

# ORDERED OPTIMAL SOLUTIONS AND APPLICATIONS

By

Li Liu

B. Sc. (Computer Science) Peking University

M. A. (Economics) Peking University

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF  
THE REQUIREMENTS FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES  
MANAGEMENT SCIENCE

We accept this thesis as conforming  
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

August 1, 1996

© Li Liu, 1996

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Management Science  
The University of British Columbia  
2075 Wesbrook Place  
Vancouver, Canada  
V6T 1Z1

Date:

August 2, 1996

## Abstract

The Monotone Response and Selection (MRS) Theorems in lattice programming provide a sufficient condition under which the optimal solutions of a parametric optimization problem are monotone in the parameter. The MRS Theorems have many applications. In particular, they explain the monotonicity of minimum cuts in parametric maximum flow problems in certain networks. However, in some parametric optimization problems, monotonicity does not hold. Rather, a totally ordered selection of optimal solutions exists. In Chapter 2, I present some conditions under which there exists a selection of optimal solutions that are totally ordered, but not necessarily monotone. Based on this result, I present necessary and sufficient conditions for some natural classes of parametric maximum flow problems to ensure the existence of a totally ordered selection of minimum cuts.

In Chapter 3, I extend the original selection problem studied by Rhys (1970) and Balinski (1970). I introduce the notion of tasks and assume that, facilities are capable of performing multi-tasks. A job then, instead of requiring a fixed set of facilities, requires a set of tasks, which can be performed by different facilities. The objective is to maximize the net return. Although this extended model is NP-hard, by introducing some natural restrictions in the extended selection problem, I obtain two variants thereof, which can be solved as minimum cut problems. A forest-harvest and road-construction scheduling problem can be formulated as one of the two variants and solved as a minimum cut problem.

Recently, new legislations have been introduced which require that a logging company cannot harvest a block of forest if an adjacent block has been harvested within the past

20 years. In order to harvest a block of forest, a road must be ready leading from a saw mill to the block. Roads must be constructed from a tree structured network of potential links. Due to the long planning horizon, forest-harvesting and road-construction must be scheduled together in order to maximize the present value of net income. In Chapter 4, I present a tabu search heuristic to generate schedules which compare favorably with previous studies.

## Table of Contents

<b>Abstract</b>	<b>ii</b>
<b>List of Tables</b>	<b>vi</b>
<b>List of Figures</b>	<b>vii</b>
<b>Acknowledgement</b>	<b>ix</b>
<b>1 Introduction and Preliminaries</b>	<b>1</b>
1.1 Introduction . . . . .	2
1.1.1 Monotonicity of Optimal Solutions . . . . .	2
1.1.2 Facility Selection Problems . . . . .	3
1.1.3 Forest Harvest and Road Scheduling Problem . . . . .	5
1.1.4 Contributions of the Thesis . . . . .	7
1.1.5 Organization of the Thesis . . . . .	10
1.2 Preliminaries . . . . .	11
1.2.1 Lattice Programming . . . . .	11
1.2.2 Parametric Maximum Flow Problems . . . . .	12
<b>2 Totally Ordered Optimal Solutions and Parametric Maximum Flows</b>	<b>15</b>
2.1 Totally Ordered Selections . . . . .	15
2.1.1 The Monotone Response and Selection Theorems . . . . .	16
2.1.2 Totally Ordered Response and Selection Theorems . . . . .	18
2.2 Ordered Minimum Cuts in Parametric Maximum Flow Problems . . . . .	27

2.2.1	Monotone minimum cuts . . . . .	27
2.2.2	Parallel Arcs in Maximum Flow Problems . . . . .	29
2.2.3	Totally Ordered Selections of Minimum Cuts . . . . .	32
2.3	Ordering of Minimum Cuts in Some Parametric Flow Problems . . . . .	52
<b>3</b>	<b>An Extended Selection Problem</b>	<b>61</b>
3.1	The Extended Selection Model . . . . .	61
3.2	Some Variants of the ES Model . . . . .	64
3.2.1	The ES Model with Nested Facilities . . . . .	65
3.2.2	An ES Model with Nested Facilities and Jobs . . . . .	71
3.3	Applications of the NFJ model . . . . .	80
3.3.1	A Network Scheduling Problem . . . . .	80
3.3.2	A Forest Harvest Scheduling Problem . . . . .	85
3.3.3	An Open Pit Mining Scheduling Problem . . . . .	86
3.4	Some Qualitative Properties . . . . .	90
<b>4</b>	<b>Forest Harvest and Road Construction Scheduling</b>	<b>105</b>
4.1	The Model . . . . .	105
4.2	The Tabu Search Heuristic . . . . .	109
4.3	Scheduling Problems in the Tangier Watershed . . . . .	112
	<b>Bibliography</b>	<b>121</b>
	<b>Appendices</b>	<b>126</b>
<b>A</b>	<b>Pseudo-Code for the Tabu Heuristic</b>	<b>126</b>

## List of Tables

3.1	Jobs in Example 3.1 . . . . .	66
3.2	Facilities in Example 3.1 . . . . .	67
3.3	Jobs in Example 3.2 . . . . .	73
3.4	Facilities and tasks performed by them . . . . .	82
3.5	Jobs and tasks required . . . . .	83
3.6	Summary of monotone results . . . . .	103
4.7	Total volumes at the ends of the planning horizons . . . . .	113
4.8	Solutions for 219 blocks with 6 periods . . . . .	115
4.9	Solutions for 219 blocks with 12 periods . . . . .	116
4.10	Solutions for 491 blocks with 6 periods . . . . .	117
4.11	Solutions for 491 blocks with 12 periods . . . . .	118
4.12	Optimal net income for the relaxed problems . . . . .	118

## List of Figures

2.1	The sets of minimum cuts are not totally ordered . . . . .	19
2.2	An infinite lattice $\mathcal{L}$ . . . . .	23
2.3	An example for which a sub-selection cannot be extended . . . . .	26
2.4	Parallel arcs . . . . .	29
2.5	Front and back arcs . . . . .	31
2.6	$G'$ for Case 1 . . . . .	35
2.7	$G'$ for Case 1 . . . . .	36
2.8	$G'$ for Case 2 . . . . .	37
2.9	$G'$ for Case 3 . . . . .	37
2.10	Partition of the node set . . . . .	40
2.11	An example where totally ordered selections of minimum cuts does not exist	53
2.12	Two rays in opposite directions . . . . .	58
3.13	The facility tree for Example 3.1 . . . . .	68
3.14	Network $G$ for Example 3.1 . . . . .	69
3.15	Tree of jobs in Example 3.2 . . . . .	74
3.16	Network $G$ for Example 3.2 . . . . .	75
3.17	Possible links from the suppliers to the consumers . . . . .	81
3.18	The original network . . . . .	84
3.19	The modified network . . . . .	84
3.20	Predecessors in open pit mining . . . . .	88
3.21	An example of ore body . . . . .	88



3.22	Dagdelen's network $G$ . . . . .	89
3.23	A sparser network $G'$ . . . . .	89

## Acknowledgement

I cannot become what I am today without the selfless help and devotion from many people, including my parents, grand-parents, and my teachers. It is impossible to enumerate all these people, yet I am deeply grateful to them.

Special thanks should go to my supervisors, Professor Shelby Brumelle and Professor Daniel Granot, for their mentorship and inspiration, for bringing my attention to the problems addressed in the thesis and for their comments and discussions which were crucial in every step in my Ph.D. programme. Further, it was from the numerous corrections that they patiently made on the drafts of my thesis that I learned the basics of academic writing.

I would like to thank my dear wife Haihua Zheng. Without her support and tolerance, my programme could not have been successfully completed.

I have received valuable comments from other members in my supervisory committee, Professor Frieda Granot and Professor Nicholas Pippenger, and from members of the examination committee, Professor Richard Anstee, Professor Eric Denardo, Professor Joel Friedman and Professor Maurice Queyranne.

I am grateful to my friends and fellow graduate students for keeping me feel alive. The sports, hiking trips, even discussions in the corridors made my life more enjoyable and memorable.

I dedicate my thesis to my grand-parents and parents.

## Chapter 1

### Introduction and Preliminaries

In this thesis, I will discuss three related topics. First, I will present a qualitative study of optimal solutions and apply the qualitative results to the study of parametric maximum flow problems. Therein, I intend to answer questions such as (1) when are the optimal solutions totally ordered? and (2) on what kind of parametric problems can a well known parametric maximum flow algorithm (the GGT algorithm) possibly apply? Then I will present a natural extension of a well studied facility selection model, which accommodates the fact that modern facilities are often multi-functional. Although this extended model is generally NP-hard to solve, I will show that under some additional assumptions, it can be solved as a minimum cut problem. As a special case of the extended selection model, a forest harvesting and road construction scheduling problem can be solved as a minimum cut problem, when some constraints that occur in real world operations are relaxed. The unrelaxed forest harvesting and road construction scheduling problem is NP-hard and can only be solved to near optimality with known technology. I will present a tabu search heuristic to solve this unrelaxed problem and present some experimental results on real harvest scheduling problems for the Tangier watershed in British Columbia.

To facilitate presentation, in this chapter I present a brief introduction to the three topics that are addressed in the rest of the thesis.

## 1.1 Introduction

### 1.1.1 Monotonicity of Optimal Solutions

Qualitative analysis provides, without data collection or computation, the direction of change of optimal solutions when the problem parameters are changed. As such, it provides the decision makers with valuable insight into the underlying problem. In some cases, qualitative analysis facilitates the solution of an optimization problem. For example, such an analysis may limit the solution space to a manageable size. Sometimes it is impossible to solve an optimization problem and qualitative analysis may be the only feasible approach to gain some insight to the problem.

Lattice programming, developed by Topkis and Veinott<sup>1</sup>, is devoted to sensitivity analysis of optimization problems. Fundamental results in lattice programming are the Monotone Response Theorem and the Monotone Selection Theorem. They are concerned with the optimization of a parametric function, for which partial orders are defined on the domain of the function and on the space of the parameters. The first theorem asserts that under certain conditions, the sets of optimal solutions will form a series of ascending sublattices when the parameter increases. The second theorem provides sufficient conditions under which it is possible to select an optimal solution for each instance of the parameter so that the selected solutions are monotone in the parameter. Milgrom and Shannon (1994) presented a necessary and sufficient condition for the optimal sets to be ascending sublattices when the optimization problem is parameterized on both a parameter and the feasible set.

---

<sup>1</sup>See Topkis (1978) and Veinott (forthcoming). For earlier work, see Topkis (1968), Topkis and Veinott (1973) and Veinott (1974).

The Monotone Response and Selection Theorems have many applications. For example, they were used by Denardo, Huberman and Rothblum (1982) to show that optimal partitions of a base interval into  $k$  sub-intervals are interleaved when  $k$  changes; they were used by Granot and Veinott (1985) to justify some monotonicity results in a parametric study of network flow problems; they were used by Lovejoy (1987) to obtain monotone results in a study of the partially observed Markov decision processes; and they were used by Milgrom and Roberts (1990) to characterize some trends in modern manufacturing. Brumelle and Granot (1993) used the Monotone Response Theorem in their parametric study of the repair kit problem<sup>2</sup>, to show, among other results, that the search for an optimal kit can be reduced to  $n + 1$  monotone subsets of tools instead of  $2^n$  subsets, where  $n$  is the total number of tools. Topkis (1995) used the Monotone Response Theorem to provide monotone comparative statics results in a model of the firm and to unify and extend previous research that appeared in the literature.

### 1.1.2 Facility Selection Problems

The original selection model, introduced by Rhys (1970), is concerned with the selection of a set of facilities in order to perform a subset of potential jobs. Therein, a decision maker is given a set of potential jobs  $\mathcal{J}$  and a set of facilities  $\mathcal{F}$ . He must choose a subset of jobs  $J \subseteq \mathcal{J}$  to perform and a subset of facilities  $F \subseteq \mathcal{F}$  to acquire. Each job  $j$  requires a specific set  $F(j)$  of facilities, and the job can be performed if all the required facilities in  $F(j)$  have been acquired. A facility does not perish after being used to complete a job and can be reused for another job, therefore, a set  $J$  of jobs can be performed if  $\cup_{j \in J} F(j) \subseteq F$ . A cost  $c(f)$  is incurred for each facility  $f$  acquired, and a revenue  $r(j)$  is earned for each job  $j$  performed. Thus the profit from selecting facilities in  $F$  and performing jobs in  $J$  is  $\sum_{j \in J} r(j) - \sum_{f \in F} c(f)$ . The decision maker's

---

<sup>2</sup>See also Smith, Chambers and Shlifer (1980) and Mamer and Smith (1982).

objective is to maximize the total profit. Clearly, a trade off exists between acquiring more facilities and incurring higher facility cost, and foregoing some jobs and losing some revenues. Rhys (1970) showed that the above optimization problem can be formulated as a linear programming problem. Balinski (1970) further showed that it can be solved as a minimum cut problem in a capacitated network.

Several applications of the original selection problem have been studied. For example, Eisner and Severance (1976) showed that the selection problem can be used to find an optimal partition of data blocks between a fast storage and a slow storage. Therein, the cost for storing a block of data in the fast memory is higher than that in the slow memory. However, if a job ever needs to access a block of data stored in the slow memory, then a penalty cost is incurred. The objective is to minimize the sum of the data storage cost and the penalty cost.

Smith, Chambers and Shlifer (1980) and Mamer and Smith (1982) studied a single period multi-item inventory control problem known as the repair kit problem. Therein, a repair crew must select a kit of tools that they carry to geographically separate locations for equipment repairs. If a repair job needs a tool that they do not carry, then a penalty cost is incurred, possibly from the delays in fulfilling the job. The repair crew must properly select the kit so as to minimize the sum of the holding cost of the tools and the penalty cost. Mamer and Smith (1982) showed that this optimization problem can be solved by the original selection model.

In many real world problems, additional constraints must be considered. For example, there may be a budget constraint so that the total cost of the facilities acquired cannot exceed a certain limit, or there may be a limit on the number of facilities acquired, possibly due to the limited number of slots available to store the facilities. The selection problem with these additional constraints are referred to as *capacitated selection problems* (CSP), which are usually NP-hard to solve. Crama (1995) provided a comprehensive survey of

the literature on CSP and its variants. See also de Matta, Hsu and Lowe (1995).

### 1.1.3 Forest Harvest and Road Scheduling Problem

A distinct change in public attitudes toward the environment has led to demands for a forest management paradigm shift from one of a dominant timber use to one in which forests are managed for multiple values. Among the most significant implications of such a shift in policy on forest management was the introduction of adjacency or green-up constraints. These constraints attempt to eliminate large clear cut areas. For example, legislation in British Columbia requires that the trees in a block must reach a mean height of three meters before any adjacent blocks can be harvested. While environmental issues are more and more recognized by forest resource managers, any policy would not be sustainable if it does not allow logging companies to remain profitable and to continuously provide employment for the community. Thus harvest schedules must be made that do not violate adjacency constraints and at the same time, yield reasonably high income. In order to harvest a block, a road to a saw mill must be ready by the time the block is harvested. Since a planning horizon is at least 40 years and can be as long as 120 years, if a road is built later, then the present value of the cost is much less than if it is built earlier. Besides, a road will degenerate if it is built and left unused for a long period of time. Thus the construction of roads must be scheduled simultaneously with harvesting.

Although a lot of work has been done to optimize harvest schedules with adjacency constraints, road scheduling has largely been ignored. Early work incorporating adjacency constraints but not road scheduling includes a mixed integer programming approach by Kirby *et al.* (1980). Several alternative approaches have been tried to deal with the size of the problem, including random search, Lagrangian relaxation, simulated annealing, and other heuristics. Random search methods have often been applied to problems with

constraints on harvesting adjacent blocks of timber or with other harvest dispersal constraints (O'Hara *et al.* 1989; Clements *et al.* 1990; Nelson and Howard 1991; Sessions and Sessions 1990). Monte Carlo integer programming can overcome some of the limitations of random search methods (Daust *et al.* 1993; Jamnick *et al.* 1993; Nelson *et al.* 1991). Lagrangian relaxation was first applied in forestry by Hoganson and Rose (1984) to maximize the harvest of a large number of stands subject to a non-declining flow constraint. A variety of heuristic methods have been applied to the formulation of adjacency constraints in conjunction with relaxation methods (Gross and Dykstra 1988; Meneghin *et al.* 1988; Torres and Brodie 1990). Other methods which have been tried include multiple-criteria decision making algorithms (Howard 1991; Howard and Nelson 1993; Nelson *et al.* 1991), dynamic programming (Ludwig 1993), fuzzy control (Bare and Mendoza 1992) and simulated annealing (Lockwood *et al.* 1993). There is evidence that the Biased Random Search Method proposed by O'Hara *et al.* is one of the most effective methods to solve the harvest scheduling problem with adjacency constraints. Their method showed excellent results with objective function values at most 8% from linear programming upper bound, and at most 4% from the upper bound of a virtually 100% confidence interval they derived for the true optimum. Brumelle, Granot, Halme and Vertinsky (1996) presented a tabu search heuristic, and when tested on a real world problem in the Tangier watershed in British Columbia, their results compared favorably against the heuristic by O'Hara *et al.*

When the adjacency constraints are relaxed, the forest harvest and road construction scheduling problem can be formulated as an extended selection problem presented in Chapter 3 and be solved as a minimum cut problem. However, with the adjacency constraints, the problem is NP-hard. In Chapter 4, I will present a tabu search heuristic to solve this unrelaxed problem to near optimality.



### 1.1.4 Contributions of the Thesis

My study in Chapter 2 was motivated by a study carried out by Stone (1978), a parametric maximum flow algorithm developed by Gallo, Grigoriadis and Tarjan (GGT) (1989), and by a subsequent extension by Arai, Ueno and Kajitani (AUK) (1993). GGT considered a maximum network flow problem in which the capacities of arcs incident to the source or the sink may change as a function of a parameter, and showed that for  $k = O(n)$  instances of the parameter, the maximum flows can be found in a time bound of one maximum flow, or, in  $O(mn \log(n^2/m) + km \log(n^2/m))$  elementary operations, where  $n$  is the number of nodes and  $m$  is the number of arcs. It was observed by Stone (1978) that for the parametric maximum flow problem that they studied, the minimum cuts move monotonically with the change of the parameter. As is shown later in this thesis, the Monotone Response and Selection Theorems offer a simple explanation for this monotonicity property. AUK extended the result of GGT and showed that when the capacities of arcs incident to a single node (other than the source or the sink) change, maximum flows for  $k = O(n)$  instances of the parameter can be found in the time bound of two maximum flows. In their study, the minimum cuts are no longer monotone in the parameter. They may move “back and forth” as the parameter changes, but there always exists a selection of minimum cuts that are totally ordered.

In light of the wide applicability of the Monotone Response and Selection Theorems, it seems appropriate to conduct a “parallel” study of the algebraic properties that lead to totally ordered optimal solutions. The Monotone Response and Selection Theorems address two closely related issues, namely, “response” and “selection”<sup>3</sup>. When the optimal solutions are monotone, the “selection” result follows from the “response” result under some mild topological conditions. When the optimal solutions are only totally ordered,

---

<sup>3</sup>Details are provided in Section 2.1.

the “selection” result is more complex. However, this result is important as it can be used to reduce the search space for optimal solutions in parametric optimization problems.

The objective of this thesis is to present algebraic sufficient conditions that ensure the existence of totally ordered selections of optimal solutions and to identify some classes of parametric maximum flow problems for which the GGT algorithmic approach can be possibly applied. Indeed, we present Totally Ordered Response and Selection Theorems which provide sufficient conditions for the totally ordered “response” and “selection” results to hold. Furthermore, we present some necessary and sufficient conditions which ensure the existence of a totally ordered selection of minimum cuts for some classes of parametric maximum flow problems. Since the existence of a totally ordered selection of minimum cuts is a common property for all parametric maximum flow problems that can be solved by the GGT algorithmic approach, our characterization provides some insight into the class of problems for which extensions of the GGT algorithm could be possible.

The classes of parametric maximum flow problems that we characterize, for which a totally ordered selection of minimum cuts exists, subsume the class of problems studied by AUK as a special case. I also present some other interesting special cases. For example, I show that one can further allow the capacity of the “center node” in the study of AUK to change and still maintain the existence of a totally ordered selection of minimum cuts. In another special case, I show that a totally ordered selection of minimum cuts exists when the capacities of arcs on a path are parametrically changed, given that no two arcs on the path are parallel<sup>4</sup>. Finally, I study the ordering of minimum cuts in some parametric maximum flow problems. It is shown that under certain conditions, minimum cuts form an ordered sequence of cuts when the capacities in the network change.

In Chapter 3 of the thesis, I present an extended selection model, which is based on the observation that many modern facilities are multi-functional. In this extension, a job

---

<sup>4</sup>A definition of parallel arcs is given in Subsection 2.2.2.

does not require a specific facility to be present. It can be accomplished as long as some required “functions” can be performed by one facility or another. I introduce the notion of tasks and assume that each job requires a set of tasks, and each facility can perform a set of tasks. Thus in my extended model, a job can be performed if the set of tasks required is covered by the set of facilities acquired. In general, the extended selection problem is NP-hard. However, by introducing some natural restrictions in the extended selection model I derive two variants thereof, which can be formulated and solved as minimum cut problems.

Apparently, the two variants of the ES model can be used to formulate classes of problems that could not be formulated by the original selection model. Indeed, the original selection model has mainly been used to study some single-period problems. For example, it has been used to model a network construction problem in which one seeks to construct an optimal set of roads. However, it cannot model the associated scheduling problem, to accommodate for the fact that the revenues and costs vary over time. That is, it cannot be used to determine an optimal schedule for the construction of these roads. By contrast, I show in this thesis that the network construction scheduling problem can be formulated as one of the variants of the ES model and can then be solved as a minimum cut problem. It is further shown that the network construction scheduling problem has applications in telecommunications, forest harvest scheduling and open pit mining. For related research on open pit mining, see Dagdelen (1979).

Brumelle and Granot (1993) studied some qualitative properties of the repair kit problem, which are also valid for the original selection problem. For example, it was shown that as the weight on the revenue from the jobs increases, the optimal collection of facilities will increase monotonically. In Chapter 3, I carry out a similar qualitative analysis and show that analogous properties exist for the two variants of the ES model.

In Chapter 4, I study a forest harvest and road construction scheduling problem.

A tabu search heuristic is used to optimize the net present value obtained from the schedules. This algorithm is tested on a real world problem in the Tangier watershed of British Columbia. The schedules generated in this experiment yield much higher net incomes compared with a previous experiment on the same data<sup>5</sup>.

### **1.1.5 Organization of the Thesis**

The remaining of the thesis contains three chapters. Chapter 2 presents a study of the totally ordered selection problem and its application to parametric maximum flow problems. In particular, Section 2.1 presents the Totally Ordered Selection Theorems. Using these theorems, in Section 2.2, I present a class of parametric maximum flow problems for which totally ordered selections of minimum cuts exist. Some special cases are also studied in this section. Section 2.3 studies the ordering of minimum cuts for some special parametric maximum flow problems.

Chapter 3 extends the original selection problem studied by Rhys (1970) and Balinski (1970). Section 3.1 formally introduces the Extended Selection (ES) model. Section 3.2 presents some sub-classes of the ES model that can be formulated and solved as minimum cut problems. Section 3.3 presents some applications of a restricted model discussed in Section 3.2. In particular, a relaxed forest harvest and road scheduling problem can be formulated as the Extended Selection model and be solved with a minimum cut problem. Section 3.4 presents some qualitative properties of the restricted models.

Chapter 4 presents a tabu search heuristic for solving the unrelaxed forest harvest and road construction scheduling problem. Therein, Section 4.1 formally presents the model, Section 4.2 presents the tabu search method used, and Section 4.3 presents the results for a scheduling problem in the Tangier watershed in British Columbia. Appendix A contains the pseudo-code for the tabu search heuristic.

---

<sup>5</sup>See Brumelle, Granot, Halme and Vertinsky (1996)

## 1.2 Preliminaries

### 1.2.1 Lattice Programming

Consider a partially ordered set  $(\mathcal{L}, \preceq)$ . I define  $z = x \vee y$  to be the *supremum* of two elements  $x$  and  $y$  in  $\mathcal{L}$  if  $x \preceq z$ ,  $y \preceq z$ , and  $x \preceq w$ ,  $y \preceq w$ ,  $w \in \mathcal{L}$  imply  $z \preceq w$ . Similarly, I define  $z = x \wedge y$  to be the *infimum* of  $x$  and  $y$  if  $z \preceq x$ ,  $z \preceq y$ , and  $w \preceq x$ ,  $w \preceq y$ ,  $w \in \mathcal{L}$  imply  $w \preceq z$ .

The set  $\mathcal{L}$  is called a *lattice* if for every pair  $x \in \mathcal{L}$  and  $y \in \mathcal{L}$ ,  $x \wedge y$  and  $x \vee y$  are also in  $\mathcal{L}$ . A subset  $S$  of  $\mathcal{L}$  is a *sublattice* of  $\mathcal{L}$  if  $x \in S$  and  $y \in S$  imply  $x \wedge y \in S$  and  $x \vee y \in S$ .

Sublattice  $S$  is *lower* than sublattice  $T$ , or,  $S \preceq_V T$ , if for every  $x \in S$  and  $y \in T$  we have  $x \vee y \in T$  and  $x \wedge y \in S$ . This order is reflexive, antisymmetric and transitive, and so partially orders sublattices of a lattice. Note that the  $\preceq_V$  order can be defined for arbitrary subsets of  $\mathcal{L}$ , although its reflexivity will be lost. This order was introduced by Veinott and will be referred to as the *Veinott ordering*.

Suppose  $(\Lambda, \leq)$  is a partially ordered set,  $(\mathcal{L}, \preceq)$  is a lattice and  $\{S_\lambda\}$  is a family of subsets of  $\mathcal{L}$  indexed in  $\Lambda$ . If for any  $\lambda_1$  and  $\lambda_2$  in  $\Lambda$ ,  $\lambda_1 \leq \lambda_2$  implies that  $S_{\lambda_1}$  is lower than  $S_{\lambda_2}$  (respectively,  $S_{\lambda_2}$  is lower than  $S_{\lambda_1}$ ), then  $\{S_\lambda\}$  is *ascending* (respectively, *descending*) in  $\lambda$  on  $\Lambda$ . If  $\lambda_1 \leq \lambda_2$  implies that  $x \preceq y$  (respectively,  $y \preceq x$ ) for each  $x \in S_{\lambda_1}$  and  $y \in S_{\lambda_2}$ , then  $\{S_\lambda\}$  is *strongly ascending* (respectively, *strongly descending*) in  $\lambda$  on  $\Lambda$ . See Topkis (1978), Veinott (forthcoming) and Zimmermann (1981) for a more detailed discussion on lattice programming.

In a finite sublattice  $S$ , there always exists a unique *minimum element*  $u$  such that for each  $x \in S$ ,  $u \preceq x$ . This is true since  $u$  is simply the infimum of all elements in  $S$ . Similarly, a unique *maximum element* exists in  $S$ .

If a finite subset  $S$  of  $\mathcal{L}$  is not a sublattice, at least one *minimal element*  $u$  exists in

$S$  such that  $x \in S$  and  $x \preceq u$  imply  $x = u$ . Such a minimal element can be found by descending along a totally ordered chain in  $S$  until the bottom of the chain is reached. When  $S$  is not a sublattice, a minimal element may not be unique. Similarly, there exists at least one *maximal element* in a finite set  $S$ .

A function  $f$  defined on a lattice  $\mathcal{L}$  is *submodular* if for every two elements  $x$  and  $y$  in  $\mathcal{L}$ ,

$$f(x \vee y) + f(x \wedge y) \leq f(x) + f(y),$$

and *supermodular* if

$$f(x \vee y) + f(x \wedge y) \geq f(x) + f(y).$$

A function  $f$  is *modular* if it is both submodular and supermodular.

A function  $f(x, \lambda)$  on  $X \times \Lambda$  has the *isotone differences* property with respect to  $(x, \lambda)$  if for any  $\lambda_1 < \lambda_2$ ,  $f(x, \lambda_2) - f(x, \lambda_1)$  is increasing in  $x$ , and has the *antitone differences* property if  $f(x, \lambda_2) - f(x, \lambda_1)$  is decreasing in  $x$ . Topkis (1978) showed that the isotone and antitone differences properties are symmetric between  $x$  and  $\lambda$ ; i.e., the above definitions of isotone and antitone differences can be equivalently defined by the monotonicity of  $f(x_2, \lambda) - f(x_1, \lambda)$  in  $\lambda$  for any  $x_1 \preceq x_2$ .

### 1.2.2 Parametric Maximum Flow Problems

Let  $G(N, A)$  denote a directed network, with nodes  $N$  and arcs  $A$ . Let  $s$  and  $t$  be the source node and the sink node, respectively. An arc capacity,  $c_{ij}$ , is associated with each arc  $(i, j)$ . The *maximum flow* problem in  $G(N, A)$  is concerned with finding a flow from  $s$  to  $t$  of maximum value which does not violate the capacity constraints. Formally, the

problem can be stated as follows:

$$\begin{aligned}
 & \max \quad \sum_{i:(s,i) \in A} x_{si} \\
 & \text{subject to} \\
 & \quad \sum_{i:(i,k) \in A} x_{ik} = \sum_{j:(k,j) \in A} x_{kj} \quad \text{for each } k \in N \setminus \{s, t\}, \\
 & \quad 0 \leq x_{ij} \leq c_{ij} \quad \text{for each } (i, j) \in A.
 \end{aligned}$$

In this study, I assume that arc capacities are strictly positive. This allows me to avoid some degenerate cases.

A *path* in a network is a sequence,  $\langle n_1, (n_1, n_2), n_2, (n_2, n_3), n_3, \dots, n_{k-1}, (n_{k-1}, n_k), n_k \rangle$ , of nodes and arcs. When there is no ambiguity, the above path will also be denoted as  $\langle n_1, n_2, \dots, n_k \rangle$ . A *simple path* is one in which the nodes do not repeat. An *s-t path* is a simple path from the source node  $s$  to the sink node  $t$ . For simplicity of presentation, I will write  $p_1 \cap p_2 = \emptyset$  to denote the fact that paths  $p_1$  and  $p_2$  are vertex disjoint;  $e \in p_1$  to denote the fact that arc  $e$  is in path  $p_1$ ; and  $u \in p_1$  to denote the fact that node  $u$  is in path  $p_1$ .

The flow in a network can alternatively be formulated in the *arc-chain form*. Let  $\mathcal{P} = \{p_1, p_2, \dots, p_m\}$  be the collection of all directed simple s-t paths in  $G(N, A)$ . A flow in  $G(N, A)$  is a function  $h$  from  $\mathcal{P}$  to the nonnegative reals, and the maximum flow problem can be formulated as:

$$\begin{aligned}
 & \max \quad \sum_{p \in \mathcal{P}} h(p) \\
 & \text{subject to} \\
 & \quad \sum_{p \in \mathcal{P}} a_{ij}(p) h(p) \leq c_{ij} \quad \text{for each } (i, j) \in A,
 \end{aligned}$$

where

$$a_{ij}(p) = \begin{cases} 1 & \text{if } (i, j) \in p, \\ 0 & \text{if } (i, j) \notin p. \end{cases}$$

The two formulations are equivalent; see for example, Ford and Fulkerson (1962).

In a node-and-arc capacitated network, node capacity constraints are introduced:

$$\sum_{i:(i,k) \in A} x_{ik} \leq c_k \quad \text{for each node } k \in N \setminus \{s, t\},$$

where  $c_k$  is the capacity of node  $k$ . Generally, in a maximum flow problem over a network  $G(N, A)$  both arcs and nodes may have capacities. However, a node-and-arc capacitated network can be transformed into an equivalent arc-capacitated network<sup>6</sup>.

A *cut* is a bi-partition  $(X, \bar{X})$  of the node set  $N$ , where  $s \in X$  and  $t \in \bar{X}$ .<sup>7</sup> A partial order can be defined on the set of cuts in a network. Namely,  $(X, \bar{X}) \preceq_c (Y, \bar{Y})$  if  $X \subseteq Y$ . Under this partial order, the set of cuts in a network is a lattice.

$(X, \bar{X})$  may be interpreted as the set of arcs between the two sets of nodes  $X$  and  $\bar{X}$ . That is,  $(X, \bar{X}) = \{(i, j) \in A \mid i \in X, j \in \bar{X}\}$ . More generally, for two subsets  $X$  and  $Y$  of  $N$ , define  $(X, Y)$  to be the set of arcs  $\{(i, j) \in A \mid i \in X, j \in Y\}$  and the capacity of  $(X, Y)$  to be  $c(X, Y) = \sum_{i \in X, j \in Y} c_{ij}$ . In particular, the capacity of cut  $(X, \bar{X})$  is defined as  $c(X, \bar{X}) = \sum_{i \in X, j \in \bar{X}} c_{ij}$ . A *minimum cut* in a network is one whose capacity is minimum among all cuts in the network.

A *parametric maximum flow* problem is a maximum flow problem in which the capacities of the arcs may change as functions of a parameter  $\lambda$ . In such a problem, the notation is modified to include the parameter, so the capacity of an arc  $(i, j)$  is denoted by  $c(\lambda; i, j)$  and the capacity of  $(X, Y)$  is written as  $c(\lambda; X, Y)$ . Again, I assume that the capacities on the arcs are positive to avoid degenerate cases.

---

<sup>6</sup>Transforming a node-and-arc capacitated network into an equivalent arc capacitated network may destroy some nice properties of the original network. For example, such a transformation may destroy the planarity of the original network. Therefore, algorithmically it may be more efficient to consider the original node-and-arc capacitated network rather than its equivalent arc capacitated network. However, since I do not deal with algorithmic issues in this thesis, I will consider only arc capacitated networks for ease of presentation.

<sup>7</sup>Cuts defined in this way are frequently called node partition cuts. Since cuts in this article are always node partition cuts, they are simply referred to as cuts.



## Chapter 2

### Totally Ordered Optimal Solutions and Parametric Maximum Flows

#### 2.1 Totally Ordered Selections

Let  $(\mathcal{L}, \preceq)$  be a lattice. Let  $f(x, \lambda)$  be a function defined on  $\mathcal{L} \times \Lambda$ , where  $\lambda$  is the parameter. The objective is to minimize  $f(\cdot, \lambda)$  for each  $\lambda$ . Let  $F^*(\lambda)$  be the set of minimum solutions of  $f(\cdot, \lambda)$  for a specific  $\lambda$ ; i.e.,  $F^*(\lambda) = \operatorname{argmin}_x \{f(x, \lambda) : x \in \mathcal{L}\}$ . For an arbitrary subset  $\Lambda_0$  of  $\Lambda$ , a *monotone sub-selection of optimal solutions* is a mapping,  $x(\cdot)$ , from  $\Lambda_0$  to  $\mathcal{L}$ , such that  $x(\lambda) \in F^*(\lambda)$  for each  $\lambda \in \Lambda_0$  and  $x(\lambda)$  is monotone in  $\lambda$ . Similarly, a *totally ordered sub-selection of optimal solutions* is a mapping,  $x(\cdot)$ , from  $\Lambda_0$  to  $\mathcal{L}$ , such that  $x(\lambda) \in F^*(\lambda)$  for each  $\lambda \in \Lambda_0$  and the collection  $\{x(\lambda) | \lambda \in \Lambda_0\}$  is totally ordered by  $\preceq$ . A monotone (respectively, totally ordered) selection of optimal solutions is a sub-selection when  $\Lambda_0 = \Lambda$ .

If  $F^*(\lambda) = \emptyset$  for any  $\lambda \in \Lambda$ , then  $\lambda$  can be ignored in this study. Therefore, I assume that  $F^*(\lambda) \neq \emptyset$  for every  $\lambda \in \Lambda$ .

My study is based on lattice programming theory and is closely related to the problem of monotone selections studied by Topkis and Veinott. In this section, I first present the Monotone Response and Selection Theorems and then introduce some sufficient conditions under which a totally ordered selection of optimal solutions exists.

### 2.1.1 The Monotone Response and Selection Theorems

The Monotone Response and Selection Theorems assume that a partial order  $\leq$  exists on the set  $\Lambda$  of parameters, and present sufficient conditions under which the set of optimal solutions,  $F^*(\lambda)$ , is ascending in  $\lambda$  with respect to the Veinott ordering, and under which a monotone selection of optimal solutions exists.

In order to facilitate further development, I provide a proof of the Monotone Response Theorem. My proof, which is slightly different from the original one, focuses on the validity of two inequalities which play a prominent role in this paper. These two inequalities are

$$f(x, \lambda_1) + f(y, \lambda_2) \geq f(x \vee y, \lambda_1) + f(x \wedge y, \lambda_2) \quad (2.1)$$

and

$$f(x, \lambda_1) + f(y, \lambda_2) \geq f(x \wedge y, \lambda_1) + f(x \vee y, \lambda_2). \quad (2.2)$$

**THEOREM 2.1 (Monotone Response Theorem)** *Suppose  $(\mathcal{L}, \preceq)$  is a lattice and  $(\Lambda, \leq)$  is a partially ordered set,  $f(x, \lambda)$  is submodular in  $x$  for each  $\lambda \in \Lambda$ , and  $f(x, \lambda)$  has antitone (respectively, isotone) differences in  $(x, \lambda)$  on  $\mathcal{L} \times \Lambda$ . Then,  $F^*(\lambda)$  is an ascending (respectively, descending) sublattice in  $\lambda$ .*

**Proof.** The fact that  $F^*(\lambda)$  is a sublattice for each  $\lambda$  follows from the submodularity of  $f$ ; see, e.g., Theorem 4.1 of Topkis (1978). Consider  $\lambda_1 \leq \lambda_2$  in  $\Lambda$ . For all  $x$  and  $y$  in  $\mathcal{L}$ ,

$$\begin{aligned} & f(x, \lambda_1) - f(x \wedge y, \lambda_1) \\ & \geq f(x, \lambda_2) - f(x \wedge y, \lambda_2) \quad \text{by the antitone differences property of } f(\cdot, \cdot) \\ & \geq f(x \vee y, \lambda_2) - f(y, \lambda_2) \quad \text{by the submodularity of } f(\cdot, \lambda_2). \end{aligned}$$

Thus (2.2) is valid for arbitrary  $x, y$  and  $\lambda_1 \leq \lambda_2$ . When  $x \in F^*(\lambda_1)$  and  $y \in F^*(\lambda_2)$ , (2.2) must be satisfied with equality, and  $x \wedge y \in F^*(\lambda_1)$  and  $x \vee y \in F^*(\lambda_2)$ . This establishes that  $F^*(\lambda_1) \preceq_V F^*(\lambda_2)$ .

Similarly, if  $f$  has isotone differences, then (2.1) holds for all  $x, y$  and  $\lambda_1 \leq \lambda_2$ . Moreover, if  $x \in F^*(\lambda_1)$  and  $y \in F^*(\lambda_2)$ , then (2.1) is satisfied with equality. Thus  $x \wedge y \in F^*(\lambda_2)$ ,  $x \vee y \in F^*(\lambda_1)$  and  $F^*(\lambda_2) \preceq_V F^*(\lambda_1)$ .  $\square$

**THEOREM 2.2 (Monotone Selection Theorem)** *Suppose the conditions in Theorem 2.1 hold.*

- *If it is further assumed that  $\Lambda$  is countable, then a monotone selection of optimal solutions exists.*
- *Alternatively, if it is further assumed that there exists a minimum element  $s(\lambda)$  (respectively, a maximum element  $S(\lambda)$ ) in  $F^*(\lambda)$  for every  $\lambda \in \Lambda$ , then  $s(\lambda)$  (respectively,  $S(\lambda)$ ) is monotone in  $\lambda$ .*

The original Monotone Selection Theorem (Theorem 6.2 in Topkis, 1978) corresponds to the second half of Theorem 2.2. It asserts that if  $f(x, \cdot)$  is lower semi-continuous in  $x$  and the parameterized feasible set is compact, then there exists a minimum element,  $s(\lambda)$ , and a maximum element,  $S(\lambda)$ , in  $F^*(\lambda)$ , which are both monotone in  $\lambda$ . The proof can be found in Topkis (1978). Below I augment the original theorem by presenting the proof of the first half of Theorem 2.2.

**Proof.** We will only prove Theorem 2.2 for the case when the conditions in Theorem 2.1 are such that  $F^*(\lambda)$  is ascending in  $\lambda$ . Let  $\Lambda = \{\lambda_1, \lambda_2, \dots\}$ . Suppose a monotone sub-selection,  $\{x(\lambda) \mid \lambda \in \Lambda_k\}$ , has been found for a subset of parameters  $\Lambda_k = \{\lambda_1, \lambda_2, \dots, \lambda_k\}$  such that  $\lambda_i \leq \lambda_j$  implies  $x(\lambda_i) \preceq x(\lambda_j)$ ,  $i, j = 1, 2, \dots, k$ . To prove the first claim in the theorem, it is enough to show that the monotone sub-selection can be extended to  $\Lambda_k \cup \{\lambda_{k+1}\}$ . Let  $\Lambda_k^1 = \{\lambda \in \Lambda_k \mid \lambda > \lambda_{k+1}\}$  and  $\Lambda_k^2 = \{\lambda \in \Lambda_k \mid \lambda < \lambda_{k+1}\}$ . Let  $x_{k+1}^1$  be an arbitrary element in  $F^*(\lambda_{k+1})$ . Let  $x_{k+1}^2$  be the infimum of  $\{x(\lambda) \mid \lambda \in$

$\Lambda_k^1\} \cup \{x_{k+1}^1\}$  and let  $x(\lambda_{k+1})$  be the supremum of  $\{x(\lambda) | \lambda \in \Lambda_k^2\} \cup \{x_{k+1}^2\}$ . Applying the Monotone Response Theorem inductively, it can be seen that  $x(\lambda_{k+1}) \in F^*(\lambda_{k+1})$ . By the definition of supremum,  $x(\lambda_{k+1}) \succeq x(\lambda)$  for each  $\lambda \in \Lambda_k^2$ . Consider an arbitrary  $\lambda \in \Lambda_k^1$ . By the definition of  $\Lambda_k^1$  and  $\Lambda_k^2$ , and the inductive assumption,  $x(\lambda) \succeq x(\lambda')$  for any  $\lambda' \in \Lambda_k^2$ . Furthermore, by the definition of  $x_{k+1}^2$ ,  $x(\lambda) \succeq x_{k+1}^2$ . Thus  $x(\lambda)$  is an upper-bound of  $\{x(\lambda) | \lambda \in \Lambda_k^2\} \cup \{x_{k+1}^2\}$ . Since  $x(\lambda_{k+1})$  is the least upper-bound of  $\{x(\lambda) | \lambda \in \Lambda_k^2\} \cup \{x_{k+1}^2\}$ ,  $x(\lambda_{k+1}) \preceq x(\lambda)$  must be valid. Therefore, introducing  $x(\lambda_{k+1})$  extends the original monotone sub-selection to  $\Lambda_k \cup \{\lambda_{k+1}\}$ .  $\square$

It can be seen from the proof of Theorem 2.1 that when  $f(\cdot, \cdot)$  satisfies the conditions in the theorem, either (2.1) holds if  $f(\cdot, \cdot)$  has isotone differences or (2.2) holds if  $f(\cdot, \cdot)$  has antitone differences. It is shown in the next section that a parametric function  $f(x, \lambda)$  satisfying either (2.1) or (2.2) for any pair of  $\lambda_1$  and  $\lambda_2$  ensures the existence of a totally ordered selection of optimal solutions.

### 2.1.2 Totally Ordered Response and Selection Theorems

If it is further assumed in the second half of the Monotone Selection Theorem that  $\Lambda$  is totally ordered, then a totally ordered selection of optimal solutions will exist. However, the hypothesis of Theorem 2.1 is not necessary for a totally ordered selection of optimal solutions to exist. Indeed, consider the example shown in Figure 2.1. The directed network  $G(N, A)$  is composed of 5 nodes on a single directed path. The capacity on arcs  $e_1$  and  $e_4$  is  $1.1 - \lambda$  and the capacity on arcs  $e_2$  and  $e_3$  is  $0.1 + \lambda$ . Let  $f(X, \lambda)$  be the capacity of  $(X, \bar{X})$  under parameter  $\lambda$ , where  $X \subseteq N$  is a subset of nodes containing  $s$ . Then  $(2^N, \subseteq)$  is a lattice. When  $\lambda = 1$ , there are two minimum cuts,  $(X_1, \bar{X}_1)$  and  $(X_2, \bar{X}_2)$ , where  $X_1 = \{s\}$  and  $X_2 = \{s, a, b, c\}$ , so  $F^*(1) = \{\{s\}, \{s, a, b, c\}\}$ . When  $\lambda = 0$ , there are also two minimum cuts,  $(Y_1, \bar{Y}_1)$  and  $(Y_2, \bar{Y}_2)$ , where  $Y_1 = \{s, a\}$  and

$Y_2 = \{s, a, b\}$ , so  $F^*(0) = \{\{s, a\}, \{s, a, b\}\}$ . It can be seen that  $F^*(0)$  and  $F^*(1)$  are not ordered by the Veinott ordering.

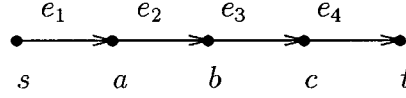


Figure 2.1: The sets of minimum cuts are not totally ordered

However, in the above example, a totally ordered selection of optimal solutions does exist. In fact, one can pick any solution from  $F^*(0)$  and any solution from  $F^*(1)$ , and the two cuts will be ordered. Thus a totally ordered selection of optimal solutions is possible even though  $\{F^*(\lambda) | \lambda \in \Lambda\}$  is not totally ordered.

In the remainder of this subsection, I consider three successively weaker conditions. Even the weakest is sufficient for the existence of a totally ordered selection of optimal solutions. However, they differ in their ability to ensure that totally ordered sub-selections can be extended to a complete one on  $\Lambda$ .

**CONDITION A** For every  $\lambda_1, \lambda_2 \in \Lambda$ , either (2.1) holds for every  $x \in F^*(\lambda_1)$  and  $y \in F^*(\lambda_2)$ , or (2.2) holds for every  $x \in F^*(\lambda_1)$  and  $y \in F^*(\lambda_2)$ .

**CONDITION B** For every  $\lambda_1, \lambda_2 \in \Lambda$  and arbitrary  $x \in F^*(\lambda_1)$ ,  $y \in F^*(\lambda_2)$ , either (2.1) or (2.2) holds.

Note that for a fixed pair  $\lambda_1$  and  $\lambda_2$ , Condition A requires inequalities (2.1) or (2.2) to hold “uniformly” for every  $x \in F^*(\lambda_1)$  and  $y \in F^*(\lambda_2)$ . In Condition B however, it is permissible for (2.1) to hold for some pairs of  $x$  and  $y$  and for (2.2) to hold for some other pairs. Clearly, Condition A implies Condition B.

**CONDITION C** For every  $\lambda_1, \lambda_2 \in \Lambda$  and  $x \in F^*(\lambda_1)$ ,  $y \in F^*(\lambda_2)$ , there exist  $u, v \in \mathcal{L}$  satisfying  $u \preceq x \wedge y$  and  $x \vee y \preceq v$  such that either

$$f(x, \lambda_1) + f(y, \lambda_2) \geq f(u, \lambda_1) + f(v, \lambda_2), \quad (2.3)$$

or

$$f(x, \lambda_1) + f(y, \lambda_2) \geq f(v, \lambda_1) + f(u, \lambda_2). \quad (2.4)$$

Condition B implies Condition C, since when the former is satisfied, one can choose  $u = x \wedge y$  and  $v = x \vee y$ .

**THEOREM 2.3 (Totally Ordered Response Theorem)** *Let  $\mathcal{L}$  be a lattice and  $\Lambda$  be an arbitrary set, and let  $f$  be a function defined on  $\mathcal{L} \times \Lambda$ . The collection  $\{F^*(\lambda) | \lambda \in \Lambda\}$  is totally ordered by  $\preceq_V$  if and only if Condition A holds.*

**Proof.** I first show the “if” part of the theorem. Consider  $\lambda_1, \lambda_2 \in \Lambda$ . Suppose (2.1) is valid for every  $x \in F^*(\lambda_1)$  and  $y \in F^*(\lambda_2)$ . Since  $f(x \vee y, \lambda_1) \geq f(x, \lambda_1)$  and  $f(x \wedge y, \lambda_2) \geq f(y, \lambda_2)$ , (2.1) and the above two inequalities are satisfied as equations. Thus  $x \vee y \in F^*(\lambda_1)$  and  $x \wedge y \in F^*(\lambda_2)$ , implying that  $F^*(\lambda_2) \preceq_V F^*(\lambda_1)$  is valid. Note that for an arbitrary  $\lambda \in \Lambda$ , one can show that  $F^*(\lambda)$  is a sublattice by letting  $\lambda_1 = \lambda_2 = \lambda$  in (2.1).

Similarly, if (2.2) holds for every  $x$  and  $y$ , then  $F^*(\lambda_1) \preceq_V F^*(\lambda_2)$ . This shows that the collection  $\{F^*(\lambda) | \lambda \in \Lambda\}$  is totally ordered.

To show the “only if” part of the theorem, let  $\lambda_1$  and  $\lambda_2$  be arbitrary parameters in  $\Lambda$ . Since  $\{F^*(\lambda) | \lambda \in \Lambda\}$  is totally ordered, either  $F^*(\lambda_1) \preceq_V F^*(\lambda_2)$  or  $F^*(\lambda_2) \preceq_V F^*(\lambda_1)$ . First suppose  $F^*(\lambda_1) \preceq_V F^*(\lambda_2)$  so that for any  $x \in F^*(\lambda_1)$  and  $y \in F^*(\lambda_2)$ ,  $x \vee y \in F^*(\lambda_2)$  and  $x \wedge y \in F^*(\lambda_1)$ . Thus  $f(x \vee y, \lambda_2) = f(y, \lambda_2)$  and  $f(x \wedge y, \lambda_1) = f(x, \lambda_1)$ , and (2.2) is valid. Similarly, if  $F^*(\lambda_2) \preceq_V F^*(\lambda_1)$ , then (2.1) is valid.  $\square$

The Totally Ordered Response Theorem bears some resemblance to the Monotone Response Theorem. However, the former theorem assumes a weaker condition, and it does not require  $\Lambda$  to be partially ordered.

**LEMMA 2.4** *Let  $\mathcal{L}$  be a lattice,  $\Lambda$  be an arbitrary set, and  $f$  be a function defined on  $\mathcal{L} \times \Lambda$ . Condition B holds if and only if for every  $\lambda_1$  and  $\lambda_2$  in  $\Lambda$  and every  $x \in F^*(\lambda_1)$  and  $y \in F^*(\lambda_2)$ , either  $x \wedge y \in F^*(\lambda_1)$  and  $x \vee y \in F^*(\lambda_2)$ , or  $x \wedge y \in F^*(\lambda_2)$  and  $x \vee y \in F^*(\lambda_1)$ .*

Since the proof of this lemma is similar to that of Theorem 2.3, it is omitted.

**LEMMA 2.5** *Suppose Condition C holds for  $f(\cdot, \cdot)$  defined on a lattice  $\mathcal{L}$  and an arbitrary set  $\Lambda$ . Let  $\lambda_1$  and  $\lambda_2$  be arbitrary elements in  $\Lambda$ . Then for an arbitrary  $x \in F^*(\lambda_1)$ , either there exists an  $m \in F^*(\lambda_2)$  such that  $m \preceq x$ , or there exists an  $M \in F^*(\lambda_2)$  such that  $x \preceq M$ .*

**Proof.** Pick an arbitrary element  $y \in F^*(\lambda_2)$ . By Condition C, one can find  $u \preceq x \wedge y$  and  $v \succeq x \vee y$  such that either  $u \in F^*(\lambda_2)$  or  $v \in F^*(\lambda_2)$ . If the former holds, then let  $m = u$ , otherwise the latter holds and let  $M = v$ .  $\square$

**LEMMA 2.6** *Let  $\mathcal{L}$  be a lattice, let  $\lambda_1$  and  $\lambda_2$  be arbitrary values in  $\Lambda$ , let  $x \in F^*(\lambda_1)$  be an arbitrary optimal solution, and let  $s(\lambda_2)$  and  $S(\lambda_2)$  be, respectively, the minimum element and the maximum element in  $F^*(\lambda_2)$ . If Condition C is valid, then either  $x \preceq S(\lambda_2)$  or  $s(\lambda_2) \preceq x$ .*

**Proof.** The lemma follows directly from Lemma 2.5, since  $M \preceq S(\lambda_2)$  and  $s(\lambda_2) \preceq m$ .  $\square$

**THEOREM 2.7 (Strong Totally Ordered Selection Theorem)** *Let  $\mathcal{L}$  be a lattice, let  $f(\cdot, \cdot)$  be defined on  $\mathcal{L} \times \Lambda$ , and suppose Condition B is satisfied.*

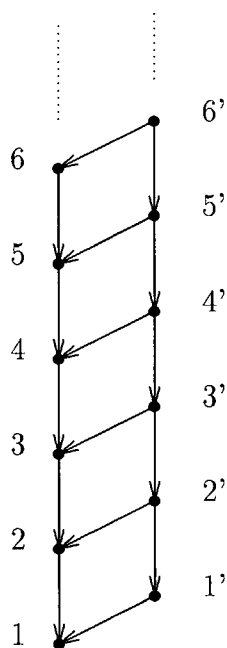
- Assume that  $\Lambda$  is a countable set, then a totally ordered sub-selection of optimal solutions  $\{x(\lambda) \in F^*(\lambda) | \lambda \in \Lambda_0\}$  for a finite set  $\Lambda_0$  can be extended to a totally ordered selection on  $\Lambda$ .
- Alternatively, if it is assumed that a minimum element  $s(\lambda)$  (respectively, maximum element  $S(\lambda)$ ) exists in every  $F^*(\lambda)$ , then the collection  $\{s(\lambda)\}$  (respectively,  $\{S(\lambda)\}$ ) is totally ordered.

**Proof.** I first prove the first half of the theorem. The mapping  $x(\cdot)$  can be constructed by the following inductive procedure. Suppose that at a certain step, one has obtained a totally ordered sub-selection for  $\Lambda_k$ , which consists of  $k$  elements and contains  $\Lambda_0$  as a subset. Since  $\Lambda_k$  is finite, the elements in  $\Lambda_k$  can be denoted by  $\lambda_1, \lambda_2, \dots, \lambda_k$  so that  $x_1 \preceq x_2 \preceq \dots \preceq x_k$ , where  $x_i = x(\lambda_i)$  for  $i = 1, 2, \dots, k$ . Choose an arbitrary  $\lambda_{k+1}$  in  $\Lambda \setminus \Lambda_k$ . I wish to find  $x_{k+1} = x(\lambda_{k+1})$  such that  $x_{k+1} \in F^*(\lambda_{k+1})$  and the collection  $\mathcal{C}_{k+1} = \{x_1, x_2, \dots, x_k, x_{k+1}\}$  is totally ordered.

By Lemma 2.5,  $F^*(\lambda_{k+1})$  either contains an element  $m_1$  such that  $m_1 \preceq x_1$  or an element  $M_1$  such that  $x_1 \preceq M_1$ . If the former is true, let  $x_{k+1} = m_1$  and  $\mathcal{C}_{k+1} = \mathcal{C}_k \cup \{x_{k+1}\}$  is totally ordered. If the latter is true, let  $l$  be the largest index such that there exists an element  $M_l$  in  $F^*(\lambda_{k+1})$  for which  $x_l \preceq M_l$ . If  $l = k$ , let  $x_{k+1} = M_l$  and  $\mathcal{C}_{k+1} = \mathcal{C}_k \cup \{x_{k+1}\}$  is again totally ordered. If  $l < k$ , then there does not exist an element  $M_{l+1}$  in  $F^*(\lambda_{k+1})$  such that  $x_{l+1} \preceq M_{l+1}$ . Consider  $x_{l+1} \vee M_l$  and  $x_{l+1} \wedge M_l$ . By Condition B, one of them must be in  $F^*(\lambda_{k+1})$ . The former cannot be in  $F^*(\lambda_{k+1})$ , since otherwise, there exists an element  $M_{l+1} = x_{l+1} \vee M_l$  in  $F^*(\lambda_{k+1})$  satisfying  $x_{l+1} \preceq M_{l+1}$ , contradicting the observation that no such  $M_{l+1}$  exists. So  $x_{l+1} \wedge M_l$  is in  $F^*(\lambda_{k+1})$ . Let  $x_{k+1} = x_{l+1} \wedge M_l$ . Clearly,  $x_{k+1} \preceq x_{l+1}$ . Since  $x_l \preceq M_l$  and  $x_l \preceq x_{l+1}$ , it follows from the definition of  $\wedge$  that  $x_l \preceq x_{l+1} \wedge M_l = x_{k+1}$ . Thus the collection  $\mathcal{C}_{k+1} = \mathcal{C}_k \cup \{x_{k+1}\}$  is totally ordered. This completes the proof of the first half of the theorem.



To prove the second half of the theorem, consider two arbitrary instances of the parameter  $\lambda_1$  and  $\lambda_2$ . Since  $s(\lambda_1) \wedge s(\lambda_2)$  is either in  $F^*(\lambda_1)$  or in  $F^*(\lambda_2)$ ,  $s(\lambda_1) \wedge s(\lambda_2)$  must be equal to either  $s(\lambda_1)$  or  $s(\lambda_2)$ . In the former case,  $s(\lambda_1) \preceq s(\lambda_2)$  and in the latter case,  $s(\lambda_2) \preceq s(\lambda_1)$ . Similarly,  $S(\lambda_1)$  and  $S(\lambda_2)$  must be ordered by  $\preceq$ .  $\square$

Figure 2.2: An infinite lattice  $\mathcal{L}$ 

When a parametric optimization problem needs to be solved for many instances of the parameter, optimal solutions for a subset of parametric instances are obtained first. If one knows that an arbitrary totally ordered sub-selection of optimal solutions can be extended to a complete one, then the search for optimal solutions can be limited to the subset of feasible solutions that are totally ordered with the solutions already obtained. Since the parametric instances must be solved successively, they form a countable sequence. Therefore, the countability assumption for  $\Lambda$  is not restrictive.

**COROLLARY 2.8** *If  $\mathcal{L}$  is finite, then the first claim in Theorem 2.7 still holds without requiring  $\Lambda$  to be countable.*

**Proof.** Observe that if for two values of the parameter,  $\lambda_1$  and  $\lambda_2$ ,  $F^*(\lambda_1) = F^*(\lambda_2)$ , then one only has to select an optimal solution  $x(\lambda_1)$  for  $\lambda_1$  and let  $x(\lambda_2) = x(\lambda_1)$ . Let  $n = |\mathcal{L}|$  be the cardinality of  $\mathcal{L}$ . For each  $\lambda \in \Lambda$ ,  $F^*(\lambda)$  must be one of the  $2^n$  subsets of  $\mathcal{L}$ . Therefore, one has to consider at most  $2^n$  representative instances in  $\Lambda$ . Since this set of representative instances is finite and thus countable, the first claim in Theorem 2.7 applies and the proof is complete.  $\square$

The first half of Theorem 2.7 is not valid without assuming that  $\Lambda_0$  is finite. Indeed, consider the counter example illustrated in Figure 2.2. The figure illustrates an infinite lattice  $\mathcal{L}$  where  $i \preceq j$  if there exists a directed path from  $j$  to  $i$ . Let  $\Lambda = \{\lambda_0, \lambda_1, \lambda_2, \dots\}$ . Define a function  $f(\cdot, \cdot)$  on  $\mathcal{L} \times \Lambda$  such that for  $i \geq 1$

$$f(x, \lambda_i) = \begin{cases} 0 & \text{for } x = 1, 2, \dots, i, \\ 1 & \text{for other } x \in \mathcal{L}, \end{cases}$$

and

$$f(x, \lambda_0) = \begin{cases} 0 & \text{for } x = 1', 2', \dots \\ 1 & \text{for other } x \in \mathcal{L}. \end{cases}$$

Clearly,  $F^*(\lambda_i) = \{1, 2, \dots, i\}$  for  $i = 1, 2, \dots$ , and  $F^*(\lambda_0) = \{1', 2', \dots\}$ . Condition B holds in this example. For instance, consider  $5 \in F^*(\lambda_5)$  and  $2' \in F^*(\lambda_0)$ .  $5' = 5 \vee 2'$  is in  $F^*(\lambda_0)$  and  $2 = 5 \wedge 2'$  is in  $F^*(\lambda_5)$ . Suppose that initially one has a totally ordered sub-selection  $x(\cdot)$  on the infinite set  $\Lambda_0 = \{\lambda_1, \lambda_2, \dots\}$ , where  $x(\lambda_i) = i$  for  $i = 1, 2, \dots$ . Notice that  $\lambda_0$  is not in  $\Lambda_0$ . It can be seen that for any  $x(\lambda_0) \in F^*(\lambda_0)$ , the collection  $\{x(\lambda_i) | i = 0, 1, 2, \dots\}$  cannot be totally ordered. On the other hand, consider a finite  $\Lambda_0$ , say,  $\{\lambda_1, \lambda_2, \lambda_3\}$ , and again let  $x(\lambda_i) = i$  for  $i = 1, 2, 3$ . Now one can select  $x(\lambda_0) = 3'$

and  $x(\lambda) = 3$  for  $\lambda = \lambda_4, \lambda_5, \dots$ , and the collection  $\{x(\lambda_i) | i = 0, 1, 2, \dots\}$  is totally ordered.

The next theorem uses the weaker Condition C, and ensures that the mapping  $x(\cdot)$  can be extended when  $\Lambda_0$  contains a single element.

**THEOREM 2.9 (Weak Totally Ordered Selection Theorem)** *Let  $\mathcal{L}$  be a lattice,  $\Lambda$  be an arbitrary set, and let  $f(\cdot, \cdot)$  be defined on  $\mathcal{L} \times \Lambda$ . Suppose Condition C holds, and a minimum element  $s(\lambda)$  and a maximum element  $S(\lambda)$  exist in every  $F^*(\lambda)$ . Then the collections  $\{s(\lambda)\}$  and  $\{S(\lambda)\}$  are totally ordered. Furthermore, for an arbitrary  $\lambda_0 \in \Lambda$  and an arbitrary  $x_0 \in F^*(\lambda_0)$ , a mapping  $x(\lambda)$  from  $\Lambda$  to  $\mathcal{L}$  can be constructed such that  $x(\lambda_0) = x_0$ ,  $x(\lambda) \in F^*(\lambda)$ , and the collection  $\mathcal{C} = \{x(\lambda) | \lambda \in \Lambda\}$  is totally ordered.*

**Proof.** To prove the first claim, let  $s(\lambda_1)$  and  $s(\lambda_2)$  be minimum elements in  $F^*(\lambda_1)$  and  $F^*(\lambda_2)$ , respectively. By Condition C, one can find  $u \preceq s(\lambda_1) \wedge s(\lambda_2)$  such that either  $u \in F^*(\lambda_1)$  or  $u \in F^*(\lambda_2)$ . If the former holds, then since  $u \preceq s(\lambda_1)$ , it follows that  $u = s(\lambda_1) = s(\lambda_1) \wedge s(\lambda_2)$  and so  $s(\lambda_1) \preceq s(\lambda_2)$ . Similarly, if  $u \in F^*(\lambda_2)$ , then  $s(\lambda_2) \preceq s(\lambda_1)$ . This argument shows that the collection  $\{s(\lambda)\}$  is totally ordered, and one can similarly demonstrate that the collection  $\{S(\lambda)\}$  is totally ordered.

I now prove the second claim in the theorem. For each  $\lambda \neq \lambda_0$ , by Lemma 2.6, either  $s(\lambda) \preceq x_0$  or  $x_0 \preceq S(\lambda)$ . If the former is valid, then let  $x(\lambda) = s(\lambda)$ , otherwise, let  $x(\lambda) = S(\lambda)$ . The minimum elements precede  $x_0$  which in turn, precedes all the maximum elements with respect to the  $\preceq$  order. From the first half of this theorem, all the minimum elements selected are ordered among themselves and the same is true for all the maximum elements. Thus the selection  $x(\cdot)$  is totally ordered.  $\square$

Condition C is not sufficient to ensure that a totally ordered sub-selection of optimal solutions on an arbitrary  $\Lambda_0$  can be extended to a complete one. Indeed, consider the

counter example stemming from Figure 2.3. Again, elements of  $\mathcal{L}$  are represented by the solid nodes, and if two elements satisfy  $i \preceq j$ , then there is a directed path from  $j$  to  $i$ .  $\mathcal{L}$  is clearly a lattice. Suppose that  $\Lambda = \{1, 2, 3\}$ ,  $F^*(1) = \{x_1, x_3\}$ ,  $F^*(2) = \{m, M\}$  and  $F^*(3) = \{x_2, x_4\}$ . It is not difficult to verify that Condition C is valid. For example, for  $x_1 \in F^*(1)$  and  $m \in F^*(2)$ , one can let  $u = x_3 = x_1 \wedge m$  and  $v = M \succeq x_5 = x_1 \vee m$ , thus  $u \in F^*(1)$  and  $v \in F^*(2)$ , and either (2.3) or (2.4) is satisfied as an equality. Suppose initially one is given a mapping  $x(\cdot)$  on  $\Lambda_0 = \{1, 3\}$  where  $x(1) = x_1$  and  $x(3) = x_2$ . No matter how one chooses  $x(2)$  (either  $m$  or  $M$ ), the collection  $\mathcal{C} = \{x(\lambda) | \lambda = 1, 2, 3\}$  cannot be totally ordered.

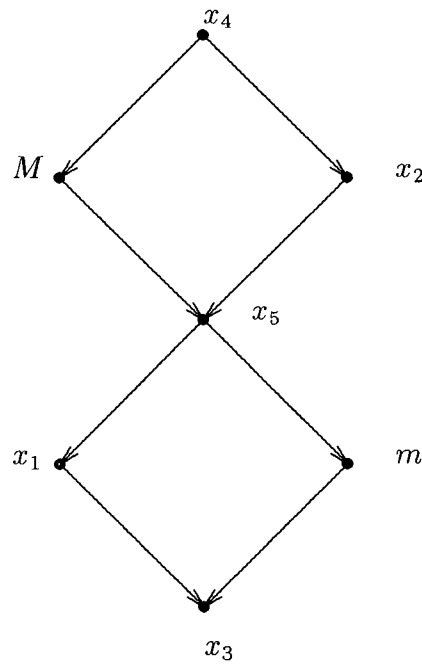


Figure 2.3: An example for which a sub-selection cannot be extended

## 2.2 Ordered Minimum Cuts in Parametric Maximum Flow Problems

In this section, I apply the Totally Ordered Selection Theorems to derive a necessary and sufficient condition for the existence of a totally ordered selection of minimum cuts in a parametric maximum flow problem.

### 2.2.1 Monotone minimum cuts

My result is related to a parametric maximum flow problem studied by GGT (1989). Prior to the development of the GGT algorithm in this study, Stone (1978) observed that the sets of minimum cuts in this parametric maximum flow problem are ascending in the parameter. Theorem 2.10 below presents this result. Here I show that it follows easily from the Monotone Selection Theorem.

**THEOREM 2.10** *Let  $G_\lambda(N, A)$  be a network in which the capacity,  $c(\lambda; s, i)$ , of each arc  $(s, i)$  leaving the source is a nonnegative, nondecreasing function of  $\lambda \in \mathbb{R}$ , the capacity,  $c(\lambda; j, t)$ , of each arc  $(j, t)$  entering the sink is a nonnegative, nonincreasing function of  $\lambda$ , and the capacities of all other arcs are constant. Let  $F^*(\lambda)$  be the set of minimum cuts for an instance of the parameter  $\lambda$ . Then  $F^*(\lambda)$  is a sublattice that is ascending in  $\lambda$ .*

My alternative proof<sup>1</sup> of Theorem 2.10 is based on the following two lemmas, where  $f(X, \lambda)$  denotes the capacity of  $(X, \overline{X})$  under parameter  $\lambda$ . Lemma 2.11 appears in Ore (1962).

**LEMMA 2.11**  *$f(X, \lambda)$  is submodular in  $X$  for a fixed  $\lambda$ .*

---

<sup>1</sup>I suspect that this alternative proof is not new, but I do not have a reference for it. I present this proof because it is closely related to my study.

**Proof.** Let  $(X, \overline{X})$  and  $(Y, \overline{Y})$  be two cuts in  $G$ . I partition the set of nodes  $N$  as follows:  $N_1 = X \cap Y$ ,  $N_2 = X \setminus Y$ ,  $N_3 = Y \setminus X$ ,  $N_4 = \overline{X} \cap \overline{Y}$ . Then  $(X \cap Y, \overline{X \cap Y}) = (N_1, \overline{N_1})$  and  $(X \cup Y, \overline{X \cup Y}) = (\overline{N_4}, N_4)$ . Observe that

$$f(X, \lambda) = c(\lambda; N_1, N_3) + c(\lambda; N_1, N_4) + c(\lambda; N_2, N_3) + c(\lambda; N_2, N_4),$$

$$f(Y, \lambda) = c(\lambda; N_1, N_2) + c(\lambda; N_1, N_4) + c(\lambda; N_3, N_2) + c(\lambda; N_3, N_4),$$

$$f(X \cap Y, \lambda) = c(\lambda; N_1, N_2) + c(\lambda; N_1, N_3) + c(\lambda; N_1, N_4),$$

$$f(X \cup Y, \lambda) = c(\lambda; N_1, N_4) + c(\lambda; N_2, N_4) + c(\lambda; N_3, N_4).$$

Then,  $f(X, \lambda) + f(Y, \lambda) - f(X \cap Y, \lambda) - f(X \cup Y, \lambda) = c(\lambda; N_2, N_3) + c(\lambda; N_3, N_2) \geq 0$ , which implies the submodularity of  $f(\cdot, \lambda)$  for a fixed  $\lambda$ .  $\square$

**LEMMA 2.12**  $f(X, \lambda)$  has antitone differences.

**Proof.** For two cuts  $(X, \overline{X})$  and  $(Y, \overline{Y})$  such that  $X \subseteq Y$ ,

$$f(Y, \lambda) - f(X, \lambda) = c(\lambda; Y \setminus X, \overline{Y}) - c(\lambda; X, Y \setminus X). \quad (2.5)$$

In  $(X, Y \setminus X)$ , the only arcs with variable capacities are those incident to  $s$  which have nondecreasing capacities as functions of  $\lambda$ . In  $(Y \setminus X, \overline{Y})$ , the only arcs with variable capacities are those incident to  $t$  which have nonincreasing capacities as functions of  $\lambda$ . Thus (2.5) is nonincreasing in  $\lambda$ .  $\square$

**Proof of Theorem 2.10.** The conditions set forth in Theorem 2.1, namely the submodularity and antitone differences conditions for  $f(X, \lambda)$ , are verified in Lemmas 2.11 and 2.12.  $\square$

### 2.2.2 Parallel Arcs in Maximum Flow Problems

In this section, I introduce the notion of parallel arcs, which is independent of arc capacities and only depends on the network topology. My study of ordered minimum cuts in the next subsection will depend on this relationship.

**DEFINITION 2.1** Two arcs  $(i, j)$  and  $(k, l)$  are *parallel* if there exist simple paths  $p_1$  from  $s$  to  $i$ ,  $p_2$  from  $s$  to  $k$ ,  $p_3$  from  $j$  to  $t$ , and  $p_4$  from  $l$  to  $t$  such that  $p_1 \cap p_4 = \emptyset$ ,  $p_2 \cap p_3 = \emptyset$ ,  $p_1 \cap p_3 = \emptyset$  and  $p_2 \cap p_4 = \emptyset$ . In this case, I call the subgraph of  $G$  spanned by  $(i, j)$ ,  $(k, l)$ ,  $p_1$ ,  $p_2$ ,  $p_3$  and  $p_4$  a *bypass* between  $(i, j)$  and  $(k, l)$ .

Intuitively, if two arcs  $(i, j)$  and  $(k, l)$  are parallel in  $G$ , then there exists a *simple* s-t path  $p_a = \langle p_1, (i, j), p_3 \rangle$  from  $s$  to  $t$  that bypasses  $(k, l)$  and another *simple* s-t path  $p_b = \langle p_2, (k, l), p_4 \rangle$  from  $s$  to  $t$  that bypasses  $(i, j)$ .

Parallel arcs are illustrated in Figure 2.4, where arcs  $(i, j)$  and  $(k, l)$  (shown in bold lines) are parallel.

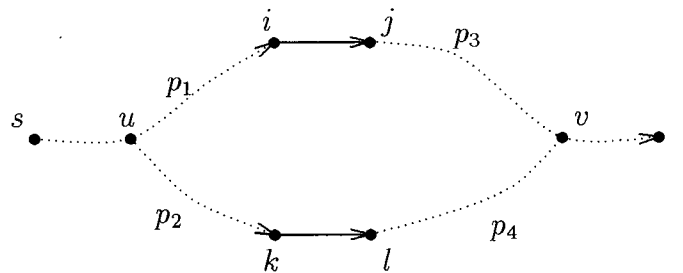


Figure 2.4: Parallel arcs

For an arbitrary directed network, verifying whether a pair of arcs are parallel is NP-hard. This is due to the fact that finding two vertex-disjoint paths between two pairs of nodes in directed graphs is NP-hard. See Fortune, Hopcroft and Wyllie (1980).

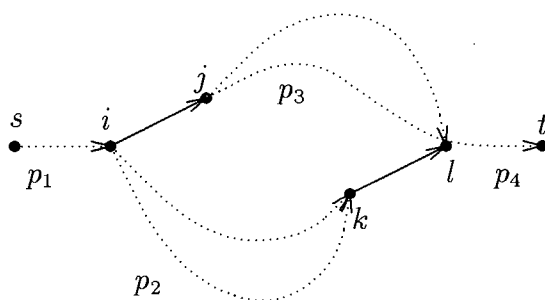
**DEFINITION 2.2** Let  $(i, j)$  and  $(k, l)$  be two parallel arcs in a directed network  $G$ . Arc  $(i, j)$  is called a *front arc* with respect to  $(k, l)$  if in every bypass between  $(i, j)$  and  $(k, l)$ ,  $i$  is the last common node in  $p_1$  and  $p_2$ . Arc  $(k, l)$  is called a *back arc* with respect to  $(i, j)$  if in every bypass between  $(i, j)$  and  $(k, l)$ ,  $l$  is the first common node in  $p_3$  and  $p_4$ .

Figure 2.5 illustrates the notion of front and back parallel arcs. In part (a),  $(i, j)$  is a front arc with respect to  $(k, l)$  and  $(k, l)$  is a back arc with respect to  $(i, j)$ . In part (b),  $(i, j)$  is a front arc with respect to  $(k, l)$ , but  $(k, l)$  is not a back arc with respect to  $(i, j)$ , since in a bypass spanned by the two arcs and  $p_1, p_2, p'_3$  and  $p_4$ ,  $l$  is not on  $p'_3$ . In part (c), arc  $(i, j)$  is both a front arc and a back arc with respect to  $(k, l)$ .

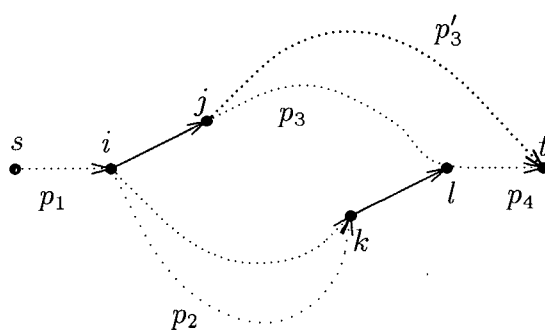
**LEMMA 2.13** Let  $G(N, A)$  be a directed network with positive arc capacities. If an arc  $(i, j)$  is in a minimum cut  $(X, \overline{X})$ , then there exist simple paths  $p_1$  from  $s$  to  $i$  and  $p_2$  from  $j$  to  $t$  such that  $p_1$  is contained in  $X$  and  $p_2$  is contained in  $\overline{X}$ .

**Proof.** In a maximum flow, all forward arcs in a minimum cut are saturated. Thus any arc  $(i, j)$ ,  $i \in X$  and  $j \in \overline{X}$ , is saturated in a maximum flow. Since the capacities are strictly positive, the flow on arc  $(i, j)$  is positive. By the arc-chain representation of the flow, there must exist a simple path  $p$  from  $s$  to  $t$  containing  $(i, j)$ , such that the flow  $h(p)$  on this path is strictly positive. Since in a maximum flow, the flow on every arc crossing a minimum cut backwards is zero, path  $p$  can only cross  $(X, \overline{X})$  exactly once. Let  $p_1$  be the sub-path of  $p$  from  $s$  to  $i$  and  $p_2$  be the sub-path of  $p$  from  $j$  to  $t$ . Since  $p$  crosses  $(X, \overline{X})$  only once,  $p_1$  and  $p_2$  have the desired property.  $\square$

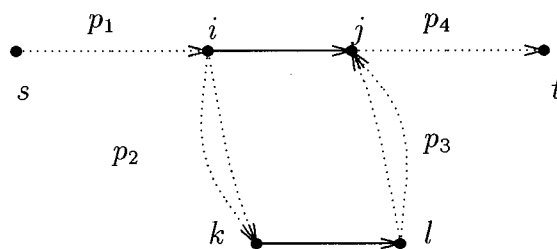




(a)



(b)



(c)

Figure 2.5: Front and back arcs

### 2.2.3 Totally Ordered Selections of Minimum Cuts

In this subsection, I consider parametric maximum flow problems in a directed network  $G(N, A)$  with positive arc capacities. Each instance of the parametric maximum flow problem corresponds to a capacity function  $c(\cdot; \cdot) : \Lambda \times A \mapsto \mathbb{R}^+$  such that  $c(\lambda; e)$  is the capacity of arc  $e$  under parameter  $\lambda$ .

GGT (1989) showed that for any parametric maximum flow problem in which the capacities of arcs leaving the source node are increasing functions of  $\lambda$ , and the capacities of arcs entering the sink node are decreasing functions of  $\lambda$ , the sets of minimum cuts are ascending in  $\lambda$ . Notice that this result does not pertain to any specific parametric capacity function  $c(\cdot; \cdot)$  on  $G$ . Rather, a class of parametric capacity functions is specified such that the aforementioned property holds for every capacity function in this class.

My main concern in this subsection is to characterize classes of parametric capacity functions such that for *every* capacity function in these classes a totally ordered selection of minimum cuts exists. To this end, I use the following two ways to specify classes of parametric capacity functions.

**Type I.** For a directed network  $G(N, A)$ , a class of parametric capacity functions can be determined by specifying the set of arcs  $A_v \in A$ , whose capacities may change. Formally, a *Type I class* of parametric functions is defined by:

$$\mathcal{F}(A_v) = \{c(\cdot; \cdot) \mid c(\cdot; e) \text{ is constant for each arc } e \in A \setminus A_v\}.$$

**Type II.** For a network  $G(N, A)$ , a class of parametric capacity functions can be determined by specifying the set of arcs  $A_v \in A$ , whose capacities may change, the set of arcs  $A_v^+ \subseteq A_v$ , whose capacities may increase in  $\lambda$ , and the set of arcs  $A_v^- \subseteq A_v$ , whose capacities may decrease in  $\lambda$ . Formally, a *Type II class* of parametric capacity

functions is defined as follows:

$$\begin{aligned} \mathcal{F}(A_v, A_v^+, A_v^-) \\ = \{c(\cdot; \cdot) \mid & c(\cdot; e) \text{ is constant for each arc } e \in A \setminus A_v, \\ & c(\lambda; e) \text{ is a nondecreasing function of } \lambda \text{ for each arc } e \in A_v^+, \\ & c(\lambda; e) \text{ is a nonincreasing function of } \lambda \text{ for each arc } e \in A_v^-\}, \end{aligned}$$

where  $A_v^+$  and  $A_v^-$  are both subsets of  $A_v$ , and  $A_v^+ \cap A_v^- = \emptyset$ . For any  $c(\cdot; \cdot) \in \mathcal{F}(A_v, A_v^+, A_v^-)$ , there is no restriction on the parametric changes on any arc  $e \in A_v \setminus (A_v^+ \cup A_v^-)$ . Thus for an arc  $e \in A_v \setminus (A_v^+ \cup A_v^-)$ ,  $c(\lambda; e)$  can be an arbitrary function of  $\lambda$ .

Although the Type I class is a special case of Type II class, I distinguish between them for clarity of presentation. Special cases of the class of parametric capacity functions  $\mathcal{F}(A_v, A_v^+, A_v^-)$  were considered in the literature. Indeed, GGT considered a class of parametric capacity functions  $\mathcal{F}(A_v, A_v^+, A_v^-)$ , where  $A_v^+ = \{(s, i) \mid (s, i) \in A, \forall i \in N\}$ ,  $A_v^- = \{(i, t) \mid (i, t) \in A, \forall i \in N\}$  and  $A_v = A_v^+ \cup A_v^-$ . (I assume that arc  $(s, t)$  does not exist in the network.) They showed that for every capacity function in  $\mathcal{F}(A_v, A_v^+, A_v^-)$ , the sets of minimum cuts are ascending in  $\lambda$ . AUK examined the class of capacity functions  $\mathcal{F}(A_v, A_v^+, A_v^-)$ , where  $A_v^+ = \{(v, i) \mid (v, i) \in A, \forall i \in N\} \cup \{(i, v) \mid (i, v) \in A, \forall i \in N\}$ ,  $A_v^- = \emptyset$  and  $A_v = A_v^+$  for a specific node  $v$  called the “center node”. They showed that for every capacity function in  $\mathcal{F}(A_v, A_v^+, A_v^-)$ , a totally ordered selection of minimum cuts exists.

The classes of parametric capacity functions that I propose to study are broad enough. They subsume the classes of parametric capacity functions studied by GGT and AUK as special cases. Observe that in the above two types of classes, I only restrict the set of arcs whose capacities may change and the direction of such a change. Another possible way of defining a class of parametric functions is to further specify restrictions on the

magnitude of the changes<sup>2</sup>, which is not studied in this thesis.

My goal is to characterize the sets  $A_v$ ,  $A_v^+$  and  $A_v^-$  which will ensure the existence of a totally ordered selection of minimum cuts. Indeed, Theorem 2.14 below provides a necessary and sufficient condition for the existence of a totally ordered selection of minimum cuts for every capacity function in the class  $\mathcal{F}(A_v, A_v^+, A_v^-)$ .

**THEOREM 2.14** *Let  $G(N, A)$  be a directed network and  $\mathcal{F}(A_v, A_v^+, A_v^-)$  be a Type II class of parametric capacity functions. A totally ordered selection of minimum cuts exists for every capacity function  $c(\cdot, \cdot)$  in  $\mathcal{F}(A_v, A_v^+, A_v^-)$  if and only if for each pair of parallel arcs  $e_1$  and  $e_2$  in  $A_v$ , at least one of the following conditions is valid:*

- (i)  $e_1$  and  $e_2$  are either both front arcs or both back arcs, and  $e_1$  and  $e_2$  are either both in  $A_v^+$  or both in  $A_v^-$ .
- (ii) Either  $e_1$  or  $e_2$  is a front arc and the other is a back arc, and one of them is in  $A_v^+$  and the other is in  $A_v^-$ .
- (iii) Either  $e_1$  or  $e_2$  is both a front arc and a back arc.

Furthermore, if any of the above conditions are satisfied, any totally ordered sub-selection of minimum cuts can be extended to a totally ordered selection.

**Proof of the “only if” Part.** Suppose there exist a pair of parallel arcs  $e_1 = (i, j)$  and  $e_2 = (k, l)$  in  $A_v$  such that none of conditions (i), (ii) or (iii) is satisfied. It suffices to construct a capacity function  $c(\cdot; \cdot) \in \mathcal{F}(A_v, A_v^+, A_v^-)$  for which a totally ordered selection of minimum cuts does not exist. Since  $e_1$  and  $e_2$  do not satisfy condition (iii), neither of them is both a front arc and a back arc. Since neither condition (i) nor condition (ii) holds, at least one of the following three cases must be valid.

---

<sup>2</sup>See McCormick (1995) for an application of the GGT method to some parametric maximum flow problems where the magnitude of parametric capacity change is restricted.

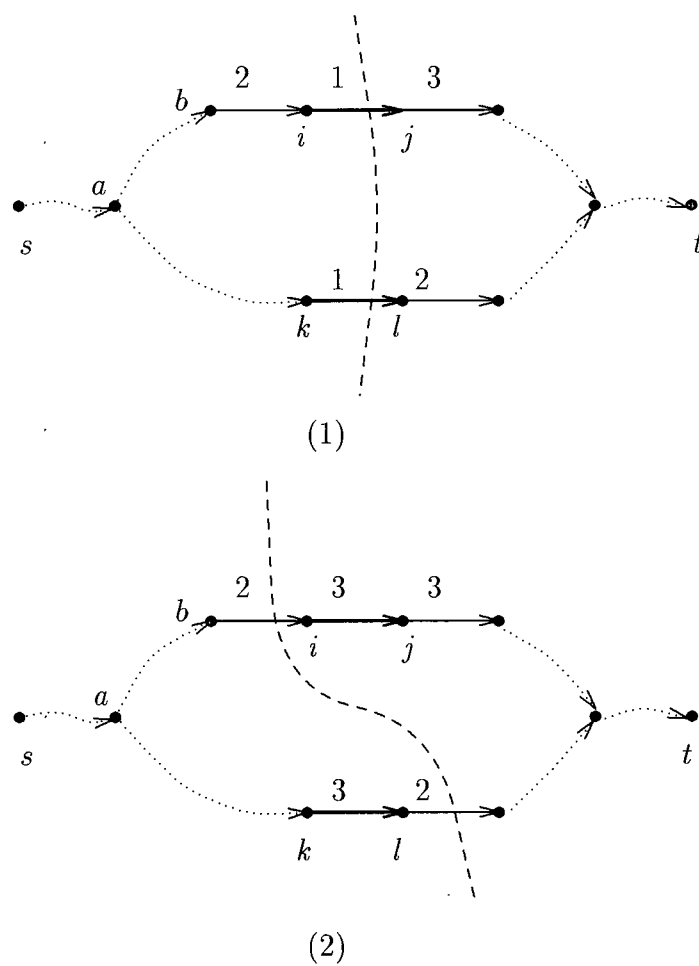
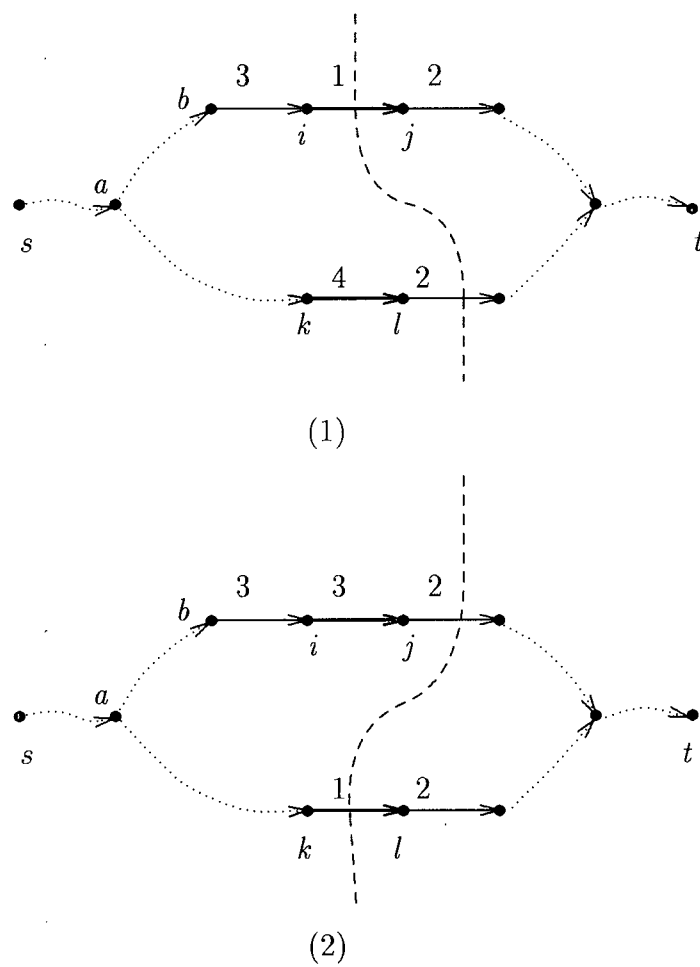


Figure 2.6:  $G'$  for Case 1

Figure 2.7:  $G'$  for Case 1

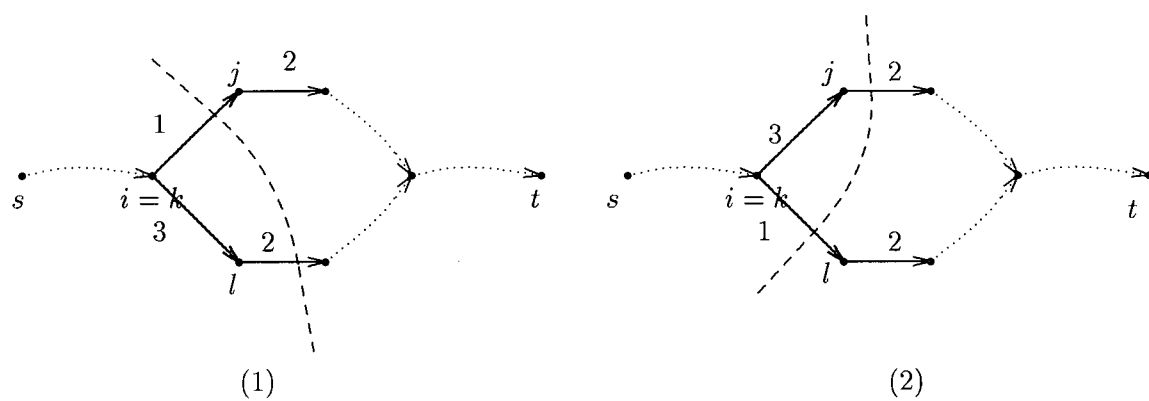


Figure 2.8:  $G'$  for Case 2

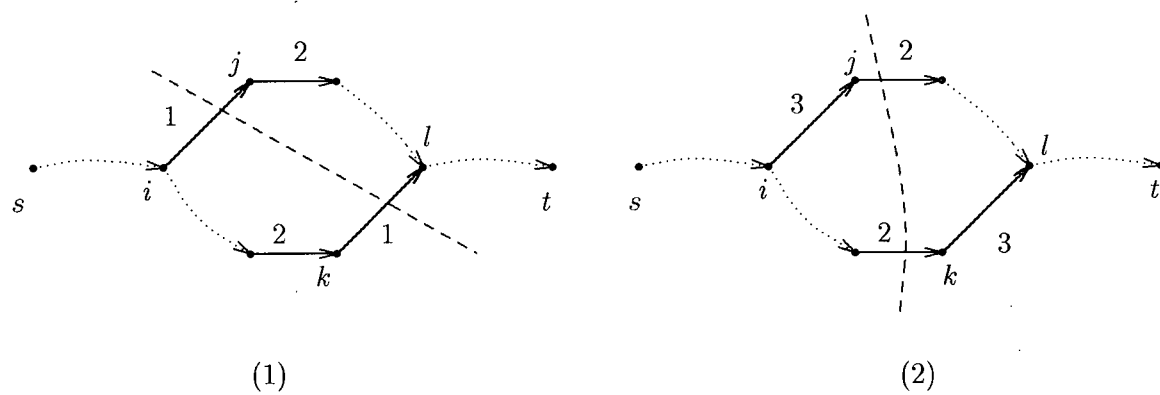


Figure 2.9:  $G'$  for Case 3

**Case 1** At least one of the arcs  $e_1$  and  $e_2$  is neither a front arc nor a back arc.

**Case 2**  $e_1$  and  $e_2$  are either both front arcs or both back arcs (i.e.  $i = j$  or  $k = l$ ), and the capacities of  $e_1$  and  $e_2$  are allowed to change in opposite directions.

**Case 3** Either  $e_1$  or  $e_2$  is a front arc and the other is a back arc, and their capacities are allowed to change in the same direction.

I will now construct a parametric capacity function  $c(\cdot; \cdot)$  for each of the above three cases. Since  $e_1 = (i, j)$  and  $e_2 = (k, l)$  are parallel, I can find simple paths  $p_1$  from  $s$  to  $i$ ,  $p_2$  from  $s$  to  $k$ ,  $p_3$  from  $j$  to  $t$  and  $p_4$  from  $l$  to  $t$  such that  $p_1 \cap p_4 = \emptyset$ ,  $p_1 \cap p_3 = \emptyset$ ,  $p_2 \cap p_3 = \emptyset$  and  $p_2 \cap p_4 = \emptyset$ . Let  $G'$  be the subgraph of  $G$  spanned by arcs  $(i, j)$ ,  $(k, l)$  and paths  $p_1$ ,  $p_2$ ,  $p_3$  and  $p_4$ . Assign very small constant capacities to arcs not in  $G'$  so that the total capacity of these arcs is less than 1. Consider a minimum cut,  $(X, \overline{X})$ , in  $G$ . It disconnects  $t$  from  $s$  in  $G'$  as well as in  $G$ . Thus  $(X, \overline{X})$  contains a subset of arcs,  $A'$ , which forms a cut set of arcs for  $G'$ . In the capacity function that I will construct, each arc in  $G'$  will be assigned an integer capacity greater than or equal to 1. Therefore,  $A'$  must be a minimum cut set of arcs for  $G'$ . Otherwise, I can augment a minimum cut in  $G'$  with all arcs in  $G \setminus G'$  to form a new cut in  $G$ , whose capacity is smaller than the capacity of  $(X, \overline{X})$ .

If for two instances of the parameter,  $\lambda_1 < \lambda_2$ , the corresponding minimum cuts are unique and cannot be ordered, then the minimum cuts in  $G$  for  $\lambda_1$  and  $\lambda_2$  cannot be ordered. Thus it suffices to show that unique minimum cuts cannot be ordered in  $G'$  for two values,  $\lambda_1$  and  $\lambda_2$  in  $\Lambda$ .

In the following counter example, I assign constant values to capacities of arcs other than  $e_1$  and  $e_2$  in  $G'$ .  $G'$  corresponding to Case 1 is illustrated in Figures 2.6 and 2.7.  $G'$  corresponding to Cases 2 and 3 is illustrated in Figures 2.8 and 2.9, respectively. The dotted lines represent paths in  $G'$ . Each such path may be empty. Let  $c(\lambda; e) = \infty$  for



each arc  $e$  in the paths represented by the dotted lines and for each  $\lambda \in \mathbb{R}$ . Values of  $c(\lambda_1; \cdot)$  and  $c(\lambda_2; \cdot)$  for other arcs in  $G'$  are as follows.

In Case 1, if the capacities of  $e_1$  and  $e_2$  are allowed to change in the same direction, then the values of  $c(\lambda_1; \cdot)$  and  $c(\lambda_2; \cdot)$  are shown in Figure 2.6. If the capacities of  $e_1$  and  $e_2$  are allowed to change in opposite directions, then the values of  $c(\lambda_1; \cdot)$  and  $c(\lambda_2; \cdot)$  are shown in Figure 2.7. In Case 2 and Case 3, the values of  $c(\lambda_1; \cdot)$  and  $c(\lambda_2; \cdot)$  are illustrated, respectively, in Figures 2.8 and 2.9. In each figure, part (1) corresponds to  $\lambda_1$  and part (2) corresponds to  $\lambda_2$ . It can be seen that the direction of change in capacities of  $e_1$  and  $e_2$  is consistent with the one set forth in the corresponding case.

The dashed lines represent the minimum cuts. In each case, the unique minimum cuts for  $\lambda_1$  and  $\lambda_2$  are not ordered. This completes the “only if” part of the proof.  $\square$

**Proof of the “if” Part.** Suppose  $\mathcal{F}(A_v, A_v^+, A_v^-)$  satisfies the hypothesis in the theorem. Consider an arbitrary parametric capacity function  $c(\cdot; \cdot) \in \mathcal{F}(A_v, A_v^+, A_v^-)$ . I will show that Condition B holds for an arbitrary pair of minimum cuts  $(X, \bar{X}) \in F^*(\lambda_1)$  and  $(Y, \bar{Y}) \in F^*(\lambda_2)$ , and by Corollary 2.8, it would then follow that a totally ordered selection of minimum cuts can be obtained by extending an arbitrary totally ordered sub-selection of minimum cuts.

Let  $N_1 = X \cap Y$ ,  $N_2 = X \setminus Y$ ,  $N_3 = Y \setminus X$ ,  $N_4 = \bar{X} \cap \bar{Y}$ . It will be convenient to assume that for every node  $i$  in  $N_1$ , there exists a path from  $s$  to  $i$  that is contained in  $N_1$ ; and similarly, for every node  $j$  in  $N_4$  there exists a path from  $j$  to  $t$  that is contained in  $N_4$ . (The motivation will be clear later.) This assumption does not hold in general. However, sets  $N'_1$  and  $N'_4$  with the desired property can be constructed from  $N_1, \dots, N_4$  as follows. Let  $N'_1$  be the set of nodes in  $N_1$  such that if  $i \in N'_1$  then there exists a path from  $s$  to  $i$  which is contained in  $N_1$ . Similarly, let  $N'_4$  be the set of nodes in  $N_4$  such that if  $i \in N'_4$  then there exists a path from  $i$  to  $t$  which is contained in  $N_4$ . Let

$N'_2 = N_2 \cup (N_1 \setminus N'_1)$  and  $N'_3 = N_3 \cup (N_4 \setminus N'_4)$ . Let  $X' = N'_1 \cup N'_2$  and  $Y' = N'_1 \cup N'_3$ . Clearly,  $X' = X$  and  $(X', \overline{X'}) = (X, \overline{X})$ . The sets  $N_1$  through  $N_4$  and  $N'_1$  through  $N'_4$  are illustrated in Figure 2.10. Observe that  $N'_2 \setminus N_2 = N_1 \setminus N'_1$  and  $N'_3 \setminus N_3 = N_4 \setminus N'_4$ .

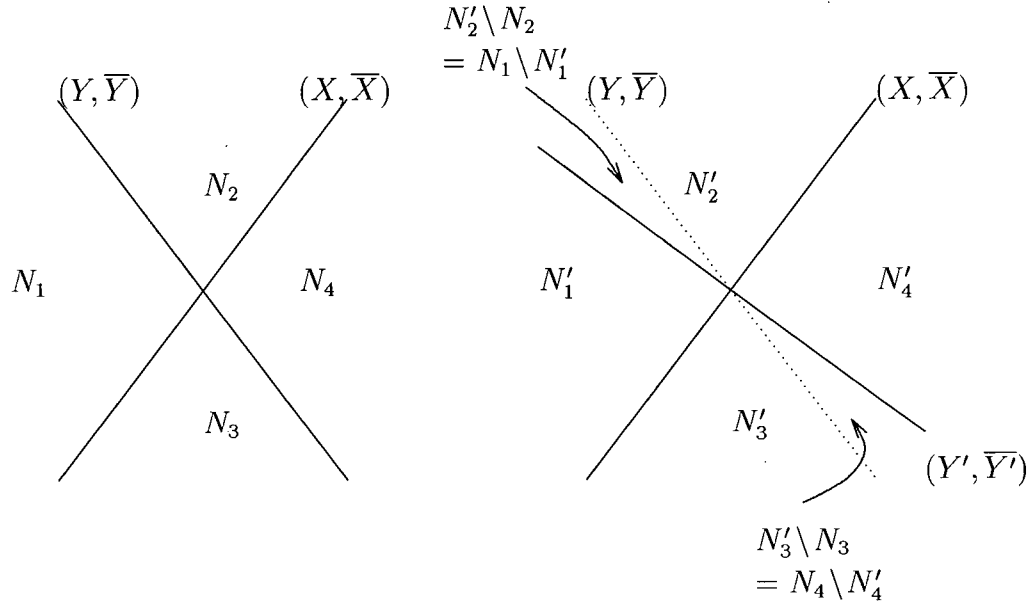


Figure 2.10: Partition of the node set

**Claim 2.1** *There does not exist an arc  $(i, j)$  such that  $i \in N'_1$  and  $j \in (N_1 \setminus N'_1)$ . Similarly, there does not exist an arc  $(i, j)$  such that  $i \in N_4 \setminus N'_4$  and  $j \in N'_4$ .*

**Proof.** Consider the first half of the claim. If an arc  $(i, j)$  exists such that  $i \in N'_1$  and  $j \in (N_1 \setminus N'_1)$ , then by the definition of  $N'_1$ , there exists a path  $p$  from  $s$  to  $i$  that is contained in  $N_1$ . Then the path  $\langle p, (i, j) \rangle$  is contained in  $N_1$  too, and node  $j$  should also be in  $N'_1$ , leading to a contradiction. The second half of the claim can be proved analogously.  $\square$

**Claim 2.2** *For each node  $i \in N'_1$ , there exists a path from  $s$  to  $i$  that is contained in  $N'_1$ . Similarly, for each node  $j \in N'_4$ , there exists a path from  $j$  to  $t$  that is contained in  $N'_4$ .*

**Proof.** By the construction of  $N'_1$ , there exists a path  $p$  from  $s$  to  $i$  that is contained in  $N_1$ . If  $p$  is not contained in  $N'_1$ , it must contain a forward arc in either  $(X', \overline{X'})$  or  $(Y', \overline{Y'})$ . Since  $p$  is contained in  $N_1$ , it cannot contain an arc in  $(X, \overline{X}) = (X', \overline{X'})$ . Suppose  $p$  contains an arc  $(i, j)$  in  $(Y', \overline{Y'})$ . Since  $p$  is contained in  $N_1$ ,  $i$  and  $j$  are also in  $N_1$ . Since  $N_1 \cap Y' = N'_1$  and  $N_1 \cap \overline{Y'} = N_1 \setminus N'_1$ , it follows that  $i \in N'_1$  and  $j \in N_1 \setminus N'_1$ . However, this contradicts Claim 2.1 and the first half of Claim 2.2 is established. Proof of the second half of the Claim 2.2 is similar.  $\square$

Let  $A'_1 = (N'_1, N'_3) \cup (N'_1, N'_4) \cup (N'_2, N'_4)$  and let  $A'_2 = (N'_1, N'_2) \cup (N'_1, N'_4) \cup (N'_3, N'_4)$ . Similarly, let  $A_1 = (N_1, N_3) \cup (N_1, N_4) \cup (N_2, N_4)$  and let  $A_2 = (N_1, N_2) \cup (N_1, N_4) \cup (N_3, N_4)$ . Clearly,  $A'_1 = (X', \overline{X'}) \setminus (N'_2, N'_3)$  and  $A'_2 = (Y', \overline{Y'}) \setminus (N'_3, N'_2)$ . Similarly,  $A_1 = (X, \overline{X}) \setminus (N_2, N_3)$  and  $A_2 = (Y, \overline{Y}) \setminus (N_3, N_2)$ .

**Claim 2.3**

- (i)  $(N'_1, N'_2) \subseteq (N_1, N_2)$ ,
- (ii)  $(N'_1, N'_3) \subseteq (N_1, N_3) \cup (N_1, N_4)$ ,
- (iii)  $(N'_2, N'_4) \subseteq (N_2, N_4) \cup (N_1, N_4)$ ,
- (iv)  $(N'_3, N'_4) \subseteq (N_3, N_4)$ ,
- (v)  $(N'_1, N'_4) \subseteq (N_1, N_4)$ ,
- (vi)  $A'_1 \subseteq A_1$ ,
- (vii)  $A'_2 \subseteq A_2$ ,

$$(viii) \quad (N'_1, \overline{N'_1}) \subseteq (N_1, N_2) \cup (N_1, N_3) \cup (N_1, N_4) \text{ and } (\overline{N'_4}, N'_4) \subseteq (N_1, N_4) \cup (N_2, N_4) \cup (N_3, N_4).$$

**Proof.** To prove (i), consider an arc  $(i, j) \in (N'_1, N'_2)$ . If  $j \in N_2$ , then since  $i \in N'_1 \subseteq N_1$ ,  $(i, j) \in (N_1, N_2)$ . Otherwise,  $j \in N'_2 \setminus N_2 = N_1 \setminus N'_1$ , but this is impossible since it violates Claim 2.1.

To prove (ii), suppose arc  $(i, j) \in (N'_1, N'_3)$ . Since  $i \in N'_1 \subseteq N_1$ ,  $i \in N_1$ . If  $j \in N_3$ , then  $(i, j) \in (N_1, N_3)$ . Otherwise,  $j \in N'_3 \setminus N_3 = N_4 \setminus N'_4 \subseteq N_4$ , and  $(i, j) \in (N_1, N_4)$ .

(iii) and (iv) can be proved analogously. (v) follows from the fact that  $N'_1 \subseteq N_1$  and  $N'_4 \subseteq N_4$ . (vi), (vii) and (viii) follow directly from (i) through (v). This completes the proof of Claim 2.3.  $\square$

Recall that for two arbitrary sets of nodes  $I$  and  $J$ ,  $c(\lambda; I, J)$  denotes the total capacity of arcs in  $(I, J) = \{(i, j) | i \in I, j \in J\}$  under the parameter  $\lambda$ . Now let  $c'(\lambda; I, J)$  be the total capacity of arcs in  $(I, J) \setminus (N_1, N_4)$  under the parameter  $\lambda$ . For example,  $c'(\lambda_1; N'_1, N'_3)$  would be the total capacity of arcs in  $(N'_1, N'_3) \setminus (N_1, N_4)$  under the parameter  $\lambda_1$ .

#### Claim 2.4

$$c(\lambda_1; A'_1) \leq c(\lambda_1; A_1), \quad (2.6)$$

$$c(\lambda_2; A'_2) \leq c(\lambda_2; A_2), \quad (2.7)$$

$$c'(\lambda_1; A'_1) \leq c'(\lambda_1; A_1), \quad (2.8)$$

$$c'(\lambda_2; A'_2) \leq c'(\lambda_2; A_2). \quad (2.9)$$

**Proof.** The proof follows from the fact that  $A'_1 \subseteq A_1$ ,  $A'_2 \subseteq A_2$ ,  $A'_1 \setminus (N_1, N_4) \subseteq A_1 \setminus (N_1, N_4)$  and  $A'_2 \setminus (N_1, N_4) \subseteq A_2 \setminus (N_1, N_4)$ .  $\square$

**Claim 2.5** *Consider the following inequalities:*

$$c(\lambda_1; N'_1, \overline{N'_1}) + c(\lambda_2; \overline{N'_4}, N'_4) \leq c(\lambda_1; A_1) + c(\lambda_2; A_2) \leq c(\lambda_1; X, \overline{X}) + c(\lambda_2; Y, \overline{Y}), \quad (2.10)$$

$$c(\lambda_2; N'_1, \overline{N'_1}) + c(\lambda_1; \overline{N'_4}, N'_4) \leq c(\lambda_1; A_1) + c(\lambda_2; A_2) \leq c(\lambda_1; X, \overline{X}) + c(\lambda_2; Y, \overline{Y}), \quad (2.11)$$

$$c'(\lambda_1; N'_1, \overline{N'_1}) + c'(\lambda_2; \overline{N'_4}, N'_4) \leq c'(\lambda_1; A_1) + c'(\lambda_2; A_2) \leq c'(\lambda_1; X, \overline{X}) + c'(\lambda_2; Y, \overline{Y}), \quad (2.12)$$

$$c'(\lambda_2; N'_1, \overline{N'_1}) + c'(\lambda_1; \overline{N'_4}, N'_4) \leq c'(\lambda_1; A_1) + c'(\lambda_2; A_2) \leq c'(\lambda_1; X, \overline{X}) + c'(\lambda_2; Y, \overline{Y}). \quad (2.13)$$

*Either (2.10) and (2.12) are valid, or (2.11) and (2.13) are valid.*

**Proof.** The second half of each inequality follows from  $A_1 \subseteq (X, \overline{X})$  and  $A_2 \subseteq (Y, \overline{Y})$ . Thus I only need to prove the first half of the inequalities.

By the definition of  $A_1$  and  $A_2$ , the capacity of each arc in  $(N_1, N_4)$  appears exactly once in  $c(\lambda_1; A_1)$  and once in  $c(\lambda_2; A_2)$ . Thus by the definition of  $c'$ ,

$$c'(\lambda_1; A_1) + c'(\lambda_2; A_2) + c(\lambda_1; N_1, N_4) + c(\lambda_2; N_1, N_4) = c(\lambda_1; A_1) + c(\lambda_2; A_2). \quad (2.14)$$

By Claim 2.3 (viii), the capacity of each arc in  $(N_1, N_4)$  appears at most once in  $c(\lambda_1; N'_1, \overline{N'_1})$  and at most once in  $c(\lambda_2; \overline{N'_4}, N'_4)$ . Thus

$$\begin{aligned} & c(\lambda_1; N'_1, \overline{N'_1}) + c(\lambda_2; \overline{N'_4}, N'_4) \\ & \leq c'(\lambda_1; N'_1, \overline{N'_1}) + c'(\lambda_2; \overline{N'_4}, N'_4) + c(\lambda_1; N_1, N_4) + c(\lambda_2; N_1, N_4), \end{aligned} \quad (2.15)$$

and

$$\begin{aligned} & c(\lambda_2; N'_1, \overline{N'_1}) + c(\lambda_1; \overline{N'_4}, N'_4) \\ & \leq c'(\lambda_2; N'_1, \overline{N'_1}) + c'(\lambda_1; \overline{N'_4}, N'_4) + c(\lambda_2; N_1, N_4) + c(\lambda_1; N_1, N_4). \end{aligned} \quad (2.16)$$

If (2.12) is valid, then the first half of (2.10) can be obtained by summing the first half of (2.12), (2.14) and (2.15). Similarly, if (2.13) is valid, then the first half of (2.11) can be obtained by summing the first half of (2.13), (2.14) and (2.16). Thus I only need to show that either (2.12) or (2.13) is valid.

To show the first half of (2.12) it is sufficient to establish that

$$c'(\lambda_1; N'_1, \overline{N'_1}) + c'(\lambda_2; \overline{N'_4}, N'_4) \leq c'(\lambda_1; A'_1) + c'(\lambda_2; A'_2), \quad (2.17)$$

since the first half of (2.12) can be obtained by summing (2.8), (2.9) and (2.17). Similarly, to prove the first half of (2.13), it is sufficient to show that

$$c'(\lambda_2; N'_1, \overline{N'_1}) + c'(\lambda_1; \overline{N'_4}, N'_4) \leq c'(\lambda_1; A'_1) + c'(\lambda_2; A'_2). \quad (2.18)$$

This is done by contradiction. If both (2.17) and (2.18) are invalid, then

$$c'(\lambda_1; N'_1, \overline{N'_1}) + c'(\lambda_2; \overline{N'_4}, N'_4) > c'(\lambda_1; A'_1) + c'(\lambda_2; A'_2)$$

and

$$c'(\lambda_2; N'_1, \overline{N'_1}) + c'(\lambda_1; \overline{N'_4}, N'_4) > c'(\lambda_1; A'_1) + c'(\lambda_2; A'_2).$$

Expanding the terms in the above two inequalities yields

$$\begin{aligned} & c'(\lambda_1; N'_1, N'_2) + c'(\lambda_1; N'_1, N'_3) + c'(\lambda_1; N'_1, N'_4) + \\ & c'(\lambda_2; N'_1, N'_4) + c'(\lambda_2; N'_2, N'_4) + c'(\lambda_2; N'_3, N'_4) \\ & > c'(\lambda_1; N'_1, N'_3) + c'(\lambda_1; N'_1, N'_4) + c'(\lambda_1; N'_2, N'_4) + \\ & c'(\lambda_2; N'_1, N'_2) + c'(\lambda_2; N'_1, N'_4) + c'(\lambda_2; N'_3, N'_4) \end{aligned}$$

and

$$\begin{aligned} & c'(\lambda_2; N'_1, N'_2) + c'(\lambda_2; N'_1, N'_3) + c'(\lambda_2; N'_1, N'_4) + \\ & c'(\lambda_1; N'_1, N'_4) + c'(\lambda_1; N'_2, N'_4) + c'(\lambda_1; N'_3, N'_4) \\ & > c'(\lambda_1; N'_1, N'_3) + c'(\lambda_1; N'_1, N'_4) + c'(\lambda_1; N'_2, N'_4) + \\ & c'(\lambda_2; N'_1, N'_2) + c'(\lambda_2; N'_1, N'_4) + c'(\lambda_2; N'_3, N'_4). \end{aligned}$$

Eliminating common terms in the above inequalities results in:

$$c'(\lambda_2; N'_2, N'_4) - c'(\lambda_1; N'_2, N'_4) > c'(\lambda_2; N'_1, N'_2) - c'(\lambda_1; N'_1, N'_2) \quad (2.19)$$

and

$$c'(\lambda_2; N'_1, N'_3) - c'(\lambda_1; N'_1, N'_3) > c'(\lambda_2; N'_3, N'_4) - c'(\lambda_1; N'_3, N'_4). \quad (2.20)$$

From (2.20), either (i),  $c'(\lambda_2; N'_3, N'_4) - c'(\lambda_1; N'_3, N'_4) < 0$ , or (ii),  $c'(\lambda_2; N'_1, N'_3) - c'(\lambda_1; N'_1, N'_3) > 0$ . From (2.19), either (iii),  $c'(\lambda_2; N'_1, N'_2) - c'(\lambda_1; N'_1, N'_2) < 0$ , or (iv),  $c'(\lambda_2; N'_2, N'_4) - c'(\lambda_1; N'_2, N'_4) > 0$ . Combining them together, one of the following four cases must be valid.

**Case 1** (i) and (iii) hold. That is,  $c'(\lambda_2; N'_3, N'_4) - c'(\lambda_1; N'_3, N'_4) < 0$  and  $c'(\lambda_2; N'_1, N'_2) - c'(\lambda_1; N'_1, N'_2) < 0$ .

**Case 2** (ii) and (iii) hold. That is,  $c'(\lambda_2; N'_1, N'_3) - c'(\lambda_1; N'_1, N'_3) > 0$  and  $c'(\lambda_2; N'_1, N'_2) - c'(\lambda_1; N'_1, N'_2) < 0$ .

**Case 3** (i) and (iv) hold. That is,  $c'(\lambda_2; N'_3, N'_4) - c'(\lambda_1; N'_3, N'_4) < 0$  and  $c'(\lambda_2; N'_2, N'_4) - c'(\lambda_1; N'_2, N'_4) > 0$ .

**Case 4** (ii) and (iv) hold. That is,  $c'(\lambda_2; N'_1, N'_3) - c'(\lambda_1; N'_1, N'_3) > 0$  and  $c'(\lambda_2; N'_2, N'_4) - c'(\lambda_1; N'_2, N'_4) > 0$ .

In Case 1, there must exist at least one arc  $(i, j)$  in  $(N'_1, N'_2) \setminus (N_1, N_4)$  such that  $c'(\lambda_2; i, j) - c'(\lambda_1; i, j) < 0$  and one arc  $(k, l)$  in  $(N'_3, N'_4) \setminus (N_1, N_4)$  such that  $c'(\lambda_2; k, l) - c'(\lambda_1; k, l) < 0$ . By (i) and (iv) of Claim 2.3,  $(i, j) \in (N_1, N_2)$ ,  $(k, l) \in (N_3, N_4)$  and thus are both forward arcs in the minimum cut  $(Y, \bar{Y})$ . From Lemma 2.13, I can find paths  $p_2$  from  $s$  to  $k$  and  $p_3$  from  $j$  to  $t$  such that  $p_2$  is contained in  $Y$  and  $p_3$  and is contained in  $\bar{Y}$ . Since  $i \in N'_1$ , there exists a path  $p_1$  from  $s$  to  $i$  that is contained in  $N_1$ . Similarly, since  $l \in N'_4$ , there exists a path  $p_4$  from  $l$  to  $t$  that is contained in  $N_4$ . Since  $p_1, p_2$  are

contained in  $Y$  and  $p_3, p_4$  are contained in  $\bar{Y}$ ,  $p_1, p_2, p_3$  and  $p_4$  form a bypass between  $(i, j)$  and  $(k, l)$ , and  $(i, j)$  and  $(k, l)$  are parallel. By assumption, arcs  $(i, j)$  and  $(k, l)$  must satisfy at least one of the conditions (i), (ii) or (iii) in Theorem 2.14. In (i), the two arcs must share a common head or a common tail. This is impossible since  $i \in N'_1, k \in N'_3, j \in N'_2$  and  $l \in N'_4$ . In (ii), the capacities of these two arcs must change in opposite directions, which is not satisfied in Case 1. Now suppose (iii) is satisfied. If  $(i, j)$  is both a front arc and a back arc, then  $p_4$  must contain  $j$ , but this is impossible since  $j \in N'_2$  and  $p_4$  is contained in  $N_4$ , and  $N'_2 \cap N_4 = \emptyset$ . If  $(k, l)$  is both a front arc and a back arc, then  $p_1$  must contain  $k$ , but this is also impossible since  $k \in N'_3$  and  $p_1$  is contained in  $N_1$ .

In Case 2, there exists at least one arc  $(i, j) \in (N'_1, N'_2) \setminus (N_1, N_4)$  such that  $c'(\lambda_2; i, j) - c'(\lambda_1; i, j) < 0$  and there exists at least one arc  $(k, l) \in (N'_1, N'_3) \setminus (N_1, N_4)$  such that  $c'(\lambda_2; k, l) - c'(\lambda_1; k, l) > 0$ . Since  $i$  and  $k$  are in  $N'_1$ , by Claim 2.2 there exist a path  $p_1$  from  $s$  to  $i$  and a path  $p_2$  from  $s$  to  $k$  that are contained in  $N'_1 \subseteq N_1$ . By Claim 2.1,  $j$  cannot be in  $N_1 \setminus N'_1$ . Thus  $j \in N_2$ . If  $l \in (N_4 \setminus N'_4)$ , then  $(k, l) \in (N_1, N_4)$ , which is impossible. Thus by part (ii) of Claim 2.3,  $l \in N_3$ . Then  $(i, j) \in (Y, \bar{Y})$  and  $(k, l) \in (X, \bar{X})$ . By Lemma 2.13, there exists a path  $p_3$  from  $j$  to  $t$  that is contained in  $\bar{Y}$ , and similarly, there exists a path  $p_4$  from  $l$  to  $t$  that is contained in  $\bar{X}$ . Since  $p_1$  and  $p_2$  are contained in  $N_1$ ,  $p_1 \cap p_3 = \emptyset, p_2 \cap p_4 = \emptyset, p_1 \cap p_4 = \emptyset$  and  $p_2 \cap p_3 = \emptyset$ . Thus  $(i, j)$  and  $(k, l)$  are in parallel. By assumption, arcs  $(i, j)$  and  $(k, l)$  must satisfy at least one of conditions (i), (ii) or (iii) in Theorem 2.14. Since the capacities on the two arcs change in opposite directions, (i) cannot be satisfied. In (ii), it is required that one of the arcs  $(i, j)$  and  $(k, l)$  be a front arc and the other be a back arc. If  $(i, j)$  is a front arc and  $(k, l)$  is a back arc, then in the bypass  $p_1, p_2, p_3$  and  $p_4$ ,  $p_3$  must contain  $l$ . But this is impossible since  $p_3$  is contained in  $\bar{Y}$  and  $l \in N_3 \subseteq Y$ . If  $(i, j)$  is back arc and  $(k, l)$  is a front arc, then  $p_4$  must pass through  $j$ . This is also impossible since  $p_4$  is contained in  $\bar{X}$  and  $j \in N_2 \subseteq X$ . Condition



(iii) is not satisfied either. Indeed, if on the contrary  $(i, j)$  is both a front arc and a back arc,  $p_4$  must pass through node  $j$ , otherwise  $(k, l)$  must be both a front arc and a back arc and  $p_3$  must contain  $l$ . Neither case is possible.

Case 3 is similar to Case 2 and Case 4 is similar to Case 1. This can be seen by exchanging the roles of  $\lambda_1$  with  $\lambda_2$ ,  $X'$  with  $Y'$ , and/or by letting the flow go from the sink node to the source node. Thus I have established a contradiction for each case, and either (2.17) or (2.18) must hold. This completes the proof of Claim 2.5.  $\square$

Note that if I let  $x = (X, \overline{X})$ ,  $y = (Y, \overline{Y})$ ,  $u = (N'_1, \overline{N'_1})$  and  $v = (\overline{N'_4}, N'_4)$ , then the extreme parts of inequalities (2.10) and (2.11) coincide with (2.3) and (2.4) in Condition C. Furthermore, notice that  $u \preceq_c (X \cap Y, \overline{X \cap Y})$  and  $(X \cup Y, \overline{X \cup Y}) \preceq_c v$ , thus Condition C holds. Since the lattice of cuts in the network  $G$  is finite, and a minimal and maximal minimum cut exist for each instance of  $\lambda$ , it follows from the Weak Totally Ordered Selection Theorem that a totally ordered selection of minimum cuts can be obtained by extending a single minimum cut. To extend a finite totally ordered sub-selection of minimum cuts, I need the following claim.

**Claim 2.6** *Either*

$$c(\lambda_1; X \cap Y, \overline{X \cap Y}) + c(\lambda_2; X \cup Y, \overline{X \cup Y}) \leq c(\lambda_1; X, \overline{X}) + c(\lambda_2; Y, \overline{Y}) \quad (2.21)$$

*or*

$$c(\lambda_1; X \cup Y, \overline{X \cup Y}) + c(\lambda_2; X \cap Y, \overline{X \cap Y}) \leq c(\lambda_1; X, \overline{X}) + c(\lambda_2; Y, \overline{Y}). \quad (2.22)$$

**Proof.** By the definition of  $c'(\cdot; \cdot)$ ,

$$c(\lambda_1; (N'_1, \overline{N'_1}) \cup (N_1, N_4)) \leq c'(\lambda_1; N'_1, \overline{N'_1}) + c(\lambda_1; N_1, N_4), \quad (2.23)$$

$$c(\lambda_2; (\overline{N'_4}, N'_4) \cup (N_1, N_4)) \leq c'(\lambda_2; \overline{N'_4}, N'_4) + c(\lambda_2; N_1, N_4), \quad (2.24)$$

$$c'(\lambda_1; X, \overline{X}) + c(\lambda_1; N_1, N_4) = c(\lambda_1; X, \overline{X}), \quad (2.25)$$

$$c'(\lambda_2; Y, \overline{Y}) + c(\lambda_2; N_1, N_4) = c(\lambda_2; Y, \overline{Y}). \quad (2.26)$$

From Claim 2.5, either (2.12) or (2.13) is valid. Suppose (2.12) is valid. Adding the inequality consisting of the two extreme parts of (2.12) and (2.23) through (2.26) yields

$$\begin{aligned} & c(\lambda_1; (N'_1, \overline{N'_1}) \cup (N_1, N_4)) + c(\lambda_2; (\overline{N'_4}, N'_4) \cup (N_1, N_4)) \\ & \leq c(\lambda_1; X, \overline{X}) + c(\lambda_2; Y, \overline{Y}). \end{aligned} \quad (2.27)$$

Thus the set of arcs  $(N'_1, \overline{N'_1}) \cup (N_1, N_4)$  is a minimum cut for  $\lambda_1$  and the set of arcs  $(\overline{N'_4}, N'_4) \cup (N_1, N_4)$  is a minimum cut for  $\lambda_2$ , and (2.27) must be satisfied as an equation.

Since  $(X, \overline{X})$  and  $(Y, \overline{Y})$  are minimum cuts for  $\lambda_1$  and  $\lambda_2$ , respectively, (2.10) must be satisfied as an equation. Since the first half of (2.10) can be obtained by summing (2.8), (2.9), (2.14), (2.15) and (2.17), all of them must be satisfied as equations. Then (2.8) and (2.9) can be rewritten as:

$$c(\lambda_1; A'_1 \setminus (N_1, N_4)) = c(\lambda_1; A_1 \setminus (N_1, N_4)) \quad (2.28)$$

and

$$c(\lambda_2; A'_2 \setminus (N_1, N_4)) = c(\lambda_2; A_2 \setminus (N_1, N_4)). \quad (2.29)$$

By parts (vi) and (vii) of Claim 2.3,  $A'_1 \setminus (N_1, N_4) \subseteq A_1 \setminus (N_1, N_4)$  and  $A'_2 \setminus (N_1, N_4) \subseteq A_2 \setminus (N_1, N_4)$ . For (2.28) and (2.29) to hold,

$$A'_1 \setminus (N_1, N_4) = A_1 \setminus (N_1, N_4) \quad (2.30)$$

and

$$A'_2 \setminus (N_1, N_4) = A_2 \setminus (N_1, N_4) \quad (2.31)$$

must be valid.

We further claim that  $(N_1, \overline{N_1}) \setminus (N_1, N_4) \subseteq (N'_1, \overline{N'_1}) \setminus (N_1, N_4)$ . Indeed, if, on the contrary,  $(N_1, \overline{N_1}) \setminus (N_1, N_4) \not\subseteq (N'_1, \overline{N'_1}) \setminus (N_1, N_4)$ , then there exists an arc  $(i, j) \in (N_1, \overline{N_1})$

such that,  $(i, j) \notin (N_1, N_4)$  and  $(i, j) \notin (N'_1, \overline{N'_1})$ . Since  $j \in \overline{N_1} \subseteq \overline{N'_1}$ ,  $i \in (N_1 \setminus N'_1)$ . But then,  $i \in N'_2$  because  $(N_1 \setminus N'_1) \subseteq N'_2$ . Since  $j \in \overline{N_1}$ , and  $(i, j) \notin (N_1, N_4)$ ,  $j$  must be either in  $N_2$  or in  $N_3$ . Therefore, there are only two possible cases:

**Case 1**  $j \in N_2 \subseteq N'_2$ . Then  $(i, j) \in (N_1, N_2) \subseteq A_2$  and  $(i, j) \in (N'_2, N'_2)$ . Thus  $(i, j) \notin A'_2$ , contradicting (2.31).

**Case 2**  $j \in N_3 \subseteq N'_3$ . Then  $(i, j) \in (N_1, N_3) \subseteq A_1$  and  $(i, j) \in (N'_2, N'_3)$ . Thus  $(i, j) \notin A'_1$ , contradicting (2.30).

In both cases, a contradiction results; thus

$$(N_1, \overline{N_1}) \setminus (N_1, N_4) \subseteq (N'_1, \overline{N'_1}) \setminus (N_1, N_4)$$

and

$$(N_1, \overline{N_1}) \subseteq (N'_1, \overline{N'_1}) \cup (N_1, N_4). \quad (2.32)$$

Similarly, one can show that

$$(\overline{N_4}, N_4) \subseteq (\overline{N'_4}, N'_4) \cup (N_1, N_4). \quad (2.33)$$

By (2.27), (2.32) and (2.33),

$$c(\lambda_1; N_1, \overline{N_1}) + c(\lambda_2; \overline{N_4}, N_4) \leq c(\lambda_1; X, \overline{X}) + c(\lambda_2; Y, \overline{Y})$$

and (2.21) is valid.

By an analogous argument, if (2.13) in Claim 2.5 holds, then (2.22) in Claim 2.6 holds.

This completes the proof for Claim 2.6.  $\square$

I now return to the “if” part of the proof of Theorem 2.14. I have been assuming that  $\Lambda$  is countable. By Claim 2.6, Condition B is satisfied. Thus by Corollary 2.8, a finite

totally ordered sub-selection of minimum cuts can be extended into a complete one and the proof of Theorem 2.14 is complete.  $\square$

The following corollary presents a necessary and sufficient condition for the existence of a totally ordered selection of minimum cuts for every capacity function in  $\mathcal{F}(A_v)$ .

**COROLLARY 2.15** *Let  $A_v$  be a set of arcs in a directed network  $G(N, A)$ . Then a totally ordered selection of minimum cuts exists for any parametric capacity function  $c(\cdot; \cdot)$  in  $\mathcal{F}(A_v)$ , if and only if for each pair of parallel arcs in  $A_v$ , one arc must be both a front arc and a back arc.*

**Proof.** For any  $c(\cdot; \cdot) \in \mathcal{F}(A_v)$ , only arcs in  $A_v$  can change in  $\lambda$ , but the direction of change is unrestricted. Thus  $\mathcal{F}(A_v) = \mathcal{F}(A_v, A_v^+, A_v^-)$  where  $A_v^+ = A_v^- = \emptyset$ . By Theorem 2.14, a totally ordered selection of minimum cuts exists for every  $c(\cdot; \cdot)$  in  $\mathcal{F}(A_v)$ , if and only if each pair of parallel arcs in  $A_v$  satisfy at least one of the conditions (i), (ii) or (iii) in Theorem 2.14. (i) and (ii) require the pair of parallel arcs to be in  $A_v^+$  or  $A_v^-$ , which are empty sets in this setting. Thus (i) and (ii) cannot be satisfied and (iii) must hold. This completes the proof.  $\square$

Next, I present some special cases of the class of parametric flow problems presented in Theorem 2.14.

**COROLLARY 2.16** *Let  $G_\lambda(N, A)$  be a capacitated network and let  $v_1$  and  $v_2$  be two arbitrary nodes therein. Suppose capacities of all arcs terminating at  $v_1$  are either all nondecreasing functions of  $\lambda$  or all nonincreasing functions of  $\lambda$ , and similarly, capacities of all arcs originating from  $v_2$  are either all nondecreasing functions of  $\lambda$  or all nonincreasing functions of  $\lambda$ . Further, the capacities of the arcs on a directed path  $p$*

from  $v_1$  to  $v_2$  are arbitrary positive functions of  $\lambda$ , and capacities of all other arcs are constant. Let  $A_v$  be the set of all variable arcs. If each pair of parallel arcs in  $A_v$  either both originate from  $v_2$  or both terminate at  $v_1$ , then there exists a totally ordered selection of minimum cuts when  $\lambda$  changes.

**Proof.** From the assumptions, each pair of parallel arcs must share the same head or the same tail, and their capacities change in the same direction. By Theorem 2.14, a totally ordered selection of minimum cuts exists.  $\square$

For the parametric flow problem studied by AUK (1993), if the capacities of arcs incident to a single node  $v$  are all nondecreasing functions of  $\lambda$ , then a totally ordered selection of minimum cuts exists. This result is a special case of Corollary 2.16 as can be seen by identifying nodes  $v_1$  and  $v_2$  and letting the path  $p$  from  $v_1$  to  $v_2$  to be empty. For the same problem, if node  $v$  is allowed to have a capacity which changes in  $\lambda$ , a totally ordered selection of minimum cuts still exists. Indeed, one can transform this node-and-arc capacitated network into an arc capacitated network by splitting node  $v$  into nodes  $v_1$  and  $v_2$ . All arcs terminating at  $v$  will now terminate at  $v_1$  and all arcs originating from  $v$  will now originate from  $v_2$ . A directed arc  $(v_1, v_2)$  is added which has the capacity of node  $v$ . This arc capacitated network is a special case of Corollary 2.16, since in this case the path  $p$  is the single arc  $(v_1, v_2)$ .

In the problem considered by AUK, a totally ordered selection of minimum cuts does not exist if we further allow capacities on nodes adjacent to  $v$  to change. A counter example is given in Figure 2.11. Therein, the capacities of nodes  $v_2, v_3, v_4, v_5$  and  $v_6$  in a network  $G$  are parametrically changed. Part (a) of Figure 2.11 shows the original node-and-arc capacitated network. Parts (b) and (c) of Figure 2.11 present the arc-capacitated equivalent of  $G$ , with two sets of capacities. The unique minimum cut in each case is

shown with dashed lines. Clearly, the minimum cuts under the two values of the parameter are not totally ordered. This is not surprising in light of Theorem 2.14, since arcs  $(v_3, v'_3)$  and  $(v_5, v'_5)$  in Figure 2.11 are parallel to each other and the conditions in Theorem 2.14 are not satisfied. Carstensen (1983) constructed an example where an exponential number of minimum cuts exist which are not totally ordered when the capacities are parametrically changed.

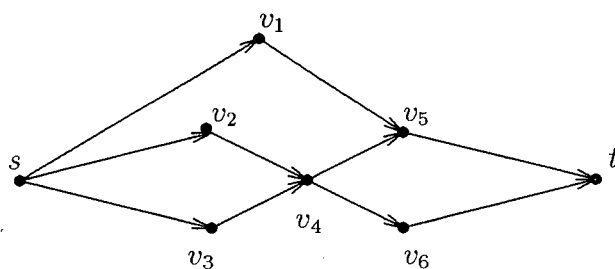
In Corollary 2.16, if one let  $v_1$  be  $s$  and  $v_2$  be  $t$ , then the set of arcs terminating at  $v_1$  or originating from  $v_2$  will be empty, and the following result can be obtained.

**COROLLARY 2.17** *Let  $p$  be a path from the source  $s$  to the sink  $t$  in a network  $G(N, A)$ . Suppose no two arcs in  $p$  are parallel to each other. Then, when the capacities of arcs on  $p$  are parametrically changed, a totally ordered selection of minimum cuts exists*

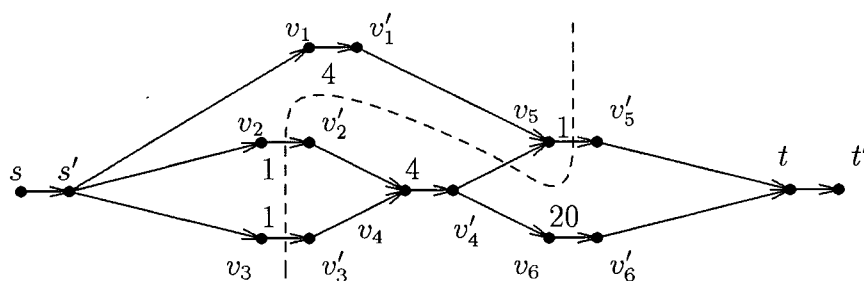
In an  $s$ - $t$  series-parallel network, no pairs of arcs on a path from  $s$  to  $t$  are parallel. Thus by Corollary 2.17, when the capacities on an  $s$ - $t$  path are arbitrarily changed in such a network, a totally ordered selection of minimum cuts exists.

### 2.3 Ordering of Minimum Cuts in Some Parametric Flow Problems

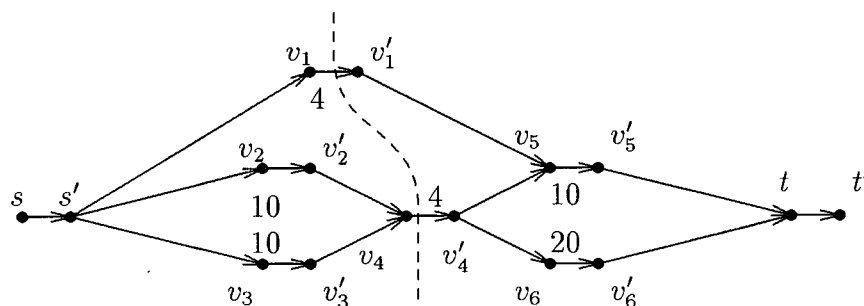
As a natural continuation of the study on ordered minimum cuts, the ordering of minimum cuts in some parametric maximum flow problems is studied in this section. My study is motivated by the following observation. If minimum cuts are monotone increasing in the parameter, then an increasing sequence of minimum cuts can be identified which contains a minimum cut for each value of the parameter and, as the parameter increases, the minimum cut will move to the next cut in the sequence. As shown in the last section, in general, minimum cuts are not monotone with respect to the natural order of cuts induced by set inclusion. Can one, then, construct a partial order, other than the natural



(a) The original network



(b) One set of capacities



(c) Another set of capacities

Figure 2.11: An example where totally ordered selections of minimum cuts does not exist

set inclusion order, with respect to which the minimum cut will be monotone increasing? I will address this question when the capacities of arcs are affine functions of a single parameter  $\lambda$ .

Suppose that in a directed network  $G(N, A)$ , the capacity of arc  $(i, j)$  is  $c(i, j) = c_0(i, j) + \lambda d(i, j)$ , where  $\lambda \geq 0$  is a parameter. For simplicity of notation, let  $\mathbf{c} = (c(i, j) : (i, j) \in A)$  be the vector of arc capacities for  $G(N, A)$ , let  $\mathbf{c}_0$  denote the vector  $(c_0(i, j) : (i, j) \in A)$ , and let  $\mathbf{d}$  denote the vector  $(d(i, j) : (i, j) \in A)$ . Then  $\mathbf{c} = \mathbf{c}_0 + \lambda \mathbf{d}$ , and clearly the vectors of arc capacities form a ray. In the sequel, we assume that all vectors are columnar unless transposed. Therefore, for two vectors  $\mathbf{a}$  and  $\mathbf{b}$ ,  $\mathbf{a}^T \mathbf{b}$  denotes the inner product of  $\mathbf{a}$  and  $\mathbf{b}$ .

For each cut  $(X, \overline{X})$  in  $G(N, A)$ , let  $\mathcal{C}(X)$  be the set of capacity vectors for which  $(X, \overline{X})$  is a minimum cut. I call each such set  $\mathcal{C}(X)$  a *region* in the space of capacity vectors. Regions corresponding to different cuts may overlap, since more than one cut may be minimum for a fixed capacity vector. The region  $\mathcal{C}(X)$  is defined by the following constraints:

$$\mathcal{C}(X) = \{ \mathbf{c} \mid \sum_{(i,j) \in (X, \overline{X})} c(i, j) \leq \sum_{(i,j) \in (Y, \overline{Y})} c(i, j) \text{ for all cuts } (Y, \overline{Y}) \}.$$

Since each constraint in the above definition is linear,  $\mathcal{C}(X)$  is a polyhedron.

The ray  $\mathbf{c} = \mathbf{c}_0 + \lambda \mathbf{d}$  can be partitioned into non-overlapping intervals  $I_0, I_1, \dots, I_r$  such that an interval  $I_{i+1}$  is farther away from the origin of the ray than  $I_i$ , and that each interval  $I_i$  is contained in  $\mathcal{C}(X_i)$ . Then  $(X_i, \overline{X}_i)$  is a minimum cut for capacity vectors in  $I_i$ . The ray thus induces a sequence of minimum cuts  $(X_0, \overline{X}_0), (X_1, \overline{X}_1), \dots, (X_r, \overline{X}_r)$ .

The next lemma shows that no two cuts in a sequence induced by a ray can be the same; i.e., if the ray leaves a region, it will never return to it. The validity of the lemma is due to the fact that each region  $\mathcal{C}(X)$  is convex. If two points on a ray are in the same region, then the segment between these two points must also be in this region.



**LEMMA 2.18** *In an arbitrary sequence induced by a ray, no two cuts are the same.*

Since the cuts in the sequence of cuts generated by a ray do not repeat, one can define a partial order among the cuts that totally orders the cuts in this sequence. My next question then is, suppose there are many rays of capacity vectors, can one find a sequence of minimum cuts which “works” for all these rays? In other words, are the partial orders induced by different rays consistent with one another? To this end, I first introduce the notion of consistent sequences. Consider two sequences of cuts. I say that the two sequences are *consistent* if there do not exist two cuts  $(X_i, \overline{X}_i)$  and  $(X_j, \overline{X}_j)$  contained in both sequences, such that the former precedes the latter in one sequence whereas the opposite is true in the other sequence. The next theorem states that, if two rays are parallel to each other or if they share the same origin, then the sequences induced by the two rays are always consistent.

**THEOREM 2.19** *Consider the following two rays:*

$$\mathbf{c} = \mathbf{c}_1 + \lambda \mathbf{d}_1$$

and

$$\mathbf{c} = \mathbf{c}_2 + \lambda \mathbf{d}_2.$$

*If  $\mathbf{c}_1 = \mathbf{c}_2$  or  $\mathbf{d}_1 = \mathbf{d}_2$ , then the sequences of cuts induced by these two rays are consistent.*

**Proof.** For each cut  $(X, \overline{X})$ , I define an index vector  $\mathbf{e}_X$  in  $\mathbb{R}^{|A|}$ , where  $A$  is the set of arcs in the network. An element in  $\mathbf{e}_X$  is 1 if the corresponding arc is a forward arc in  $(X, \overline{X})$ . All other elements in  $\mathbf{e}_X$  are 0. Thus  $\mathbf{e}_X^T \mathbf{c}$  is the capacity of  $(X, \overline{X})$ .

I will prove the theorem first for the case where  $\mathbf{c}_1 = \mathbf{c}_2$ . Let  $\mathbf{c}_0 = \mathbf{c}_1 = \mathbf{c}_2$ . Consider the two rays originating from  $\mathbf{c}_0$  with directions  $\mathbf{d}_1$  and  $\mathbf{d}_2$ . If the sequences induced by the two rays are inconsistent, then there exist  $\lambda_1 < \lambda_2$  and  $\lambda_3 < \lambda_4$  such that

- $(X, \overline{X})$  is a minimum cut for  $\mathbf{c}_0 + \lambda_1 \mathbf{d}_1$  and  $\mathbf{c}_0 + \lambda_4 \mathbf{d}_2$  but not for  $\mathbf{c}_0 + \lambda_2 \mathbf{d}_1$  or  $\mathbf{c}_0 + \lambda_3 \mathbf{d}_2$ , whereas
- $(Y, \overline{Y})$  is a minimum cut for  $\mathbf{c}_0 + \lambda_2 \mathbf{d}_1$  and  $\mathbf{c}_0 + \lambda_3 \mathbf{d}_2$  but not for  $\mathbf{c}_0 + \lambda_1 \mathbf{d}_1$  or  $\mathbf{c}_0 + \lambda_4 \mathbf{d}_2$ .

Then:

$$\mathbf{e}_X^T(\mathbf{c}_0 + \lambda_1 \mathbf{d}_1) < \mathbf{e}_Y^T(\mathbf{c}_0 + \lambda_1 \mathbf{d}_1), \quad (2.34)$$

$$\mathbf{e}_X^T(\mathbf{c}_0 + \lambda_2 \mathbf{d}_1) > \mathbf{e}_Y^T(\mathbf{c}_0 + \lambda_2 \mathbf{d}_1), \quad (2.35)$$

$$\mathbf{e}_X^T(\mathbf{c}_0 + \lambda_3 \mathbf{d}_2) > \mathbf{e}_Y^T(\mathbf{c}_0 + \lambda_3 \mathbf{d}_2), \quad (2.36)$$

$$\mathbf{e}_X^T(\mathbf{c}_0 + \lambda_4 \mathbf{d}_2) < \mathbf{e}_Y^T(\mathbf{c}_0 + \lambda_4 \mathbf{d}_2). \quad (2.37)$$

Subtracting  $\lambda_1$  times (2.35) from  $\lambda_2$  times (2.34) yields:

$$(\lambda_2 - \lambda_1)(\mathbf{e}_X^T - \mathbf{e}_Y^T)\mathbf{c}_0 < 0.$$

Since  $(\lambda_2 - \lambda_1) > 0$ ,

$$(\mathbf{e}_X^T - \mathbf{e}_Y^T)\mathbf{c}_0 < 0. \quad (2.38)$$

Similarly, subtracting  $\lambda_3$  times (2.37) from  $\lambda_4$  times (2.36) yields:

$$(\lambda_4 - \lambda_3)(\mathbf{e}_X^T - \mathbf{e}_Y^T)\mathbf{c}_0 > 0.$$

Since  $(\lambda_4 - \lambda_3) > 0$ ,

$$(\mathbf{e}_X^T - \mathbf{e}_Y^T)\mathbf{c}_0 > 0. \quad (2.39)$$

However, (2.38) and (2.39) lead to a contradiction.

Now I prove the theorem when  $\mathbf{d}_1 = \mathbf{d}_2$ . I denote the common direction by  $\mathbf{d}$ . If two inconsistent sequences exist, then there exist two cuts  $(X, \overline{X})$  and  $(Y, \overline{Y})$  such that  $(X, \overline{X})$  is minimum for  $\mathbf{c}_1 + \lambda_1 \mathbf{d}$  and  $\mathbf{c}_2 + \lambda_4 \mathbf{d}$  but not for  $\mathbf{c}_1 + \lambda_2 \mathbf{d}$  and  $\mathbf{c}_2 + \lambda_3 \mathbf{d}$ , whereas the opposite is true for  $(Y, \overline{Y})$ . Therefore,

$$\mathbf{e}_X^T(\mathbf{c}_1 + \lambda_1 \mathbf{d}) < \mathbf{e}_Y^T(\mathbf{c}_1 + \lambda_1 \mathbf{d}), \quad (2.40)$$

$$\mathbf{e}_X^T(\mathbf{c}_1 + \lambda_2 \mathbf{d}) > \mathbf{e}_Y^T(\mathbf{c}_1 + \lambda_2 \mathbf{d}), \quad (2.41)$$

$$\mathbf{e}_X^T(\mathbf{c}_2 + \lambda_3 \mathbf{d}) > \mathbf{e}_Y^T(\mathbf{c}_2 + \lambda_3 \mathbf{d}), \quad (2.42)$$

$$\mathbf{e}_X^T(\mathbf{c}_2 + \lambda_4 \mathbf{d}) < \mathbf{e}_Y^T(\mathbf{c}_2 + \lambda_4 \mathbf{d}). \quad (2.43)$$

Subtracting inequality (2.41) from (2.40) yields:

$$(\lambda_1 - \lambda_2)(\mathbf{e}_X^T - \mathbf{e}_Y^T)\mathbf{d} < 0.$$

Since  $\lambda_1 - \lambda_2 < 0$ ,

$$(\mathbf{e}_X^T - \mathbf{e}_Y^T)\mathbf{d} > 0. \quad (2.44)$$

Similarly, subtracting (2.43) from (2.42) yields:

$$(\lambda_3 - \lambda_4)(\mathbf{e}_X^T - \mathbf{e}_Y^T)\mathbf{d} > 0;$$

and since  $\lambda_3 - \lambda_4 < 0$ ,

$$(\mathbf{e}_X^T - \mathbf{e}_Y^T)\mathbf{d} < 0. \quad (2.45)$$

Again, (2.44) and (2.45) lead to a contradiction.  $\square$

Thus each ray in a collection of rays which share the same origin or direction, induces a sub-sequence of some fixed sequence of minimum cuts. However, if I allow both the

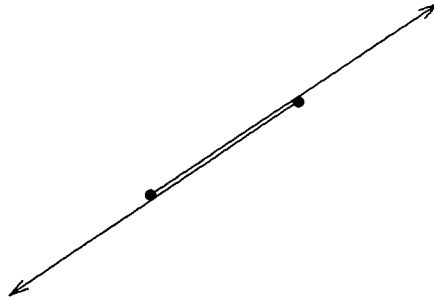


Figure 2.12: Two rays in opposite directions

origin and the direction of a ray to change, the induced sequences will not be consistent. To see this, consider two rays in opposite directions, as illustrated in Figure 2.12, which induce two sequences that are reversed in order.

Generally, in a sequence of cuts induced by a ray, the cuts are not totally ordered by the natural ordering of cuts. However, if one places some restrictions on the ray, then a sequence of minimum cuts can be found which are totally ordered. This is exactly the case when a ray satisfies Theorem 2.14. Formally,

**COROLLARY 2.20** *Suppose  $\mathbf{c} = \mathbf{c}_0 + \lambda \mathbf{d}$  is a ray in the space of the capacity vectors. Let  $A_v$  be the set of arcs that correspond to non-zero elements in  $\mathbf{d}$ . If no pairs of arcs in  $A_v$  are parallel, then the ray will induce a sequence of minimum cuts that are totally ordered by the natural ordering of cuts.*

I further note that the collection of minimum cuts derived from merging the two totally ordered sequences of minimum cuts induced by two rays is in general, not totally ordered. Indeed, suppose that in one ray, capacities of arcs in  $A_v$  change and in another ray, capacities of arcs in another set  $A'_v$  change. Suppose that no parallel arcs exist in either  $A_v$  or  $A'_v$ , thus each ray induces a totally ordered sequence of minimum cuts. However, there may be a pair of parallel arcs in  $A_v \cup A'_v$ . So when the capacities change

from a point in one ray to a point in the other, capacities on a pair of parallel arcs may change. The minimum cuts corresponding to these two points may not be ordered.

However, suppose that for each ray in a collection of rays, the set of arcs whose capacities change is a subset of  $A_v$ , which does not contain any parallel arcs. Then, when the sequences induced by the different rays are merged, the resultant sequence is still totally ordered.

When a ray induces a totally ordered sequence of cuts, there are at most  $n-1$  minimum cuts in the sequence, where  $n$  is the number of nodes in the network. Further recall that if one only changes the direction of a ray and not its point of origin, the sequences of minimum cuts induced are consistent. Thus the following theorem is valid:

**THEOREM 2.21** *Let  $A_v$  be a subset of arcs that does not contain parallel arcs and let  $\mathcal{R}$  be a collection of rays. Suppose among all points on the rays, only capacities of arcs in  $A_v$  vary. If all the rays in  $\mathcal{R}$  share a common origin, then there exists a sequence of at most  $n-1$  totally ordered minimum cuts and each ray in  $\mathcal{R}$  induces a consistent subsequence of it.*

Similarly, if all the rays have the same direction, I have

**THEOREM 2.22** *Let  $A_v$  be a subset of arcs that does not contain parallel arcs and let  $\mathcal{R}$  be a collection of rays. Suppose among all points on the rays, only capacities of arcs in  $A_v$  vary. If all the rays in  $\mathcal{R}$  have the same direction, then there exists a sequence of at most  $n-1$  totally ordered minimum cuts and each ray in  $\mathcal{R}$  induces a consistent subsequence of it.*

If I allow both the origin and the direction of the rays to differ, then even when the change of capacities is restricted to arcs in  $A_v$ , which does not contain parallel arcs, the rays will not induce consistent sequences. In fact, in the next example, I demonstrate that  $(n-1)!$  inconsistent sequences may exist.

Let the network  $G(N, A)$  be a single path from  $s$  to  $t$  where  $A = \{e_1, e_2, \dots, e_q\}$  and  $A_v = A$ . Evidently,  $A_v$  does not contain parallel arcs. Since each minimum cut in this network is composed of a single arc, a sequence of minimum cuts corresponds to a sequence of arcs in the network. I wish to construct  $q!$  rays such that the sequence induced by each ray corresponds to a different permutation of the  $q$  arcs in the network. Consider an arbitrary permutation of arcs  $(e_{\pi_1}, e_{\pi_2}, \dots, e_{\pi_q})$ . I construct the ray that induces this sequence by letting  $\mathbf{c}_0 = (\pi_1, \pi_2, \dots, \pi_q)^T$  and  $\mathbf{d} = (4^{1-\pi_1}, 4^{1-\pi_2}, 4^{1-\pi_3}, \dots, 4^{1-\pi_q})^T$ . Instead of a formal proof, I use the following example to show that this ray indeed induces the sequence  $e_{\pi_1}, e_{\pi_2}, \dots, e_{\pi_q}$ .

Consider the permutation  $(e_q, e_{q-1}, \dots, e_2, e_1)$ . The ray corresponding to this permutation has an origin  $\mathbf{c}_0 = (q, q-1, \dots, 2, 1)^T$  and a direction  $\mathbf{d} = (\frac{1}{4^{q-1}}, \frac{1}{4^{q-2}}, \dots, \frac{1}{4}, 1)^T$ . When  $\lambda$  is 0, the minimum cut is at  $e_q$ , since the capacities at this point are  $\mathbf{c}_0 = (q, q-1, \dots, 2, 1)^T$  and  $e_q$  has the smallest capacity. When  $\lambda$  increases to 4,  $e_{q-1}$  becomes the new minimum cut, since the capacities are now  $(q + \frac{1}{4^{q-2}}, q-1 + \frac{1}{4^{q-3}}, \dots, 3\frac{1}{4}, 3, 5)^T$ . When  $\lambda = 4^i$ , the capacities are  $(q + \frac{1}{4^{q-1-i}}, \dots, i+3 + \frac{1}{4^2}, i+2\frac{1}{4}, i+2, i+4, \dots, 3 + 4^{i-2}, 2 + 4^{i-1}, 1 + 4^i)^T$  and the minimum cut is at  $e_{q-i}$ . Thus the sequence of minimum cuts induced by the ray is  $e_q, e_{q-1}, \dots, e_2, e_1$ .

## Chapter 3

### An Extended Selection Problem

#### 3.1 The Extended Selection Model

In this chapter, I consider a problem faced by a decision maker, who has to choose a collection of facilities to acquire in order to undertake a subset of jobs. Each job requires a set of tasks, and each facility can perform a set of tasks. A job can be performed if the tasks it requires can be performed by the facilities that were acquired. I assume that the facilities are non-perishable. Thus a facility can be used by more than one job. A revenue is associated with each job and a cost is associated with each facility. The net return obtained by the decision maker is the total revenue from the jobs performed minus the total cost of the facilities acquired. The objective is to maximize the net return. Without loss of generality, I will assume that all job revenues and all facility costs are positive.

I now present the notation used in the model.

- $\mathcal{F} = \{f_1, f_2, \dots, f_n\}$  is the set of possible facilities.
- $\mathcal{J} = \{j_1, j_2, \dots, j_m\}$  is the set of potential jobs.
- $\mathcal{T} = \{t_1, t_2, \dots, t_k\}$  is the set of tasks.
- $T(j)$  is the set of tasks required in order to perform job  $j$ , and for a subset of jobs  $J$ ,  $T(J)$  is the set of tasks required to perform all jobs in  $J$ , where  $T(J) = \cup_{j \in J} T(j)$ .
- $T(f)$  is the set of tasks that can be performed by facility  $f$ , and for a subset of

facilities  $F$ ,  $T(F)$  is the set of tasks that can be performed by the collection  $F$ , where  $T(F) = \cup_{f \in F} T(f)$ .

- $c(f)$  is the cost of facility  $f$ , and for a collection of facilities  $F$ ,  $c(F) = \sum_{f \in F} c(f)$  is the total cost of this collection.
- $r(j)$  is the revenue from a job  $j$ , and for a set of jobs  $J$ ,  $r(J) = \sum_{j \in J} r(j)$  is the total revenue from jobs in  $J$ .

I refer to  $(F, J)$  as a *selection plan*, where  $F$  is the collection of facilities acquired and  $J$  is the set of jobs performed. A selection plan  $(F, J)$  is *feasible* if the tasks required by  $J$  can be performed by the facilities in  $F$ . I will refer to this feasibility constraint as the *sufficient tasks constraint*.

Similar to the original selection model, the objective in the ES model is to maximize the net return from a feasible selection plan. In an application of the original selection model, Eisner and Severance (1976) introduced a relative weight between the revenue and the cost terms, which reflects the conversion factor between a unit of storage and a unit of user value. Brumelle and Granot (1993) presented some monotonicity results with respect to this relative weight. Sometimes the facility costs may increase by roughly the same factor due to inflation, whereas the job revenues can only increase by a smaller factor due to competition. In such a case, a relative weight between the revenue and the cost terms is also helpful. To take account of situations like this, to be consistent with earlier papers mentioned above and for the sake of flexibility, I retain the relative weight between costs and revenues in the ES model. Thus the net return associated with a selection plan  $(F, J)$  is defined as  $g(F, J; \lambda) = \lambda r(J) - c(F)$ , where  $\lambda$  is the relative weight on the revenue, and the extended selection (ES) model can be formally presented



as

$$\begin{aligned} \text{ES : } & \max_{J \subseteq \mathcal{J}, F \subseteq \mathcal{F}} g(F, J; \lambda) \\ & \text{subject to} \\ & T(J) \subseteq T(F). \end{aligned}$$

The original selection problem proposed by Rhys (1970) is a special case of the ES problem. That is, any instance of the original selection model can be formulated as an ES model. Indeed, for an arbitrary instance of the original selection model, let the set of facilities and the set of jobs in the ES model be the same as in the original model. Each facility  $f$  in the ES model performs a single task, which is  $f$  itself. A job  $j$  in the ES model requires a certain task if and only if it requires the corresponding facility in the original selection problem. Clearly, the original model coincides with this instance of the ES problem.

The ES problem is generally NP-hard. Indeed, I claim that the set covering problem, which is known to be NP-hard, can be formulated as a special case of the ES problem. To show that, recall that the set covering problem is concerned with a finite ground set  $S$ , and a collection of subsets,  $\mathcal{C} = \{D_1, D_2, \dots, D_n\}$ , of  $S$ , where  $D_k \subseteq S$  for  $k = 1, 2, \dots, n$ . The objective is to find a sub-collection,  $C$ , of  $\mathcal{C}$  with minimum total cost, such that the union of subsets in  $C$  covers  $S$ . Formally, the set covering problem can be stated as

$$\begin{aligned} \min_{C \subseteq \mathcal{C}} & \sum_{D \in C} c(D) \\ & \text{subject to} \\ & \bigcup_{D \in C} D = S, \end{aligned}$$

where  $c(D)$  is the cost associated with subset  $D \in \mathcal{C}$ .

The above set covering problem can be formulated as a special case of the ES problem. Indeed, let the set of jobs in the ES problem be  $S$ , and assign a very large revenue  $\Pi$  to each job in  $S$ . Each job  $j \in S$  requires a single task, which is also denoted by  $j$ . Thus the ground set of tasks in the ES problem is also  $S$ . The ground set of facilities is  $\mathcal{C}$ . For

each facility  $D \in \mathcal{C}$ , the tasks performed by  $D$  are precisely the elements contained in  $D$ . The cost of facility  $D$  is  $c(D)$ . Because the common revenue  $\Pi$  is very large, every job will be performed in any optimal selection plan. The set of tasks required is then the ground set  $S$ . The minimal cost collection of facilities that can perform  $S$  will be a minimum cost cover in the set covering problem. The ES problem is thus NP-hard.

### 3.2 Some Variants of the ES Model

Even though the ES problem is NP-hard, under some natural assumptions it can be solved efficiently. Indeed, in some cases, the tasks performed by different facilities are nested. That is, a higher level facility can perform all the tasks that can be performed by a lower level one. For example, a 486 CPU performs all the tasks performed by a 386 CPU and a 387 co-processor. Such a nested structure also exists in a forest harvest scheduling problem, which will be formally presented in the next section. To further motivate the analysis for the rest of this section, let us briefly consider this problem. Therein, a network of roads is predetermined which forms a tree. The root of the tree is a saw mill and each other node is a block of forest to be harvested. In order to harvest a block, the road leading from the saw mill to the block must be ready. The value of a block varies over time, due to the growth of trees and the discount of value over time. Similarly, the cost of roads also varies in time. In order to maximize the present value of the net income, the decision maker must find an optimal schedule to build the roads and to harvest the blocks. In this problem, the availability of a leg of road  $l$  at a time period  $p$  can be considered as a task. The construction of a leg of road  $l$  at a period  $p$  can be considered as a facility which can be viewed as performing the tasks that correspond to the availability of leg  $l$  in time periods since  $p$ . If a leg of road is built earlier, then it is available in more periods and can perform more tasks. Thus a nested structure exists among the facilities.

For the rest of this section, I will analyze two models in which facilities and jobs are nested, and in the next section, I will elaborate on the application of these models.

### 3.2.1 The ES Model with Nested Facilities

In the first model, I assume that the tasks performed by different facilities are either nested or disjoint. Formally, I make the following assumption:

**ASSUMPTION 3.1 (The Nested or Disjoint Facilities Assumption)** *The sets of tasks,  $T(f_1)$  and  $T(f_2)$ , performed by an arbitrary pair of facilities,  $f_1$  and  $f_2$  in  $\mathcal{F}$ , are either disjoint or contained in one another.*

By Assumption 3.1, the sets of tasks performed by an arbitrary pair of facilities  $f_1$  and  $f_2$  in  $\mathcal{F}$  satisfy  $T(f_1) \cap T(f_2) = \emptyset$ , or  $T(f_1) \subseteq T(f_2)$ , or  $T(f_2) \subseteq T(f_1)$ . When the sets of tasks performed by an arbitrary pair of facilities are either nested or disjoint, I say that the facilities are *nested*. I will refer to the ES model in which Assumption 3.1 holds as the Nested Facilities (NF) model.

Observe that the original selection problem introduced by Rhys can still be viewed as a special case of the NF model. Indeed in the original selection problem, each facility performs a single task which is identical to itself. Thus the sets of tasks performed by different facilities are disjoint, and the Nested or Disjoint Facilities Assumption is satisfied.

I define the following partial order among the facilities. For  $f_1$  and  $f_2$  in  $\mathcal{F}$ ,  $f_1 \preceq_f f_2$  if  $T(f_1) \subseteq T(f_2)$ , i.e.  $f_2$  performs all the tasks performed by  $f_1$ . If  $f_1 \preceq_f f_2$ , then I say that  $f_2$  is a *super-facility* of  $f_1$ , and that  $f_1$  is a *sub-facility* of  $f_2$ . I say that a facility  $f_1$  is a *parent* of facility  $f_2$  if  $f_2 \preceq_f f_1$  and there does not exist another facility  $f_3$  such that  $f_2 \preceq_f f_3 \preceq_f f_1$ . Since the tasks performed by the facilities are either disjoint or nested, the parent of a facility  $f$ , denoted by  $\pi(f)$ , is unique. I add an artificial facility,  $\hat{f}$ , which is a

super-facility of every real facility. I assume that the cost of  $\hat{f}$  is infinitely large and refer to this facility as the *root facility*. With the introduction of  $\hat{f}$ , a unique parent exists for every real facility.

A collection of facilities  $F$  is *redundant* if there exist a pair of facilities  $f_1$  and  $f_2$  in  $F$  such that  $f_1 \preceq_f f_2$ . In an optimal selection plan, one would never select redundant facilities. Indeed, if  $f_1$  and  $f_2$  are both in the collection of facilities to be acquired, then  $f_1$  can be deleted without affecting the set of tasks that can be performed by the collection.

A network,  $G_F(\mathcal{F}, A_F)$ , in which the nodes are facilities in  $\mathcal{F}$ , can be constructed to represent the relationship among the facilities in an NF model. For ease of presentation, I will use the same notation to denote the facility and the corresponding node. A directed arc  $(f_2, f_1)$  is in  $A_F$  if and only if  $f_1$  is the parent of  $f_2$ . Clearly,  $G_F$  is a directed tree with  $\hat{f}$  as its root and all the arcs directed towards the root.

The following is an example of an NF model:

**EXAMPLE 3.1** Let  $\mathcal{J} = \{j_1, j_2, j_3, j_4\}$  be the set of possible jobs, and let  $\mathcal{F} = \{f_1, f_2, f_3, f_4, f_5, f_6\}$  be the set of possible facilities. Table 3.1 presents the revenues associated with the jobs and the tasks required by the jobs, and Table 3.2 presents the costs of the facilities and the tasks performed by them.

Jobs	Revenues	Tasks Required
$j_1$	12	$t_1, t_2$
$j_2$	6	$t_2, t_4, t_5$
$j_3$	9	$t_3, t_5$
$j_4$	8	$t_7$

Table 3.1: Jobs in Example 3.1

The tree  $G_F(\mathcal{F}, A_F)$  representing the facilities in the above example is shown in Figure 3.13.

Facilities	Costs	Tasks Performed
$f_1$	4	$t_1, t_2$
$f_2$	3	$t_3$
$f_3$	11	$t_1, t_2, t_3, t_4$
$f_4$	2	$t_5, t_6$
$f_5$	14	$t_1, t_2, t_3, t_4, t_5, t_6$
$f_6$	13	$t_7$
$\hat{f}$	0	all tasks

Table 3.2: Facilities in Example 3.1

For each task  $t$ ,  $\tau(t)$  denotes the least sophisticated facility that can perform task  $t$ . Formally,  $T(\tau(t)) \subseteq T(f)$  for any facility  $f$  such that  $t \in T(f)$ . For instance, in Example 3.1,  $\tau(t_4) = f_3$ ,  $\tau(t_1) = f_1$  and  $\tau(t_6) = f_4$ . Without loss of generality, I assume that any task  $t$  performed by  $\hat{f}$  can be performed by at least a real facility in  $\mathcal{F}$ . Therefore, there does not exist a task  $t$  such that  $\tau(t) = \hat{f}$ .

I now show that the NF model can be solved by finding a minimum cut in a capacitated network  $G(V, A)$  constructed as follows.  $G(V, A)$  contains  $G_F(\mathcal{F}, A_F)$  as a subgraph, and can be constructed from  $G_F(\mathcal{F}, A_F)$  by adding the following nodes and arcs. For each job  $j$  in  $\mathcal{J}$ , add a node to  $V$ , and denote it also by  $j$ . Further add a source node  $s$  to  $V$ . Thus  $V = \mathcal{F} \cup \mathcal{J} \cup \{s\}$ . For each job  $j$ , add a directed arc  $(s, j)$  with capacity  $\lambda r(j)$ , and for each job  $j$  and each task  $t$  required by  $j$ , add a directed arc  $(j, \tau(t))$  with an infinite capacity. Recall that for each real facility  $f$ , arc  $(f, \pi(f))$  is already in  $A_F$  and is thus contained in  $A$ . Assign capacity  $c(f)$  to arc  $(f, \pi(f))$ . Finally, identify node  $\hat{f}$  as the sink node of the network and for convenience, denote it by  $t$ .

The network  $G(V, A)$  for Example 3.1 is shown in Figure 3.14 for  $\lambda = 1$ . The capacities of the arcs are shown beside the arcs, and the dotted arcs have infinite capacities.

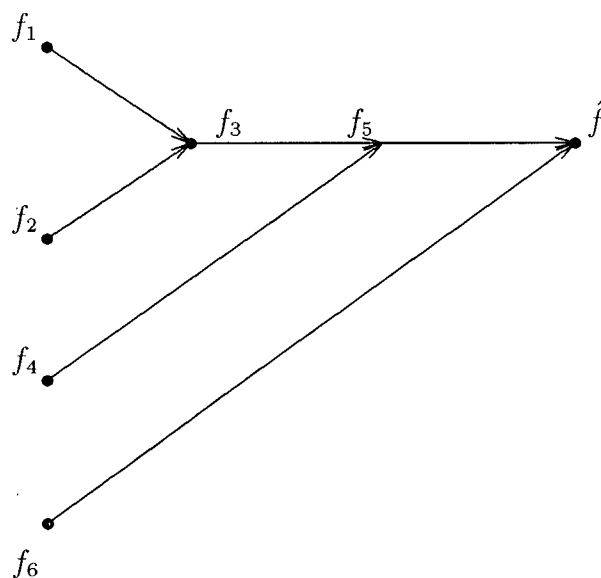


Figure 3.13: The facility tree for Example 3.1

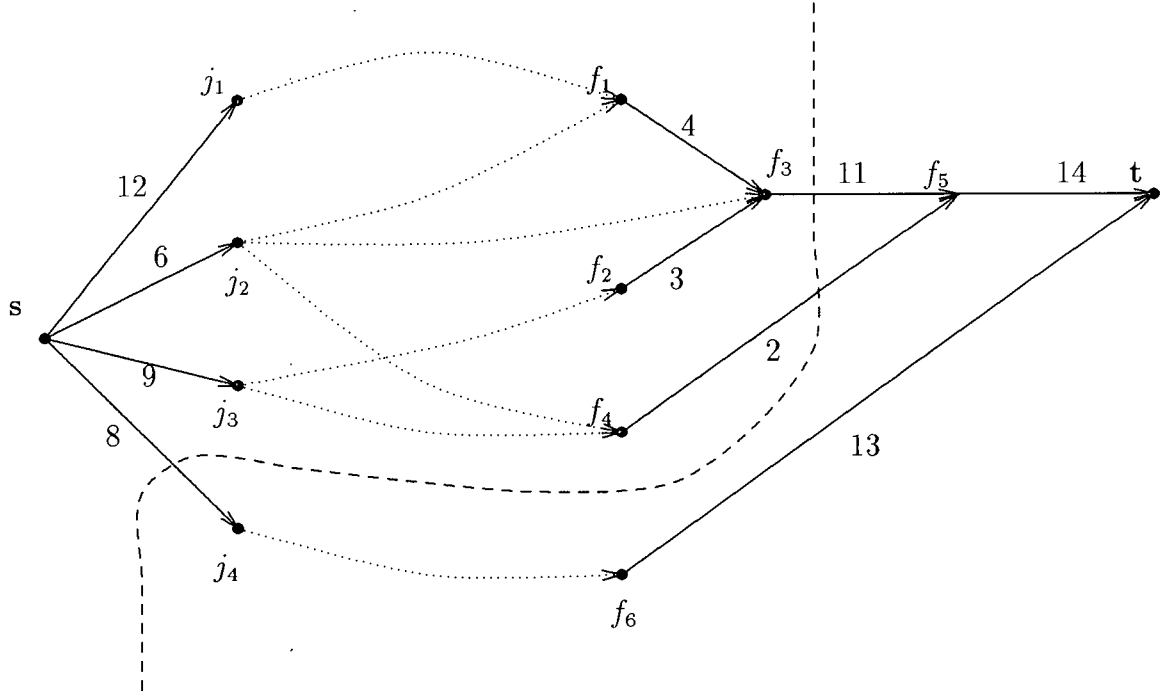
For a cut  $(X, \overline{X})$  in  $G$ , let  $J(X) = X \cap \mathcal{J}$  be the set of jobs in  $X$ , and let

$$F(X) = \{f \in X \cap \mathcal{F} \mid \nexists f_1 \in X \cap \mathcal{F} \text{ such that } f \preceq_f f_1\}.$$

That is,  $F(X)$  contains the undominated facilities in  $X$ .

**THEOREM 3.1** *Let  $(X^*, \overline{X}^*)$  be a minimum cut in  $G(V, A)$  separating  $\mathbf{t}$  from  $\mathbf{s}$ . Then  $(F(X^*), J(X^*))$  is an optimal selection plan for the NF model.*

**Proof.** The proof is done in two steps. In Step 1, I show that a cut in  $G(V, A)$  is associated with every feasible selection plan in which the collection of facilities is non-redundant, and the capacity of the cut is a constant minus the return associated with the selection plan. A collection of cuts is thus identified such that each cut in this collection is associated with a feasible (and non-redundant) selection plan. In Step 2, I show that if a cut is of minimum capacity, then it belongs to this collection of cuts. Thus a minimum cut will surely yield an optimal selection plan.

Figure 3.14: Network  $G$  for Example 3.1

**Step 1.** Suppose  $(F, J)$  is a feasible selection plan. Assume that the collection  $F$  is non-redundant, and let

$$F_1 = F \cup \{f \in \mathcal{F} \mid \exists f_1 \in F \text{ such that } f \preceq_f f_1\}.$$

That is,  $F_1$  consists of facilities in  $F$  and their sub-facilities. Let  $X = J \cup F_1 \cup \{s\}$ . Since  $F$  is non-redundant, none of the super-facilities of any facility  $f \in F$  is also in  $F$ , and by the construction of  $X$ , a facility  $f$  in  $F$  is not dominated by other facilities in  $X$ . Thus  $F(X) = F$ . The total capacity of forward arcs of the form  $(f, \pi(f))$  in  $(X, \bar{X})$ , where  $f$  is a facility, is the total cost of  $F$ . If a job  $j$  is not in  $J$ , then  $j \in \bar{X}$  and  $(s, j)$  is a forward arc in  $(X, \bar{X})$ . The total capacity of these arcs is  $\sum_{j \notin J} \lambda r(j) = \lambda \mathcal{R} - \sum_{j \in J} \lambda r(j)$ , where  $\mathcal{R}$  is the total revenue for all the jobs in  $\mathcal{J}$  and is a constant. The only arcs not considered so far are those with infinite capacities, and I claim that no such arc can be

a forward arc in  $(X, \overline{X})$ . Indeed, suppose, on the contrary, that such an arc  $(j, f)$  is a forward arc in  $(X, \overline{X})$ . Then  $f \in \overline{X}$  and there must exist a task  $t$  required by job  $j$  such that  $f = \tau(t)$ . Since  $j \in J$  and  $(F, J)$  is a feasible selection plan, there exists a facility  $f_1 \in F$  such that  $f_1$  can perform  $t$ . By the definition of  $\tau(\cdot)$ ,  $f \preceq_f f_1$ . Since  $f_1 \in F$  and  $f \preceq_f f_1$ , it follows from the definition of  $F_1$  and  $X$ , that  $f$  is contained in  $X$ , leading to a contradiction. Thus arcs with infinite capacities cannot be forward arcs in  $(X, \overline{X})$ . The capacity of  $(X, \overline{X})$  is then equal to  $\lambda\mathcal{R} - g(F, J; \lambda)$ .

**Step 2.** Consider a minimum cut  $(X^*, \overline{X^*})$ . I first show that  $(F(X^*), J(X^*))$  is a feasible selection plan. Let  $j$  be an arbitrary job in  $J(X^*)$ . Since  $(X, \overline{X})$  has a finite capacity,  $\tau(t)$  must also be in  $X^*$  for each task  $t$  required by job  $j$ . Because  $G_F$  is a tree, there is a unique directed path from  $\tau(t)$  to the sink node  $\mathbf{t}$ , and this path must cross  $(X^*, \overline{X^*})$  at least once. Let  $(f_1, \pi(f_1))$  be the last arc that crosses  $(X^*, \overline{X^*})$ .  $f_1$  is then undominated by any other facility in  $X^*$ . By the definition of  $F(\cdot)$ ,  $f_1$  is in  $F(X^*)$ .  $f_1$  is a super-facility of  $\tau(t)$  and can also perform  $t$ . Thus  $(F(X^*), J(X^*))$  is a feasible selection plan.

If a job  $j$  is not in  $X^*$ , then arc  $(\mathbf{s}, j)$  is a forward arc in  $(X^*, \overline{X^*})$ , which has capacity  $\lambda r(j)$ . For each facility  $f$  in  $F(X^*)$ , there is a forward arc  $(f, \pi(f))$  in  $(X^*, \overline{X^*})$  with capacity  $c(f)$ . The above two types of arcs are the only forward arcs in  $(X^*, \overline{X^*})$ . Thus the capacity of  $(X^*, \overline{X^*})$  is  $\lambda\mathcal{R} - g(F, J; \lambda)$ .  $\square$

For Example 3.1, the minimum cut is shown with a dashed line in Figure 3.14. The optimal collection of facilities is  $F(X^*) = \{f_3, f_4\}$  and the optimal set of jobs is  $J(X^*) = \{j_1, j_2, j_3\}$ .

As previously discussed, the set covering problem can be viewed as a special instance of the ES problem. The fact that the NF model can be solved in polynomial time suggests that if the subsets in a set covering problem are either nested or disjoint, then the set



covering problem can be solved easily. Indeed, as illustrated in the last section, one can transform such a set covering problem into an ES problem with nested facilities, which can then be solved as a minimum cut problem.

### 3.2.2 An ES Model with Nested Facilities and Jobs

In the previous section, I assume that the facilities can be ordered in some sense. In many real world problems, jobs can also be naturally ordered. For example, an automobile service package may include a tune-up and an oil change. In this case, the service package is a super-job of either the oil change or the tune-up. In the forest harvest scheduling problem, jobs also appear to be naturally ordered. Therein, a job is the harvesting of a block  $b$  in time period  $p$ . Therefore, such a job requires the availability of all legs of road in the tree network from the saw mill to this block in time period  $p$ . However, a leg of road available at period  $p$  is also available at time periods after  $p$ . Therefore, I can view the job of harvesting block  $b$  in period  $p$  as if it requires the availability of all legs of road leading to it from period  $p$  and beyond. Thus harvesting a block earlier is a super-job of harvesting the same block later.

Based on this observation, I present another model in this subsection, which incorporates the additional assumption that a partial order  $\preceq_j$  exists on the set of jobs  $\mathcal{J}$ . If  $j_1 \preceq_j j_2$ , I call  $j_2$  a *super-job* of  $j_1$ , and  $j_1$  a *sub-job* of  $j_2$ . If  $j_2$  is an immediate super-job of  $j_1$ , then  $j_2$  is called the parent of  $j_1$  and be denoted by  $\pi(j_1)$ .

In addition to the sufficient tasks constraint introduced earlier, I will assume that feasible selection plans satisfy the *non-redundant jobs constraint*. The *non-redundant jobs constraint* stipulates that in a feasible selection plan  $(F, J)$ , the set of jobs  $J$  cannot contain a pair of jobs that are ordered by  $\preceq_j$ . This assumption is realistic because if a super-job provides all the utilities of a sub-job, one does not need to perform both of them. With the introduction of the non-redundant jobs constraint, the model can be

formulated as follows.

$$\max_{J \subseteq \mathcal{J}, F \subseteq \mathcal{F}} g(F, J; \lambda)$$

subject to

$$T(J) \subseteq T(F)$$

$$j_1 \not\preceq_j j_2 \text{ for arbitrary } j_1, j_2 \in J$$

I retain the Nested or Disjoint Facilities Assumption introduced for the NF model and introduce the following additional assumptions.

**ASSUMPTION 3.2 (The Increasing Requirements Assumption)** *A super-job requires all the tasks required by its sub-jobs. That is, if  $j_1 \preceq_j j_2$ , then  $T(j_1) \subseteq T(j_2)$ .*

**ASSUMPTION 3.3 (The Nested or Disjoint Jobs Assumption)** *Let  $j_1$  and  $j_2$  be an arbitrary pair of jobs. Then either  $j_1$  and  $j_2$  are ordered by  $\preceq_j$ , or they do not have a common sub-job.*

In view of the fact that a nested structure exists among both facilities and jobs, I will refer to this model as the Nested Facilities and Jobs (NFJ) model.

The NF model can be viewed as a sub-class of the NFJ model, since any instance of the NF model can be formulated as an instance of the NFJ model. Indeed, for any instance of the NF model, one can assume that an empty partial order  $\preceq_j$  exists among the potential jobs, and under this empty partial order, no pairs of jobs are ordered. Thus the non-redundant jobs constraint and Assumptions 3.2 and 3.3 are all trivially satisfied. Although the NF model is a special case of the NFJ model, I still present them separately since the two models can be solved using different techniques, which in themselves are interesting.

Similar to the representation of nested facilities in the NF model, the partially ordered jobs can also be represented by a tree, denoted by  $G_J(\mathcal{J}, A_J)$ . I assume that a dummy

job, denoted by  $\hat{j}$ , exists which is a super-job of all other jobs. Job  $\hat{j}$  is the root of  $G_J(\mathcal{J}, A_J)$  and is referred to as the *root job*. With the introduction of this dummy job, a parent job  $\pi(j)$  exists for each job  $j$ . Let  $A_J = \{(j, \pi(j)) \mid \text{for each job } j \neq \hat{j}\}$ . Since the Nested or Disjoint Facilities Assumption still holds in the NFJ model, the facilities in this model can still be represented by the tree  $G_F(\mathcal{F}, A_F)$ , introduced in the last subsection.

The following is an example of an NFJ model.

**EXAMPLE 3.2** *The facilities in this example are identical to those in Example 3.1 and are shown in Table 3.2. The jobs in this example are shown in Table 3.3. The partial order  $\preceq_j$  among the jobs is shown in Figure 3.15, where two jobs  $j_1$  and  $j_2$  satisfy  $j_1 \preceq_j j_2$  if there exists a directed path from  $j_1$  to  $j_2$ .*

Jobs	Revenues	Tasks Required
$j_1$	12	$t_1, t_2$
$j_2$	24	$t_1, t_2, t_4, t_5$
$j_3$	6	$t_4$
$j_4$	9	$t_3, t_5$
$j_5$	8	$t_3, t_5, t_7$
$\hat{j}$	0	all tasks

Table 3.3: Jobs in Example 3.2

Although the Nested or Disjoint Facilities Assumption and the Nested or Disjoint Jobs Assumption are very similar in appearance, they are quite different in nature. In the Nested or Disjoint Facilities Assumption, the nested or disjoint restriction is on the set of tasks performed by the facilities. That is, if the sets of tasks performed by two facilities are not contained in one another, then they must be disjoint. In the Nested or Disjoint Jobs Assumption however, the restriction is on the set of sub-jobs. That is, if two jobs are not ordered by  $\preceq_j$ , then they cannot share a common sub-job, but they can

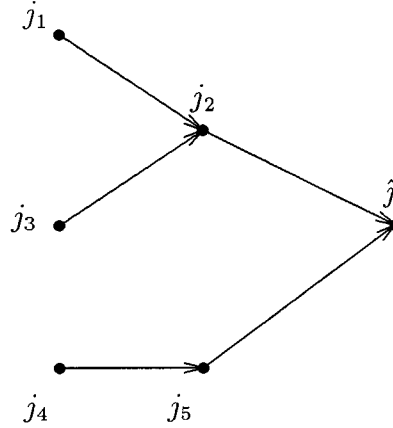


Figure 3.15: Tree of jobs in Example 3.2

require a common task. Consider for example, jobs  $j_2$  and  $j_4$  in Example 3.2. They are not ordered by  $\preceq_j$ , but they both require a common task  $t_5$ . Notice that in Example 3.2 the revenue from  $j_4$  is higher than that from  $j_5$ . That is, the revenues are not necessarily monotone in jobs ordered by  $\preceq_j$ .

If a job does not have any sub-jobs, then I call it an *elementary job*. We define the span,  $\psi(j)$ , of a job  $j$  as follows: If  $j$  is an elementary job, then  $\psi(j) = 1$ . If  $j$  is the parent of other jobs, then  $\psi(j) = \sum_{j_1: \pi(j_1)=j} \psi(j_1)$ . Intuitively,  $\psi(j)$  is the number of elementary sub-jobs of  $j$ , or, the number of leaf nodes in the subtree of  $G_J$  rooted at  $j$ . In Example 3.2,  $j_1$  is an elementary job, thus  $\psi(j_1)=1$ .  $j_2$  has two *immediate* sub-jobs,  $j_1$  and  $j_3$ . Thus  $\psi(j_2)=\psi(j_1)+\psi(j_3)=2$ .

Similar to the NF model, the NFJ model can also be solved as a minimum cut problem in a directed network  $G(V, A)$ , constructed as follows.  $G(V, A)$  contains the trees of jobs and facilities,  $G_F(\mathcal{F}, A_F)$  and  $G_J(\mathcal{J}, A_J)$ , as subgraphs. All the nodes and arcs in  $G_F$  and  $G_J$  remain in  $G$ , except that node  $\hat{j}$  in  $G_J$  and node  $\hat{f}$  in  $G_F$  are identified as a single node, which is denoted by  $\mathbf{t}$ . All arcs terminating at either  $\hat{j}$  or  $\hat{f}$  will now terminate at  $\mathbf{t}$ .

Add a source node  $s$  to  $V$ , and for each elementary job  $j$  in  $\mathcal{J}$ , add a directed arc  $(s, j)$ . For each job  $j$  in  $\mathcal{J}$ , add a directed arc  $(j, \tau(t))$  if  $j$  requires task  $t$  which, in turn, is not required by any of the sub-jobs of  $j$ . Let  $\Pi$  be a large constant, whose magnitude will be clarified later. Assign capacity  $\Pi$  to arc  $(s, j)$  for each elementary job  $j$ , and for each job  $j \neq \hat{j}$ , assign capacity  $\psi(j)\Pi - \lambda r(j)$  to arc  $(j, \pi(j))$ . For each facility  $f \neq \hat{f}$ , I assign the capacity  $c(f)$  to arc  $(f, \pi(f))$ . Finally, I assign an infinite capacity to each arc  $(j, \tau(t))$ .

The network  $G(V, A)$  for Example 3.2 with  $\lambda=1$  is shown in Figure 3.16. Similar to Figure 3.14, the capacities in  $G$  are shown beside the arcs, and the arcs in dotted lines have infinite capacities.

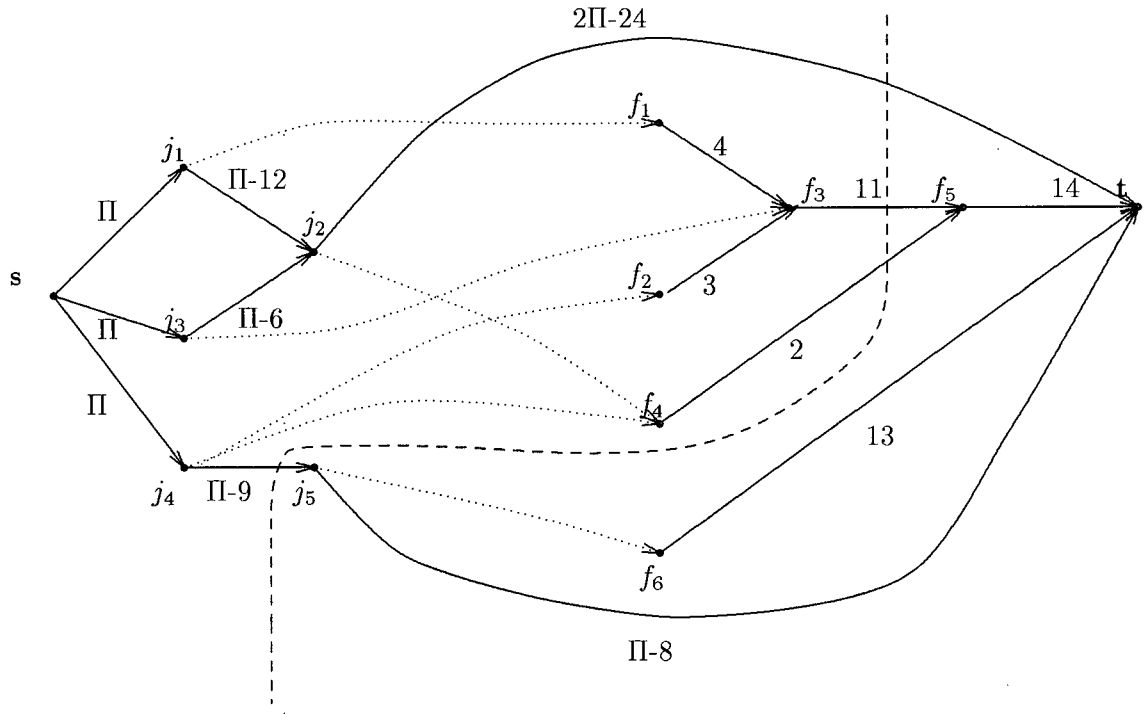


Figure 3.16: Network  $G$  for Example 3.2

For the NFJ model,  $J(X)$  for a cut  $(X, \overline{X})$  in  $G$  is redefined to consist of the set of

undominated jobs in  $X$ . That is,

$$J(X) = \{j \in X \cap \mathcal{J} \mid \nexists j_1 \in X \cap \mathcal{J} \text{ such that } j \preceq_j j_1\}.$$

The definition for  $F(X)$  is the same as that for the NF model, i.e.,

$$F(X) = \{f \in X \cap \mathcal{F} \mid \nexists f_1 \in X \cap \mathcal{F} \text{ such that } f \preceq_f f_1\}.$$

**THEOREM 3.2** *Suppose  $(X^*, \overline{X^*})$  is a minimum cut separating  $\mathbf{t}$  from  $\mathbf{s}$  in the network  $G(V, A)$ . Then,  $(F(X^*), J(X^*))$  is an optimal selection plan.*

To prove the theorem, I need the following lemma.

**LEMMA 3.3** *Let  $(X, \overline{X})$  be a cut in  $G(V, A)$  with finite capacity. If for each job  $j$  in  $X$ , all the sub-jobs of  $j$  are also in  $X$ , and for each facility  $f$  in  $X$ , all the sub-facilities of  $f$  are also in  $X$ , then the capacity of  $(X, \overline{X})$  is equal to  $K\Pi - \lambda r(J(X)) + c(F(X))$ , where  $K$  is the total number of elementary jobs in  $\mathcal{J}$ .*

**Proof.** Since the capacity of  $(X, \overline{X})$  is finite, no arcs of the form  $(j, \tau(t))$  can be forward arcs in  $(X, \overline{X})$ . Each job  $j \in J(X)$  is not dominated by any other job in  $X$ . Thus  $\pi(j)$  is in  $\overline{X}$ , and  $(j, \pi(j))$  is a forward arc in  $(X, \overline{X})$  for every job  $j \in J(X)$ . On the other hand, consider an arbitrary forward arc of the form  $(j, \pi(j))$  in  $(X, \overline{X})$ . Clearly,  $\pi(j) \in \overline{X}$ . If  $j$  is dominated by another job  $j_1$  in  $X$ , then either  $\pi(j)=j_1$  or  $\pi(j)$  is dominated by  $j_1$ . By assumption, all sub-jobs of  $j_1$  are also in  $X$ , implying that  $\pi(j)$  must also be in  $X$ , which leads to a contradiction. Thus  $j$  is not dominated by any other job in  $X$  and  $j \in J(X)$ . I have just shown that there is a one to one correspondence between forward arcs of the form  $(j, \pi(j))$  in  $(X, \overline{X})$  and jobs in  $J(X)$ . Similarly, there is a one to one correspondence between forward arcs of the form  $(f, \pi(f))$  in  $(X, \overline{X})$  and

facilities in  $F(X)$ . Finally, for each elementary job  $j \in \overline{X}$ ,  $(s, j)$  is a forward arc in  $(X, \overline{X})$ . Thus the set of forward arcs in  $(X, \overline{X})$  is

$$\{(j, \pi(j)) | j \in J(X)\} \cup \{(f, \pi(f)) | f \in F(X)\} \cup \{(s, j) \in A | j \in \overline{X}\}.$$

Let  $K_1$  be the number of elementary jobs that are either in  $J(X)$  or dominated by a job in  $J(X)$ . I now show that the total capacity of arcs in  $\{(j, \pi(j)) | j \in J(X)\}$  is  $K_1\Pi - \lambda r(J(X))$ . Recall that the capacity of an arc  $(j, \pi(j))$  is  $\psi(j)\Pi - \lambda r(j)$ , and  $\psi(j)$  is the number of elementary sub-jobs of  $j$ . Thus it suffices to show that those  $K_1$  elementary jobs are counted exactly once in  $\sum_{j \in J(X)} \psi(j)\Pi$ . Clearly, each such elementary job is counted at least once. For two arbitrary jobs  $j_1$  and  $j_2$  in  $J(X)$ , by the definition of  $J(\cdot)$ ,  $j_1$  and  $j_2$  do not dominate each another. The sets of elementary sub-jobs of  $j_1$  and  $j_2$  are then disjoint. Thus these elementary sub-jobs are never double counted.

Next, consider the set of arcs  $\{(s, j) \in A | j \in \overline{X}\}$ . For each arc  $(s, j)$  in this set,  $j$  is clearly an elementary job. The total capacity of arcs in this set is  $K_2\Pi$ , where  $K_2$  is the number of elementary jobs that are not in  $X$ . Since all sub-jobs of a job in  $X$  are also in  $X$ , these elementary jobs cannot be dominated by jobs in  $X$ . Thus  $K_1 + K_2 = K$ .

Finally, the total capacity of arcs in  $\{(f, \pi(f)) | f \in F(X)\}$  is  $c(F(X))$ . This completes the proof of Lemma 3.3.  $\square$

I now return to the proof of Theorem 3.2.

**Proof of Theorem 3.2.** The proof is in two steps, similar to those used in the proof of Theorem 3.1. The reader is referred to Theorem 3.1 for an outline of the proof.

**Step 1.** In this step, I wish to show that for each feasible selection plan  $(F, J)$ , a cut  $(X, \overline{X})$  in  $G$  can be constructed such that  $F = F(X)$ ,  $J = J(X)$  and the capacity of  $(X, \overline{X})$  is  $K\Pi - g(F, J; \lambda)$ . Suppose  $(F, J)$  is a feasible selection plan with no redundant facilities or redundant jobs. Let  $J_1 = \{j \in \mathcal{J} | \exists j_1 \in J \text{ such that } j \preceq_j j_1\}$ , i.e.,  $J_1$  is the set

of jobs in  $J$  and their sub-jobs. Similarly, let  $F_1 = \{f \in \mathcal{F} \mid \exists f_1 \in F \text{ such that } f \preceq_f f_1\}$ , i.e.,  $F_1$  is the set of facilities in  $F$  and their sub-facilities. Let  $X = F_1 \cup J_1$ . From the construction of  $F_1$  and  $J_1$  and by the definition of  $F(\cdot)$  and  $J(\cdot)$ ,  $F = F(X)$  and  $J = J(X)$ .

Next I show that the capacity of  $(X, \overline{X})$  is finite. Consider an arc  $(j, \tau(t))$ , which has an infinite capacity. If  $j \in X$ , then either  $j \in J$ , or  $j$  is dominated by some other job  $j_1$  in  $J$ . In the latter case,  $t$  is required by  $j_1$  due to the Increasing Requirements Assumption. Thus in either case, task  $t$  is required by jobs in  $J$ . Since  $(F, J)$  is feasible, either  $\tau(t)$  is in  $F$  or a super-facility of  $\tau(t)$  is in  $F$ , and in either case  $\tau(t)$  is in  $X$ . Thus  $(j, \tau(t))$  cannot be a forward arc in  $(X, \overline{X})$  and the capacity of  $(X, \overline{X})$  is finite. By Lemma 3.3, the capacity of  $(X, \overline{X})$  is  $K\Pi - \lambda r(J(X)) + c(F(X))$ , or  $K\Pi - g(F, J; \lambda)$ .

**Step 2.** In this step, I wish to show that if  $(X^*, \overline{X}^*)$  is a minimum cut, then  $(F(X^*), J(X^*))$  is a feasible selection plan, and the capacity of  $(X^*, \overline{X}^*)$  is  $K\Pi - g(F(X^*), J(X^*); \lambda)$ .

First I show that if job  $j$  is in  $X^*$ , then all its sub-jobs must also be in  $X^*$ . Suppose otherwise, then I can find a minimal sub-job  $j_1$  of  $j$  such that  $j_1 \in \overline{X}^*$  and if  $j_1$  has a sub-job  $j_2$ , then  $j_2 \in X^*$  and  $(j_2, j_1) \in (X^*, \overline{X}^*)$ . If  $j_1$  does not have such a sub-job, then  $(s, j_1) \in (X^*, \overline{X}^*)$ . I only consider the first case, as the second case is similar.

Let  $\mathcal{A}_0 = \{(f, t) \in A \mid f \in \mathcal{F}\}$ , and let  $\mathcal{A} = (X^*, \overline{X}^*) \cup \mathcal{A}_0 \setminus \{(j_2, j_1)\}$ . That is,  $\mathcal{A}$  is the set of forward arcs in  $(X^*, \overline{X}^*)$ , with  $(j_2, j_1)$  therein being replaced by arcs in  $\mathcal{A}_0$ . We will show that  $\mathcal{A}$  is a cut set of arcs separating  $s$  and  $t$ , though not necessarily a node partition cut. Moreover, observe that the capacity of arc  $(j_2, j_1)$  is  $\psi(j_2)\Pi - \lambda r(j_2)$ . For a large enough  $\Pi$ , the capacity of  $(j_2, j_1)$  is strictly larger than the total capacity of arcs in  $\mathcal{A}_0$ . For such a  $\Pi$ , the total capacity of arcs in  $\mathcal{A}$  is then strictly smaller than the capacity of  $(X^*, \overline{X}^*)$ .

If the removal of arcs in  $\mathcal{A}$  instead of those in  $(X^*, \overline{X}^*)$  results with  $s$  and  $t$  being connected, then it must be due to the fact that arc  $(j_2, j_1)$  is not in  $\mathcal{A}$ . Let  $p$  be a directed



path from  $s$  to  $t$  that contains arc  $(j_2, j_1)$ . If the last arc in the path  $p$  is  $(f, t)$  for some facility  $f$ , then  $p$  is disconnected with the removal of arcs in  $\mathcal{A}$ . Otherwise, the last arc in  $p$  is  $(j_3, t)$  for some job  $j_3$ . Since there does not exist a task  $t$  such that  $\tau(t) = \hat{f}$ ,  $(j_3, t)$  must be an arc in the tree of jobs. Thus  $j_3$  is a super-job of  $j_1$  and must be on path  $p$ . Since  $j \in X^*$ ,  $j$  is disconnected from  $t$  when arcs in  $(X^*, \overline{X^*})$  are removed. When arcs in  $\mathcal{A}$  are removed instead of those in  $(X^*, \overline{X^*})$ ,  $(j_2, j_1)$  remains. But  $(j_2, j_1)$  is not on any path leading from  $j$  to  $t$ . Thus  $j$  is still not connected to  $t$  when arcs in  $\mathcal{A}$  are removed, and  $p$  is still disconnected. Thus  $\mathcal{A}$  is a cut set of arcs, which contradicts the fact that  $(X^*, \overline{X^*})$  is a minimum cut, since for a large enough  $\Pi$ , the total capacity of arcs in  $\mathcal{A}$  is strictly smaller than the capacity of  $(X^*, \overline{X^*})$ .

I can assume that all the sub-facilities of an arbitrary facility  $f \in X^*$  are also in  $X^*$ . This can be shown as follows. Suppose there exists a sub-facility  $f_1$  of  $f$  that is in  $\overline{X^*}$ . Since every path from  $f_1$  to  $t$  must contain  $f$ ,  $f_1$  is already disconnected from  $t$  after the removal of forward arcs in  $(X^*, \overline{X^*})$ . So if all nodes corresponding to sub-facilities of  $f$  are moved into  $X^*$ , no new forward arc will be introduced into  $(X^*, \overline{X^*})$ . Since all arc capacities are positive, and the original  $(X^*, \overline{X^*})$  is a minimum cut, this move cannot have eliminated any forward arc either. Since the facilities that are moved into  $X^*$  are all dominated by other facilities in the original  $X^*$ ,  $F(X^*)$  is not affected by this move either. So, as far as the facility selection problem is concerned, the original  $(X^*, \overline{X^*})$  is equivalent to the new one. Thus I can assume that in the original  $(X^*, \overline{X^*})$  all sub-facilities of any facility  $f$  in  $X^*$  are also in  $X^*$ . By Lemma 3.3, the capacity of  $(X^*, \overline{X^*})$  is  $K\Pi - g(F(X^*), J(X^*); \lambda)$ .

Next, I show that the collection  $F(X^*)$  of facilities is sufficient to perform jobs in  $J(X^*)$ . Let  $j$  be an arbitrary job in  $J(X^*)$ . Suppose  $j$  requires a task  $t$ . I can find the smallest sub-job  $j_1$  of  $j$  that requires  $t$ . As I have shown before,  $j_1$  must be in  $X^*$ . By the construction of  $G$ , there exists an arc  $(j_1, \tau(t))$  in  $G$  which has an infinite capacity.

Therefore,  $\tau(t)$  must also be in  $X^*$ . Let  $f$  be the largest super-facility of  $\tau(t)$  that is in  $X^*$ . Since  $f$  is undominated in  $X^*$ ,  $f \in F(X^*)$ . Thus  $F(X^*)$  can perform  $t$ .

Finally, because  $J(X^*)$  is composed of undominated jobs in  $X^*$ , two jobs that are ordered by  $\preceq_j$  cannot be both in  $J(X^*)$ . Thus the non-redundant jobs constraint is satisfied, and  $(F(X^*), J(X^*))$  is a feasible selection plan. This completes Step 2.  $\square$

It can be seen from the proof that I can choose  $\Pi$  to be a number that is larger than the total cost of the facilities plus  $\lambda$  times the largest revenue of a job. In Example 3.2, let  $\Pi=100$ . (Observe that  $\Pi$  is larger than 71, the sum of all facility costs, 47, plus the largest revenue, 24.) The minimum cut  $(X^*, \overline{X^*})$  is represented by the dashed line in Figure 3.16, and  $J(X^*)=\{j_2, j_4\}$ ,  $F(X^*)=\{f_3, f_4\}$ . By Theorem 3.2,  $(F(X^*), J(X^*))$  is an optimal selection plan for this example.

### 3.3 Applications of the NFJ model

In this section, I first present an application of the NFJ model to a network scheduling problem, which may arise in telecommunication, forest harvesting, and open pit mining.

#### 3.3.1 A Network Scheduling Problem

The network scheduling problem studied in this section is concerned with a forest structured network. The root of each tree in the network represents a supplier of a single commodity, for example, cable TV programs. Other nodes represent potential consumers who may need the commodity, for example, townships that may need cable TV. Each arc in the network represents a potential link between two nodes, for example, a TV cable, and will be referred to as a *leg*. In order to make the commodity available to a client at a certain time period, all the legs on the unique path from the supplier to the client must be

built by this time period. A cost is incurred from constructing a leg, which, due to time discounting, depends on the time period in which it is constructed. A one-time lump-sum revenue is associated with the initiation of service to a client at a certain time, which also varies over time. For example, this one-time revenue may be the present value of the total revenue from cable TV consumers in a township *ever since* it is made available. In order to maximize the net income, one needs an optimal schedule specifying the time periods in which to build the legs and to make the commodity available to the clients. I assume that the planning horizon consists of a finite number of discrete time periods, and I ignore the capacities of the links. For the cable TV example, since all viewers receive the same signals, the capacity of a TV cable does not depend on the number of users linked to it. Figure 3.17 illustrates an example where there are three clients,  $C_1$ ,  $C_2$  and  $C_3$ , and a supplier  $S$ .

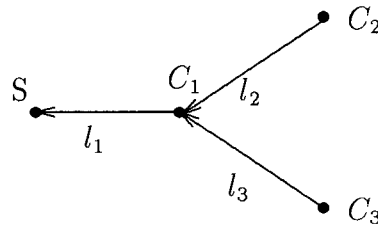


Figure 3.17: Possible links from the suppliers to the consumers

The network scheduling problem can be formulated as an NFJ model constructed as follows. I associate a task  $t(l, p)$  with each leg  $l$  and each time period  $p$ . Such a task indicates that leg  $l$  is *available* at time period  $p$ . Further, I associate a facility  $f(l, p)$  with each leg  $l$  and each time period  $p$ , and such a facility indicates that leg  $l$  is *constructed* in time period  $p$ . The cost of  $f(l, p)$  is the present value of building  $l$  in period  $p$ . A facility  $f(l, p)$  can perform tasks  $t(l, p_1)$  for  $p_1 = p, p + 1, p + 2, \dots$ . Clearly, the collections of tasks performed by facilities corresponding to the same leg are nested, whereas the

collections of tasks performed by facilities corresponding to different legs are disjoint. Thus the Nested or Disjoint Facilities Assumption is satisfied.

Consider the network shown in Figure 3.17. For simplicity, I only consider two periods, i.e.,  $p=1$  or  $2$ . Table 3.4 presents the facilities in the model and the tasks performed by the facilities.

Facilities	Tasks Performed
$f(l_1, 1)$	$t(l_1, 1), t(l_1, 2)$
$f(l_1, 2)$	$t(l_1, 2)$
$f(l_2, 1)$	$t(l_2, 1), t(l_2, 2)$
$f(l_2, 2)$	$t(l_2, 2)$
$f(l_3, 1)$	$t(l_3, 1), t(l_3, 2)$
$f(l_3, 2)$	$t(l_3, 2)$

Table 3.4: Facilities and tasks performed by them

I associate a job  $j(b, p)$  with each consumer  $b$  and each time period  $p$ . If  $j(b, p)$  is performed, then the service is opened to  $b$  in time period  $p$ . The revenue from this job is the present value,  $v(b, p)$ , of making the commodity available to  $b$  in time period  $p$ . Job  $j(b, p)$  requires task  $t(l, p_1)$  for each leg  $l$  on the path from the supplier to  $b$ , and for each  $p_1 \geq p$ . That is, in order to make the commodity available to  $b$  in time  $p$ , each leg on the path from the supplier must be available at time  $p$ . Of course, if the leg is available at time  $p$ , it will be available in each period thereafter. The jobs in the above example are presented in Table 3.5.

I define the following partial order among the jobs. For each consumer  $b$ ,  $j(b, p_1) \preceq_j j(b, p_2)$  if  $p_2 \leq p_1$ . Clearly, a job corresponding to making the commodity available earlier requires more tasks. Jobs associated with the same consumer are totally ordered, whereas jobs associated with different consumers are never ordered by  $\preceq_j$ . Thus two jobs that are not ordered by  $\preceq_j$  can never share a common sub-job and the Nested or Disjoint Jobs

Jobs	Tasks Required
$j(C_1, 1)$	$t(l_1, 1), t(l_1, 2)$
$j(C_1, 2)$	$t(l_1, 2)$
$j(C_2, 1)$	$t(l_2, 1), t(l_1, 1), t(l_2, 2), t(l_1, 2)$
$j(C_2, 2)$	$t(l_2, 2), t(l_1, 2)$
$j(C_3, 1)$	$t(l_3, 1), t(l_1, 1), t(l_3, 2), t(l_1, 2)$
$j(C_3, 2)$	$t(l_3, 2), t(l_1, 2)$

Table 3.5: Jobs and tasks required

Assumption holds. Note that jobs associated with the same client are totally ordered, and two ordered jobs are never contained in the same feasible selection plan. This prevents opening service to the same client twice. It is not hard to see that a feasible selection plan in the above NFJ model corresponds to a feasible schedule, and thus the network scheduling problem can be solved as a minimum cut problem in a directed network. This network contains  $O(nt)$  nodes and  $O(Dnt)$  arcs, where  $n$  is the number of consumers,  $t$  is the number of periods in the planning horizon and  $D$  is the maximum depth of the tree of links.

I can slightly modify the network and reduce the number of arcs therein to  $O(nt)$ . The modified network is the same as the original network except that the arcs with infinite capacities are different. Therein, the set of arcs with infinite capacities consists of the following arcs: For each job  $j(b, p)$ , there is an arc from node  $j(b, p)$  to node  $f(l, p)$ , where  $l$  is the leg terminating at  $b$ . For each pair of nodes  $f(l, p)$  and  $f(l', p)$ , there is an arc from  $f(l, p)$  to  $f(l', p)$  if leg  $l'$  immediately precedes  $l$ . The modified network has  $(2nt + 2)$  nodes and  $(4nt + n)$  arcs and is smaller than the original one. The original and the modified networks for the example displayed in Figure 3.17 are shown in Figures 3.18 and 3.19, respectively. The modified network is equivalent to the original one as far as maximum flow and minimum cut are concerned, since from the original network

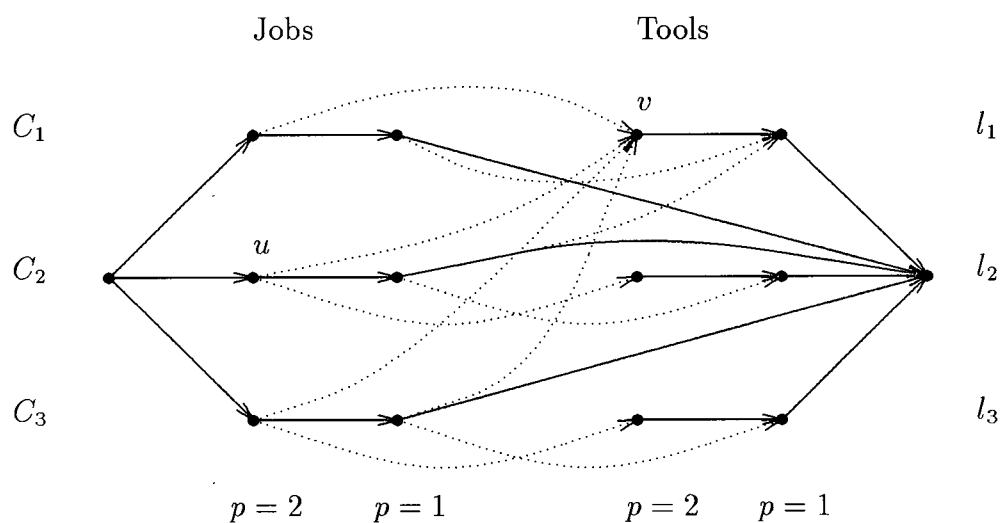


Figure 3.18: The original network

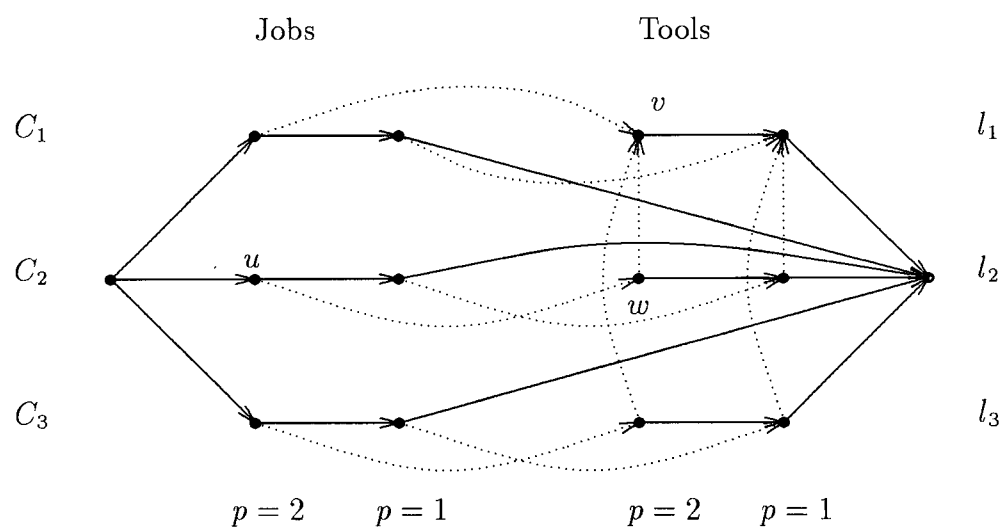


Figure 3.19: The modified network

to the modified network, I merely replace some *arcs* of infinite capacities with *paths* of infinite capacities. For example, arc  $(u, v)$  in Figure 3.18 is replaced by path  $\langle u, w, v \rangle$  in Figure 3.19.

### 3.3.2 A Forest Harvest Scheduling Problem

In logging operations, a watershed is usually divided into smaller blocks, which are contiguous areas and in which the tree species and their maturity are similar. In order to harvest a block, roads must be built from a saw mill to this block. The set of possible routes are predetermined, mainly by the landscape of the area. The possible routes can be represented by a network in which each node is either a saw mill or a block of forest, and each arc is a segment of road between the two nodes. These segments of road are called legs. I assume that capacities of the roads and of the saw mills are large enough so that these capacities can be ignored. With this assumption, it is unnecessary to deliver logs from a block to more than one saw mill or to build multiple roads from a block to a saw mill. Thus I can assume that the network of roads is a forest, with a saw mill as the root of each tree of roads therein.

Schedules must be made to determine which blocks should be cut and if so, at which time period. Each time period is usually 10 years, and the planning time horizon can be as long as 12 periods, or 120 years. A good harvest schedule must take into account the growth of trees, the cost of building the roads, and environmental factors. Naturally, it would be unwise to cut a block of trees when it is still in a rapidly growing stage. Further, due to the lengthy planning horizon, the value of harvested wood and the cost of constructing the roads must be discounted to their present values.

I assume that the present value of revenue from harvesting each block of forest in each time period is known, and so is the cost in present value of building each leg of road in each time period. The problem is to find a forest-harvest and road-construction schedule

that yields the maximum net present income. A schedule is feasible if the roads from a saw mill to each block is available by the time the block is to be harvested. Since the planning horizon is usually no longer than the growth cycle of a tree, a block cannot be harvested twice within the planning horizon.

The above scheduling problem is a special case of the network scheduling problem discussed in Subsection 3.3.1. Indeed, harvesting a block in a certain time period in the former problem corresponds to making the commodity available to a consumer in this period in the latter problem. Thus this scheduling problem can be formulated as a minimum cut problem in the modified network presented in Subsection 3.3.1.

In real world logging operations, usually additional constraints must be considered. For example, adjacency constraints have been introduced recently in forestry legislations. They require that after a block is harvested, its neighboring blocks cannot be harvested until a certain green-up period has elapsed. With the introduction of these constraints, the problem becomes NP-hard. Indeed, any instance of the stable set problem can be formulated as a forest harvest scheduling problem with adjacency constraints in which the planning horizon is only one period.

### 3.3.3 An Open Pit Mining Scheduling Problem

In open pit mining, a body of ore deposit and waste is divided into blocks. Due to the difference in mineral concentration (grade), each block has a different value, which may change over time due to time discount and variations in ore price. Let  $c_{bp}$  be the present value of the cost to remove block  $b$  in period  $p$ , and let  $r_{bp}$  be the present value of the revenue from processing and selling block  $b$  in period  $p$ . To limit the slope of the pit, blocks in a cone above  $b$  must be excavated no later than the time period in which block  $b$  is excavated. I call the blocks in the cone above  $b$  the *predecessors* of  $b$ . For example, in Figure 3.20, blocks 1, 2 and 3 must be excavated no later than block 5, as indicated by the



arrows. Due to the lengthy planning horizon, excavation of the blocks must be properly scheduled in order to maximize the net present value. In real mining operations, such a schedule must take account of excavation capacity, proper mix of ore grades, and so on. With the introduction of these constraints, the problem is NP-hard. However, when the latter constraints are ignored, the problem can be formulated as an NFJ problem.

Indeed, for each block  $b$  and each time period  $p$ , job  $j(b, p)$  corresponds to processing and selling block  $b$  in time period  $p$ , facility  $f(b, p)$  corresponds to excavating block  $b$  in time period  $p$ , and task  $t(b, p)$  means that block  $b$  is already excavated in period  $p$ . Job  $j(b, p)$  yields a revenue  $r_{bp}$  and requires tasks  $t(b_1, p_1)$  for each  $p_1 \geq p$  and for each predecessor  $b_1$  of  $b$ . Facility  $f(b, p)$  costs  $c_{bp}$  and can perform tasks  $t(b, p_1)$  for  $p_1 \geq p$ . Thus, the open pit mining scheduling problem is a special case of the NFJ model.

In the above problem, I assume that a mining company can remove a block of material in one period and process it in a later period. If a block must be processed in the same period it is excavated, then the problem cannot be formulated as an NFJ model. Dagdelen (1979) proposed a method to solve this latter version of the open pit mining scheduling problem by finding a minimum cut in a directed network. (See also Dagdelen 1992.) Based on similar ideas used in solving the NFJ model, I construct an alternative network to solve this version of the problem. My alternative network contains the same number of nodes but fewer arcs compared to Dagdelen's network.

I now briefly describe the method proposed by Dagdelen. Since excavation and processing must occur in the same period, I denote by  $u_{bp} = r_{bp} - c_{bp}$  the present value of the profit from excavating block  $b$  in period  $p$ . For each block  $b$  and each time period  $p$ , let  $x_{bp} = 1$  if block  $b$  is excavated in period  $p$  and 0 if it is not, let  $w_{bp} = x_{b1} + x_{b2} + \cdots + x_{bp}$ ,  $\bar{u}_{bp} = u_{bp} - u_{b,p+1}$  for  $p = 1, 2, \dots, L-1$  and let  $\bar{u}_{bL} = u_{bL}$ , where  $L$  is the last period. Dagdelen's network contains a node  $v_b^p$  for each block  $b$  and each period  $p$ , a source node  $s$  and a sink node  $t$ . For each period  $p$ , if block  $b_1$  is an immediate predecessor of  $b$ , then

there is an arc  $(v_b^p, v_{b_1}^p)$  in  $G$  with an infinite capacity. For each block  $b$  and each period  $p = 1, 2, \dots, L-1$ , there is an arc  $(v_b^p, v_b^{p+1})$  in  $G$  with an infinite capacity. For each node  $v_b^p$ , if  $\bar{u}_{bp} > 0$ , then there is an arc  $(s, v_b^p)$  in  $G$  with capacity  $\bar{u}_{bp}$ , otherwise, there is an arc  $(v_b^p, t)$  with capacity  $|\bar{u}_{bp}|$ . An optimal schedule for the open pit mining problem is derived from a minimum cut  $(X, \bar{X})$  in  $G$  by letting  $w_{bp} = 0$  for each  $v_b^p \in \bar{X}$  and  $w_{bp} = 1$  for each  $v_b^p \in X$ . To illustrate this network, consider a body of ore deposit shown in Figure 3.21, which is divided into three blocks. Suppose that the planning horizon consists of three time periods. The corresponding network  $G$  is shown in Figure 3.22.

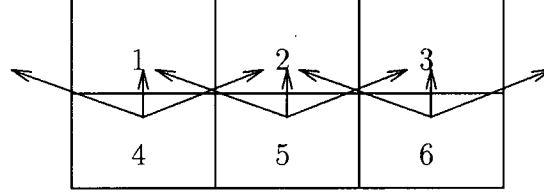


Figure 3.20: Predecessors in open pit mining

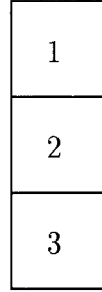


Figure 3.21: An example of ore body

My sparser network  $G'$  contains a node  $v_b^p$  for each block  $b$  and each time period  $p$ , a source node  $s$  and a sink node  $t$ . For each time period  $p$ , if  $b_1$  is an immediate predecessor of  $b$ , then  $G'$  contains an arc  $(v_b^p, v_{b_1}^p)$  with an infinite capacity. For each block  $b$  and each

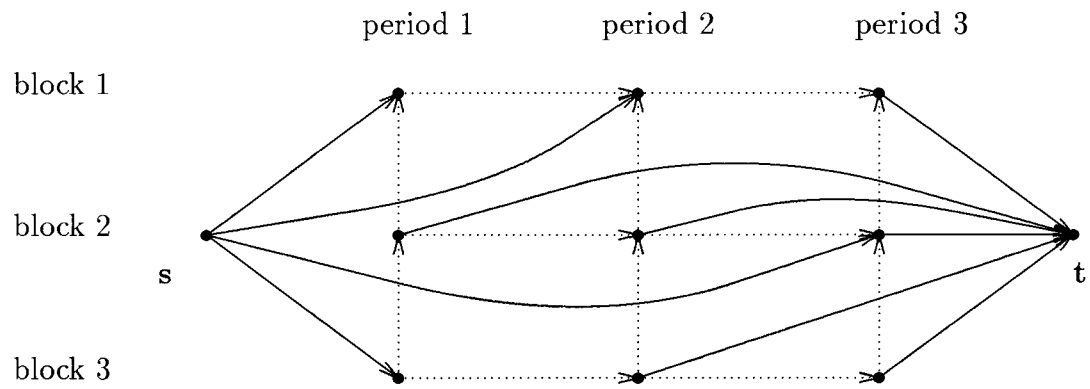


Figure 3.22: Dagdelen's network  $G$

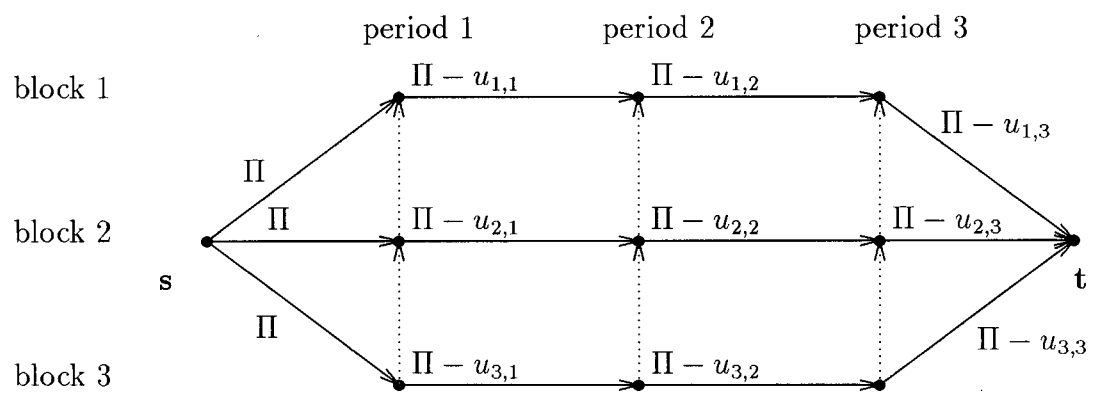


Figure 3.23: A sparser network  $G'$

period  $p = 1, \dots, L - 1$ ,  $G'$  contains an arc  $(v_b^p, v_b^{p+1})$  with capacity  $\Pi - u_{bp}$ , where  $\Pi$  is a constant that is larger than  $\max_{b,p} \{u_{bp}\}$ . Further for each block  $b$ ,  $G'$  contains arc  $(v_b^L, \mathbf{t})$  with capacity  $\Pi - u_{bL}$  and arc  $(\mathbf{s}, v_b^1)$  with capacity  $\Pi$ . The network  $G'$  corresponding to the example shown in Figure 3.21, again with three time periods, is given in Figure 3.23.

An optimal schedule for the open pit mining problem is derived from a minimum cut,  $(X, \overline{X})$ , in  $G'$  as follows. If arc  $(v_b^p, v_b^{p+1})$  (or when  $p = L$ , arc  $(v_b^L, \mathbf{t})$ ) is a forward arc in  $(X, \overline{X})$ , then block  $b$  should be excavated in time period  $p$ . If arc  $(\mathbf{s}, v_b^1)$  is a forward arc in  $(X, \overline{X})$ , then block  $b$  should never be excavated. I will not formally prove that the schedule thus obtained is optimal, but the validity of this result can be seen to follow from the following two arguments.

(1) For each block  $b$ , let  $P(b)$  be the path  $(\mathbf{s}, v_b^1, v_b^2, \dots, v_b^L, \mathbf{t})$ . If a cut  $(X, \overline{X})$  is a minimum cut, then it contains exactly one forward arc from each  $P(b)$ . Thus the capacity of any cut  $(X, \overline{X})$  with a finite capacity is  $n\Pi$  minus the payoff from the schedule, where  $n$  is the number of blocks in the ore deposit.

(2) For any cut  $(X, \overline{X})$  with a finite capacity, if node  $v_b^p$  is in  $X$ , then  $v_{b_1}^p$  must also be in  $X$  for any predecessor  $b_1$  of  $b$ . This is because there is a path from  $v_b^p$  to  $v_{b_1}^p$  with an infinite capacity. Therefore, if a block  $b$  is excavated in time period  $p$ , its predecessors must be excavated no later than  $p$ .

For a problem with  $n$  blocks  $r$  precedence relations<sup>1</sup> and  $t$  periods, my network has  $nt - 2n$  fewer arcs than in the original network, which contains  $rt + 2nt - n$  arcs.

### 3.4 Some Qualitative Properties

In this section, I derive some qualitative properties of optimal solutions in parametric NF and NFJ models. As shown in Subsection 3.2.2, the NF model is a special case of

<sup>1</sup>A precedence relation is a pair of blocks such that one immediately precedes the other.

the NFJ model. Therefore, whenever a result in this section holds for both the NF and NFJ models, I only prove it for the latter model. My study in this section uses lattice programming methodology. The reader is referred to Section 1.2 and Subsection 2.1.1 for a detailed discussion on related research in lattice programming.

In the parametric optimization problems considered in this section,  $\mathcal{L}$  will be the set of all selection plans and  $g(x, \lambda)$  will be the net return function. Thus in all these problems,  $\mathcal{L}$  is compact and  $g(x, \lambda)$  is lower semi-continuous, although I will not verify this point to avoid unnecessary mathematical details. Therefore, in the sequel, whenever the set of optimal solutions is ascending (respectively, descending), an increasing (respectively, decreasing) selection of optimal solutions exists.

To present my monotonicity results, I first introduce the following partial orders. In the NFJ model, for two subsets of jobs  $J_1$  and  $J_2$ ,  $J_1 \preceq_J J_2$  if for every job  $j_1 \in J_1$ , there exists a job  $j_2 \in J_2$  such that  $j_1 \preceq_j j_2$ . In both the NFJ and NF models, for two subsets of facilities  $F_1$  and  $F_2$ ,  $F_1 \preceq_F F_2$  if for every facility  $f_1 \in F_1$ , there exists a facility  $f_2 \in F_2$  such that  $f_1 \preceq_f f_2$ .

In the NFJ model, for each elementary job  $j$ , an artificial *null job*  $\underline{j}(j)$  is added, which has no utility to the customer and yields no revenue. I assume that  $\underline{j}(j) \preceq_j j$  is valid. If an elementary job  $j$  is not in a collection of jobs  $J$ , and neither are super-jobs of  $j$ , then I assume that  $\underline{j}(j)$  is in  $J$ . When an NF model is viewed as a special case of the NFJ model,  $J_1 \preceq_J J_2$  if and only if  $J_1 \subseteq J_2$ . Similarly, an artificial *null facility*  $\underline{f}(f)$  can be added which performs no tasks and has zero cost. I assume that  $\underline{f}(f) \preceq_f f$  is valid. If an elementary facility  $f$  is not in a collection of facilities  $F$ , and neither are super-facilities of  $f$ , then I assume that  $\underline{f}(f)$  is in  $F$ . The following propositions are needed to prove the main results in this section.

**PROPOSITION 3.4** *In the NF or NFJ models, let  $f$  be an arbitrary facility in  $\mathcal{F}$ ,*

and let  $F$  be an arbitrary non-redundant collection of facilities. Then,  $F$  must contain at least one facility that is totally ordered with  $f$ .

**Proof.** If  $F$  does not contain a real facility that is totally ordered with  $f$ , then all the null sub-facilities of  $f$  will be contained in  $F$ . Each such null facility is totally ordered with  $f$ . □

**PROPOSITION 3.5** *In the NF or NFJ models, let  $j$  be an arbitrary job in  $\mathcal{J}$ , and let  $J$  be an arbitrary non-redundant collection of facilities. Then,  $J$  must contain at least one job that is totally ordered with  $j$ .*

**Proof.** Similar to the proof of Proposition 3.4 and thus omitted. □

The following proposition considers the union and intersection of two sets of facilities. Recall that a *chain* in a partially ordered set is a totally ordered subset thereof.

**PROPOSITION 3.6** *In the NF or NFJ models, let  $F_1$  and  $F_2$  be two non-redundant collections of facilities. Then the following hold.*

- (i) *A chain in  $F_1 \cup F_2$  contains at most two facilities, one from  $F_1$  and the other from  $F_2$ .*
- (ii) *A chain in  $F_1 \cup F_2$  contains a single facility  $f_1$ , if and only if  $f_1$  is in both  $F_1$  and  $F_2$ .*

**Proof.** Since the collections  $F_1$  and  $F_2$  are non-redundant, no two facilities in  $F_1$  can be totally ordered, and the same is valid for  $F_2$ . Thus (i) holds. To show (ii), suppose  $f_1$  is the unique facility in a maximal chain in  $F_1 \cup F_2$  and, without loss of generality, assume  $f_1 \in F_1$ . By Proposition 3.4,  $F_2$  contains a facility  $f_2$  that is totally ordered with

$f_1$ . If  $f_2 \neq f_1$ , then  $f_1$  cannot be a maximal chain in  $F_1 \cup F_2$ . Thus  $f_1 \in F_2$ . On the other hand, if  $f_1 \in F_1 \cap F_2$ , then by the non-redundancy of  $F_1$  and  $F_2$ ,  $f_1$  must be the unique element in a chain containing  $f_1$ .  $\square$

Similarly, the following proposition is valid.

**PROPOSITION 3.7** *In the NF or NFJ models, let  $J_1$  and  $J_2$  be two non-redundant collections of jobs. Then the following hold.*

- (i) *A chain of jobs in  $J_1 \cup J_2$  contains at most two jobs, one from  $J_1$  and the other from  $J_2$ .*
- (ii) *A maximal chain in  $J_1 \cup J_2$  contains a single job  $j_1$ , if and only if  $j_1$  is in both  $J_1$  and  $J_2$ .*

Let  $F_1 \wedge F_2$  and  $F_1 \vee F_2$  be, respectively, the supremum and infimum of two collections of facilities  $F_1$  and  $F_2$  with respect to the  $\preceq_F$  order among the collections of facilities. Similarly, let  $J_1 \wedge J_2$  and  $J_1 \vee J_2$  be, respectively, the supremum and infimum of two collections of jobs  $J_1$  and  $J_2$  with respect to the  $\preceq_J$  order among the collections of jobs. In view of Proposition 3.6,  $F_1 \wedge F_2$  can be constructed by picking the lower facility in each maximal chain in  $F_1 \cup F_2$ . If a maximal chain contains only one facility, then  $F_1 \wedge F_2$  contains this facility. Similarly,  $F_1 \vee F_2$  can be constructed by picking the higher facility in each maximal chain in  $F_1 \cup F_2$ . Again, if a maximal chain contains only one facility, then  $F_1 \vee F_2$  contains this facility. Similarly,  $J_1 \wedge J_2$  can be constructed by picking the lower job in each maximal chain in  $J_1 \cup J_2$  and  $J_1 \vee J_2$  can be constructed by picking the higher job in each maximal chain.

**PROPOSITION 3.8** *In the NF or NFJ models, let  $F_1$  and  $F_2$  be two non-redundant collections of facilities. Then,*

- (i)  $F_1 \cup F_2 = (F_1 \vee F_2) \cup (F_1 \wedge F_2)$ , and  $F_1 \cap F_2 = (F_1 \vee F_2) \cap (F_1 \wedge F_2)$ ,
- (ii)  $c(F_1) + c(F_2) = c(F_1 \wedge F_2) + c(F_1 \vee F_2)$ , i.e., the cost is modular in the collection of facilities,
- (iii)  $T(F_1) \cup T(F_2) = T(F_1 \vee F_2)$  and  $T(F_1) \cap T(F_2) = T(F_1 \wedge F_2)$ .

**Proof.** Part (i) of the proposition follows from the construction of  $F_1 \wedge F_2$  and  $F_1 \vee F_2$ . In constructing  $F_1 \wedge F_2$  or  $F_1 \vee F_2$ , one only need to use facilities in  $F_1 \cup F_2$ . Furthermore, any facility in  $F_1 \cup F_2$  will either go into  $F_1 \wedge F_2$ , if it is the lower facility in a maximal chain, or go into  $F_1 \vee F_2$ , if it is the higher facility in a maximal chain. Thus  $F_1 \cup F_2 = (F_1 \vee F_2) \cup (F_1 \wedge F_2)$ .

If  $f$  is a maximal chain by itself in  $F_1 \cup F_2$ , then  $f$  is in both  $F_1 \cap F_2$  and  $(F_1 \vee F_2) \cap (F_1 \wedge F_2)$ , otherwise it is in neither of them. Thus  $F_1 \cap F_2 = (F_1 \vee F_2) \cap (F_1 \wedge F_2)$ .

Part (ii) of the proposition follows directly from part (i).

To prove part (iii), consider  $t$  in  $T(F_1) \cup T(F_2)$ . Without loss of generality, assume that  $t \in T(F_1)$ . Since  $T(F_1) \subseteq T(F_1 \vee F_2)$ ,  $t \in T(F_1 \vee F_2)$ . On the other hand, if  $t \in T(F_1 \vee F_2)$ , then there exists a facility  $f$  in  $F_1 \vee F_2$  that can perform  $t$ . From the construction of  $F_1 \vee F_2$ ,  $f$  is either in  $F_1$  or in  $F_2$ . Thus  $t \in T(F_1) \cup T(F_2)$ .

The second half of part (iii) can be proved similarly. □

**PROPOSITION 3.9** *In the NF or NFJ models, let  $J_1$  and  $J_2$  be two non-redundant collections of jobs. Then*

- (i)  $J_1 \cup J_2 = (J_1 \vee J_2) \cup (J_1 \wedge J_2)$ , and  $J_1 \cap J_2 = (J_1 \vee J_2) \cap (J_1 \wedge J_2)$ ,
- (ii)  $r(J_1) + r(J_2) = r(J_1 \wedge J_2) + r(J_1 \vee J_2)$ , i.e., the revenue is modular in the collection of jobs,
- (iii)  $T(J_1) \cup T(J_2) = T(J_1 \vee J_2)$  and  $T(J_1) \cap T(J_2) \supseteq T(J_1 \wedge J_2)$ .



**Proof.** The proof of part (i) is similar to that of part (i) in Proposition 3.8 and is thus omitted. Part (ii) follows directly from part (i).

To show the first half of part (iii), consider a task  $t \in T(J_1) \cup T(J_2)$ . Without loss of generality, assume  $t \in T(J_1)$ . Suppose  $t$  is required by a job  $j$  in  $J_1$ . By the construction of  $J_1 \vee J_2$ , either  $j$  is in  $J_1 \vee J_2$ , or there exists another job  $j_1$  such that  $j \preceq_j j_1$  and  $j_1$  is in  $J_1 \vee J_2$ . Since  $j_1$  also requires  $t$ , in either case,  $t \in T(J_1 \vee J_2)$ . On the other hand, suppose  $t \in T(J_1 \vee J_2)$ . Then it is required by a job  $j \in J_1 \vee J_2$ . Since each job in  $J_1 \vee J_2$  is either in  $J_1$  or in  $J_2$ , without loss of generality, I can assume that  $j$  is in  $J_1$ . Then  $t$  is in  $T(J_1)$ , and the first half of part (iii) is established.

To prove the second half of part (iii), consider a task  $t \in T(J_1 \wedge J_2)$ . Then there exists a job  $j \in J_1 \wedge J_2$  such that  $t$  is required by  $j$ . By the construction of  $J_1 \wedge J_2$ ,  $j$  belongs to either  $J_1$  or  $J_2$ . Without loss of generality, assume that  $j \in J_1$ . Then there exists a job  $j_1 \in J_2$  such that  $j \preceq_j j_1$ , and therefore, both  $j$  and  $j_1$  require  $t$ . Thus  $t \in T(J_1) \cap T(J_2)$ .  $\square$

In general,  $T(J_1) \cap T(J_2) \subseteq T(J_1 \wedge J_2)$  does not hold. Indeed, consider the example where  $J_1 = \{j_1, j_4\}$ ,  $J_2 = \{j_2, j_3\}$ ,  $j_1 \preceq_j j_2$ ,  $j_3 \preceq_j j_4$ ,  $T(j_1) = T(j_3) = \emptyset$ , and  $T(j_2) = T(j_4) = \{t\}$ . Clearly,  $J_1 \wedge J_2 = \{j_1, j_3\}$  and  $T(J_1 \wedge J_2) = \emptyset$ . However,  $T(J_1) \cap T(J_2) = \{t\}$ .

For a given collection of facilities  $F$ , let  $J^*(F)$  be the (unique) set of jobs that can be performed by  $F$  and which yields the largest revenue. Similarly, the least expensive collection of facilities that can perform a set of jobs  $J$  is denoted by  $F^*(J)$ . In the NFJ model,  $J^*(F)$  and  $F^*(J)$  can be obtained easily.

**PROPOSITION 3.10** *In the NF or NFJ models,  $c(F^*(J))$  is submodular in  $J$  with respect to the  $\preceq_J$  order.*

**Proof.** Consider two sets of jobs  $J_1$  and  $J_2$ .

$$\begin{aligned}
 T(J_1 \wedge J_2) &\subseteq T(J_1) \cap T(J_2) && \text{by Proposition 3.9(iii)} \\
 &\subseteq T(F^*(J_1)) \cap T(F^*(J_2)) && \text{by the feasibility of the selection plans} \\
 &= T(F^*(J_1) \wedge F^*(J_2)) && \text{by Proposition 3.8(iii),}
 \end{aligned}$$

and

$$\begin{aligned}
 T(J_1 \vee J_2) &= T(J_1) \cup T(J_2) && \text{by Proposition 3.9(iii)} \\
 &\subseteq T(F^*(J_1)) \cup T(F^*(J_2)) && \text{by the feasibility of the selection plans} \\
 &= T(F^*(J_1) \vee F^*(J_2)) && \text{by Proposition 3.8(iii).}
 \end{aligned}$$

Thus  $F^*(J_1) \wedge F^*(J_2)$  can perform  $J_1 \wedge J_2$ , and since  $F^*(J_1 \wedge J_2)$  is the least expensive collection of facilities that can perform  $J_1 \wedge J_2$ ,  $c(F^*(J_1 \wedge J_2)) \leq c(F^*(J_1) \wedge F^*(J_2))$ . Similarly,  $c(F^*(J_1 \vee J_2)) \leq c(F^*(J_1) \vee F^*(J_2))$ . The last two inequalities imply:

$$\begin{aligned}
 c(F^*(J_1 \wedge J_2)) + c(F^*(J_1 \vee J_2)) &\leq c(F^*(J_1) \wedge F^*(J_2)) + c(F^*(J_1) \vee F^*(J_2)) \\
 &= c(F^*(J_1)) + c(F^*(J_2)) \\
 &\quad \text{by Proposition 3.8(ii).}
 \end{aligned}$$

**PROPOSITION 3.11** *In the NF or NFJ models,  $r(J^*(F))$  is supermodular in  $F$  with respect to the  $\preceq_F$  order.*

**Proof.** Consider two collections of facilities  $F_1$  and  $F_2$ .

$$\begin{aligned}
 T(F_1 \wedge F_2) &= T(F_1) \cap T(F_2) && \text{by Proposition 3.8(iii)} \\
 &\supseteq T(J^*(F_1)) \cap T(J^*(F_2)) && \text{by the feasibility of the selection plans} \\
 &\supseteq T(J^*(F_1) \wedge J^*(F_2)) && \text{by Proposition 3.9(iii),}
 \end{aligned}$$

and

$$\begin{aligned}
 T(F_1 \vee F_2) &= T(F_1) \cup T(F_2) && \text{by Proposition 3.8(iii)} \\
 &\supseteq T(J^*(F_1)) \cap T(J^*(F_2)) && \text{by the feasibility of the selection plans} \\
 &= T(J^*(F_1) \wedge J^*(F_2)) && \text{by Proposition 3.9(iii).}
 \end{aligned}$$

Thus  $J^*(F_1) \wedge J^*(F_2)$  can be performed by  $F_1 \wedge F_2$  and by the definition of  $J^*(\cdot)$ ,  $r(J^*(F_1 \wedge F_2)) \geq r(J^*(F_1) \wedge J^*(F_2))$ . Similarly,  $r(J^*(F_1 \vee F_2)) \geq r(J^*(F_1) \vee J^*(F_2))$ .

The last two inequalities imply:

$$\begin{aligned} r(J^*(F_1 \wedge F_2)) + r(J^*(F_1 \vee F_2)) &\geq r(J^*(F_1) \wedge J^*(F_2)) + r(J^*(F_1) \vee J^*(F_2)) \\ &= r(J^*(F_1)) + r(J^*(F_2)) \\ &\quad \text{by Proposition 3.9(ii).} \end{aligned}$$

□

With the introduction of  $F^*(J)$  and  $J^*(F)$ , the NFJ model can be rewritten as

$$\max_{F \in \mathcal{F}} g(F, J^*(F); \lambda),$$

or,

$$\max_{J \in \mathcal{J}} g(F^*(J), J; \lambda).$$

For convenience, in the rest of the chapter, the objective function  $g$  may have different arguments. For example,  $g(J; \lambda)$  or  $g(J)$  is the return of a selection plan  $(F^*(J), J)$  under parameter  $\lambda$ ,  $g(F; c)$  is the return of collection  $(J^*(F), F)$  when the costs of the facilities are given by a function  $c(\cdot)$ .

**LEMMA 3.12** *In the NF or NFJ models,  $g(J)$  and  $g(F)$  are supermodular in  $J$  and  $F$ , respectively.*

**Proof.**  $g(J) = \lambda r(J) - c(F^*(J))$ . By Proposition 3.9(ii),  $r(J)$  is modular, and by Proposition 3.10,  $c(F^*(J))$  is submodular in  $J$ , implying that  $-c(F^*(J))$  is supermodular. As the sum of supermodular functions is supermodular, the result follows. One can similarly prove that  $g(F)$  is supermodular. □

**LEMMA 3.13** *In the NF or NFJ models,  $g(F; \lambda)$  has isotone differences in  $F$  and  $\lambda$ .*

**Proof.** Consider  $\lambda_1 \leq \lambda_2$  and  $g(F; \lambda_2) - g(F; \lambda_1) = (\lambda_2 - \lambda_1)r(J^*(F))$ . Since  $r(J^*(F))$  is nondecreasing in  $F$ ,  $g(F; \lambda)$  has isotone differences in  $F$  and  $\lambda$ .  $\square$

Suppose  $g(F; \lambda)$  is parameterized on  $\lambda$ . Let  $\mathcal{F}^*(\lambda) = \{F | (F, J^*(F)) \text{ is optimal for } g(F; \lambda)\}$ , and let  $\mathcal{J}^*(\lambda) = \{J | (F^*(J), J) \text{ is optimal for } g(J; \lambda)\}$ . Similarly, if  $g(F; c)$  is parameterized on the functional  $c(\cdot)$ , then let  $\mathcal{F}^*(c) = \{F | (F, J^*(F)) \text{ is optimal for } g(F; c)\}$ , and so on.

**THEOREM 3.14** *In the NF or NFJ models,  $\mathcal{F}^*(\lambda)$  and  $\mathcal{J}^*(\lambda)$  are both sublattices that are ascending in  $\lambda$ .*

**Proof.** By Lemmas 3.12 and 3.13,  $g(F; \lambda)$  is supermodular in  $F$  and has isotone differences in  $F$  and  $\lambda$ . By Theorem 2.1,  $\mathcal{F}^*(\lambda)$  is an ascending sublattice in  $\lambda$ .

To prove that  $\mathcal{J}^*(\lambda)$  is an ascending sublattice in  $\lambda$ , consider  $\lambda_1 \leq \lambda_2$  and let  $J_1$  and  $J_2$  be optimal collections of jobs for  $\lambda_1$  and  $\lambda_2$ , respectively. Let  $F_1 = F^*(J_1)$  and  $F_2 = F^*(J_2)$ . Then,  $F_1 \in \mathcal{F}^*(\lambda_1)$  and  $F_2 \in \mathcal{F}^*(\lambda_2)$ . By the first half of the proof,  $\mathcal{F}^*(\lambda)$  is ascending in  $\lambda$ , and it follows that  $F_1 \vee F_2 \in \mathcal{F}^*(\lambda_2)$  and  $F_1 \wedge F_2 \in \mathcal{F}^*(\lambda_1)$ . By Propositions 3.8(iii), 3.9(iii) and the fact that  $T(J_1) \subseteq T(F_1)$  and  $T(J_2) \subseteq T(F_2)$ ,  $T(J_1 \wedge J_2) \subseteq T(J_1) \cap T(J_2) \subseteq T(F_1) \cap T(F_2) = T(F_1 \wedge F_2)$ . Thus,  $F_1 \wedge F_2$  is sufficient to perform  $J_1 \wedge J_2$ .

Since  $(J_1, F_1)$  is optimal for  $\lambda_1$ ,  $\lambda_1 r(J_1 \wedge J_2) - c(F_1 \wedge F_2) \leq \lambda_1 r(J_1) - c(F_1)$ , or,

$$\lambda_1(r(J_1) - r(J_1 \wedge J_2)) - (c(F_1) - c(F_1 \wedge F_2)) \geq 0. \quad (3.46)$$

On the other hand, both  $(J_1, F_1)$  and  $(J_1 \wedge J_2, F_1)$  are feasible selection plans, and the former is optimal for  $\lambda_1$ . Therefore,  $r(J_1) - r(J_1 \wedge J_2) \geq 0$ . Since  $\lambda_2 \geq \lambda_1 > 0$ , (3.46)

still holds if  $\lambda_1$  therein is replaced with  $\lambda_2$ . Therefore,

$$\lambda_2(r(J_1) - r(J_1 \wedge J_2)) - c(F_1) + c(F_1 \wedge F_2) \geq 0. \quad (3.47)$$

By adding  $\lambda_2 r(J_2) - c(F_2)$  to both sides of (3.47), it follows from the modularity of  $r(\cdot)$  and  $c(\cdot)$  that,

$$\begin{aligned} & \lambda_2(r(J_1) + r(J_2) - r(J_1 \wedge J_2)) - c(F_1) - c(F_2) + c(F_1 \wedge F_2) \\ = & \lambda_2 r(J_1 \vee J_2) - c(F_1 \vee F_2) \\ \geq & \lambda_2 r(J_2) - c(F_2). \end{aligned} \quad (3.48)$$

Thus  $(J_1 \vee J_2, F_1 \vee F_2)$  is also an optimal selection plan for  $\lambda_2$  and

$$\lambda_2 r(J_1 \vee J_2) - c(F_1 \vee F_2) = \lambda_2 r(J_2) - c(F_2).$$

Substituting the latter equality back to the equality part of (3.48) yields

$$\lambda_2(r(J_1) - r(J_1 \wedge J_2)) - c(F_1) + c(F_1 \wedge F_2) = 0. \quad (3.49)$$

Since  $r(J_1) - r(J_1 \wedge J_2) \geq 0$  and  $\lambda_2 \geq \lambda_1 > 0$ , replacing  $\lambda_2$  with  $\lambda_1$  in (3.49) yields

$$\lambda_1(r(J_1) - r(J_1 \wedge J_2)) - c(F_1) + c(F_1 \wedge F_2) \leq 0, \quad (3.50)$$

or

$$\lambda_1 r(J_1) - c(F_1) \leq \lambda_1 r(J_1 \wedge J_2) - c(F_1 \wedge F_2).$$

It follows that  $(J_1 \wedge J_2, F_1 \wedge F_2)$  is an optimal selection plan for  $\lambda_1$ , which completes the proof.  $\square$

Now I introduce a partial order among the revenue functions. For two revenue functions  $r_1$  and  $r_2$ ,  $r_1 \preceq r_2$  if  $r_1(j) \leq r_2(j)$  for every  $j \in \mathcal{J}$ . Similarly, I define a partial order among the cost functions. For two cost functions  $c_1$  and  $c_2$ ,  $c_1 \preceq c_2$  if  $c_1(f) \leq c_2(f)$  for every  $f \in \mathcal{F}$ .

**LEMMA 3.15** *In the NF model,  $g(F; r)$  has isotone differences in  $F$  and  $r$ , and  $g(J; r)$  has isotone differences in  $J$  and  $r$ .*

**Proof.** I first prove the first half of the lemma. Consider two revenue functions  $r_1(\cdot)$  and  $r_2(\cdot)$  such that  $r_1(j) \leq r_2(j)$  for every  $j \in \mathcal{J}$ . Consider the following difference:

$$\begin{aligned} & g(F; r_2) - g(F; r_1) \\ &= r_2(J^*(F)) - c(F) - r_1(J^*(F)) + c(F) \\ &= \sum_{j \in J^*(F)} (r_2(j) - r_1(j)). \end{aligned}$$

Since the revenues are positive,  $J^*(F)$  is increasing in  $F$  with respect to the partial order  $\preceq_F$ . By assumption,  $r_2(j) - r_1(j)$  is nonnegative for each  $j$ . Thus  $g(F; r_2) - g(F; r_1)$  is nondecreasing in  $F$ .

To prove the second half of the lemma, consider the following difference.

$$\begin{aligned} & g(J; r_2) - g(J; r_1) \\ &= r_2(J) - c(F^*(J)) - r_1(J) + c(F^*(J)) \\ &= \sum_{j \in J} (r_2(j) - r_1(j)). \end{aligned}$$

Since  $r_2(j) - r_1(j)$  is nonnegative,  $g(J; r_2) - g(J; r_1)$  is nondecreasing in  $J$ , which completes the proof.  $\square$

**THEOREM 3.16** *In the NF model, both  $\mathcal{F}^*(r)$  and  $\mathcal{J}^*(r)$  are ascending sublattices in  $r$ .*

**Proof.** Proof of the theorem follows from Lemmas 3.12 and 3.15. Indeed,  $g(F; r)$  is supermodular in  $F$  and has isotone differences in  $F$  and  $r$ . Thus by Theorem 2.1,  $\mathcal{F}^*(r)$  is an ascending sublattice in  $r$ . Similarly, since  $g(J; r)$  is supermodular in  $J$  and has isotone differences in  $J$  and  $r$ ,  $\mathcal{J}^*(r)$  is an ascending sublattice in  $r$ .  $\square$

The above theorem does not hold for the NFJ model. Indeed, consider an example with three jobs,  $\{j_1, j_2, j_3\}$ , and three facilities  $\{f_1, f_2, f_3\}$ . Each facility performs a single task, which is identical to the facility itself;  $j_1$  requires  $f_1$ ,  $j_2$  requires  $\{f_1, f_2\}$ , and  $j_3$  requires  $f_3$ ;  $j_1 \preceq_j j_2$ , and only one of these two jobs may appear in a feasible selection plan due to the Non-Redundant Jobs Assumption. The costs of facilities  $f_1$ ,  $f_2$  and  $f_3$  are 1, 2 and 2, respectively. Initially, the revenues for  $j_1$ ,  $j_2$  and  $j_3$  are 1, 4 and 1, respectively, and the unique optimal selection plan is to perform job  $j_2$  and to acquire facilities  $f_1$ ,  $f_2$ . Now increase the revenues for jobs  $j_1$ ,  $j_2$  and  $j_3$  to 4, 4, 3, respectively. The unique optimal selection plan under the new revenue function is to perform jobs  $j_1$  and  $j_3$  and to acquire facilities  $f_1$  and  $f_3$ .

It was shown by Brumelle and Granot (1993) that for the original selection problem,  $\mathcal{F}^*(c)$  and  $\mathcal{J}^*(c)$  are ascending sublattices in  $c$ , the facility cost function. This, however, is not valid for the NF or NFJ models. Indeed, consider an NF model which contains three facilities and three jobs. The three facilities,  $f_1$ ,  $f_2$  and  $f_3$  perform tasks  $\{t_1\}$ ,  $\{t_1, t_2\}$  and  $\{t_3\}$ , respectively. The three jobs,  $j_1$ ,  $j_2$  and  $j_3$  require tasks  $\{t_1\}$ ,  $\{t_2\}$  and  $\{t_3\}$ , and their revenues are 10, 2 and 2, respectively. Note that facilities  $f_1$  and  $f_2$  are nested. Initially, the three facilities cost 1, 10 and 1, respectively, and the unique optimal selection plan is to acquire facilities  $f_1$  and  $f_3$ , and to perform jobs  $j_1$  and  $j_3$ . Now increase the costs of  $f_1$ ,  $f_2$  and  $f_3$  to 10, 10 and 10, respectively. Then the unique optimal selection plan is to acquire facility  $f_2$ , and to perform jobs  $j_1$  and  $j_2$ . Clearly, the two optimal selection plans are neither monotone nor totally ordered.

When the sets of tasks performed by the facilities increase, the optimal selection plan is neither monotone nor totally ordered for either the NF model or the NFJ model. This is illustrated in the following example of an NF model, in which the potential jobs are  $j_1$ ,  $j_2$  and  $j_3$ , with corresponding revenues 1, 10 and 2, respectively. The jobs require the sets of tasks  $\{t_1\}$ ,  $\{t_2\}$  and  $\{t_3\}$ , respectively. There are three facilities,  $f_1$ ,  $f_2$  and  $f_3$ ,

which cost 0.5, 10 and 1, respectively. Facilities  $f_1$  and  $f_2$  are nested and perform the sets of tasks  $\emptyset$  and  $\{t_1, t_2\}$ , respectively. Facility  $f_3$  performs  $\{t_4\}$ . One can verify that the unique optimal selection plan is to acquire facility  $f_2$ , and to perform jobs  $j_1$  and  $j_2$ . The profit from the selection plan is 1. Now increase the set of tasks performed by the facilities so that  $f_1$  performs  $\{t_2\}$ ,  $f_2$  still performs  $\{t_1, t_2\}$ , and  $f_3$  performs  $\{t_3, t_4\}$ . Then the unique optimal selection plan is to acquire facilities  $f_1$  and  $f_3$  and to perform jobs  $j_2$  and  $j_3$ , with an associated profit of 10.5. It can be seen that the two optimal selection plans are not ordered.

Now I construct an example to show that when the tasks required by the jobs increase, the optimal selection plans are neither monotone nor totally ordered for either the NF or the NFJ model. In the following NF problem, there are three jobs,  $j_1, j_2, j_3$ , with revenues 20, 1, and 10, respectively. They require tasks  $\{t_1\}$ ,  $\{t_2\}$  and  $\{t_3\}$ , respectively. There are four potential facilities,  $f_1, f_2, f_3$  and  $f_4$ , with costs 1, 2, 1 and 20, respectively, and they perform tasks  $\{t_1\}$ ,  $\{t_1, t_2\}$ ,  $\{t_3\}$  and  $\{t_3, t_4\}$ , respectively. Note that facilities  $f_1, f_2$  are nested and facilities  $f_3, f_4$  are nested. The optimal selection plan is to acquire facilities  $f_1$  and  $f_3$  and to perform jobs  $j_1$  and  $j_3$ . Now increase the sets of tasks required by the jobs so that  $j_1$  requires  $\{t_1, t_2\}$  and  $j_3$  requires  $\{t_3, t_4\}$ . The tasks required by the other jobs remain the same. The new optimal selection plan is to acquire facility  $f_2$  and to perform jobs  $j_1$  and  $j_2$ . Clearly, the two optimal selection plans are not totally ordered.

Table 3.6 summarizes the monotone results for the NF, NFJ and the original selection model<sup>2</sup>. In Table 3.6, the monotonicity results are listed by different parameterizations. For example, " $T_1(j) \subseteq T_2(j), \forall j \in \mathcal{J}$ " means that the parameterization is on the tasks required by the facilities and that for two instances,  $T_1$  and  $T_2$ , of the parameter  $T(\cdot)$ , the set of tasks required by an arbitrary facility for the latter instance includes that for

---

<sup>2</sup>See Brumelle and Granot (1993) for details of the monotonicity results for the original selection problem.



Parameterization	NF	NFJ	Original
$\lambda_1 \leq \lambda_2$	ascending	ascending	ascending
$c_1(f) \leq c_2(f), \forall f \in \mathcal{F}$	no	no	descending
$r_1(j) \leq r_2(j), \forall j \in \mathcal{J}$	ascending	no	ascending
$\mathcal{F}_1 \subseteq \mathcal{F}_2$	no	no	ascending
$\mathcal{J}_1 \subseteq \mathcal{J}_2$	ascending	no	ascending
$T_1(j) \subseteq T_2(j), \forall j \in \mathcal{J}$	no	no	NA
$T_1(f) \subseteq T_2(f), \forall f \in \mathcal{F}$	no	no	NA

Table 3.6: Summary of monotone results

the former instance. “Ascending” in the table means that the optimal set  $\mathcal{F}^*$  and  $\mathcal{J}^*$  are ascending sublattices in the parameter. “No” indicates that monotonicity results do not exist for this parameterization. “NA” indicates that the parameterization is not applicable for this model. Note that the parameterization represented by “ $\mathcal{F}_1 \subseteq \mathcal{F}_2$ ” in the table is a special case of “ $c_2 \leq c_1$ ”, and the parameterization represented by “ $\mathcal{J}_1 \subseteq \mathcal{J}_2$ ” is a special case of “ $r_1 \leq r_2$ ”.

Other sensitivity analysis results can be derived. For example, it often occurs that the costs of facilities in a nested collection change together. This may be caused by a price change for a common component used in all the nested facilities.

In Chapter 2, I have shown that if in a capacitated network, a path from the source node to the sink node contains no parallel arcs, then there exists a totally ordered selection of minimum cuts when the capacities on the path are parametrically changed. Consider two nested facilities  $f_1 \preceq_f f_2$ . Since the unique path from  $f_1$  to  $\mathbf{t}$  must pass  $f_2$ , the unique arcs originating from  $f_1$  and  $f_2$  cannot be parallel. Recall that these two arcs have capacities equal to the facility costs, thus when the costs of a collection of nested facilities change, a totally ordered selection of optimal selection plans exists. Finally, McCormick (1995) showed that when the costs or revenues in the NF or NFJ problems are parameterized

subject to some magnitude restrictions, the parametric optimization problem can be solved efficiently by the GGT algorithm (Gallo, Grigoriadis and Tarjan, 1989).

## Chapter 4

### Forest Harvest and Road Construction Scheduling

#### 4.1 The Model

In forest resource management, a watershed is divided into *blocks* (also referred to as cut blocks, forest management units, or units in other literature), each of which is a geographically contiguous area in which the topography, forest composition, etc. are consistent. A planning horizon is divided into a finite number of periods. For example, in solving a scheduling problem for the Tangier watershed in British Columbia, either 6 periods or 12 periods are used with each period being 10 years. In order to harvest a block, a road must be available from a saw mill or a transportation center to the block. Roads are constructed from a network of potential legs, which are assumed to form a graphical tree or a forest. The root of each tree in the network is a saw mill or a transportation center, and all other nodes in the network are cut blocks. An edge in the network is called a *leg* of road. A forest harvest and road construction scheduling problem is an assignment of a time period to each block in which it is to be harvested and to each leg of road in which it is to be built.

Two blocks are *neighbors* of each other if they are geographically adjacent. The *adjacency constraints* (also referred to as green-up constraint, spatial constraint, etc. in other literature) require that, if a block is harvested in a period  $p$ , then its neighbors can only be harvested after period  $p + g$  or before  $p - g$ , where  $g$  is a constant and is referred to as the *green-up time*. In solving the scheduling problem for the Tangier watershed, a

green-up time of 2 periods is used.

In order to maintain a steady level of employment, it is desirable for a schedule to yield about the same volume of timber in different periods. Indeed, the government of British Columbia requires that the volume harvested in each period by a logging company does not deviate more than 10% from a licensed amount, which is usually the same for all the periods.

The following is some notation used in the harvesting scheduling model:

- $B$  denotes the set of blocks in the forest.
- $P$  denotes the set of time periods, which includes period  $\infty$ . If the harvest time of block of forest is  $\infty$ , then it is never harvested in the planning horizon.
- $L$  denotes the set of legs of roads.
- $v(b, p)$  denotes the volume of timber on block  $b$  if it is harvested in time period  $p$ .  
Clearly,  $v(b, \infty) = 0$ .
- $c(l, p)$  denotes the cost in present value for building a leg of road  $l$  in time period  $p$ .  
Similarly,  $c(l, \infty) = 0$ .
- $m(b, p)$  denotes the present value of the contribution margin for harvesting a block  $b$  in time period  $p$ .

The contribution margin of a block harvested in a time period is the difference between the revenue from the timber in the block and the harvesting cost (arising from felling, hauling, etc.). The revenue is calculated by first multiplying the volume of timber in the block in time period  $p$  and an average price of timber of the corresponding species and quality, then discounting to the present value. Due to the growth of trees, both the volume and price may change in time. Since a leg of road is on the path leading to various blocks,

its cost cannot be attributed to any single block and thus the above mentioned harvesting cost does not include road cost.

A harvest schedule is a function  $h(\cdot)$  from  $B$  to  $P$ . For a block  $b$ ,  $h(b)$  is the time period in which the block is to be harvested. Thus it is assumed that the harvesting of a block starts and is completed within a single time period.

Associated with a harvest schedule  $h(\cdot)$ , there is a minimum cost road construction schedule allowing the scheduled harvest activities to take place. This road construction schedule is represented by a function  $r_h(\cdot)$  from  $L$  to  $P$ . Since each period in this model is 10 years, it can be assumed that a leg of road is available in the same time period as it is built. To ensure that the roads are ready when they are needed, a constraint

$$r_h(l) \leq h(b) \quad (4.51)$$

is introduced for each block  $b$  and each leg  $l$  on the road leading to  $b$ .

It is assumed that if the cost for building a leg of road in period 0 is  $C_0$ , then the cost (in present value) for building the leg in period  $p$  is  $C_0\beta^p$ , where  $\beta$  is a discount factor. Thus road cost is decreasing in time, and for a given harvest schedule  $h$ , a leg of road should be built at the latest period without violating (4.51). It then follows that for each leg  $l$ ,

$$r_h(l) = \min\{h(b) : l \text{ is on the road leading to } b\}.$$

The function  $r_h(\cdot)$  is then uniquely determined by  $h$ , and the minimum road cost associated with a harvesting schedule  $h$  is

$$C(h) = \sum_{l \in L} c(l, r_h(l)).$$

For a harvesting schedule  $h$ , the total volume of timber harvested during period  $p$  is

$$V(h, p) = \sum_{b \in B} v(b, h(b)),$$

and the total volume of timber harvested during the whole planning horizon is

$$V(h) = \sum_{p \in P} V(h, p).$$

The present value of income associated with a schedule  $h(\cdot)$  is

$$N(h) = \sum_{b \in B} m(b, h(b)) - \sum_{l \in L} c(l, r_h(l)).$$

The *unevenness* of flow for a schedule  $h$  is defined as

$$u(h) = \max_{p \in P} V(h, p) - \min_{p \in P} V(h, p).$$

Recall that I assume that harvesting of a block must start and finish in a single time period, however, a logging company can take a schedule generated by my heuristic and further reduce unevenness by shifting the harvesting time of a block by a fraction of the time period and thus allocating the volume of this block to two consecutive time periods.

The adjacency constraint requires that

$$|h(b_1) - h(b_2)| > 2 \tag{4.52}$$

for all pairs  $(b_1, b_2)$  such that  $b_1$  and  $b_2$  are adjacent to each other. In some cases, slight violations of the adjacency constraints are allowed. Therefore, for a schedule  $h$ , the *number of adjacency violations* is defined as

$$a(h) = \frac{1}{2} ||\{(b_1, b_2) : b_1, b_2 \in B, b_1 \sim b_2, |h(b_1) - h(b_2)| \leq 2\}||,$$

where  $b_1 \sim b_2$  means that  $b_1$  and  $b_2$  are adjacent to each other and  $|| \cdot ||$  denotes the cardinality of a set. The cardinality of the set is divided by 2 in the above definition to avoid double counting of  $(b_1, b_2)$  and  $(b_2, b_1)$ . A schedule is *feasible* if the number of violations is no larger than a bound  $A$ . If adjacency violations are not allowed, then  $A$  is set to 0.

The harvesting scheduling problem is a multi-criteria optimization problem in which the present value of net income is maximized and the unevenness in harvest volumes is minimized. Let  $w_u$  be the weight on unevenness, and without loss of generality, let the weight on the net income be 1. Then the two criteria can be combined into a single objective function and the problem can be formulated as follows:

$$\begin{aligned} \text{HP1: } \max & N(h) - w_u u(h) \\ & \text{subject to} \\ & a(h) \leq A. \end{aligned}$$

The number of adjacency violations could also be incorporated as an additional criterion in the objective function. However, since this number is a small integer<sup>1</sup> whereas the net income is usually in the magnitude of millions, the weight on the number of adjacency violations must be a very large number, otherwise the change in the number of adjacency violations will have little effect on the objective value. Then the algorithm would be very sensitive to the weight on the number of adjacency violations. Indeed in my computational experiments, I have observed that the heuristic is much more robust when the number of adjacency violations is incorporated in a constraint.

## 4.2 The Tabu Search Heuristic

I first briefly describe some of the characteristics of the tabu search method. Since tabu search is a general methodology and heuristic algorithms that fall within this scope may vary significantly, the following introduction is not intended to be a thorough review of this methodology. Rather, it merely covers some of the features used in my heuristic. The reader is referred to Glover and Laguna (1993) for a detailed review of tabu search.

---

<sup>1</sup>In my experiments on the Tangier watershed example, no more than 10 adjacency violations are allowed.

Tabu search is a neighborhood search method that solves combinatorial optimization problems to near optimality. In each iteration, a *current solution* is identified, which is usually feasible to the optimization problem. The heuristic compares a collection of *neighboring solutions* in each iteration and chooses the best one to be used as the current solution of the next iteration. Neighboring solutions differ from the current solution only slightly. For example, if the current solution is a vector, then one may define its neighbors to be the vectors that differ from the current one in at most one element. The transition from a current solution to the next one is called a *move*. To encourage the heuristic to search through a diverse collection of solutions, and to avoid being trapped at or near a local optima, when the current solution leaves a neighborhood, an early return to the same neighborhood should be discouraged. Thus after a move is made, a number of iterations is specified which is known as the *tabu tenure*. During this tabu tenure, moves which are likely to cause the current solution to return to the old neighborhood are prohibited, and these moves are said to be *tabu*. If however, a tabu move will generate a solution that is better than all the currently known ones, then an exception is made to allow this move to happen, and such a move is called *aspiration*.

In the heuristic presented in this chapter, each current solution is a harvest schedule  $h$  satisfying the bound on the number of adjacency violations. Since the minimum cost road construction schedule is uniquely determined by the harvest schedule, a harvest and road construction schedule is identified only by the harvest schedule component. A neighbor of a schedule is obtained by changing the harvest time of one block. Once a move is made that changes the harvesting time of a block  $b$ , the harvest time of  $b$  cannot be changed during the tabu tenure, which consists of a constant number of iterations specified by a parameter to the whole heuristic.

In solving a forest harvest and scheduling problem without considering road construction, Brumelle, Granot, Halme and Vertinsky (1996) implemented a simple aspiration



criterion. They observed that the introduction of this aspiration criterion did not improve the solution by much. Thus in my heuristic, aspiration is not used, and yet the solutions generated appears to be very satisfactory.

In each iteration, the change in objective value for each feasible move is calculated and is then multiplied by a uniform random number between 0.8 and 1.2 to yield a score for this move. Among all the non-tabu moves that do not cause the number of adjacency violations to exceed the limit  $A$ , a move with the best score is chosen and the current solution for the next iteration is generated. This process is repeated until a certain number of iterations is completed.

A solution *dominates* another if the former is not inferior in any criterion and superior in at least one criterion. In each iteration, the newly generated schedule is compared with a collection of *undominated solutions*. If it is not dominated by any schedule in the collection, then it is introduced to the collection, and if it dominates existing solutions in the collection, the dominated ones will be removed.

During the search, the parameters are systematically changed, which include the weight on unevenness and the upper bound on the number of adjacency violations. By this means, a reasonably diverse collection of undominated solutions can be obtained. In forest resource management, a decision maker may wish to know the trade off between the net income and the number of adjacency violations. For example, he may ask, how much net income must be sacrificed in order to reduce the number of adjacency violations by 1. Although the collection of undominated solutions does not provide a precise answer to such a question, it does provide a useful estimate.

In calculating the change of unevenness, a surrogate measure is used to speed up computation. This surrogate measure focuses on periods with the largest and the smallest harvest volumes. Intuitively, if harvesting of a block is moved away from the period with the largest volume, then the unevenness should probably decrease by the volume of the

block. See Appendix A for a detailed description of this surrogate measure.

### 4.3 Scheduling Problems in the Tangier Watershed

The Tangier watershed is located just north of the Trans-Canada highway between Mount Revelstoke and Glacier National Park. It is a part of the British Columbia Revelstoke Timber Supply Area. The watershed has a total area of about 28,000 hectares, of which slightly over 10,000 hectares are covered by forests. Half of this forested area is suitable for commercial forestry. While the primary land use has been commercial logging, potential conflicts are apparent between logging and other land uses, such as wildlife habitat, recreation and old growth or biodiversity preservation (Thompson et al. 1993). A variety of species are growing in the Tangier watershed forest, including fir, cedar, hemlock, balsam and spruce. A significant number of the blocks are old growth. Some blocks consist of juvenile trees which will be ready for harvesting only in the last periods of the planning horizon.

Based on a detailed set of harvest guidelines, Blake and Price (1993) identified 491 blocks suitable for harvesting. They specified the method to be used to harvest each block, but did not provide any harvest schedules. Among the 491 blocks suitable for harvesting, 272 blocks lie in a proposed provincial park<sup>2</sup> and cannot be used for harvesting if the park eventuates. Thus two areas are considered, one contains all the 491 blocks and the other contains the 219 blocks outside the proposed provincial park. For each area, a planning horizon of either 6 periods or 12 periods is considered. Thus a total of four instances need to be solved.

The legislation in British Columbia requires that the trees reach an average height of 3 meters before any adjacent blocks can be harvested, which corresponds to roughly 20

---

<sup>2</sup>The Serenity Peaks Wilderness Area.

years of seedling growth. Thus a green-up time of 2 periods is used.

The volume  $v(b,p)$  of timber obtained from harvesting block  $b$  at time period  $p$  is calculated using the British Columbia Ministry of Forests growth model TIPSy v.2 (Mitchell *et al.*1992). The volume of timber available in a block is a function of species, site quality, age of trees and size of the block. The harvesting cost of a block depends on the choice of harvesting systems<sup>3</sup> and volume of timber. Based on the assignment of harvesting system by Blake and Price (1993), the contribution margin for each block and each period is calculated, and discounted using a factor of 0.676 per 10 year period, which is equivalent to a 4.0% annual interest rate. A potential road network is predetermined and the road cost for each leg in the first period is calculated based on the length of the leg and the harvest system to be used where the leg traverses. The costs of a leg in later periods are obtained by discounting its cost in the first period.

Table 4.7 presents the total volumes of timber available in the two areas in the sixth period or in the twelfth period. For example, the total volume of timber for the 219 blocks in the sixth period is 606,082 cubic meters. Since in the Tangier watershed, the timber volume does not decrease in time during the planning horizon, these volumes reflect the highest volume of timber that can ever exist in the planning horizons, and are upper-bounds of the total harvest volumes generated by the harvest schedules.

	219 blocks problem		491 blocks problem	
	6 <sup>th</sup> period	12 <sup>th</sup> period	6 <sup>th</sup> period	12 <sup>th</sup> period
Total Vol. ( $m^3$ )	606,082	668,995	1,881,862	1,984,539

Table 4.7: Total volumes at the ends of the planning horizons

For the small area, the total contribution margin is 9.825 million dollars if all the

---

<sup>3</sup>Harvesting system refers to the method of harvesting. Possible choices include crawler tractor, low pressure ground skidding, short distance cable, mid-distance cable, standing skyline and helicopter.

profitable blocks (those with positive contribution margins) are cut in the first period, 10.766 million dollars (or 1.515 million dollars in present value) if harvested in the sixth period and 13.062 million dollars (or 0.175 million dollars in present value) if harvested in the twelfth period. For the large area, the total contribution margin is 35.847 million dollars if all the profitable blocks are cut in the first period, 37.694 million dollars (or 5.304 million dollars in present value) if harvested in the sixth period and 40.527 million dollars (or 0.542 million dollars in present value) if harvested in the twelfth period.

The total road cost for the small area in the first period is 2.96 million dollars and 8.99 million dollars for the large area.

To solve the scheduling problems, I implemented the tabu search heuristic in C++ on a HP9000/735 workstation. Tables 4.8 to 4.11 present the undominated solutions for the four problems. For each schedule, the number of adjacency violations, the net income, the total harvest volume, the unevenness, and the road cost are reported. The time lag from the moment the heuristic is started till the moment the solution is generated is also attached.

As discussed in Chapter 3, if evenness of flow and adjacency constraints are ignored, the scheduling problem can be solved as a minimum cut problem, which yields an upper-bound for the unrelaxed problem. I generated directed networks corresponding to these relaxed problems and used the preflow algorithm (Goldberg and Tarjan, 1988) to obtain the minimum cuts. The net income for the two areas and two planning horizons are shown in Table 4.12.

Unfortunately, this upper bound is not tight for the scheduling problems in the Tangier watershed. This is mainly due to the fact that the trees in the majority of the blocks have past the rapid growing stage and therefore, the present value of contribution margins decrease rapidly in time. Thus there is a strong incentive to harvest the blocks as early as possible. For example, when the even flow objective and adjacency constraints are

Number of Adj. Viola- tions	Net Income (in \$10 <sup>6</sup> )	Total Har- vest Volume (m <sup>3</sup> )	Unevenness (m <sup>3</sup> )	Road Cost (\$10 <sup>6</sup> )	Time Gen- erated (sec.)
0	2.3561	453249	921	1.5977	890.4
0	2.6454	428247	2465	1.4937	889.7
1	2.5752	442859	1098	1.6393	814.1
1	2.6174	436818	1482	1.5404	824.3
1	2.6712	436955	3795	1.5257	824.5
2	2.3453	466650	834	1.6703	350.2
3	2.6319	453276	2344	1.5701	276.0
3	2.7303	448029	4332	1.6153	718.0
4	2.3003	460406	619	1.7260	686.7
4	2.6363	449803	1648	1.5822	699.2
4	2.7354	466937	3962	1.6034	717.8
5	2.5611	451942	829	1.5065	216.2
5	2.8213	495498	2922	1.6145	212.7
6	2.7770	469430	2595	1.6177	591.3
7	2.6788	440779	1058	1.5849	581.8
7	2.7075	440779	1170	1.5888	581.7
8	2.5651	457270	874	1.6251	474.8
8	2.7111	464886	1063	1.6047	518.8

Table 4.8: Solutions for 219 blocks with 6 periods

Number of Adj. Viola- tions	Net Income (in \$10 <sup>6</sup> )	Total Har- vest Volume (m <sup>3</sup> )	Unevenness (m <sup>3</sup> )	Road Cost (\$10 <sup>6</sup> )	Time Gen- erated (sec.)
0	2.1105	639064	3968	1.4492	1865.2
0	2.1186	635229	4294	1.3736	922.9
1	2.1536	621669	2534	1.4423	758.6
1	2.1592	629127	3952	1.4462	758.8
2	1.9486	628039	1992	1.3962	1648.9
3	2.1925	640165	1915	1.5140	1593.0
3	2.1957	640165	2633	1.4940	1592.8
3	2.2446	641571	3858	1.4184	1591.2
4	2.2518	622732	3706	1.3815	1508.8
4	2.3202	625872	4992	1.3735	1328.3
5	2.0228	631268	1890	1.2503	401.8
5	2.3540	616079	4500	1.3423	1325.9
6	2.0894	624749	1621	1.3360	1315.5
8	2.3443	645016	2775	1.3068	1085.1
8	2.3731	638872	3638	1.3397	1086.4
8	2.4165	638245	4957	1.3417	1086.3
10	1.4400	669478	1527	1.3724	136.1

Table 4.9: Solutions for 219 blocks with 12 periods

Number of Adj. Viola- tions	Net Income (in \$10 <sup>6</sup> )	Total Har- vest Volume (m <sup>3</sup> )	Unevenness (m <sup>3</sup> )	Road Cost (\$10 <sup>6</sup> )	Time Gen- erated (sec.)
0	10.1929	1475590	924	5.4151	1676.0
0	10.5249	1484150	2862	5.5118	1664.6
0	10.5364	1469030	4447	5.3871	1715.9
1	10.1493	1474340	561	5.5285	3473.4
1	10.5496	1482950	1413	5.4129	1547.4
1	10.6139	1522130	3923	5.4450	3322.9
1	10.6695	1523240	4249	5.5281	3341.7
2	10.3448	1476950	679	5.3483	1403.2
2	10.6270	1511760	779	5.4244	1256.2
3	10.7303	1527410	2524	5.3856	2862.6
4	10.6542	1523180	1462	5.4869	2827.3
4	10.7415	1498890	2478	5.3976	2783.4
4	10.7706	1498890	4262	5.3872	2783.2
5	10.4276	1530960	589	5.4416	761.5
5	10.8477	1542070	2212	5.4634	774.2
5	10.8503	1542070	4850	5.4634	774.4
6	10.5706	1567160	520	5.4370	2465.8
6	10.8188	1551840	1923	5.4976	2501.4
6	10.8938	1567630	3511	5.4724	2491.3
7	10.6199	1490620	751	5.4196	2218.3
8	10.8353	1520240	2041	5.5157	2131.5
8	10.9743	1520190	4762	5.4182	2007.1

Table 4.10: Solutions for 491 blocks with 6 periods

Number of Adj. Viola- tions	Net Income (in $\$10^6$ )	Total Har- vest Volume ( $m^3$ )	Unevenness ( $m^3$ )	Road Cost ( $\$10^6$ )	Time Gen- erated (sec.)
0	7.5030	1956420	1821	4.5722	7934.3
0	7.6425	1946670	2117	4.5516	7982.8
0	7.9859	1969320	2867	4.5325	7822.4
0	7.9893	1971520	4987	4.5271	7826.4
1	7.7691	1956460	2650	4.5533	7711.8
2	7.0591	1898470	936	4.4286	2975.9
2	7.6720	1935570	2482	4.4824	2948.9
2	7.6988	1916860	2630	4.6320	7288.5
3	7.6523	1942940	1288	4.5841	6577.4
6	7.8413	1936510	2337	4.5834	5335.1
6	7.8650	1936510	2478	4.5824	5334.7
6	7.8886	1947060	2613	4.5959	5305.8
8	7.4807	1927840	916	4.5381	4722.9
8	7.7580	1939320	2239	4.5376	4805.1

Table 4.11: Solutions for 491 blocks with 12 periods

	219 blocks		491 blocks	
	6 periods	12 periods	6 periods	12 periods
Net Income (in $\$10^6$ )	8.707	8.729	31.467	31.489

Table 4.12: Optimal net income for the relaxed problems



ignored, one of the feasible schedules is to harvest all the blocks and build all the roads in the first period. For the area with 219 blocks, this schedule yields a net income of 6.87 million dollars, which is much larger than the net income values for the unrelaxed problem. Since the above schedule is a feasible solution to the relaxed problem, the optimal net income value for the relaxed problem should be even larger than 6.87 million dollars.

Since in some blocks the trees are still young in the early periods, the contribution margins for these blocks are negative throughout the first 6 periods. Thus in the schedules for the six-period problems, these blocks are not harvested at all in the planning horizon, and the harvest volumes are significantly less than the highest volume of timber in the planning horizon. Trees in most of these blocks will mature in later periods, making the contribution margins positive later on. Thus the harvest volumes for the twelve period problems are very close to the highest volume of timber that can ever exist in the area. For example, the highest volume of timber for the 491 blocks in the 12 periods planning horizon is 1.984 million cubic meters, and the total harvest volume for the first schedule in Table 4.11 is 1.956 million cubic meters, 1.6% less than the largest total volume.

Brumelle, Granot, Halme and Vertinsky (1996) used a tabu search heuristic to solve the forest harvest scheduling problem for the same set of data but without considering road scheduling. Rather than maximizing the net income, their objective is to maximize total harvest volume. Therefore, in their solutions, the total harvest volumes are higher, especially for the 6 period problems. However, the net income associated with their schedules are much lower. For example, for the 219 blocks and 6 period problem, the total harvest volumes in their solutions range from 568,160 cubic meters to 579,752 cubic meters with an average of 575,733 cubic meters. The total volumes in my study for the same problem range from 428,247 to 495,498 cubic meters with an average of 453,545 cubic meters. The net incomes in their solutions range from 1.48 million dollars to 1.88 million dollars, with an average of 1.62 million dollars. The net incomes in my solutions

range from 2.30 to 2.82 million dollars with an average of 2.61 million dollars. Their results are inferior in terms of net income because (1) they did not consider the road cost and (2) their focus is on the harvest volume instead of net income, and in some blocks the trees are so young that the contribution margins for these blocks are negative even though the volumes are positive.

It can be seen that the net incomes does not change significantly in the number of adjacency violations. For example, in Table 4.11, the first solution has no adjacency violation and the net income is 5.55 million dollars. The last solution has 9 adjacency violations and the net income is 5.98 million dollars, which is only 7.7% higher than that of the first solution. This suggests that with proper scheduling, steady and profitable production can be sustained while at the same time, the damage to the environment can be significantly reduced.

## Bibliography

- [1] T. Arai, S. Ueno, and Y. Kajitani. Generalization of a theorem on the parametric maximum flow problem. *Discrete Applied Mathematics*, 41:69–74, September 1993.
- [2] M. L. Balinski. On a selection problem. *Management Science*, 17(13):230–231, 1970.
- [3] B. B. Bare and G. A. Mendoza. Timber harvest scheduling in a fuzzy decision environment. *Canadian Journal of Forest Research*, 22:423–428, 1992.
- [4] J. Blake and L. Price. Revelstoke timber supply area gis-based multi-resource analysis. *The 7<sup>th</sup> annual symposium on geographic information system in forestry, environment and natural resources management, Vancouver, B.C.*, 1:227–240, 1993.
- [5] Shelby Brumelle and Daniel Granot. The repair kit problem revisited. *Operations Research*, 41:994–1006, 1993.
- [6] Shelby Brumelle, Daniel Granot, Mirja Halme, and Ilan Vertinsky. A tabu search algorithm for finding a good forest harvest schedule satisfying green-up constraints. Working Paper, Faculty of Commerce, The University of British Columbia, 1996.
- [7] Shelby Brumelle, Daniel Granot, and Li Liu. An extended selection problem. Working Paper 96-MSC-003, Faculty of Commerce, The University of British Columbia, 1996.
- [8] Shelby Brumelle, Daniel Granot, and Li Liu. Totally ordered optimal solutions and parametric maximum flows. Working Paper 96-MSC-002, Faculty of Commerce, The University of British Columbia, 1996.
- [9] P. J. Carstensen. Complexity of some parametric integer and network programming problems. *Mathematical Programming*, 26:64–75, 1983.
- [10] S.E. Clements, P. L. Dallein, and M. S. Jamnick. An operational spatially constrained harvest scheduling model. *Canadian journal of Forest Research*, 20:1438–1447, 1990.
- [11] Y. Crama. Combinatorial optimization models for production scheduling in automated manufacturing systems. In *Semi-plenary papers of EURO XIV conference*, pages 237–259, Jerusalem, 1995.

- [12] Y. Crama and J. Mazzola. Valid inequalities and facets for a hypergraph model of the nonlinear knapsack and FMS part selection problems. *Annals of Operations Research*, 1995.
- [13] Kadri Dagdelen. *Optimum multi period open pit mine production scheduling*. PhD thesis, Colorado School of Mines, 1979.
- [14] Kadri Dagdelen. Cutoff grade optimization. In *23<sup>rd</sup> Application of Computers and Operations Research in the Mineral Industry*, page 157. Richardson, TX, USA Soc. of Petroleum Engineers of AIME, 1992.
- [15] D. K. Daust and J. D. Nelson. Spatial reduction factors for strata-based harvest schedules. *Forest Science*, 39:152–165, 1993.
- [16] Renato de Matta, Vernon Ning Hsu, and Timothy J. Lowe. The capacitated selection problem. Technical report, Department of Management Science, College of Business Administration, University of Iowa, Iowa City, IA 52242, 1995.
- [17] E. V. Denardo, G. Huberman, and U. G. Rothblum. Optimal locations on a line are interleaved. *Operations Research*, 30(4):745–759, 1982.
- [18] Mark J. Eisner and Dennis G. Severance. Mathematical techniques for efficient record segmentation in large shared databases. *J. Association for Computing Machinery*, 23(4):619–635, 1976.
- [19] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, N.J., 1962.
- [20] S. Fortune, J. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10:111–121, 1980.
- [21] G. Gallo, H. Grigoriadis, and R. E. Tarjan. A fast parametric network flow algorithm. *SIAM Journal on Computing*, 18(1):30–35, 1989.
- [22] F. Glover. Tabu search: Part i. *ORSA Journal on Computing*, 1(3):190–206, 1989.
- [23] F. Glover and M. Laguna. Tabu search. In Colin R. Reeves, editor, *Modern Heuristic Techniques for Combinatorial Problems*, pages 70–150. 1993.
- [24] Andrew V. Goldberg and Robert E. Tarjan. A new approach to the maximum flow problem. *J. Association for Computing Machinery*, 35:4:921–940, October 1988.
- [25] F. Granot and Jr A. F. Veinott. Substitutes, complements and ripples in network flows. *Mathematics of Operations Research*, 10(3):471–497, 1985.

- [26] T. E. Gross and D. O. Dykstra. Harvest scheduling with non-adjacency constraints. *Proceedings of the Society of American Foresters National Convention*, pages 310–315, 1988.
- [27] Dan Gusfield and Éva Tardos. A fast parametric minimum-cut algorithm. *Algorithmica*, 11(3):278–290, 1994.
- [28] Dan Gusfield and Charles Martel. A fast algorithm for the generalized parametric minimum cut problem and applications. *Algorithmica*, 7:499–519, 1992.
- [29] H. M. Hoganson and D. W. Rose. A simulation approach for optimal timber management scheduling. *Forest Science*, 30:220–238, 1984.
- [30] A. F. Howard. A critical look at multiple criteria decision making techniques with reference to forestry applications. *Canadian Journal of Forest Research*, 21:1649–1659, 1991.
- [31] A. F. Howard and J. D. Nelson. Area-based harvest scheduling and allocation of forest land using methods for multiple-criteria decision making. *Canadian Journal of Forest Research*, 23:151–158, 1993.
- [32] M. S. Jamnick and K. R. Walters. Spatial and temporal allocation of stratum-based harvest schedules. *Canadian Journal of Forest Research*, 23:402–413, 1993.
- [33] K. N. Johnson and H. L. Scheurman. Techniques for prescribing optimal timber harvest and investment under different objectives - discussion and synthesis. *Forest Science Monograph*, 18, 1977.
- [34] M. W. Kirby, P. Wong, W. A. Hager, and M. E. Huddleston. A guide to the integrated resources planning model. Technical report, USDA Forest Service, Berkeley, CA, 1980.
- [35] C. Lockwood and T. Moore. Harvest scheduling with spatial constraints: A simulated annealing approach. *Canadian Journal of Forest Research*, 23:468–478, 1993.
- [36] William S. Lovejoy. Some monotonicity results for partially observed markov decision processes. *Operations Research*, 35(5):736–743, 1987.
- [37] D. Ludwig. Forest management strategies that account for short-term and long-term consequences. *Canadian Journal of Forest Research*, 23:563–572, 1993.
- [38] John W. Mamer and Stephen A. Smith. Optimizing field repair kits based on job completion rate. *Management Science*, 28(11):1328–1333, 1982.

- [39] S.T. McCormick. Fast algorithms for parametric scheduling come from extensions to parametric maximum flow. Working Paper 95-MSC-005, Division of Management Science, Faculty of Commerce, University of British Columbia, 1995.
- [40] B. J. Meneghin, M. W. Kirby, and J. G. Jones. An algorithm for writing adjacency constraints efficiently in linear programming models. Technical report, USDA Forest Service General Technical Report RM-161, Fort Collins, CO, 1988.
- [41] Paul Milgrom and John Roberts. The economics of modern manufacturing technology, strategy, and organization. *American Economic Review*, 80(3):511–528, June 1990.
- [42] Paul Milgrom and Chris Shannon. Monotone comparative statics. *Econometrica*, 62(1):157–180, January 1994.
- [43] K. J. Mitchell, S. E. Grout, R. N. MacDonald, and D. A. Watmough. User's guide for tipsy: A table interpolation program for stand yields. Technical report, B. C. Ministry of Forests, Victoria., 1992.
- [44] J. D. Nelson and S. T. Finn. The influence of cut-block size and adjacency rules on harvest levels and road networks. *Canadian Journal of Forest Research*, 21:595–600, 1991.
- [45] J. D. Nelson and A. F. Howard. A three-stage heuristic procedure for allocating spatially and temporally feasible cutting rights on public lands. *Canadian Journal of Forest Research*, 21:762–768, 1991.
- [46] B.C. Ministry of Forests. Revelstoke tsa timber supply analysis. Technical report, Integrated Resource Branch, B.C. Ministry of Forests, Victoria, 1993.
- [47] O. Ore. *Theory of Graphs*, volume 38. American Mathematical Society Colloquium Publications, Providence, Rhode Island, 1962.
- [48] A. J. Q'Hara, B. H. Faaland, and B. B. Bare. Spatially constrained timber harvest scheduling. *Canadian Journal of Forest Research*, 19:715–724, 1989.
- [49] J. M. W. Rhys. A selection problem of shared fixed costs and network flows. *Management Science*, 17(3):200–207, November 1970.
- [50] J. Sessions and J. B. Sessions. Scheduling and network analysis program (snap ii), user guide version 1.0. Technical report, Oregon State University, Corvallis, OR, 1990.

- [51] Stephen A. Smith, John C. Chambers, and Eli Shlifer. Optimal inventories based on job completion rate for repairs requiring multiple items. *Management Science*, 26(8):849–852, 1980.
- [52] H. S. Stone, Critical load factors in two-processor distributed systems, *IEEE Trans. Software Engineering*, 4:254–258, 1978.
- [53] W. A. Thompson, G. C. van Kooten, I. Vertinsky, S. Brown, and H. Schreier. A preliminary economic model for evaluation of forest management in a geo-referenced framework. *The 7<sup>th</sup> Annual Symposium on Geographic Information Systems in Forestry Environment and Natural Resources Management, Vancouver, B.C.*, 1:209–220, 1993.
- [54] D. M. Topkis and Jr. A. F. Veinott. Monotone solutions of extremal problems on lattices. In *Abstracts of talks, VIII International Symposium on Mathematical Programming*, page 131, Stanford University, 1973.
- [55] Donald M. Topkis. *Ordered optimal solutions*. PhD thesis, Department of Operations Research, Stanford University, 1968.
- [56] Donald M. Topkis. Minimizing a submodular function on a lattice. *Operations Research*, 26(2):305–321, 1978.
- [57] Donald M. Topkis. Comparative statics of the firm. *Journal of Economic Theory*, 67:370–401, 1995.
- [58] J. M. Torres and J. D. Brodie. Adjacency constraints in harvest scheduling: An aggregation heuristic. *Canadian Journal of Forest Research*, 20:978–986, 1990.
- [59] A. F. Jr Veinott. Lattice programming. Forthcoming.
- [60] A. F. Jr. Veinott. Class notes, course on inventory theory. Operations Research Department, Stanford University, 1974.
- [61] U. Zimmerman. *Linear and combinatorial optimization in ordered algebraic structures*. North Holland, 1981.

## Appendix A

### Pseudo-Code for the Tabu Heuristic

#### THE TABU SEARCH HEURISTIC

Construct an initial schedule  $h$ .

FOR each record in a parameter file

{

    Read a set of parameters:

        adj\_violation\_bound,

        uneven\_weight,

        road\_cost\_weight,

        iterations;

    FOR iterations TIMES REPEAT:

    {

        FOR ALL pairs  $(b, p) \in B \times P$  SUCH THAT

        ( (block  $b$  is not tabu)

          AND  $(h(b) \neq p)$

          AND (changing  $h(b)$  to  $p$  will not cause the

          number of violation to be more than adj\_viol\_bound) )

    {



```

/*
Compute the change in objective value
if harvesting time of  $b$  is changed to  $p$ :
*/
 $\Delta(b,p)=\text{Delta}(h, b, p)$ ;

/*
Perturb the change:
*/
 $\Delta(b,p) = \Delta(b,p) * \text{Random}(1-\epsilon, 1+\epsilon, \text{Uniform})$ ;

}

Among the above  $(b, p)$  pairs, find one
with the best  $\Delta(b,p)$  and change
the harvesting time of  $b$  to  $p$  by
calling  $\text{Move}(h, b, p)$ .

}

}

/* End of the tabu search heuristic. */

```

**FUNCTIONS USED IN THE TABU SEARCH HEURISTIC:**

/\*

Compute the change in objective value

if harvesting time of  $b$  is changed to  $p$ :

\*/

FUNCTION Delta( $h, b, p$ )

{

RETURN\_VALUE =

    Delta\_contribution( $h, b, p$ ) -    Delta\_road\_cost( $h, b, p$ ) -    uneven\_weight\*Delta\_uneven( $h, b, p$ ) -

}

/\*

Change in contribution margin is simply the difference

between the contribution margins in two time periods:

\*/

FUNCTION Delta\_volume( $h, b, p$ )

{

RETURN\_VALUE =

 $m(b, p) - m(b, h(b));$ 

}

```
/*
A function to calculate changes in unevenness precisely:
*/
FUNCTION Delta_uneven( $h, b, p$ )
{
    /*
    First delete the volume  $v(b, h(p))$ 
    */
    IF ( $h(b) == \text{period\_with\_max\_volume}$ )
    {
        max_volume_after_delete =
        MAX((max_volume_of_all_periods -  $v(b, h(b))$ ),
        second_largest_volume_of_all_periods);

    } ELSE
        max_volume_after_delete = max_volume_of_all_periods;

    temp = volume_of_period[ $h(b)$ ] -  $v(b, h(b))$ ;
    IF (min_volume_of_all_periods <= temp)
    {
        min_volume_after_delete = min_volume_of_all_periods;
        min_period_after_delete = period_with_min_volume;
        second_smallest_volume_after_delete =
```

```
        MIN(temp, second_smallest_volume_of_all_periods);

    } ELSE
    {

        min_volume_after_delete=temp;
        min_period_after_delete=h(b);
        IF (h(b) != period_with_smallest_volume)
            second_smallest_volume_after_delete=
            min_volume_of_all_periods;
        ELSE
            second_smallest_volume_after_delete=
            second_smallest_volume_of_all_periods;

    }

    /*
    Then add volume  $v(b,p)$ 
    */

    new_max_volume=
    MAX(max_volume_after_delete,
    (volume_in_period[p] +  $v(b,p)$ ))

    IF (p==min_period_after_delete)
        new_min_volume=
```

```
        MIN((min_volume_after_delete +  $v(b,p)$ ),
            second_smallest_volume_after_delete);
    ELSE
        new_min_volume=min_volume_after_delete;
        new_uneven=new_max_volume-new_min_volume;

}

/*
A function to calculate a surrogate of changes in unevenness:
*/
FUNCTION Delta_uneven( $h,b,p$ )
{
    d=0;
    IF ( $h(b)$ ==period_with_max_volume)
        d=d -  $v(b,h(b))$ ;
    IF ( $h(b)$ ==period_with_min_volume)
        d=d +  $v(b,h(b))$ ;

    IF ( $p$ ==period_with_max_volume)
        d = d +  $v(b,p)$ ;
    IF ( $p$ ==period_with_min_volume)
        d = d -  $v(b,p)$ ;

    RETURN_VALUE=d;
```

```
}
```

```
/*
```

Change in road cost. In the function, a leg is identified by the block it goes into. For example, if a leg goes from the saw mill to a block 3, then the leg is referred to as 3. If there is another leg from block 3 to block 5 (note that block 3 is closer to the saw mill than 5), then this leg will be referred to as 5. Sibling legs are those going into sibling nodes, and offspring legs are those going into offspring nodes of a node.

```
*/
```

```
FUNCTION Delta_road_cost(h, b, p)
```

```
{
```

```
    old_construction_time=construction_time[b];
```

```
    IF (p< old_construction_time)
```

```
        new_construction_time=p;
```

```
    ELSE /* construction of this leg can be delayed */
```

```
    {
```

```
        find the minimum construction time t1 for
```

```
        all offspring legs;
```

```
        new_construction_time=t1;
```

```
    }
```

```
FIND the earliest period t2 among the following:
    new_construction_time,
    periods in which sibling legs of b are constructed,
    harvesting time of Parent(b),

RETURN_VALUE=
    Leg_cost(b,t)-Leg_cost(b,t0) +
    Delta_road_cost(h, Parent(b), t2);
}
```