solution quality for each set is very poor in terms of both relative optimality gap and percent of maximally interdicted flow. There seems to be a strong relationship between increasing interval width on r_{ij} (with increasing R) and decreasing solution quality.

We observe several interesting interactions for this data set. As R increases, λ^*, which is roughly the value of an additional unit of interdiction resource, decreases. Both the upper and lower bounds are decreasing for increasing interval width on r_{ij} and increasing R, meaning more flow is interdicted. As the optimality gap decreases, the relative optimality gap balloons since the denominator, the lower bound gets smaller. We include another measure of the quality of the solution for this purpose. The percent of maximally interdicted flow remains around 50% for results C-2 through C-5.

It is interesting to compare Table 4, result C-1, with $R = 6$, to Table 3, DNET400-1 with $R = 5$. The Table 4, C-1 result has a solution that removes 546 additional units of capacity from the network by expending six units of interdiction resource versus five units for the Table 3, DNET400-1 result. With $R = 6$, the lower bound is less than with $R = 5$; this indicates a potential for an even better solution when $R = 6$. Comparing the large relative optimality gaps of 392.6% for Table 4 C-1 and 181.8% for Table 3, DNET400-1, we are less sure of the quality of the known result for $R = 6$.

The optimality gaps for Table 4 C-3, C-4 and C-5 are extremely large and out of proportion to the percent of maximally interdicted flow. The rapidly degrading quality of the solutions leads us to investigate better solutions such as adjusting R slightly. Testing the network C-4 with $R = 22$ instead of $R = 24$, we find a solution that has the same upper bound. To compare the solutions we use the percent of maximally interdicted flow, 52.5% for $R = 22$ compared to 49.6% for $R = 24$, a small improvement. This leads us to recommend that the network interdictor use the percent of maximally interdicted flow as an indicator of whether it would be useful to look for other solutions

46

by varying the amount of interdiction resource available by one or two units and re-solving the problem.

For results D in Table 4, we vary the interval width on the uniformly distributed capacities u_{ij}. The first two results have the same structure with arc capacities that are approximate scalar multiples. (Recall that we increase the capacity of each arc one unit per time period after scaling.) As expected, the interdiction sets are the same and the bounds with for D-2 are approximately 50 times those of D-1. In the static network sensitivity testing, the interdiction set was consistent across the six test networks. The dynamic test networks have interdiction sets that change for each change in capacity interval width.

For results E, we vary the interval width on the uniformly distributed arc traversal time, τ_{ij}. There is no observable relationship between the interval width and solution quality. As expected, the amount of flow through the network decreases as the average traversal times increase.

For results F, we vary the interval width on the uniformly distributed repair times q_{ij}. There is strong relationship between this interval width and the quality of the solution both in terms of the relative optimality gap and the percent of maximally interdicted flow: Solution quality declines as we increase the interval width. Result F-1 assumes repair occurs immediately after interdiction, i.e., $q_{ij} = 0$ and we find an optimal solution.

Network: DNET400		u_{ij}	r_{ij}	τ_{ij}	q_{ij}	R	T	$\bar{z}^{TE*}$	$z^{TE}(\lambda^*)$	Abs. Gap	Rel. Gap	%Max Inter.	z_o^*	Run Time
C	1	[0,96]	[1,1]	[1,10]	[1,10]	6	80	2041	433	1607	371.1	25.5	2591	553.0
	2	[0,96]	[1,2]	[1,10]	[1,10]	9	80	1713	701	1012	144.4	46.5	2591	604.6
	3	[0,96]	[1,3]	[1,10]	[1,10]	12	80	1411	418	993	237.6	54.3	2591	658.8
	4	[0,96]	[1,7]	[1,10]	[1,10]	24	80	1370	127	1243	978.7	49.6	2591	598.4
	5	[0,96]	[1,15]	[1,10]	[1,10]	48	80	1116	64	1052	1643.8	58.4	2591	583.1
D	1	[1,1]	[1,1]	[1,10]	[1,10]	6	80	100	70	30	42.9	38.8	119	741.4
	2	[50,50]	[1,1]	[1,10]	[1,10]	6	80	7114	5086	2027	39.9	30.7	8012	1091.9
	3	[40,60]	[1,1]	[1,10]	[1,10]	6	80	6560	4700	1860	39.6	33.7	7505	348.0
	4	[20,80]	[1,1]	[1,10]	[1,10]	6	80	5381	3751	1630	30.3	38.0	6379	234.6
	5	[10,90]	[1,1]	[1,10]	[1,10]	6	80	4606	3033	1573	51.9	37.3	5543	250.0
	6	[1,99]	[1,1]	[1,10]	[1,10]	6	80	3742	2242	1500	66.9	34.9	4581	271.7
E	1	[0,96]	[1,1]	[1,1]	[1,10]	6	30	8837	6119	2718	44.4	12.8	9235	399.9
	2	[0,96]	[1,1]	[0,2]	[1,10]	6	30	12472	9377	3095	33.0	14.1	12979	1557.9
	3	[0,96]	[1,1]	[1,2]	[1,10]	6	30	3375	337	3038	901.5	16.4	3972	365.9
	4	[0,96]	[1,1]	[1,3]	[1,10]	6	40	5251	2928	2323	79.3	15.5	5678	246.1
	5	[0,96]	[1,1]	[1,5]	[1,10]	6	60	8466	5818	2658	45.5	17.8	9038	284.8
F	1	[0,96]	[1,1]	[1,10]	[0,0]	6	80	2195	2195	0	0.0	100.0	2587	191.2
	2	[0,96]	[1,1]	[1,10]	[0,1]	6	80	2144	1956	188	9.6	70.2	2587	347.2
	3	[0,96]	[1,1]	[1,10]	[0,2]	6	80	2003	1702	301	17.7	66.0	2587	594.4
	4	[0,96]	[1,1]	[1,10]	[0,4]	6	80	2053	1383	670	48.4	44.3	2587	207.2
	5	[0,96]	[1,1]	[1,10]	[0,10]	6	80	2119	430	1689	392.6	21.7	2587	217.7

Legend:
  See Table3a.

Table 4. Sensitivity analysis for a Dynamic Network.

The poor quality of our results for the dynamic networks seems to be caused largely by including arc repair time in the model. However, including arc repair time is one of the major motivations for exploring dynamic networks in this thesis. There may be methods for improving the bounds and the quality of the results that take into account the width of the interval for q_{ij}. Due to time constraints, we cannot explore such possibilities in this thesis.

Our testing found several test networks, both static and dynamic, where the algorithm never finds a solution that uses all R units of interdiction resource. For $\lambda = \lambda * - \varepsilon$ ($\varepsilon > 0$ and small) we find an interdiction set that is infeasible while for $\lambda = \lambda * + \varepsilon$ we find an interdiction set that is feasible but does not use all R units of interdiction resource. There should be a solution that consumes all R units of resource for networks with $r_{ij} = 1$. However, we have been unable to find this solution using the methods proposed in this thesis.

We use a modified shortest augmenting path maximum flow algorithm by Edmonds and Karp (1972) which runs in $O(nm^2)$ time for a network with n nodes and m arcs. The run times can probably be improved significantly by using a faster pre-flow push maximum flow algorithm such as the excess scaling algorithm which runs in $O(nm + n^2 \log U)$ time (e.g., Ahuja, et al., 1993, p. 239).

V. CONCLUSION

This thesis presents optimization-based heuristic methods to solve two forms of a network interdiction problem, a "static network interdiction problem" and a "dynamic time-expanded network interdiction problem." While static network interdiction models have been studied before, we develop a new heuristic method that provides a good feasible solution and upper and lower bounds on the optimal solution value. We also develop heuristic methods to find a good feasible solution to a time-expanded dynamic network interdiction model with bounds on the optimal solution value. The dynamic model allows us to consider arc traversal times, repair of interdicted arcs and time-weighted flow — the weight is a 0 or 1 depending on the time that the flow arrives at the sink.

We model network interdiction problems as min-max models where the network user maximizes flow through a capacitated network while the network interdictor minimizes that maximum flow by interdicting (destroying) arcs using limited assets.

Both static and dynamic forms of the problem can be formulated as constrained minimum cut models that are difficult to solve. An interdiction budget constraint complicates the problem. By relaxing this constraint, we are able to use a sequence of maximum flow problems to approximately solve the original problem. For both forms of the problem, we use Lagrangian relaxation to find a lower bound on the optimal solution value. In the process of maximizing the lower bound, we find feasible solutions with corresponding upper bounds. The difference between the upper and lower bounds, the optimality gap, indicates the quality of the solution. We search for the best lower and upper bounds and hope that the difference is small.

The Lagrangian relaxation procedure for both the static and dynamic problems can have difficulty finding an optimal solution when many arcs have the same capacity. In particular, the solution methodology may interdict all of the arcs in a cut with the same capacity or none of them. To avoid this, all arc capacities are randomized by small

51

amounts. As a result the algorithm is able to differentiate between arcs in a cut, yet the optimal solution value changes only negligibly. Additionally, the dynamic network interdiction problem requires that we have a method to differentiate between copies of an arc in multiple time periods. We accomplish this by adding a small increment of capacity for each time period so that arc capacity $u_{ijt} < u_{ij(t+1)}$ for all arcs (i,j) and time periods t.

Testing with several grid networks shows that 7 of 12 instances of the static network interdiction problem are solved optimally. Four more solutions have relative optimality gaps that are less than 30%. The remaining solution has a relative optimality gap of 81.5%. The quality of this last solution is poor since only a part of the available interdiction resource is used. The result can probably be improved by further perturbing the arc capacities so that more of the interdiction resource is expended.

The dynamic networks in time-expanded form are 20 to 100 times larger than the static networks. This results in longer computation time and larger optimality gaps than seen for the static networks. Specifically, 13 of 24 solutions have relative optimality gaps that are less than 15% and six more are between 15% and 30%. The remaining five solutions are of poor quality with relative optimality gaps between 30% and 195%. Closing these optimality gaps will require further research.

We have several suggestions for further work and possible model improvements for the dynamic network interdiction problem.

1. We do not model arbitrary time-weighted flow since the decomposition methods that are probably required to solve such a problem are beyond the scope of this thesis. However, military engagements often last for long periods of time and there may be a need to assign one weight for flows arriving before a battle, another weight for replenishing expended wartime commodities such as ordnance and fuel during the battle, and another weight for flow arriving after the battle. The Bender's decomposition

52

method described by Cormican (1995) for the static network interdiction problem would probably be applicable to this problem.

2. The restriction step that converts constraints (26) to constraints (27) reduces the quality of the lower bounds obtained. It may be possible to avoid this by:

    (a) reformulating the resource constraint,

    $$\sum_{(i,j)_t \in A^T} r_{ij} \gamma_{ijt} \leq R \tag{19}$$

    into multiple constraints:

    $$\sum_{(i,j) \in A} r_{ij} \gamma_{ijt} - \delta_t = 0 \quad \text{for } t = 1, \ldots, T \tag{19a}$$

    $$\sum_{t=1}^{T} \delta_t \leq R \tag{19b}$$

    and

    (b) using Lagrangian relaxation on (19a) and (19b) with separate Lagrangian multipliers for each relaxed constraint.

    This would entail more computational effort, but the improved solution quality might be worth the effort.

3. There is an assumption that the capacity and traversal time of each arc in a network are known and fixed. In fact, environmental effects and congestion may add a stochastic element to the arc capacities and traversal times. The dynamic network interdiction model could be improved by using stochastic arc capacities (Cormican, Morton and Wood, 1996) and stochastic traversal times. These extensions would be difficult, however.

4. It is assumed in our model that the repair time of an arc is fixed and that an interdicted arc has no capacity until repair is complete. While the "all or nothing" arc capacity may be valid from some arcs such as a bridge crossing a ravine, when the interdiction on an arc is in the form of an inspection or blockade, the effect of the interdiction will tend to degrade

53

with time. Interdictions could be modeled with a decreasing effectiveness as time passes.

APPENDIX A. HEURISTIC ALGORITHM FOR THE STATIC NIP

The heuristic algorithm described here finds a feasible solution and bounds for the static network interdiction problem. Essentially, it maximizes the lower bound, $z(\lambda)$, the value of the Lagrangian relaxation, LRR-D2(λ) but it also identifies feasible solutions to the network interdiction problem and reports the best lower and upper bounds and the best set of arcs to interdict (the interdiction set). The upper bound corresponds to the maximum flow through the network when a feasible interdiction set is found.

Static Network Interdiction Heuristic

This heuristic seeks λ^*, the best value of the Lagrangian multiplier, by conducting a binary search on the interval of uncertainty for λ. The heuristic solves a maximum flow problem for each value of λ using the results to adjust λ and the endpoints of the interval. The initial left endpoint of the interval of uncertainty is slightly smaller than the smallest arc capacity, and the right endpoint is slightly larger than the largest arc capacity. λ is initially set to the value of the right endpoint. Inside a do-while loop, the heuristic defines arc capacities as min$\{ u_{ij}, \lambda r_{ij} \}$. The arc capacities and the network graph are inputs to the procedure find_max_flow, which solves the maximum flow problem, identifying a cut A_C, arc flows and the maximum flow through the network x_{ba}. The cut, arc flows, and original arc capacities are passed to the procedure find_interdiction_set. This procedure interprets the results from the maximum flow procedure returning the interdiction set, a potential upper bound and the amount of interdiction resource consumed by the interdiction set. If a feasible solution is found, the heuristic compares the incumbent upper and lower bounds with the current bounds keeping the better values. The heuristic also stores the best interdiction set. The procedure adjust_lambda takes λ and the endpoints of the interval of uncertainty, the interdiction resource consumed by the current solution, and

the amount of interdiction resource available as input, and returns a new value for λ with adjusted endpoints for the new interval of uncertainty. The heuristic re-starts the sequence with a new value of λ until either an optimal solution is found (the interdiction set expends exactly R units of interdiction resource) or the endpoints of the interval of uncertainty for λ converge to a value within one unit of λ^*. The heuristic prints the best interdiction set found, the upper and lower bounds, and the amount of interdiction resource consumed by the interdiction set.

procedure BOUND_THE_STATIC_PROBLEM

Input: Network $G = (N, A)$ with source or super-source, $a \in N$ and sink or super-sink identified, $b \in N$,

$\mathbf{u}$, integer arc capacities $u_{ij} \geq 0 \ \ \forall (i, j) \in A$,

$\mathbf{r}$, integer interdiction resource requirements $r_{ij} \geq 0 \ \ \forall (i, j) \in A$, and

R, total interdiction resource available.

Output: The best feasible interdiction set found A_I,

UB, an upper bound for the network interdiction problem,

LB, a lower bound for the network interdiction problem,

$\hat{R}$, the amount of interdiction resource required for A_I.

Begin {

$LB \leftarrow -\infty$;

$UB \leftarrow +\infty$;

$A_I \leftarrow \varnothing$;

/* find the initial endpoints for the interval of uncertainty for λ */

$\lambda_{\max} \leftarrow \max_{(i,j) \in A-(b,a)} u_{ij} + 1$;

$\lambda_{\min} \leftarrow \min_{(i,j) \in A-(b,a)} u_{ij} - 1$;

56

$\lambda \leftarrow \lambda_{\max}$;

Do{

$\qquad \hat{\mathbf{x}} \leftarrow \mathbf{0}$;   /* reset all network flows */

$\qquad \mathbf{u'} \leftarrow \min\{u_{ij}, \lambda r_{ij}\}$ for all $(i, j) \in A$;   /* reset the adjusted arc capacities*/

$\qquad (A_C, \hat{\mathbf{x}}, x_{ba}) \leftarrow$ find_max_flow(G, a, b, $\mathbf{u'}$);

$\qquad (\hat{A}_I, UB', \hat{R}) \leftarrow$ find_interdiction_set($A_C, \hat{\mathbf{x}}, \mathbf{u}$);

$\qquad$ if $(\hat{R} \le R)$ {   /* a feasible solution has been found*/

$\qquad\qquad LB \leftarrow \max\{$ LB, $x_{ba} - \lambda R\}$;

$\qquad\qquad$ if $(UB' < UB)$ {

$\qquad\qquad\qquad UB \leftarrow UB'$;

$\qquad\qquad\qquad A_I \leftarrow \hat{A}_I$;

$\qquad\qquad$ }

$\qquad$ }

$\qquad (\lambda, \lambda_{\min}, \lambda_{\max}) \leftarrow$ adjust_lambda $(\lambda, \lambda_{\min}, \lambda_{\max}, \hat{R}, R)$;

$\quad$ } while $(\lambda_{\max} - \lambda_{\min} > 1$ and $\hat{R} \ne R)$

$\quad$ print($A_I, UB, LB, \hat{R}$);

} End;

The procedure find_max_flow finds the standard maximum flow in the directed network G with source a, sink b and arc capacities $\mathbf{u'}$.

procedure find_max_flow(G, a, b, $\mathbf{u'}$)

Input: Network graph $G = (N, A)$ with source or super-source, $a \in N$ and sink or super-sink identified, $b \in N$,

$\qquad$ $\mathbf{u'}$, integer arc capacities $u_{ij} \ge 0$ $\forall (i, j) \in A$.

Output: A_C, a minimum capacity cut $A_C \subset A$,

$\hat{\mathbf{x}}$, vector of maximum arc flows $x_{ij} \geq 0 \quad \forall (i,j) \in A$,

x_{ba}, maximum flow value.

{ This procedure uses a standard shortest-augmenting path maximum flow algorithm (Edmonds and Karp, 1972) that is modified to find the maximum residual capacity among all the "shortest paths." Shortest path means the path with the minimum number of arcs.

return (A_C, $\hat{\mathbf{x}}$, x_{ba});

} End;

The procedure find_interdiction_set takes the cut, network arc flows, and arc capacities as input. It identifies and returns an interdiction set, an upper bound, and the amount of resource consumed.

procedure find_interdiction_set (A_C, $\hat{\mathbf{x}}$, $\mathbf{u}$)

Input: A_C, minimum capacity cut in G,

       $\hat{\mathbf{x}}$, vector of maximum arc flows, and

       $\mathbf{u}$, original capacity of each arc in the network.

Output: $\hat{A}_I$, an interdiction set,

       UB, a potential upper bound on the optimal solution of the network interdiction problem,

       $\hat{R}$, the amount of resource consumed by the interdiction set.

Begin {

    $\hat{A}_I \leftarrow \varnothing$;

    $UB' \leftarrow 0$;

    for (each $(i,j) \in A_C$) {

       if ($x_{ij} = u_{ij}$)

/*upper bound is the capacity of the minimum cut after interdiction*/

$UB' \leftarrow UB'+u_{ij}$;

else{

add (i,j) to $\hat{A}_I$;

$\hat{R} \leftarrow \hat{R}+r_{ij}$;

}

}

return $(\hat{A}_I, UB', \hat{R})$;

}End;

The procedure adjust_lambda returns a new value for λ and adjusted endpoints for the new interval of uncertainty. We look for $\lambda*$ using binary search until either an optimal solution is found or the endpoints for the interval of uncertainty converge. If the endpoints converge, $\lambda \leq \lambda* \leq \lambda +1$ and we have found the maximum lower bound with only negligible error.

procedure adjust_lambda $(\lambda, \lambda_{min}, \lambda_{max}, \hat{R}, R)$

Input: λ, Lagrangian multiplier,

$\lambda_{min}, \lambda_{max}$, lower and upper endpoints for interval of uncertainty on λ,

$\hat{R}$, amount of resource consumed by the interdiction set, and

R, amount of interdiction resource available.

Output: λ, new value of the Lagrangian multiplier, and

$\lambda_{min}, \lambda_{max}$, new lower and upper endpoints for λ.

Begin {

if $(\hat{R} = R)$ /*an optimal solution has been found*/

$\lambda_{min} \leftarrow \lambda_{max} \leftarrow \lambda$;

else if $(\hat{R} > R)$ /*solution is infeasible, need a larger value of λ */

      $\lambda_{min} \leftarrow \lambda$;

else   /*feasible solution has been found, try a smaller value of λ */

      $\lambda_{max} \leftarrow \lambda$;

$\lambda = \frac{1}{2}\lfloor \lambda_{max} + \lambda_{min} \rfloor$;

return $(\lambda, \lambda_{min}, \lambda_{max})$;

} End;

APPENDIX B. HEURISTIC ALGORITHM FOR THE DYNAMIC NIP

This heuristic algorithm finds a feasible solution and bounds for the dynamic network interdiction problem. Essentially, it maximizes the lower bound, $z^{TE}(\lambda)$, the value of the Lagrangian relaxation, TE-LRR-D2(λ) but it also identifies feasible solutions to the network interdiction problem and reports the best lower and upper bounds and the best set of arcs to interdict (the interdiction set). The upper bound corresponds to the maximum flow through the dynamic network when a feasible interdiction set is found.

Dynamic Network Interdiction Heuristic

This heuristic seeks λ^*, the best value of the Lagrangian multiplier, by conducting a binary search on the interval of uncertainty for λ. The heuristic solves a maximum flow problem for each value of λ using the results to adjust λ and the endpoints of the interval. The initial left endpoint of the interval of uncertainty is slightly smaller than the smallest arc capacity, and the right endpoint is slightly larger than the largest arc capacity. λ is initially set to the value of the right endpoint. Inside a do-while loop, the heuristic defines arc capacities as $\min\{u_{ij}, \lambda r_{ij}/(q_{ij}+1)\}$. The arc capacities and the time-expanded network graph are inputs to the procedure find_max_flow, which solves the maximum flow problem, identifying a cut A_C^T, arc flows and the maximum flow through the network $x_{b'a'T}$. The cut, arc flows, original arc capacities, and the time horizon are passed to the procedure find_interdiction_set. This procedure interprets the results from the maximum flow procedure returning the interdiction set, a potential upper bound and the amount of interdiction resource consumed by the interdiction set. If a feasible solution is found, the heuristic compares the incumbent upper and lower bounds with the current bounds and keeps the better values. The heuristic also stores the best interdiction set. The procedure adjust_lambda takes λ and the endpoints of the interval of uncertainty, the interdiction

resource consumed by the current solution, and the amount of interdiction resource available as input, and returns a new value for λ with adjusted endpoints for the new interval of uncertainty. The heuristic then re-starts the sequence with a new value of λ until the endpoints of the interval of uncertainty for λ converge to a value within one unit of λ^*. The heuristic prints the best interdiction set found, the upper and lower bounds, and the amount of interdiction resource consumed by the interdiction set.

procedure BOUND_THE_DYNAMIC_PROBLEM

Input: Network $G^T = (N^T, A^T)$ with super-source, $a' \in N^T$, and super-sink, $b' \in N^T$ identified,

$\mathbf{u}$, integer arc capacities $u_{ij} \geq 0 \ \forall (i,j)_t \in A^T$,

$\mathbf{r}$, integer interdiction resource requirements $r_{ij} \geq 0 \ \forall (i,j)_t \in A^T$,

R, total interdiction resource available, and

time horizon T.

Output: Best interdiction set found A_s^T,

UB, an upper bound for the network interdiction problem,

LB, a lower bound, for the network interdiction problem,

$\hat{R}$, amount of interdiction resource required for A_s^T.

Begin {

    $LB \leftarrow -\infty$;

    $UB \leftarrow +\infty$;

    $A_s^T \leftarrow \varnothing$;

    /* find the initial endpoints for the interval of uncertainty for λ */

    $\lambda_{max} \leftarrow (\max_{(i,j)_t \in A^T - (b',a')_T} u_{ij} + 1) * (q_{ij} + 1)$;

    $\lambda_{min} \leftarrow \min_{(i,j)_t \in A^T - (b',a')_T} u_{ij} - 1$;

$\lambda \leftarrow \lambda_{\max}$;

Do{

$\hat{\mathbf{x}} \leftarrow \mathbf{0}$;   /* reset all network flows */

$\mathbf{u'} \leftarrow \min\{u_{ij}, \lambda r_{ij}/(q_{ij}+1)\}$ for all $(i,j) \in A$;   /* reset the adjusted arc capacities */

$(A_C^T, \hat{\mathbf{x}}, x_{b'a'T}) \leftarrow$ find_max_flow $(G^T, a', b', \mathbf{u'})$;

$(\hat{A}_S^T, \text{UB'}, \hat{R}) \leftarrow$ find_interdiction_set $(A_C^T, \hat{\mathbf{x}}, \mathbf{u}, T)$;

if $(\hat{R} \le R)$ {   /* a feasible solution has been found*/

$LB \leftarrow \max\{LB, x_{b'a'T} - \lambda R\}$;

if $(UB' < UB)$ {

$UB \leftarrow UB'$;

$A_S^T \leftarrow \hat{A}_S^T$;

}

}

$(\lambda, \lambda_{\min}, \lambda_{\max}) \leftarrow$ adjust_lambda $(\lambda, \lambda_{\min}, \lambda_{\max}, \hat{R}, R)$;

} while $(\lambda_{\max} - \lambda_{\min} > 1)$

print $(A_S^T, UB, LB, \hat{R})$;

} End;

The procedure find_max_flow finds the standard maximum flow in the directed network G^T with super-source a', super-sink b, and arc capacities $\mathbf{u'}$.

procedure find_max_flow $(G^T, a', b', \mathbf{u'})$

Input: Network graph $G^T = (N^T, A^T)$ with super-source, $a' \in N^T$, and super-sink,

$b' \in N^T$ identified,

$\mathbf{u'}$, integer arc capacities $u_{ij} \ge 0 \ \forall (i,j) \in A$.

63

Output: A_C^T, a minimum capacity cut $A_C^T \subset A$,

$\hat{\mathbf{x}}$, vector of maximum arc flows $x_{ijt} \geq 0 \ \forall(i,j)_t \in A^T$,

$x_{b'a'T}$, maximum flow value.

{   This procedure uses a standard shortest-augmenting path maximum flow algorithm (Edmonds and Karp, 1972) that is modified to find the maximum residual capacity among all the "shortest paths." Shortest path means the path with the minimum number of arcs.

return (A_C^T, $\hat{\mathbf{x}}$, $x_{b'a'T}$);

} End;

The procedure find_interdiction_set takes the cut, network arc flows, arc capacities, and time horizon as input. It compares arcs in the cut by time period from earliest to latest to identify an interdiction set. When an interdictable arc is found, the procedure scans the cut marking that arc each time it appears from one to q_{ij} time periods after interdiction,. The procedure returns an interdiction set, an upper bound, and the amount of resource consumed.

procedure find_interdiction_set ($A_C^T, \hat{\mathbf{x}}, \mathbf{u}, T$);

Input: A_C^T, a minimum capacity cut in G^T,

$\hat{\mathbf{x}}$, vector of maximum arc flows, and

$\mathbf{u}$, the original capacity of each arc in the network.

Output: $\hat{A}_S^T$, an interdiction set,

UB, a potential upper bound on the optimal solution of the network interdiction problem,

$\hat{R}$, the amount of resource consumed by the interdiction set.

Begin {

64

$\hat{A}_s^T \leftarrow \varnothing$;

$UB' \leftarrow 0$;

$A_C^T \leftarrow \text{unmark}(A_C^T)$;   /*reset an indicator variable*/

for $(t = 0$ to $T)$ {

    for (each $(i,j)_t \in A_C^T)$ {

        if $(x_{ijt} = u_{ijt})$

            /*upper bound is the capacity of the minimum capacity cut after interdiction*/

            $UB' \leftarrow UB' + u_{ij}$;

        else if $((i,j)_t$ unmarked) {

            add $(i,j)_t$ to $\hat{A}_s^T$;

            $\hat{R} \leftarrow \hat{R} + r_{ij}$;

            for ($t' = t + 1; t' \le t + q_{ij}; t'++$) {

                if ($(i,j)_{t'} \in A_C^T)$

                    /*mark arcs under repair from interdicting $(i,j)_t$ */

                    mark $(i,j)_{t'}$;

            }

        }

        }

    }

}

return $(\hat{A}_s^T, UB', \hat{R})$;

}End;

    The procedure adjust_lambda returns a new value for λ and adjusted endpoints for the new interval of uncerainty. We look for λ^* using binary search until the endpoints of

the interval of uncertainty for λ converge. When the endpoints converge, $\lambda \leq \lambda^* \leq \lambda + 1$ and we have found the maximum lower bound with only negligible error.

procedure adjust_lambda $(\lambda, \lambda_{min}, \lambda_{max}, \hat{R}, R)$

Input: λ, Lagrangian multiplier,

         $\lambda_{min}, \lambda_{max}$, lower and upper endpoints for interval of uncertainty on λ,

         $\hat{R}$, amount of resource consumed by the interdiction set, and

         R, amount of interdiction resource available.

Output: λ, new value of the Lagrangian multiplier, and

         $\lambda_{min}, \lambda_{max}$, new lower and upper endpoints for λ.

Begin {

      if $(\hat{R} > R)$ /*solution is infeasible, need a larger value of λ */

          $\lambda_{min} \leftarrow \lambda$;

      else   /*solution is feasible, try a smaller value of λ */

          $\lambda_{max} \leftarrow \lambda$;

      $\lambda = \frac{1}{2} \lfloor \lambda_{max} + \lambda_{min} \rfloor$;

      return $(\lambda, \lambda_{min}, \lambda_{max})$;

} End;

LIST OF REFERENCES

Ahuja, R.K., Magnanti, T.L., and Orlin, J.B., *Network Flows*, Prentice Hall, 1993.

Cormican, K.J., "Computational Methods for Deterministic and Stochastic Network Interdiction Problems," Master's Thesis, Naval Postgraduate School, Monterey, California, March 1995.

Cormican, K.J., Morton, D.P., and Wood, R.K., "Stochastic Network Interdiction," working paper, October 1996. To appear in *Operations Research*.

Edmonds, J., and Karp, R.M., "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems," *Journal of the Association for Computing Machinery*, Vol. 19, No. 2, April 1972.

Ford, L.R., Jr., and Fulkerson, D.R., *Flows in Networks*, Princeton University Press, Princeton, New Jersey, 1962.

Lubore, S.H., Ratliff, H.D., and Sicilia, G.T., "Finding the n Most Vital Links in Flow Networks," *Management Science*, Vol. 21, No. 5, January, 1975.

Phillips, C.A., "The Network Destruction Problem," Report SAND-92-0186C, Sandia National Laboratories, Albuquerque, New Mexico, April 27, 1992.

Steinrauf, R.L., "Network Interdiction Models," Master's Thesis, Naval Postgraduate School, Monterey, California, September 1991.

Wollmer, R.D., "Removing Arcs from a Network," *Operations Research*, Vol. 12, pp. 807-1076, November-December 1964.

Wollmer, R.D., "Algorithm for Targeting Strikes in a Lines-of-Communication Network," *Operations Research*, Vol. 18, pp. 497-515, May-June 1970.

Wood, R.K., "Deterministic Network Interdiction," *Mathematical and Computer Modeling*, Vol. 17, No. 2, pp. 1-18, 1993.

Wood, R.K., "Notes on A Heuristic to Solve A Network Interdiction Problem," unpublished notes, 1997.

INITIAL DISTRIBUTION LIST

		No. copies
1.	Defense Technical Information Center	2
	8725 John J. Kingman Rd., STE 0944	
	Ft. Belvoir, Virginia 22060-6218	
2.	Dudley Knox Library	2
	Naval Postgraduate School	
	411 Dyer Rd.	
	Monterey, California 93943-5101	
3.	Defense Logistics Studies Information Exchange	1
	U.S. Army Logistics Management Center	
	Fort Lee, Virginia 23801	
4.	Deputy Chief of Naval Operations (Logistics)	1
	Attn: CDR Millie Simons, N421C	
	2000 Navy Pentagon	
	Washington, DC 20350-2000	
5.	Professor David Schrady	1
	Department of Operations Research	
	Naval Postgraduate School, Code OR/So	
	Monterey, California 93943-5101	
6.	Professor R. Kevin Wood	3
	Department of Operations Research	
	Naval Postgraduate School, Code OR/Wd	
	Monterey, California 93943-5101	

7. Professor Paul Bloch   1
   Department of Operations Research
   Naval Postgraduate School, Code OR/Be
   Monterey, California 93943-5101

8. Joint Warfare Analysis Center   2
   18385 Frontage Road
   Dahlgren, Virginia 22448-5500

9. LCDR Henry D. Derbes   1
   2249 Elwick Dr.
   Baton Rouge, Louisiana 70816