POLYHEDRAL STUDIES ON
SCHEDULING AND ROUTING PROBLEMS

By

Yaoguang Wang

B. Sc. Shanghai Fudan University
M. Sc. Shanghai Jiao Tong University

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
Doctor of Philosophy

in

THE FACULTY OF GRADUATE STUDIES
COMMERCE AND BUSINESS ADMINISTRATION

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA
June 1991

© Yaoguang Wang, 1991
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.

Department of Commerce and Business Administration

The University of British Columbia
Vancouver, Canada

Date June 17, 1991
Abstract

During the last decade, there have been major advances in solving a class of large-scale real world combinatorial optimization problems. Such problems are formulated as Travelling Salesman Problems (TSP), some involving up to thousands of cities. These achievements, mainly due to the use of so called polyhedral techniques, have established the importance of the polyhedral study for various combinatorial optimization problems.

This thesis studies polyhedral structures of two well known combinatorial problems: (i) precedence constrained single machine scheduling and (ii) TSP, both Symmetric TSP (STSP) and Asymmetric TSP (ATSP). These problems are of both theoretical interest and practical importance. Better knowledge of the polyhedral descriptions of these problems may facilitate the polyhedral study of more complex scheduling and routing problems.

For the scheduling problem, we present two classes of facetial inequalities, which suffice to describe the linear system of the scheduling problem when the precedence constraints are series-parallel. We also propose a cutting plane procedure based on these facet cuts. The computational results show the procedure yields feasible schedules with relative deviations from the optimum less than 0.25% on the average and less than 1% in the empirical worst case.

For TSPs, we explore a Hamiltonian path approach to the polyhedral study. We propose various facet extension techniques for deriving large classes of facets from known facets. In the STSP case, we propose new clique lifting results. In the ATSP case, we develop a Tree Composition method, which generates all non-spanning clique tree facetial inequalities.
Table of Contents

Abstract ii

List of Tables v

List of Figures vi

Acknowledgement vii

1 Introduction 1

1.1 Preliminaries ........................................... 2
1.2 Precedence constrained single machine scheduling .......... 4
1.3 Travelling Salesman Problems ............................ 7

2 Single Machine Scheduling: Polyhedral Structure 11

2.1 Introduction ............................................. 11
2.2 Definitions and preliminaries ............................ 12
2.3 Three classes of valid inequalities ......................... 17
2.4 Structure of the precedence constrained scheduling polyhedra 21
2.5 Facet Lifting ............................................. 32
2.6 Proofs of the four lemmata ............................. 36

3 Single Machine Scheduling: Polyhedral Computations 41

3.1 Introduction ............................................. 41
3.2 A cutting plane procedure ................................ 42
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.1</td>
<td>Initial Formulation</td>
<td>43</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Cutting Planes and Separation Algorithms</td>
<td>44</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Heuristics for the Upper Bounds</td>
<td>49</td>
</tr>
<tr>
<td>3.2.4</td>
<td>The Cutting Plane Procedure</td>
<td>50</td>
</tr>
<tr>
<td>3.3</td>
<td>Computational Experience</td>
<td>51</td>
</tr>
<tr>
<td>3.4</td>
<td>Computational Results for 280 problems</td>
<td>60</td>
</tr>
<tr>
<td>4</td>
<td>Hamiltonian Path and Symmetric Travelling Salesman Problem</td>
<td>67</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>67</td>
</tr>
<tr>
<td>4.2</td>
<td>Definition and Notation</td>
<td>70</td>
</tr>
<tr>
<td>4.3</td>
<td>Polyhedral Equivalence: HPP v.s. STSP</td>
<td>72</td>
</tr>
<tr>
<td>4.4</td>
<td>Clique-Lifting</td>
<td>80</td>
</tr>
<tr>
<td>4.5</td>
<td>Simple Conditions for Node-Cloning</td>
<td>89</td>
</tr>
<tr>
<td>5</td>
<td>Symmetric Inequalities for Asymmetric Travelling Salesman Polytopes</td>
<td>97</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>97</td>
</tr>
<tr>
<td>5.2</td>
<td>Definitions and Notation</td>
<td>101</td>
</tr>
<tr>
<td>5.3</td>
<td>The ATS Polytope and its Projection</td>
<td>104</td>
</tr>
<tr>
<td>5.4</td>
<td>Symmetric Facet-defining Inequalities</td>
<td>108</td>
</tr>
<tr>
<td>5.5</td>
<td>Composing Symmetric Inequalities</td>
<td>115</td>
</tr>
<tr>
<td>5.6</td>
<td>Proofs of the Main Results in Section 5.4</td>
<td>121</td>
</tr>
<tr>
<td>5.7</td>
<td>Proofs of the Main Results in Section 5.5</td>
<td>129</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>138</td>
</tr>
<tr>
<td></td>
<td>Appendices</td>
<td>145</td>
</tr>
<tr>
<td>A</td>
<td>Generating the Sample of 280 scheduling problems</td>
<td>145</td>
</tr>
</tbody>
</table>
List of Tables

3.1 Data for the 10-job example ............................................. 52
3.2 Cutting planes and the corresponding solutions .......................... 52
3.3 Data for the 30-job problem ............................................. 53
3.4 Solutions from the cutting plane procedure .............................. 54
3.5 The optimal schedule and the relevant solutions ........................ 55
3.6 A summary of the results for 280 problems .............................. 57
3.7 Computational results with respect to $P$ values ......................... 58
3.8 Computational results for $n = 30, 40$ .................................. 60
3.9 Computational results for $n = 50, 60$ .................................. 61
3.10 Computational results for $n = 70, 80$ .................................. 62
3.11 Computational results for $n = 90, 100$ ................................ 63
3.12 Computational results for $n = 110, 120$ ................................ 64
3.13 Computational results for $n = 130, 140$ ................................ 65
3.14 Computational results for $n = 150, 160$ ................................ 66

A.1 Seeds for the instances with $n = 30, 40, 50, 60, 70, 80, 90, 100$ .... 152
A.2 Seeds for the instances with $n = 110, 120, 130, 140, 150, 160$ ....... 153
## List of Figures

4.1 Equivalent envelope inequalities .................................. 75  
4.2 Facet-defining bipartition inequalities .......................... 83  
4.3 Facet-defining binested inequalities ............................. 94  

5.1 An ATSP Regular Star Inequality ................................. 109  
5.2 Symmetric facets and their projections .......................... 112  
5.3 A 9-node ATSP Chain Inequality ................................. 113  
5.4 A Tree Composition .................................................. 118  
5.5 A Path Tree obtained from the Tree Compositions .............. 120  
5.6 c-tight paths ......................................................... 122  
5.7 c-tight paths for the Chain Inequality .......................... 128
Acknowledgement

First, I wish to express my sincere gratitude to Professor Maurice Queyranne for introducing me to the area of scheduling and for his effort in effectively guiding me at every stage of my thesis research. I have benefitted a great deal from his assistance. I gratefully acknowledge the partial financial support from his NSERC Operating Grant.

Also, I wish to thank Professor D. Naddef, who gave an inspiring seminar on the travelling salesman problems during his visit in UBC; Professors L.A. Wolsey, C.N. Potts, and Van de Velde, who generously provided their test examples and computer programs for a computational study in my thesis.

I am very grateful to Professors E. Balas, F. Granot, D. Granot, R. Israel, D.G. Kirkpatrick and S.T. McCormick for their valuable efforts in examining and correcting the manuscript.

I greatly appreciate the financial support from Canada International Development Agency for my doctoral studies.

Finally, special thanks are due to my wife Liqun, my parents Meixian and Zufu, for their every effort in taking a good care of my two children and me during difficult times.
Chapter 1

Introduction

Over the past decade, we have witnessed major advances in solving a class of large-scale real world combinatorial optimization problems. Such problems are formulated as Travelling Salesman Problems (TSP), some involving up to thousands of cities, see Padberg and Rinaldi [49], and Kolata [31]. These achievements are largely due to the use of so called polyhedral techniques, which employ polyhedral descriptions of the underlying problems. The successes have motivated in recent years extensive studies on polyhedral structures and computations for various combinatorial optimization problems.

The present thesis studies polyhedral structures of two well known combinatorial problems, namely, precedence constrained single machine scheduling in Chapter 2, and TSP, both Symmetric TSP (STSP) and Asymmetric TSP (ATSP) in Chapters 4 and 5, respectively. These problems are of both practical importance and theoretical interest, because (a) they are very simple, yet fundamental, versions of machine scheduling and vehicle routing problems, which often arise in production and distribution management; (b) they are strongly NP-hard, see Garey and Johnson [22]. Better knowledge of the polyhedral descriptions of these problems may thence facilitate the polyhedral study of more complex scheduling and routing problems.

To test the facetial results derived in Chapter 2, we perform polyhedral computations for the precedence constrained single machine scheduling problem. Chapter 3 introduces a cutting plane procedure which uses effective cuts (i.e., facet-defining inequalities) derived in Chapter 2. Computational results show that the solution procedure
consistently yields feasible solutions with a guaranteed high accuracy over a broad range of instances. The procedure generates feasible schedules with percent deviation from the optimum less than 0.25% on the average and less than 1% in the empirical worst case.

The rest of the chapter is organized as follow. Section 1.1 briefly reviews terminology and notation used for the polyhedral methods. Sections 1.2 and Section 1.3 summarize the motivations and main results in this thesis for the scheduling problem and Travelling Salesman Problems, respectively.

1.1 Preliminaries

In this section, we briefly review the terminology and notation which are necessary for our exposition. For a more detailed treatment of the theory of combinatorial polyhedra, we refer to Grötschel and Padberg [28].

Let $R$ denote the real line and $R^n$ the $n$-dimensional real vector space. For $x^1, \cdots, x^k \in R^n$ and $\lambda_1, \cdots, \lambda_k \in R$, the vector $x \in R^n$ with $x = \lambda_1 x^1 + \cdots + \lambda_k x^k$ is called a linear combination of the vectors $x^1, \cdots, x^k$. If in addition, $\lambda_1 + \cdots + \lambda_k = 1$ (resp., all $\lambda_i$ are nonnegative and $\lambda_1 + \cdots + \lambda_k = 1$), then $x$ is called an affine (resp., convex) combination of the vectors $x^1, \cdots, x^k$.

If $\emptyset \neq S \subseteq R^n$, then the set of all convex combinations of finitely many vectors in $S$ is called the convex hull of $S$ and is denoted by $\text{conv}(S)$.

A nonempty set $S \subseteq R^n$ is called linearly (resp., affinely) independent, if for every finite set $\{x^1, x^2, \cdots, x^k\} \subseteq S$, the equations $\lambda_1 x^1 + \cdots + \lambda_k x^k = 0$ (resp., $\lambda_1 x^1 + \cdots + \lambda_k x^k = 0$ and $\lambda_1 + \cdots + \lambda_k = 0$) imply $\lambda_i = 0$, $i = 1, \cdots, k$; otherwise $S$ is called linearly (resp., affinely) dependent.

The rank (resp., affine rank) of a set $S \in R^n$ is the cardinality of the largest linearly (resp., affinely) independent subset of $S$, and the dimension of $S$, denoted by $\text{dim}(S)$, is
the affine rank of $S$ minus one. A set $S \subseteq R^n$ is called full-dimensional if $\dim(S) = n$. The notion of affine rank or independence is useful in the theory of polyhedra because it is invariant under translations of the origin.

To keep notation to a minimum, we denote by $ax$ the scalar product of any vectors $a, x \in R^n$. A set $H \subseteq R^n$ is said to be a halfspace if there exist a vector $a \in R^n$ and a scalar $a_0 \in R$ such that $H = \{x \in R^n : ax \leq a_0\}$. Then $H$ is the halfspace defined by the inequality $ax \leq a_0$. An inequality $ax \leq a_0$ is called valid with respect to a set $S \subseteq R^n$ if $S$ is contained in the halfspace defined by $ax \leq a_0$. A valid inequality $ax \leq a_0$ for $S$ is called tight, or supporting, if $ax = a_0$ for some $x \in S$. An inequality $ax \leq a_0$ for $S$ is called an implicit equation if $ax = a_0$ holds for all $x \in S$.

A polyhedron is the intersection of finitely many halfspaces. A bounded polyhedron is called a polytope. For any polyhedron $\mathcal{P} \subseteq R^n$, there exist an $(m, n)$-matrix $A$ and a vector $b \in R^m$ such that $\mathcal{P}$ can be represented in the form

$$\mathcal{P} = \{x \in R^n : Ax \leq b\}.$$ 

A subset $F$ of a polyhedron $\mathcal{P}$ is called a face of $\mathcal{P}$ if there exists a supporting inequality $ax \leq a_0$ with respect to $\mathcal{P}$ such that $F = \{x \in \mathcal{P} : ax = a_0\}$. We say that the inequality $ax \leq a_0$ defines $F$ or is face-defining for $\mathcal{P}$. A facet $F$ of $\mathcal{P}$ is any face of $\mathcal{P}$ with $\dim F = \dim \mathcal{P} - 1$. A valid inequality $ax \leq a_0$ is facet-defining, or facet-inducing, if the induced face $F = \{x \in \mathcal{P} : ax = a_0\}$ contains exactly $\dim \mathcal{P}$ affinely independent points.

A valid inequality $fx \leq f_0$ for $\mathcal{P}$ is said to dominate $cx \leq c_0$ if any $x \in \mathcal{P}$ satisfying $cx = c_0$ also satisfies $fx = f_0$. Any two valid inequalities $fx \leq f_0$ and $cx \leq c_0$ are equivalent w.r.t. $\mathcal{P}$ if they dominate each other. It is well-known (e.g., [28]) that two valid inequalities for $\mathcal{P}$ are equivalent iff one can be obtained by multiplying the other with a positive scalar and adding a linear combination of the implicit equations for $\mathcal{P}$. 
The polyhedral technique is a Linear Programming (LP) approach to combinatorial optimization problems. The feasible solution set to any combinatorial problem can be represented as a collection of points in an appropriate space $\mathbb{R}^n$. To apply the polyhedral method, we first need to study the polyhedral structure, i.e., a linear description, of the convex hull $P$ of the feasible solution set. We then devise cutting plane procedures using known valid inequalities which define faces (of high dimension, preferably facets) of $P$ as cutting planes. The procedures often yield very tight lower (resp., upper) bounds if it is a minimization (resp., maximization) problem. Such bounds can then be used in subsequent Branch and Bound or Branch and Cut procedures if an exact optimum is sought.

1.2 Precedence constrained single machine scheduling

We consider the following single machine nonpreemptive scheduling problem. Each of $n$ jobs, available for processing at time zero, is to be processed without interruption on a single machine. Associated with each job $j$ is its processing time $p_j$ and its linear deferral (delay) cost rate $w_j$ [53], both of which are independent of job sequencing. The machine can process only one job at a time. Precedence constraints may be assumed between jobs. A schedule is specified by the job completion times $C_j$. A schedule is feasible if it satisfies all precedence constraints.

Our objective is to find a feasible schedule so as to minimize the total weighted completion time, i.e., the sum of the deferral costs $w_jC_j$. In a production process, such costs $w_jC_j$ may measure the cost of work-in-process inventory for each job $j$. For the treatment of other objectives for single machine scheduling, such as minimizing makespan, weighted tardiness, the number of late jobs, etc., interested readers are referred to an excellent survey by Lawler et al [36].
Chapter 1. Introduction

When there are no precedence constraints, the problem can be solved, in $O(n \log n)$ computational steps, by sequencing jobs in nonincreasing order of ratios $w_j/p_j$ [54]. Special precedence relations, such as rooted trees, parallel chains, etc., were considered by Sidney [53]. Lawler [33] considered series-parallel precedence relation which subsumes all previous special cases, and derived an elegant $O(n \log n)$ optimization procedure.

With general precedence constraints, the problem becomes strongly NP-hard even if the weights or processing times are assumed to be zero or one [33,37]. This indicates that the existence of a polynomially bounded optimization algorithm is highly unlikely.

For the problems with general constraints, Potts [50] summarized various solution methods up to 1985, and presented an integer programming formulation, where a Branch and Bound solution procedure is applied using lower bounds from a Lagrangian Relaxation. Recently, Van de Velde (1990) proposed a dual decomposition method based on another Lagrangian Relaxation [56]. These methods have difficulties in solving large problems because of weak lower bounds. The reported computational results for these methods involve problems of size only up to 100 jobs.

Balas [3] pioneered the study of scheduling polyhedra on a closely related job shop scheduling problem, allowing for multiple machines, release dates and sequence-dependent changeover times. In his work, precedence constraints only arise between jobs processed on different machines. Balas gives some fundamental results about scheduling polyhedra and partial characterizations thereof. Queyranne [52] studied single machine scheduling polyhedra, and derived the class of facet-defining inequalities for the scheduling polyhedra without precedence constraints. He also proposed an $O(n \log n)$ separation algorithm for this class of inequalities.

Other polyhedral studies of scheduling problems explored alternate formulations. Dyer and Wolsey [16] used a mixed-integer programming formulation for the single machine problem with release dates. They also compared alternate relaxations in terms of
the resulting lower bounds for that problem. Sousa and Wolsey [55] used a large-scale 0-1 programming formulation for the single machine problem with release dates, deadlines and a general separable objective function. More recently, Wolsey [57] considered extended mixed-integer programming formulations, and derived very tight lower bounds, for the precedence constrained scheduling problem. His formulations involve $O(n^2)$ variables, mostly binary, and $O(n^2)$ or $O(n^3)$ constraints.

In this thesis, we consider a formulation of the precedence constrained single scheduling problem involving only $n$ variables, one per job. In Chapter 2, we propose three classes of facet defining inequalities for the scheduling problem. We show that two classes suffice to define the scheduling polyhedra when the constraints are series-parallel. The first class, introduced by Queyranne [52], consists of inequalities that result from parallel composition, and are thus called parallel inequalities. The inequalities in the second class result from series compositions and are called series inequalities.

In Chapter 3, we develop a cutting plane algorithm for the scheduling problem. The Cutting plane methods, based on facet defining inequalities, have proven to be highly successful in solving large scale TSP problems [49]. In contrast, the cutting plane approaches have not been well explored in the area of scheduling [56]. This motivates the present computational study. The proposed algorithm includes separation procedures for the above facet cuts. The separation for the class of parallel inequalities takes $O(n \log n)$ computational steps. The separation for a subclass of series inequalities, called simple series inequalities, requires $O(n^2)$ computational steps. The algorithm is tested on an IDM 386/25 Personal Computer. We use a random sample of several hundred problems with up to several hundreds of jobs. The computational results show that this algorithm consistently yields tight lower bounds and feasible schedules with maximum percent deviations from the optimum less than $0.25\%$ on the average, and less than $1\%$ in the empirical worst case.
These results indicate that the polyhedral approach has potential for solving large-size constrained scheduling problems. These tight bounds can be used for other optimization procedures, such as Branch and Bound or Branch and Cut methods if an optimal schedule is sought. The ultimate goal of this research will be to apply similar polyhedral techniques to more complex scheduling problems.

1.3 Travelling Salesman Problems

Travelling Salesman Problems (TSPs) arise when seeking a shortest tour, starting from a home city, visiting each given city exactly once and then returning to the home city. If the travel distance (or cost) from one city to another equals the return distance, the problem is called a Symmetric Travelling Salesman (STS) problem; otherwise it is an Asymmetric Travelling Salesman (ATS) problem. For more than a century, TSP has been one of the most well-known mathematical problems and is notoriously difficult.

TSPs are of both theoretical interest and practical importance. They are known to be so-called strongly NP-hard, see Garey and Johnson [22]. Therefore the existence of a polynomial optimization algorithm is highly unlikely. On the other hand, many practical problems in production and distribution can be formulated as TSPs. For example, such problems arise in the fabrication of circuit boards, where lasers must drill tens to hundreds of thousands of holes in a board. Deciding an optimal order to drill these holes is a travelling salesman problem. TSPs also represent the simple versions of vehicle routing problems. Kearny [30] reported that annual distribution costs in the United States are about 400 billion dollars. We refer to [35] for a thorough discussion of, and motivation for, the study of Travelling Salesman problems.

For convenience, we consider STS and ATS problems in complete graphs on any given node set $V'$. A feasible solution to the STS (resp., ATS) problem is a Hamiltonian
circuit $C$ on $V'$, uniquely determined by its corresponding 0-1 incidence vector $y^C$, where $y^C_e = 1$ if edge $e$ is used in $C$ and zero otherwise. The STS polytope $STSP(V')$ (resp., ATS polytope $ATSP(V')$) is the convex hull of all incidence vectors of feasible solutions to the STS (resp., ATS) problem on $V'$. In addition to being interesting mathematical objects in themselves, these polytopes are worthy of study because corresponding valid linear inequalities, in particular those that are facet-defining, play an important role in formulating and solving Travelling Salesman problems.

There have been major advances in solving many large scale real world Travelling Salesman problems over the past decade. The most notable one is the resolution of the 2392-city problem by Padberg and Rinaldi [49]. (See also Kolata [31] for a recent report on some ongoing work.) They developed an effective Branch and Cut procedure which uses some simple families of facet defining inequalities, namely the subtour elimination constraints and simple clique tree inequalities. Their success has motivated extensive study of polyhedral structure of TSP. For more details, we refer to Grötschel and Padberg [28].

To establish facial results for a given polyhedron, indirect proofs (or polyhedral proofs) [28] are often used. Such proofs are particularly effective if the polyhedron is full dimensional, because an inequality defining any given facet is unique up to a positive multiple. However, this is not the case for TSPs. Since every solution to TSPs satisfies degree constraint equations, i.e., the restrictions that the travelling salesman enters each city and leaves it exactly once, the resulting polytopes are far from being full dimensional. This considerably complicates indirect proofs for TSP polyhedral results [27,28], [29].

In recent years, two major research directions have been followed for the polyhedral study of TSPs. The first one is to reformulate TSPs by relaxing the degree constraints: the monotone relaxation [28] (resp., graphical relaxation [13],[10]) assumes that each city is visited at most once (resp., at least once). The resulting polytopes contain
TS polytopes as faces, and have full dimension. The second approach employs *clique-lifting* [27],[38],[41], [6] for extending TSP facets, and *composition* [41] techniques for extending STSP facets. Using these approaches, large classes of facet defining inequalities have been obtained. However, the derivations of these results still require very complicated and lengthy proofs, see [27,28], [29], [41].

The primary motivation for the present work is twofold. First, as the search for TSP facets becomes increasingly difficult, a simple, alternate approach is proposed. Second, while new TSP facets emerge from time to time, stronger but easily applicable facet generating operations are of particular interest.

This thesis explores a *projective approach* to the polyhedral study of Travelling Salesman (TS) problems. Suppose that we ignore the visit of the home city, and seek a shortest path that visits each city exactly once with arbitrary (and distinct) initial and final cities. This is called a *Shortest Hamiltonian Path* (SHP) problem. The associated polytope, the *Hamiltonian Path (HP) Polytope*, is the convex hull of incidence vectors of all Hamiltonian paths. First, we observe that there is a one-to-one correspondence between solutions of TSP and those of HP problems. Next, we note that in the SHP problem, all degree constraints are relaxed, because the end cities are not fixed in advance. These observations lead to a projective approach to the polyhedral study of TSPs. More precisely, we show that an HP polytope is a projection of a TS polytope. Two important properties are: (a) *near-full dimensionality*: HP polytopes are of full dimension minus one; (b) *polyhedral equivalence*: the polyhedral structures of the HP polytope and the STS polytope are isomorphic. Property (a) greatly facilitates the study of HP polytopes, and property (b) enables one to directly transfer facetial results from HP polytopes to TS polytopes.

In Chapter 4, we study Symmetric Travelling Salesman (STS) polytopes. Using the projective approach, we first derive various *Clique-Lifting* techniques to extend known STS facets. We then demonstrate how these lifting results generalize previous results.
Finally, we show that all facets with $0 - 1$ coefficients are *clique-liftable* with respect to any node.

In Chapter 5, we study Asymmetric Travelling Salesman (ATS) polytopes. As Balas and Fischetti [6] have shown that all ATS facets can be clique-lifted, we consider another type of facet extension technique, *facet composition*. In particular, we work on *symmetric inequalities* (i.e., those inequalities $ax \leq a_0$ with coefficients satisfying $a_{ij} = a_{ji}$ for all $i$ and $j$). By this facet composition technique, we show that large known classes of facets of STS polytopes also induce facets of ATS polytopes. On the other hand, all symmetric facet defining inequalities for ATS polytopes are easily shown to induce facets of the STS polytopes [18]. As a result, new facets of STS polytopes can also be derived in this fashion.
Chapter 2

Single Machine Scheduling: Polyhedral Structure

2.1 Introduction

A set $N = \{1, 2, \ldots, n\}$ of jobs is to be processed without interruption on a single machine. All jobs are released at time $t = 0$. At any instant, the machine can process at most one job. With each job $j \in N$ is associated a known positive processing time $p_j$. Precedence constraints are represented by an acyclic digraph $G(N) = (N, A(N))$, where $(i, j) \in A(N)$ denotes that job $i$ must be processed before job $j$.

We use the job completion times $C = (C_1, \ldots, C_n)^t$ to describe schedules of interest. Thus, by letting $I_N$ denote the index set of the jobs in $N$ that can be sequenced first (i.e., those jobs without predecessors), the collection of all feasible schedules can be represented as follows:

$$T(N) = \{C \in R^n : C_i \geq p_i, \forall i \in I_N; C_j - C_i \geq p_j, \forall (i, j) \in A(N); C_j - C_i \geq p_j \lor C_i - C_j \geq p_i, \forall i, j \in N, i \neq j, (i, j) \notin A(N)\}$$

We note that $T(N)$ is a disjunctive set.

Most of the research on this subject to date deals with optimization versions of the job sequencing problem with some objective function $f(C)$ over $T(N)$. For general precedence relations on the job set, the optimization problem has been shown to be NP-hard, see [33,37]. However, if the precedence constraints are series-parallel and the objective function $f(C)$ admits of a string interchange relation (linear functions are special cases), an optimal sequence can be found in $O(n \log n)$ time, see Lawler [34].
In this chapter, we study the facial structure of the *scheduling polyhedron* $P(N)$ defined as follows:

$$P(N) = \text{conv}(T(N))$$

In the next section, we introduce notation, definitions and preliminary algebraic results. We present some classes of valid inequalities for the scheduling polyhedron $P(N)$ with general precedence constraints in section 2.3, and a partial description of its facial structure in section 2.4. Three classes of valid inequalities used in this description are associated with series composition, parallel composition and induced $Z$-subgraphs, respectively. The class of inequalities used in Queyranne [52] is a special case of the parallel composition class. In section 2.5, we derive the unique minimal linear system defining $P(N)$, consisting of the first two classes of facet-inducing inequalities, if the precedence constraints are series-parallel. We also show that facet-inducing inequalities of the third class, associated with $Z$-induced subgraphs, arise whenever the precedence constraints are not series-parallel.

### 2.2 Definitions and preliminaries

Graph $G(S) = (S, A(S))$ is said to be a *subgraph* of $G(N)$ induced by $S$ if $S \subseteq N$ and $A(S) = \{(i, j) \in A(N) : i \in S, j \in S\}$. Similarly, we define the collection of all feasible *subschedules* induced by job set $S \subseteq N$:

$$T(S) = \{ C \in R^S : C_i \geq p_i, \forall i \in I^S; C_j - C_i \geq p_j, \forall (i, j) \in A(S); C_j - C_i \geq p_j \lor C_i - C_j \geq p_i, \forall i, j \in S, i \neq j, (i, j) \notin A(S) \}.$$

Note now that $G(S)$ specifies precedence relations on job set $S$ and the scheduling polyhedron induced by $S$ is

$$P(S) = \text{conv}(T(S)).$$
Chapter 2. Single Machine Scheduling: Polyhedral Structure

Node $i$ has precedence over node $j$ (or job $i$ must be processed before job $j$), denoted by $i \rightarrow j$, if there is a directed path from $i$ to $j$ in $G(N) = (N, A(N))$. Similarly, for any disjoint subsets $S_1, S_2$ of $N$, $S_1 \rightarrow S_2$ iff $i \rightarrow j$ for all $i \in S_1$ and $j \in S_2$. For any $S \subseteq N$, $A(S)$ is said to be transitivity reduced if $(i, j) \in A(S)$ implies that $(i, j)$ is the unique directed path from $i$ to $j$ in $G(S)$. For any $S \subseteq N$, $\tilde{A}(S)$ is said to be transitivity closed if

$\{(i, k), (k, j)\} \subseteq \tilde{A}(S) \Rightarrow (i, j) \in \tilde{A}(S)$.

Note that if $\tilde{A}_1(S)$ and $\tilde{A}_2(S)$ are two transitively closed arc sets containing $A(S)$, then so is $\tilde{A}_1(S) \cap \tilde{A}_2(S)$. Therefore, we may define the transitive closure of $A(S)$ as the smallest transitively closed set $\tilde{A}(S)$ satisfying $\tilde{A}(S) \supseteq A(S)$. Unless otherwise noted, we use $A(S)$ (resp., $\tilde{A}(S)$) to represent the transitively reduced arc set (resp., the transitive closure) of $A(S)$.

A set $S \subseteq N$ is said to be $Z$-inducing if $|S| \geq 4$ and there exists a numbering of the nodes in $S$ such that $S = \{1, 2, 3, ..., k\}$ and $\{(i, j) \in \tilde{A}(N) : i, j \in \{1, 2, 3, 4\}\} = \{(1, 2), (3, 2), (3, 4)\}$. Note that a graph $G(N)$ contains a $Z$-subgraph if there is a $Z$-inducing subset $S \subseteq N$. Note also that a graph is series-parallel if and only if it contains no $Z$-subgraph, see Lawler [34].

Graph $G(S)$ is series-decomposable if there exists a partition $(S_1, S_2)$ of $S$ such that $S_1 \rightarrow S_2$, in which case, we may write $S = S_1 \rightarrow S_2$.

A schedule $C \in R^S$ is a permutation schedule on $S$ if $C \in T(S)$ has no inserted idle times.

Subset $S \subseteq N$ is an initial set of $N$ if $k \rightarrow j$ and $j \in S$ imply $k \in S$.

Subset $S \subseteq N$ is a terminal set of $N$ if $i \rightarrow k$ and $i \in S$ imply $k \in S$.

Subset $S \subseteq N$ is an intermediate set of $N$ if $i \rightarrow k \rightarrow j$, $i \in S$ and $j \in S$ imply $k \in S$. 
Note that the intersection of any collection of initial (resp., terminal; resp., intermediate) sets is an initial (resp., terminal; resp., intermediate) set. Thus, for any $S \subseteq N$, let

\[
I(S) = \cap \{I : S \subseteq I, \text{I is initial}\},
\]

\[
T(S) = \cap \{T : S \subseteq T, \text{T is terminal}\},
\]

\[
Q(S) = \cap \{Q : S \subseteq Q, \text{Q is intermediate}\}.
\]

We call $I(S)$ (resp., $T(S)$, $Q(S)$) the \textit{smallest} initial (resp., terminal, intermediate) set containing $S$.

The following three propositions establish some properties of the subsets $I$, $T$ and $Q$ of $N$.

\textbf{Proposition 2.1} $S \subseteq N$ is an initial set of $N$ iff $N \setminus S$ is a terminal set of $N$.

\textbf{Proposition 2.2} If $S \subseteq N$ is an initial or a terminal set of $N$, then $S$ is an intermediate set of $N$.

Proofs of Propositions 2.1 and 2.2 are straightforward, and therefore omitted.

\textbf{Proposition 2.3} \textit{Assume}

1. $Q$ is an intermediate set of $N$; and

2. $I(Q)$ is the smallest initial set of $N$ containing $Q$.

Then, $Q$ is a terminal set of $I(Q)$.

\textit{Proof.} By contradiction, suppose that $Q$ is not a terminal set of $I(Q)$, i.e., there exists some $j \in Q$ such that $j \rightarrow k$ for some $k \in I(Q) \setminus Q$. We claim that $k$ has no successors in $Q$ since otherwise assumption (1) implies $k \in Q$ which is a contradiction. Let $V$ be
the set containing $k$ and all its successors. Thus, $I(Q) \setminus V$ is another initial set of $N$ containing $Q$ which contradicts assumption (2). \hfill \Box

For $S \subseteq N$, we let $a_S \in R^S$ be the subvector of $a \in R^N$ whose components are those in $a$ indexed by $S$. Let $(S_1, S_2)$ be a partition of $S$. Then by $C = (a_{S_1}, b_{S_2})$, where $a, b \in R^S$, we mean that

$$C_k = \begin{cases} a_k, & \text{if } k \in S_1; \\ b_k, & \text{if } k \in S_2. \end{cases}$$

For arbitrary vectors $u, v \in R^N$ and subset $S \subseteq N$, we let

$$u(S) = \sum_{j \in S} u_j, \quad u^2(S) = \sum_{j \in S} u_j^2,$$

$$u(S)^2 = (\sum_{j \in S} u_j)^2, \quad u \ast v(S) = \sum_{j \in S} u_j v_j.$$

Define also

$$g(S) = \frac{(p(S)^2 + p^2(S))}{2},$$

$$g'(S) = \frac{(p(S)^2 - p^2(S))}{2}.$$

**Proposition 2.4** Let $w_k \in R, k = 1,...,n$. Let $a^i \in R^N, i = 1,..., n = |N|$, be $n$ affinely independent vectors satisfying

$$\sum_{k=1}^n w_k a^i_k = w_0, \quad \forall i,$$  \hspace{1cm} (2.1)

and let $S = \{k : w_k \neq 0\}$.

Then,

1. There are precisely $s = |S|$ affinely independent vectors in $\{a^i_S : i = 1,..., n\}$; and

2. There are precisely $l = |V| + 1$ affinely independent vectors in $\{a^i_V : i = 1,..., n\}$ for all $V \subseteq N$ such that $S \nsubseteq V$. 
Chapter 2. Single Machine Scheduling: Polyhedral Structure

Proof.

(1) Let $M$ be an $n$ by $n + 1$ matrix whose $i$th row is $((a^i)', 1)$; $M_S$ be an $n$ by $s + 1$ submatrix of $M$ whose $i$th row is $((a^i_S)', 1)$; and $M_V$ be an $n$ by $l$ submatrix of $M$ whose $i$th row is $((a^i_V)', 1)$.

(2.1) implies that the rank of $M$ is $n$ (denoted by $r(M) = n$). Thus, there is a unique (up to multiples) solution to the system

$$Mx = 0,$$

namely, by (2.1), the solution

$$x^0_k = \begin{cases} 
  w_k, & \text{if } k \in S; \\
  -w_0, & \text{if } k = n + 1; \\
  0, & \text{otherwise.}
\end{cases}$$

Since $r(M_S) < s$ implies that $r(M) < n$ and $r(M_S) > s$ implies that $x^0$ does not solve (2.2), we must have $r(M_S) = s$. Hence, (1) follows.

(2) Obviously, $r(M_V) \leq l$. We claim that $r(M_V) \geq l$ since otherwise we have $M_Vz = 0$ for some $z \neq 0$. Using this $z$, we can construct another solution, linearly independent of $x^0$, to (2.2) which is a contradiction with (2.1). Hence, (ii) follows. □

Corollary 2.5 The facet-inducing inequalities for a full dimensional polyhedron $P \subseteq \mathbb{R}^N$ are unique up to some positive multiples.

Proof. Since $P \subseteq \mathbb{R}^N$ is full dimensional, there exists no equation satisfied by all points $x \in P \subseteq \mathbb{R}^N$. The result then follows from Theorems 2.5 and 2.6 in Pulleyblank [51]. □

Proposition 2.4 and Corollary 2.5 are used in Section 4 for proving that certain valid inequalities are facet-inducing.
2.3 Three classes of valid inequalities

To make the cutting plane method a success, valid inequalities, especially those inducing facets, play a key role. In this section, we present some classes of valid inequalities for $P(N)$. Of particular interest are those valid inequalities which, under certain conditions, are facet-inducing. Proposition 2.7 and Proposition 2.8 give three classes of valid inequalities which play an important role in the sequel. We begin with Proposition 2.6 which gives a necessary condition for an inequality to be valid for $P(N)$.

**Proposition 2.6** If

$$\sum_{k \in N} w_k C_k \geq w_0$$

is a valid inequality for $P(N)$, then $w(N) \geq 0$.

**Proof.** Suppose that $w(N) < 0$. Let $C \in P(N)$ be a schedule and $y \in R$, and let

$$\hat{C} = C + y1,$$

where $1 \in R^n$ is a vector of one's.

Clearly, $\hat{C} \in P(N)$ for all $y \geq 0$. But $\sum_{k \in N} w_k \hat{C}_k < w_0$ for $y$ sufficiently large, which produces a contradiction. \qed

We call a valid inequality for $P(N)$ a *positive-sum* (resp., *zero-sum*) valid inequality if $w(N) > 0$ (resp., $w(N) = 0$).

We now present three classes of valid inequalities for $P(N)$. The first class was introduced in Queyranne [52]. The other two are new, arising from the precedence constraints.

**Proposition 2.7** (i) For all $S \subseteq N$,

$$p^* C(S) \geq g(S) \quad (2.3)$$
is a valid inequality for \( P(N) \). Moreover, inequality (2.3) is tight if \( S \) is an initial set of \( N \).

(ii) For any \( S_1 \subseteq N, S_2 \subseteq N \) such that \( S_1 \to S_2 \),

\[
F(S_1, S_2) = -p(S_2)p \cdot C(S_1) + p(S_1)p \cdot C(S_2) \\
- p(S_2)g(S_1) - p(S_1)g(S_2) + p(S_2)p^2(S_1) \geq 0
\]

(2.4)
is a valid inequality of \( P(N) \). Moreover, inequality (2.4) is tight if \( S = S_1 \cup S_2 \) is an intermediate set of \( N \).

Proof. It follows from Theorem 3.1 of Queyranne [52] that (2.3) is a valid inequality for \( P(N) \). Next, observe that if \( S \) is initial, then any permutation schedule in \( T(N) \) with sequence \((S, N \setminus S)\) is tight for (2.3). This completes the proof for (i).

(ii) Note that for any \( C \in T(N) \), there exists a time \( t \) such that

(a) all jobs indexed by \( S_1 \) are completed before or at time \( t \), and

(b) all jobs indexed by \( S_2 \) start after or at time \( t \).

Using the results in (i), we have (a) implies

\[
p(S_2) \sum_{i \in S_1} p_i(t - C_i + p_i) \geq g(S_1)p(S_2), \tag{2.5}
\]

and (b) implies

\[
p(S_1) \sum_{j \in S_2} p_j(C_j - t) \geq g(S_2)p(S_1). \tag{2.6}
\]

Adding (2.5) to (2.6) yields (2.4).

Next, observe that if \( S \) is intermediate, then for some \( t \geq 0 \), there exists a feasible permutation schedule tight for both (2.5) and (2.6), therefore tight for (2.4).

Suppose now that \( N \) is \( Z \)-inducing. This structure is used to generate the third class of valid inequalities studied here. Because \( N \) is \( Z \)-inducing, there exists a numbering of
nodes in $N$ such that

$$\{(i, j) \in \bar{A}(N) : \forall i, j \in \{1, 2, 3, 4\} = \{(1, 2), (3, 2), (3, 4)\}.$$ 

For any $Q \subseteq \{i \in N : \{(3, i), (i, 2)\} \subseteq \bar{A}(N), \{(1, i), (i, 1), (4, i), (i, 4)\} \cap \bar{A}(N) = \emptyset\}$, let $S = \{1, 2, 3, 4\} \cup Q$, $S' = S \setminus \{3\}$, $S'' = S \setminus \{2\}$, $S'_1 = \{1, 3\} \cup Q$, and $S'_2 = \{2, 4\} \cup Q$.

Note that $Q$ may be empty.

**Proposition 2.8** The following inequalities

$$-p(S')p \star C(S'_1) + [p(\{1\} \cup Q)p(S) + p_2p_3]C_2 + p_3p_4C_4 \geq p(S')g'([1, 2, 3] \cup Q) + p_2p_3,$$  \hspace{1cm} (2.7)

$$-p_1p_2C_1 \geq p(S'')g([2, 4] \cup Q) + p_1p_2p_3$$  \hspace{1cm} (2.8)

are valid for $P(N)$. Moreover, they are tight if $S$ is intermediate.

**Proof.** Define

$$T^{31} = \{C \in T(N) : C_1 - C_3 \geq p_1\} \text{ and } T^{13} = \{C \in T(N) : C_3 - C_1 \geq p_3\}.$$  

Clearly, $T(N) = T^{31} \cup T^{13}$.

**Claim 1.** \textbf{(2.7) is valid for $T^{31}$.}

**Proof.** Note that the definition of $T^{31}$ implies that $\{3\} \to S'$. Using Proposition 2.7(ii) with $S_1 = \{3\}$ and $S_2 = S'$, the inequality

$$-p(S')p_3C_3 + p_3(p \star C(S')) \geq p_3g(S')$$  \hspace{1cm} (2.9)

is valid for $T^{31}$. Using Proposition 2.7(ii) again, but with $S_1 = \{1\} \cup Q$ and $S_2 = \{2\}$, the inequality

$$p(S)[-p_1C_1 - p \star C(Q) + p(\{1\} \cup Q)C_2] \geq p(S)g'([1, 2] \cup Q)$$  \hspace{1cm} (2.10)
is valid for $T^{31}$. The sum of (2.9) and (2.10) yields (2.7).

This completes the proof for Claim 1.

Claim 2. (2.7) is valid for $T^{13}$.

Proof. Note that for $T^{13}$, we have \{1\} $\rightarrow$ \{3\} $\rightarrow$ \{Q, 2, 4\}. The following two inequalities

\[ p_1p(S')(\!-\!C_1 + C_2) \geq p_1p(S')(p_2 + p_3 + p(Q)), \]  
\[ p_1p_3(-C_3 + C_2) \geq p_1p_3(p_2 + p(Q)), \]  

(2.11)  
(2.12)

directly follow from the induced precedence relation \{1\} $\rightarrow$ \{3\} $\rightarrow$ Q $\rightarrow$ \{2\}. Using Proposition 2.7(ii) with $S_1 = \{3\}$ and $S_2 = S_2'$, the inequality

\[ - p(S_2')p_3C_3 + p_3p * C(S_2') \geq p_3g(S_2') \]  
(2.13)

valid for $T^{13}$. Also, using Proposition 2.7(ii) with $S_1 = Q$ and $S_2 = \{2\}$, the inequality

\[ - p(S)p * C(Q) + p(S)p(Q)C_2 \geq p(S)g'(\{2\} \cup Q) \]  
(2.14)

is valid for $T^{13}$. The sum of (2.11) to (2.14) yields (2.7). This completes the proof for Claim 2.

Claim 1 and Claim 2 imply that (2.7) is valid for $T(N)$, and therefore for $P(N)$.

Similarly, we can show (2.8) is a valid inequality for $P(N)$ by considering

\[ T^{24} = \{ C \in T(N) : C_4 - C_2 \geq p_4 \} \text{ and } T^{42} = \{ C \in T(N) : C_2 - C_4 \geq p_2 \}. \]

Next, observe that if $S$ is intermediate, then any feasible schedule with uninterrupted subsequence (3,1,Q,4,2) (no idle time, no other jobs in between) is tight for (2.7) and (2.8).

The proof is complete. \qed

We call the inequality (2.3) parallel-inequality, (2.4) series-inequality, and the inequalities (2.7) and (2.8) Z-inequalities, respectively.
2.4 Structure of the precedence constrained scheduling polyhedra

In this section, we study the facial structure of $P(N)$. We begin with a discussion of some properties of the faces of $P(N)$. We then show under which conditions the valid inequalities, introduced in the previous section, are facet-inducing for $P(N)$. The proofs of all lemmata in this section are found in section 2.6.

Lemma 2.9 Let $n = |N|$. A valid inequality

\[ \sum_{i \in N} w_i C_i \geq w_0 \]  

(2.15)

is $(n - b - 1)$-dimensional face-inducing for $P(N)$, where $b = 0, 1, 2, \ldots, n - 1$, if and only if there are precisely $n - b$ affinely independent feasible schedules $C^j \in P(N), j = 1, \ldots, n - b$, all tight for (2.15).

Note. The above lemma is implicitly assumed in Balas [3], proof of Theorem 4.4.

With Lemma 2.9, we may use solutions or schedules interchangeably in the statements about faces of scheduling polyhedra.

Lemma 2.10 Set $S$ is not series-decomposable if and only if there exist at least $|S|$ affinely independent permutation schedules in $T(S)$.

The above two lemmata are used in studying the dimension of the face of $P(N)$.

We are now in a position to study the facial structure of $P(N)$.

Theorem 2.11 (Face-Lifting Theorem) Let

\[ \sum_{i \in S} w_i C_i \geq w_0, \]  

(2.16)

where $C \in R^S$, be an $(s - b - 1)$-dimensional face-inducing inequality of $P(S)$, where $0 \leq b < s$. 
(1) If \( S \) is an initial set of \( N \), then (2.16) is an \( (n - b - 1) \)-dimensional face-inducing inequality for \( P(N) \).

(2) If \( S \) is an intermediate set of \( N \) with \( w(S) = 0 \), then (2.16) is a zero-sum \( (n - b - 1) \)-dimensional face-inducing inequality for \( P(N) \).

Proof. Let \( s = |S| \) and \( w_i = 0 \), for all \( i \in N \setminus S \). Let

\[
F_S = \{ C \in P(S) : \sum_{i \in S} w_i C_i = w_0 \}, \quad S \subseteq N. \tag{2.17}
\]

First, note that since \( C \in T(N) \Rightarrow C_S \in T(S) \), any valid inequality for \( P(S) \) induces a valid inequality for \( P(N) \), and hence so does (2.16): \( \sum_{i \in S} w_i C_i \geq w_0 \) holds for all \( C \in P(N) \).

Next, let \( P(S) \) be represented by a minimal linear system, that is,

\[
P(S) = \{ C \in R^S : \sum_{i \in S} a_{hi} C_i \geq a_{h0}, h \in I \}, \tag{2.18}
\]

where \( I \) is an index set of all facet-inducing inequalities. For face \( F_S \), let

\[
I_e = \{ h \in I : \sum_{i \in S} a_{hi} C_i = a_{h0}, \forall C \in F_S \}. \tag{2.19}
\]

Face \( F_S \) is an \( (s - b - 1) \)-dimensional face of \( P(S) \) if and only if matrix \( A_e \) with rows \( a_h \), \( h \in I_e \), has rank \( b + 1 \). As observed before, the inequalities

\[
\sum_{i \in S} a_{hi} C_i \geq a_{h0}, \quad \forall h \in I_e,
\]

are also valid inequalities for \( P(N) \). Moreover, these inequalities are tight for all \( C \in F_N \). Therefore, \( F_N \) is at most \( (n - b - 1) \)-dimensional. We are done if we find \( (n - b) \) affinely independent schedules in \( F_N \).

(1) \( S \) is initial. Since \( F_S \) is \( (s - b - 1) \)-dimensional, there exist \( s - b \) affinely independent feasible schedules \( C_j^i \in F_S \), for all \( j = 1, \ldots, s - b \).
Chapter 2. Single Machine Scheduling: Polyhedral Structure

Let $C_{N\setminus S} \in T(N \setminus S)$, and $u = \max\{(C^i_S)_k : k \in S, j = 1, \ldots, s - b\}$. For all $j = 1, \ldots, s - b$, let

$$(C^i_N)_i = \begin{cases} (C^i_S)_i, & \text{if } i \in S; \\ (C_{N\setminus S})_i + u, & \text{if } i \in N \setminus S. \end{cases}$$

Since $S$ is an initial set of $N$, by Proposition 2.1, $N \setminus S$ is a terminal set of $N$. It follows that $C^j_N \in T(N), j = 1, \ldots, s - b$.

Now, define

$$C^j = \begin{cases} C^j_N, & \text{if } 1 \leq j \leq s - b; \\ C^1_N + \sum_{k=j+b}^n e_{\pi(k)}, & \text{if } s - b < j \leq n - b; \end{cases}$$

where $\pi$ is the permutation induced by $C^1_N$.

It is easy to verify that $C^j \in F_N, j = 1, \ldots, n - b$, and that they are affinely independent.

(2) $S$ is intermediate. Since $F_s$ is $(s - b - 1)$ dimensional, there are $s - b$ affinely independent schedules, $C^j_S \in F_s, j = 1, 2, \ldots, s - b$.

Let $U = \{k : k \in N \setminus S, \exists i \in S, k \rightarrow i\}$, $V = N \setminus (S \cup U)$, and $u = |U|$.

Let $C_U \in T(U), C_V \in T(V)$.

Let $m_1 = \max\{(C_U)_i : i \in U\}$, $m_2 = m_1 + \max\{(C^i_S)_i : i \in S, j = 1, \ldots, s - b\}$.

Define the following schedules for $P(N)$.

For $j = u + 1, u + 2, \ldots, u + s - b$,

$$(C^i_N)_i = \begin{cases} (C_U)_i, & \text{if } i \in U; \\ m_1 + (C^i_S)_i - u, & \text{if } i \in S; \\ m_2 + (C_V)_i, & \text{if } i \in V. \end{cases}$$

For $j = 1, 2, \ldots, u$, and $j = u + s - b + 1, u + s - b + 2, \ldots, n - b$,

$$C^j_N = C^{u+1}_N + \sum_{k=\alpha(j)}^n e_{\pi(k)}.$$
where \( \pi \) is the permutation induced by \( C_N^{u+1} \) and

\[
\alpha(j) = \begin{cases} 
    j, & \text{if } 1 \leq j \leq u; \\
    j + b, & \text{otherwise.}
\end{cases}
\]

It can be easily verified that schedules \( C_N^{i} \in F_N, j = 1, ..., n - b, \) are affinely independent.

Therefore, we have constructed \((n - b)\) affinely independent schedules in \( F_N \) in either case.

Note that if \( U \) (or \( V \)) is empty, the above proof is simplified. This completes the proof of the lifting theorem. \( \square \)

We mentioned in the previous section that valid inequalities (2.3), (2.4), (2.7) and (2.8) are tight under certain conditions. We study below the dimension of the face they induce under those conditions. Before showing this, we need some intermediate results.

**Lemma 2.12** Let

\[
\sum_{k \in N} w_k C_k \geq w_0
\]

be facet-inducing for \( P(N) \) and let \( S \) be an initial set of \( N \) such that for some \( i \in S, w_i \neq 0 \) and \( w(S) = w(N) \). Then \( w_k = 0 \), for all \( k \in N \setminus S \).

**Lemma 2.13** Let

\[
\sum_{k \in N} w_k C_k \geq w_0
\]

induce a \( d \)-dimensional face of \( P(N) \), where \( 0 \leq d \leq n - 1 \), and let \( Q \) be the smallest set containing \( S = \{i \in N : w_i \neq 0\} \), which is initial if \( w(N) > 0 \); intermediate if \( w(N) = 0 \).

Then, for every schedule \( C \in T(N) \) tight for (4.5.1), \( C_Q - t1_Q \), for some \( t \geq 0 \) \((t = 0 \text{ if } w(N) > 0)\), is a permutation schedule in \( T(Q) \), if any of the following conditions holds:

(i) there exists no proper initial set \( Q_0 \) of \( Q \) such that \( w(Q_0) = w(N) \); or

(ii) \( d = n - 1 \)
Lemma 2.13 asserts that, under the stated conditions, any schedule $C$ tight for (2.21) has the jobs in $Q$ processed consecutively, that is, without any other jobs (jobs not in $Q$) or idle time inserted in between.

Theorem 2.14 and Theorem 2.16 relate the dimension of a face to the number of series compositions in an appropriate set containing all jobs with nonzero coefficients in the corresponding inequality. In Theorem 2.14, we consider positive-sum inequalities, while in Theorem 2.16, we consider zero-sum inequalities.

**Theorem 2.14** Let

$$\sum_{i \in N} w_i C_i \geq w_0 \tag{2.22}$$

be a positive-sum tight valid inequality for $P(N)$. Let $S$ be the smallest initial set containing $\hat{S} = \{i \in N : w_i \neq 0\}$. Assume

(i) there exists no proper initial set $S_0$ of $S$ such that $w(S_0) = w(N)$; and

(ii) $S = S_1 \rightarrow S_2 \rightarrow \cdots \rightarrow S_{b+1}$, where $0 \leq b \leq n - 1$ is the total number of series-compositions in $S$.

Then, $F_N = \{C \in P(N) : w * C(S) = w_0\}$ is at most $(n - b - 1)$-dimensional.

**Proof.** Note that $b = 0$ iff $S$ is not series-decomposable, and thus, (ii) implies that (2.22) is not facet-inducing if $b > 0$.

By condition (i), Lemma 2.13 implies that for any $C \in T(N) \cap F_N$, $C_S$ is a permutation schedule in $T(S)$, and therefore $C$ is tight for all series-inequalities (2.4) induced by the $b$ pairs of sets, $(S_1 \rightarrow S_2), (S_2 \rightarrow S_3), \ldots, (S_b \rightarrow S_{b+1})$, (cf. Proposition 2.7).

Assume that we have a finite $P(N)$-defining linear system $a_h C \geq a_{h_0}, h \in I$, including (2.22) as the first inequality, and the inequalities (2.4) induced by $(S_{i-1} \rightarrow S_i)$ as the $i$-th inequality for $i = 2, ..., b + 1$. Note now that for all $C \in F_N$,

$$a_h C = a_{h_0}, \quad h = 1, ..., b + 1.$$
Chapter 2. Single Machine Scheduling: Polyhedral Structure

It suffices to show that \( a_h, h = 1, ..., b + 1 \), are linearly independent.

Let \( \tilde{A}_e \) be the matrix whose rows are \( a_h, h = 1, ..., b + 1 \). For any \( y \in \mathbb{R}^{b+1} \) such that \( y^t \tilde{A}_e = 0^t_N \), we have

\[
0 = y^t \tilde{A}_e 1_N = y^t (w(N), 0, ..., 0)^t = y_1 w(N)
\]

and therefore \( y_1 = 0 \) since \( w(N) > 0 \). Further, observe that any square submatrix of \( \tilde{A}_e \) with columns corresponding to exactly one job from each set \( S_i, i = 1, ..., b \), is a diagonal matrix with nonzero diagonal elements. Thus, rows \( 2, ..., b + 1 \) of \( \tilde{A}_e \) are linearly independent, which implies that \( y_2 = ... = y_{b+1} = 0 \). It follows that \( \tilde{A}_e \) itself is of rank \( b + 1 \). Therefore, face \( F_N \) is of dimension at most \( (n - b - 1) \). The proof is complete. □

The following theorem yields the dimension of the face induced by a parallel inequality (2.3).

**Theorem 2.15** For any initial set \( S \subseteq N \),

\[
p * C(S) \geq g(S)
\]

is an \( (n - b - 1) \)-dimensional face-inducing inequality for \( P(N) \), where \( 0 \leq b \leq n - 1 \) is the total number of series-compositions in \( S \).

**Proof.** Let \( S = S_1 \rightarrow S_2 \rightarrow ... \rightarrow S_{b+1} \).

Note that (2.23) satisfies condition (i) of Theorem 2.14, and therefore \( F_N \) is at most \( (n - b - 1) \)-dimensional. This further implies, by the Face-Lifting Theorem, that \( F_S = \{ C \in P(S) : p * C(S) = g(S) \} \) is at most \( (s - b - 1) \)-dimensional. To show that \( F_S \) is exactly \( (s - b - 1) \)-dimensional, we construct \( s - b \) affinely independent schedules in \( F_S \) as follows.

By Lemma 2.10, there exist \( s_k = |S_k| \) affinely independent permutation schedules \( \bar{C}_{S_k}^h, h = 1, ..., s_k \), in \( T(S_k) \), for \( k = 1, 2, ..., b + 1 \).
Let \( s_0 = m_0 = 0, z_0 = 1, \) and \( s = |S| = \sum_{k=1}^{b+1} s_k. \) Recursively define \( m_k = p(S_k) + m_{k-1}, z_k = z_{k-1} + s_{k-1} - 1, \) for \( k = 1, \ldots, b + 1. \) Define \( s - b \) schedules \( C^j \in T(S) \) as follows: For \( j = 1, 2, \ldots, s - b, k = 1, 2, b + 1, \) let

\[
C^j_{S_k} = \begin{cases} 
C^j_{S_k} - z_k + m_{k-1}, & \text{if } z_k < j \leq z_k + s_k; \\
C^j_{S_k} + m_{k-1}, & \text{otherwise.}
\end{cases}
\]

It is easy to verify that \( C^j \in T(S) \cap F_S. \) Moreover, all \( C^j - C^1, j = 2, 3, \ldots, s - b \) are linearly independent. This implies that all \( C^j \)s are affinely independent, and therefore \( F_S \) is \((s - b - 1)\)-dimensional. Since \( S \) is initial, by the Face-Lifting Theorem, \( F_N \) is \((n - b - 1)\)-dimensional.

We now turn to the dimension of the face induced by a zero-sum inequality.

**Theorem 2.16** Let

\[
\sum_{i \in N} w_i C_i \geq w_0 \tag{2.24}
\]

be a zero-sum tight valid inequality for \( P(N). \) Let \( S \) be the smallest intermediate set containing \( \hat{S} = \{i \in N : w_i \neq 0\}. \) Assume that

(i) there exists no proper initial set \( S_0 \) of \( S \) such that \( w(S_0) = 0; \)

(ii) \( S = S_1 \rightarrow S_2 \rightarrow \cdots \rightarrow S_{b+1}, \) where \( 0 \leq b \leq n - 1 \) is the total number of series-compositions in \( S. \)

Then

\[
F_N = \{C \in P(N) : w \ast C(S) = w_0\}
\]

is at most \( d \)-dimensional, where \( d = \min\{n - 1, n - b\}. \)

**Proof.** By condition (i), Lemma 2.13 implies that \( \forall C \in T(N) \cap F_N, C_S - t_1 S, \) for some \( t \geq 0, \) is a permutation schedule in \( T(S), \) and therefore \( C \) is tight for series-inequalities (2.4) induced by the \( b \) pairs of sets, \( (S_1 \rightarrow S_2), (S_2 \rightarrow S_3), \ldots, (S_b \rightarrow S_{b+1}). \) Moreover, it
can be easily verified that $\hat{A}_e$ (defined the same as in Theorem 2.14) is of rank at least $\max\{1, b\}$ (cf. Theorem 2.14.). It follows that $F_N$ is at most $d$-dimensional. 

The following theorem yields the dimension of the face induced by a series-inequality (2.4).

**Theorem 2.17** For any $S_1, S_2$ such that $S_1 \rightarrow S_2$ and $S = S_1 \cup S_2$ is an intermediate set of $N$, 

$$F(S_1, S_2) = \frac{-p(S_2)}{p(S_1)}p \ast C(S_1) + \frac{(p(S_1))p \ast C(S_2)}{p(S_2)}$$

$$-p(S_2)g(S_1) - p(S_1)g(S_2) + p(S_2)p^2(S_1) \geq 0$$

is a $(n - b)$-dimensional face-inducing inequality for $P(N)$, and $b \leq n - 1$ is the total number of the series-compositions in $S$.

**Proof.** Let $s = |S|$.

Note that (2.25) satisfies condition (i) of Theorem 2.16, and therefore $F_N$ is at most $(n - b)$-dimensional, since $b \geq 1$. This further implies, by the Face-Lifting Theorem, that $F_S = \{C \in P(S) : F(S_1, S_2) = 0\}$ is at most $(s - b)$-dimensional. To show that $F_S$ is exactly $(s - b)$-dimensional, we construct $s - b + 1$ affinely independent schedules in $F_S$ as follows.

Construct $s - b$ affinely independent feasible permutation schedules $C^j \in T(S)$, $j = 1, \ldots, s - b$ in the same way as we do in Theorem 2.15 and let $C^{s-b+1} = C^1 + 1_S$. Observe that $C^j \in T(S) \cap F_S$, $j = 1, \ldots, s - b + 1$, and we show below that they are affinely independent.

For any $v \in R^{s-b+1}$ such that $\sum_{j=1}^{s-b+1} v_j = 0$ and $\sum_{j=1}^{s-b+1} v_j C^j = 0_S$, we have

$$p^j \sum_{j=1}^{s-b+1} v_j C^j = \sum_{j=1}^{s-b+1} v_j g(S) + v_{s-b+1} p(S) = 0$$
implying $v_{s-b+1} = 0$, where $p^i \in \mathbb{R}^s$ is the processing time vector for job set $S$. Moreover, 

since $C^i, j = 1, ..., s - b$, are affinely independent, we have $v_j = 0, j = 1, ..., s - b$, 

which implies that $C^i, j = 1, ..., s - b + 1$, are affinely independent. Therefore $F_S$ is $(s - b)$-dimensional. Since $S$ is intermediate, by the Face-Lifting Theorem, $F_N$ is $(n - b)$-dimensional. 

**Theorem 2.18** below is one of the main results of this chapter. It implies that all facet-inducing inequalities other than parallel inequalities (2.3) must be zero-sum inequalities.

**Theorem 2.18** Let 

$$
\sum_{j \in N} w_j C_j \geq w_0 \quad (2.26)
$$

be a valid inequality of $P(N)$, and $S = \{i \in N : w_i \neq 0\}$ be its support. Then this inequality is positive-sum facet-inducing for $P(N)$, if and only if

(i) $S$ is an initial set of $N$;

(ii) $S$ is not series-decomposable; and

(iii) (2.26) is a positive multiple of (2.3).

**Proof.** *Sufficiency* trivially follows from **Theorem 2.15**.

**Necessity:** Let $S_0$ be the smallest initial set with $w(S_0) = w(S)$.

Since (2.26) is facet-inducing, there exist $n$ affinely independent schedules $C^i \in T(N)$, $j = 1, ..., n$, tight for (2.26). By **Lemma 2.13**, $\forall j, C^i_{S_0} \in T(S_0)$ is a permutation schedule. Thus, $C^i, j = 1, ..., n$, are tight for the valid inequality $p \ast C(S_0) \geq g(S_0)$. Hence, by **Corollary 2.5**, both (i), i.e., $S_0 = S$, and (iii) follow. Moreover, by **Proposition 2.4**, we have $s = |S|$ affinely independent feasible permutation schedules in $\{C^i_{S_0} : j = 1, ..., n\}$, which, by **Lemma 2.10**, implies (ii). The proof is complete. 

$\square$
Theorem 2.19 below characterizes which series-inequalities are facet-inducing for $P(N)$.

**Theorem 2.19** Let

$$
\sum_{k \in N} w_k C_k \geq w_0
$$

(2.27)

be a valid inequality for $P(N)$. Let $S$ be the smallest intermediate set containing $\hat{S} = \{i \in N : w_i \neq 0\}$. Assume $S = (S_1 \rightarrow S_2)$.

Then, (2.27) is zero-sum facet-inducing for $P(N)$ if and only if

(i) $\hat{S}$ is an intermediate set of $N$, i.e., $S = \hat{S}$;

(ii) neither $S_1$ nor $S_2$ is itself series-decomposable; and

(iii) (2.27) is a positive multiple of (2.4).

**Proof.**

*Sufficiency* trivially follows from Theorem 2.17.

*Necessity* Since (2.27) is facet-inducing, we have $n$ affinely independent schedules

$$
C_j \in F_N = \{C \in P(N) : w \ast C(S) = w_0\}, \quad j = 1, \ldots, n.
$$

Lemma 2.13 implies that $\forall j$, $C_j^{\hat{S}} - t_j 1_S \in T(S)$, for some $t_j \geq 0$, are permutation schedules, therefore all $C_j$'s are tight for the series-inequality defined by $(S_1 \rightarrow S_2)$. By Corollary 2.5, (i) and (iii) follow.

We show (ii) holds by contradiction.

Suppose (ii) does not hold. Then, $b \geq 2$. Further, Using (i),(iii), and Theorem 2.17, we derive a contradiction: $\dim(F_N) = n - b < n - 1$. The proof is complete. \hfill \Box

Thus, Theorem 2.18 characterizes all positive facet-inducing inequalities for $P(N)$ and Theorem 2.19 characterizes one class of zero-sum facet-inducing inequalities — the series-inequalities. The following theorem characterizes another class of zero-sum facet-inducing inequalities — the Z-inequalities.
Assume that $N$ is Z-inducing. Let $S$, $S'_1$, $S'_2$, $S'$, $S''$, and $Q$ be defined as in Section 3, just before Proposition 2.8.

**Theorem 2.20** Inequalities (2.7) and (2.8) are facet-inducing for $P(N)$ if and only if $S$ is an intermediate set of $N$.

**Proof.** If either (2.7) or (2.8) is facet-inducing, then by Lemma 2.13, any feasible schedule in $T(N)$ tight for either inequality has the jobs in $S$ processed consecutively without inserted idle times. This implies that $S$ is intermediate.

Conversely, since (2.7) and (2.8) are valid for $P(N)$ by Proposition 2.8, it suffices to construct $n$ affinely independent feasible schedules tight for (2.7) and (2.8).

Let $s = |S|, q = |Q|.$

Let $C^1 \in T(S)$ be a feasible permutation schedule with sequence $(1, 3, Q, 2, 4)$.

For $j = 2, \ldots, q+2$, let $C^j$ be the permutation schedule with job 1 in the $j$th-position and the rest of the jobs in the same sequence as in $C^1$. Further, let $C^{q+3}$ be the feasible permutation schedule with sequence $(3, 4, 1, Q, 2)$, and $C^s = C^{q+4} = C^1 + 1_s$.

Since $C^j - C^1, j = 2, \ldots, s$, are linearly independent, schedules $C^1, \ldots, C^s$, are affinely independent. Next, we can check that $C^{q+3}$ is tight for all inequalities (2.11), (2.12), (2.13) and (2.14), and all other schedules are tight for (2.9) and (2.10). Therefore, these schedules are tight for (2.7). Finally, since $S$ is intermediate, we can use the Face-lifting Theorem to lift these schedules and construct $n$ affinely independent schedules in $T(N)$ tight for (2.7).

The construction of the tight schedules for (2.8) is similar. The proof is complete. □

From Propositions 2.7, 2.8, and Theorems 2.15, 2.17 and 2.20, we have the following corollary.

**Corollary 2.21** (1) Parallel inequality (2.3) is face-inducing if and only if $S$ is initial; and
(2) Zero-sum valid inequalities (2.4), (2.7) and (2.8) are face-inducing if and only if \( S \) is intermediate.

### 2.5 Facet Lifting

In this section, we show that, when a job set \( S \) results from series or parallel composition of two subsets \( S_1 \) and \( S_2 \), all facets of \( P(S) \) can be derived. Some facets of \( P(S) \) are in fact obtained by lifting the facets of \( P(S_1) \) or \( P(S_2) \). These results further lead to a minimal linear system sufficient for scheduling polyhedra when the precedence constraints are series-parallel.

**Lemma 2.22 (Facet-Lowering Lemma)** Let

\[
\sum_{k \in S} w_k c_k \geq w_0 \tag{2.28}
\]

be a facet-inducing inequality for \( P(S) \) and let \( S_0 \subseteq S \) and \( S_0 \supseteq \{i \in N : w_i \neq 0\} \). Then \( w \ast C(S_0) \geq w_0 \) is a facet-inducing inequality for \( P(S_0) \), if \( S_0 \) is

(i) an initial set of \( S \) with \( w(S) > 0 \); or

(ii) an intermediate set of \( S \) with \( w(S) = 0 \).

**Proof.** Either (i) or (ii) implies that for any \( \hat{C} \in T(S_0) \), there exists \( C \in T(S) \) such that \( C_{S_0} - t1_{S_0} = \hat{C} \), for some \( t \geq 0 \) (Note. \( t = 0 \) if \( w(S) > 0 \)). Thus, \( \forall \hat{C} \in T(S_0) \), \( w \ast \hat{C}(S_0) = w \ast C(S_0) \geq w_0 \). Therefore, (2.28) is also valid for \( P(S_0) \). Further, by Proposition 2.4, we have \( |S_0| \) affinely independent schedules in \( \{C^j_{S_0} : j = 1, \ldots, s\} \), tight for \( w \ast C(S_0) \geq w_0 \), where \( C^i, j = 1, \ldots, s \) are \( s \) affinely independent schedules in \( T(S) \), tight for (2.28). The proof is complete. \( \square \)

Set \( S \) is said to be the parallel composition of \( S_1 \) and \( S_2 \), denoted by \( S = (S_1 || S_2) \), if \( S = S_1 \cup S_2 \) and neither \( i \rightarrow j \) nor \( j \rightarrow i \) for all \( i \in S_1 \) and \( j \in S_2 \).

The following theorem characterizes all facets resulting from parallel composition.
Chapter 2. Single Machine Scheduling: Polyhedral Structure

Theorem 2.23 The facet-inducing inequalities for \( P(S) \), where \( S = (S_1||S_2) \), are precisely the following:

(i) all facet-inducing inequalities for \( P(S_1) \),

(ii) all facet-inducing inequalities for \( P(S_2) \), and

(iii) all parallel inequalities \( p \ast C(I_1 \cup I_2) \geq g(I_1 \cup I_2) \),

where \( I_1 \) and \( I_2 \) are any nonempty initial sets of \( S_1 \) and \( S_2 \), respectively.

Proof. Since \( S_1, S_2, \) and \( I_1 \cup I_2 \) are initial set of \( S \), any of (i), (ii) (by the Face-Lifting Theorem), and (iii) (by Theorem 2.18) is sufficient to define a facet of \( P(S) \).

Conversely, let

\[
\sum_{k \in S} w_k C_k \geq w_0
\]  

be facet-inducing for \( P(S) \) and \( \hat{S} = \{ i \in S : w_i \neq 0 \} \).

If \( \hat{S} \subseteq S_1 \) (resp., \( \hat{S} \subseteq S_2 \)), by the Facet-Lowering Lemma, (2.29) has representation (i) (resp., (ii)).

If both \( \hat{S} \cap S_1 \) and \( \hat{S} \cap S_2 \) are nonempty, we claim that \( w(S) > 0 \) since otherwise \( w(S) = 0 \) implies, by Lemma 2.12, that \( w(S_i) < 0 \) for some \( i \in \{1, 2\} \), and (2.29) is not valid since \( S_i \) is a terminal set of \( S \). Since (5.2.1) is positive facet-inducing, by Theorem 2.18, \( \hat{S} \) is an initial set of \( S \). By Letting \( I_1 = \hat{S} \cap S_1 \) and \( I_2 = \hat{S} \cap S_2 \), (2.29) has representation (iii). The proof is complete. \( \square \)

The following theorem characterizes all facets resulting from series composition.

Theorem 2.24 The facet-inducing inequalities for \( P(S) \), where \( S = (S_1 \rightarrow S_2) \), are precisely the following:

(i) all facet-inducing inequalities for \( P(S_1) \),

(ii) all zero-sum facet-inducing inequalities for \( P(S_2) \), and
(iii)

\[-(p(I)p \ast C(T) + (p(T)p \ast C(I)) \geq p(I)g(T) + p(T)g(I) - p(I)p^2(T),\]

(2.30)

where I and T satisfy the following two conditions:

(a) T is a terminal set of $S_1$ and I is an initial set of $S_2$; and

(b) neither T nor I is itself series-decomposable.

Proof First, we need to show any of (i), (ii) and (iii) is sufficient to define a facet of $P(S)$.

(1) Since $S_i$ is an initial (therefore, intermediate) set of $S$, by the Face-Lifting Theorem, any facet of $P(S_i)$ induces a facet of $P(S)$.

(2) Since $S_2$ is an intermediate set of $S$, by the Face-Lifting Theorem, any zero-sum facet of $P(S_2)$ induces a zero-sum facet of $P(S)$.

(3) By Theorem 2.19, the given conditions imply that (2.30) induces a zero-sum facet for $P(S)$.

Conversely, let

\[\sum_{k \in S} w_k C_k \geq w_0, \quad C \in R^S,\]  

(2.31)

be facet-inducing for $P(S)$ and let $\hat{S} = \{i \in S : w_i \neq 0\}$. We need to consider only two cases:

(I) If (2.31) is a positive-sum facet, then by Theorem 2.18, $\hat{S}$ is an initial set of $S$ and is not series-decomposable which also implies $\hat{S} \subseteq S_1$. Further, by the Facet-Lowering Lemma, (2.31) has representation (i).

(II) If (2.31) is a zero-sum facet of $P(S)$, then we need to consider the following two cases:

(1) neither $\hat{S} \cap S_1$ nor $\hat{S} \cap S_2$ is empty.
Since \( S = (S_1 \rightarrow S_2) \), we have \( \hat{S} \cap S_1 \rightarrow \hat{S} \cap S_2 \). By **Theorem 2.19**, (2.31) has representation (iii) with \( T = \hat{S} \cap S_1 \) and \( I = \hat{S} \cap S_2 \).

(2) one of \( \hat{S} \cap S_1 \) and \( \hat{S} \cap S_2 \) is empty. Observe that both \( S_1 \) and \( S_2 \) are intermediate set of \( S \), by the **Facet-Lowering Lemma**, (2.31) has one of the representations (i) and (ii). The proof is complete. 

As a straightforward consequence of **Theorem 2.23** and **Theorem 2.24**, we have the following.

**Corollary 2.25** The following is the unique (up to positive multiples) minimal linear system sufficient to define \( P(N) \) with series-parallel precedence constraints:

(i) \( p * C(S) \geq g(S) \), for all non-series-decomposable initial set \( S \) of \( N \); and

(ii)

\[
-(p(S_2))p * C(S_1) + (p(S_1))p * C(S_2) \\
\geq p(S_2)g(S_1) + p(S_1)g(S_2) - p(S_2)p^2(S_1),
\]

for all non-series-decomposable sets \( S_1 \) and \( S_2 \) such that \( S = S_1 \rightarrow S_2 \) is an intermediate set of \( N \).

Given a scheduling polyhedron \( P(N) \) subject to series-parallel precedence constraints, **Corollary 2.25** gives a complete characterization of all facets of \( P(N) \). These facets can be generated by repeatedly applying the rules stated in **Theorem 2.23** and **Theorem 2.24**, starting from the leaves (singleton subsets of \( N \)) and following the compositions specified by any series-parallel decomposition tree of \( N \) (see Lawler [34], for details on the series-parallel decomposition trees).

If the precedence relations on the job set are not series-parallel, the facet-inducing inequalities of the scheduling polyhedron \( P(N) \) are not all of parallel or series inequalities, as shown by the following theorem.
**Theorem 2.26** All facet-inducing inequalities for $P(N)$ are in the form of either (2.3) or (2.4) if and only if $G(N)$ is series-parallel.

**Proof.** Sufficiency follows from Corollary 2.25.

**Necessity:** If $G(N)$ is not series-parallel, it is well known that $N$ is $Z$-inducing (see Lawler [34]).

Let $S \subseteq N$ be a minimal intermediate $Z$-inducing set.

Renumber the jobs such that $S = \{1, 2, 3, \ldots, k\}$,

$$\{(i, j) \in \bar{A}(N) : \forall i, j \in \{1, 2, 3, 4\}\} = \{(1, 2), (3, 2), (3, 4)\},$$

and $Q = \{5, \ldots, k\}$. Note that $Q$ may be empty.

Since $S$ is a minimal intermediate $Z$-inducing set, $\{1, 2\}$, $\{3, 4\}$ and $\{3, 2\} \cup Q$ are intermediate sets of $N$. Moreover, for all $j \in Q$, we have $(3, j) \in \bar{A}(N)$ and $(j, 2) \in \bar{A}(N)$.

By Theorem 2.20, $Z$-inequality (2.7) or (2.8) induced by $S$ is a facet-inducing inequality for $P(N)$. The proof is complete. □

Note that, for series-parallel precedence constraints, facets are uniquely defined by the index set of their nonzero coefficients. However, Theorems 2.20 and 2.26 show that this is never the case when the precedence constraints are not series-parallel.

### 2.6 Proofs of the four lemmata

**Proof of Lemma 2.9.** Let $S \equiv \{i : w_i \neq 0\}$, and $F \equiv \{C \in P(N) : w \ast C(S) = w_0\}$.

First, $F$ is an $(n - b - 1)$-dimensional face of $P(N)$ if and only if there are precisely $n - b$ affinely independent points $\hat{C}^k \in F, k = 1, 2, \ldots, n - b$. Let $C^j \in T(N), j = 1, 2, \ldots, m$ be a minimal collection of affinely independent feasible schedules such that each $\hat{C}^k$ can be represented by a positive convex combination of $C^j$s. Then, we have $\forall k$,

$$w_0 = w \ast \hat{C}^k(S) = \sum_{h \in S} w_h \sum_{j=1}^m v_{kj} C^j_h = \sum_{j=1}^m v_{kj} w \ast C^j(S) \geq w_0,$$
where \( v_{kj} \geq 0, \forall j \), and \( \sum_{j=1}^{m} v_{kj} = 1 \), which implies that \( C^j \in F, j = 1, \ldots, m \).

Thus, \( C^j \)'s also span the linear hull of \( F \). Hence, \( \dim(F) = n - b - 1 \) if and only if there are precisely \( n - b \) affinely independent schedules in \( F \). The proof is complete. \( \square \)

**Proof of Lemma 2.10.** Let \( s = |S| \).

**Sufficiency.** Suppose \( S \) is series-decomposable, i.e. \( S = S_1 \rightarrow S_2 \).

Let \( C^j \in R^s, j = 1, \ldots, l \), be a maximal collection of affinely independent permutation schedules in \( T(S) \). Clearly, all \( C^j \)'s are tight for both (2.3) and series-inequality (2.4), whose coefficients are linearly independent. It follows that \( l \leq s + 1 - 2 = s - 1 \).

**Necessity.** Suppose \( S \) is not series-decomposable.

By induction on \( |S| \). For \( |S| = 1 \), the necessity trivially holds. Suppose it holds for all \( |S| \leq s - 1 \).

Let \( S' = S \setminus \{i\} \), where \( \{i\} \) is a terminal set in \( S \). We have two cases:

1. \( S' \) itself is not series-decomposable. Then, by the induction hypothesis, there are \( s - 1 \) affinely independent permutation schedules in \( T(S') \), namely, \( \hat{C}^j, j = 1, \ldots, s - 1 \). Observe that there exists some terminal set \( \{k\} \) of \( S' \) such that job \( k \) can be sequenced after job \( i \), since otherwise \( S = (S' \rightarrow \{i\}) \).

   Define
   \[
   C^j_h = \begin{cases} 
   \hat{C}^j_h, & \text{if } h \in S'; \\
   p(S), & \text{if } h = i,
   \end{cases}
   \]
   for all \( j = 1, \ldots, s - 1 \), and let \( C^* \) be the permutation schedule in \( T(S) \) such that job \( k \) is sequenced last and the rest of the jobs are sequenced the same order as in \( C^1 \). We easily verify that \( C^1, \ldots, C^* \) are affinely independent permutation schedules in \( T(S) \).

2. \( S' \) is series-decomposable, i.e., \( S' = (S_1 \rightarrow S_2) \).

First, observe that \( S_1 \cup \{i\} \) is not series-decomposable, since otherwise \( S_1 \cup \{i\} = (S'' \rightarrow S''' \) with \( S''' \) containing terminal job \( i \), and thus, \( S = S'' \rightarrow (S''' \cup S_2) \), a contradiction. Therefore, by the induction hypothesis, there are \( |S_1| + 1 \) affinely independent
permutation schedules in $T(S_1 \cup \{i\})$, namely, $C^j_{S_1 \cup \{i\}}, j = 1, \ldots, |S_1| + 1$.

Let $C_{S_2} \in T(S_2)$ be any permutation schedule and define

$$C^j_h = \begin{cases} (C^j_{S_1 \cup \{i\}})_h, & \text{if } h \in S_1 \cup \{i\}; \\ (C^j_{S_2})_h + p(S_1 \cup \{i\}), & \text{if } h \in S_2, \end{cases}$$

for $j = 1, \ldots, |S_1| + 1$.

Let $\pi$ be the permutation induced by $C^1$.

For $|S_1| + 2 \leq j \leq s$, let $C^j$ be a permutation schedule with job $i$ sequenced at the $\pi(j)$-th position and the rest of the jobs are sequenced in the same order as in $C^1$.

Observe that no job $k \in S_2$ has precedence over job $i$, since otherwise $S = S_1 \rightarrow (S_2 \cup \{i\})$, a contradiction. Therefore, all $C^j, j = 1, \ldots, s$, are feasible permutation schedules in $T(S)$. Moreover, since $C^j - C^1, j = 2, 3, \ldots, s$, are linearly independent, $C^j, j = 1, 2, \ldots, s$ are affinely independent.

In either case, we can construct $s$ affinely independent permutation schedules in $T(S)$.

The proof is complete. $\Box$

**Proof of Lemma 2.12.**

1. Since (2.20) is facet-inducing, there are $n$ affinely independent schedules, $C^j \in T(N), j = 1, \ldots, n$, satisfying:

$$\sum_{k \in N} w_k C^j_k = w_0.$$

2. We claim that

$$\sum_{k \in S} w_k C^j_k = v_j = v,$$

for all $j = 1, \ldots, n$, and prove it by contradiction.

Suppose not. Then, there exist some $h, l$ such that $v_h > v_l$. Let

$$\hat{C}_i = \begin{cases} (C^j_S)_i, & \text{if } i \in S; \\ (C^j_{N \setminus S})_i + u, & \text{if } i \in N \setminus S. \end{cases}$$
Since $S$ is an initial set of $N$, $\hat{C} \in T(N)$ for $u$ sufficiently large. Moreover,

$$\sum_{k \in N} w_k \hat{C}_k = v_l + w_0 - v_h < w_0,$$

which contradicts the fact that (2.20) is a valid inequality for $P(N)$.

From (1), (2), and Corollary 2.5, it follows that $w_0 = v$ and $w_k = 0$ for all $k \in N \setminus S$. The proof is complete.

**Proof of Lemma 2.13** Let $X$ be any initial set of $N$ and $\bar{X} = N \setminus X$ such that $X \cap S \neq \emptyset$ and $\bar{X} \cap S \neq \emptyset$. Observe that $\bar{X}$ is a terminal set of $N$, we have $w(\bar{X}) \geq 0$ since otherwise (2.21) is not valid. Using condition (i), it is trivial that $w(\bar{X}) > 0$. Using condition (ii), we cannot have $w(\bar{X}) = 0$ since otherwise we have $w(X) = w(N)$ and by Lemma 2.12, $w_i = 0, \forall i \in \bar{X}$. This further implies $\exists k \in S, w_k = 0$, a contradiction. Therefore, we have $w(\bar{X}) > 0$ in either case.

Next, by contradiction, suppose $\exists C \in T(N)$ tight for (2.21) such that $C_Q - t1_Q$, for any $t > 0$, is not a permutation schedule in $T(Q)$. Then, $C$ represents a feasible schedule such that either there exists some idle time between the completion times of some jobs in $Q$, or jobs in $Q$ are not sequenced consecutively. Let $I(Q)$ be the smallest initial set containing $Q$. Let $\pi$ be a permutation induced by $C$. For $w(N) > 0$, we note that

(a) $I(Q) = Q$, by definition of $Q$ and $I(Q)$;

(b) $C_\pi(1) = p_\pi(1)$, since otherwise $\hat{C} = C - (C_\pi(1) - p_\pi(1))1_N \in T(N)$, but $w \ast \hat{C}(N) = w_0 - w(N) < w_0$, a contradiction. Hence, there is no idle time before $C_\pi(1)$ for $w(N) > 0$. Define

$$\delta(x) = \begin{cases} 1, & \text{if } x > 0; \\ 0, & \text{if } x \leq 0. \end{cases}$$

By Proposition 2.3, $Q$ is a terminal set of $I(Q)$. Let $M_e = \max\{C_k : k \in I(Q)\}$ and
define a schedule $\hat{C}$ as follows:

$$
\hat{C}_i = \begin{cases} 
C_i, & \text{if } i \in I(Q) \setminus Q; \\
C_i + M_e(1 - \delta(w(N))), & \text{if } i \in Q; \\
C_i + 2M_e, & \text{if } i \in N \setminus I(Q).
\end{cases}
$$

Observe that $\hat{C} \in T(N)$ and is also tight for (2.21) but has some idle time inserted between the completion times of some jobs in $Q$. (Note: For $w(N) > 0$, none of the completion times of jobs in $Q$ change, i.e., $C_Q = \hat{C}_Q$. Nevertheless, if $C_Q$ contains any idle time, so does $\hat{C}_Q$; else some other job is sequenced between jobs in $Q$ and thus, by definition, $\hat{C}_Q$ contains idle time). Since $Q$ is the smallest (intermediate if $w(N) = 0$; or initial if $w(N) > 0$) set containing $S$, there exists an idle time interval $(t, t + u)$, for some $t, u > 0$ (for $w(N) > 0$, cf. Note (b) above), such that some jobs in $S$ are processed before $t$ and some jobs in $S$ are processed after $t + u$.

Let $X = \{i \in N : \hat{C}_i \leq t\}$ \(\bar{X} = \{i \in N : \hat{C}_i > t + u\}\). Clearly, $N = X \cup \bar{X}$ and both $S \cap \bar{X}$ and $S \cap X$ are nonempty.

Define $\bar{C} = \hat{C} - u \sum_{i \in X} e_i$.

It is clear that $\bar{C} \in T(N)$. Since we have before $w(\bar{X}) > 0$,

$$
\sum_{i \in N} w_i \bar{C}_i = w_0 - w(\bar{X})u < w_0,
$$

a contradiction. This completes the proof for the lemma. \(\square\)
This chapter presents a cutting plane procedure, using facet cuts derived in Chapter 2. In addition, extensive computational results as well as their analysis are provided.

3.1 Introduction

For the last thirty-five years, since the work of Smith [54], the single machine scheduling problem has been one of the challenging research topics in combinatorial optimization. The problem with general precedence constraints was shown to be strongly NP-hard by Lawler [33] and Lenstra and Rinnooy Kan [37]. For the series-parallel precedence constraints, an elegant $O(n \log n)$ algorithm was derived by Lawler [33]. A discussion of the algorithms (up to 1985) for the problem with general constraints can be found in Potts [50]. Very tight formulations with $O(n^2)$ binary variables and $O(n^3)$ constraints are studied by Wolsey [57]. A new Lagrangian-based lower bounding method with $O(n)$ continuous variables (and one Lagrangian multiplier per nonredundant precedence constraint) is introduced in Van de Velde [56].

In spite of these research efforts, one is still unable to solve even medium-sized problems. The reported largest solved problem instances involve no more than 100 jobs, see Potts [50], where Branch-and-Bound methods were used for obtaining optimal solutions. The reason why the methods do not apply to the larger problems may be a combination of (1) the number $O(n^2)$ of binary variables in the integer programming formulations, and (2) weak lower bounds. To a large extent, the present polyhedral approach overcomes
both difficulties.

In this chapter, a linear programming based cutting plane algorithm is developed for the scheduling problem. The proposed algorithm works on a compact linear formulation with only \( n \) variables, one per job. Two families of facet-defining inequalities, the parallel inequalities and the so-called simple series inequalities, are used in the cutting plane procedure. Extensive computational experiments are conducted on an IDM 386/25 Personal Computer with a sample of 280 problems of sizes 30 to 160 jobs. The average computation time for the 160-job problems is less than 20 minutes. The results also show that the percent gap in objective function value between the feasible solution derived from the cutting plane algorithm and the linear programming lower bound is less than 0.25 percent on the average, and less than 1 percent in the empirical worst case. A detailed computational report can be found in Section 3.4.

3.2 A cutting plane procedure

The proposed cutting plane algorithm consists of three stages. It begins by solving an initial linear programming (LP) problem. This initial LP formulation defines a polyhedron containing the set of feasible schedules. The second stage is an iterative process: at each iteration, it generates (1) an optimal solution to the current LP relaxation, which serves as a lower bound (LB) ; (2) possibly a valid inequality or a cut, violated by the LP solution; (3) a heuristic schedule which serves as an upper bound (UB). The cuts are successively added to the LP formulation. This process shrinks the feasible region and yields increasingly tighter bounds on the value of the optimal schedules. The iteration terminates when either the current UB equals LB, (hence the corresponding schedule is optimal) or when the LP solution cannot be cut off by any other available cuts. At the third stage, a heuristic is applied to the incumbent schedule for possible improvement.
(if it is not proven optimal), and then a final solution report is produced. For a general discussion on cutting plane methods, the reader is referred to the text by Nemhauser and Wolsey [45].

3.2.1 Initial Formulation

The proper choice of an initial formulation is important in cutting plane algorithms. We first include all the following simple facets (3.1) and (3.2):

\[ \min \sum_{j \in N} w_j c_j \]
\[ \text{st. } c_j \geq p_j, \quad \text{all } \{j\} \text{ initial} \]
\[ c_j - c_i \geq p_j, \quad \text{all nonredundant } (i,j). \]

The inclusion of all these facets ensures that the LP solutions obtained from stage 2 induce feasible sequences.

Computational experiments have shown that this initial formulation is satisfactory in most cases, except for problem instances with very few precedence relations. From our computational experience, we found that, when only a few precedence constraints are present, the initial formulation yields very weak bounds. As a result, too many cuts, and thus excessive computation time, are needed to reduce the gap between UB and LB.

To speed up the computation for these sparsely precedence-constrained problems, we extend the initial formulation by including an additional set of cuts (most of them known to be facet-defining). The set of cuts consists of \( n - 1 \) parallel inequalities obtained as follows. We first use a heuristic method to find a reasonably good feasible sequence, say, \((\pi(1), \pi(2), \ldots, \pi(n))\), see Section 3.2.3 for details. Let \( S_i = \{\pi(1), \pi(2), \ldots, \pi(i)\}, \) \( i = 2, 3, \ldots, n \). Note that these sets \( S_i \) necessarily induce initial sets. Then, we add to the original initial formulation the following parallel inequalities:

\[ p \ast C(S_i) \geq g(S_i), \quad i = 1, 2, \ldots, n. \]
Recall from Chapter 2 that the inequality induced by set $S_i$ is facet-defining, unless the set $S_i$ is series-decomposable. The computational experience confirms the effectiveness of this method for sparse precedence constraints.

### 3.2.2 Cutting Planes and Separation Algorithms

Two types of cuts are used in the separation routine: parallel cuts and simple series cuts. Recall that the parallel cuts take on the following form:

$$p * C(S) \geq g(S), \quad \text{where } S \text{ is a nonempty initial set.} \quad (3.3)$$

Queyranne [52] proposed an $O(n \log n)$ separation procedure for the scheduling problem with no precedence constraints. Note that the procedure is still valid for the precedence constrained scheduling problems: the LP solution at each iteration necessarily induces a feasible sequence, because of the inclusion of all non-redundant precedence constraints in the initial formulation. Thus, any set $S = \{\pi(1), \cdots, \pi(i)\}$ identified by the algorithm below is necessarily an initial set.

**Parallel Cut Separation:**

**INPUT:** Integer $n$; processing time vector $p$; completion time vector $C$.

**RESULT:** a subset $S \subseteq \{1, \ldots, n\}$ maximizing the violation $\Gamma(S) = g(S) - p * C(S)$ of (3.3).

**begin**

1. Sort $N = \{1, \ldots, n\}$ as $N = (\pi(1), \ldots, \pi(n))$ in nondecreasing order of the $C_j$ coefficients;
2. Let $S' := \emptyset$; $\rho := 0$; $\gamma' := 0$; $\gamma := 0$; $S := \emptyset$;
3. For $i := 1$ to $n$ do
   **begin**
   $k := \pi(i)$; $S' := S' \cup \{k\}$; $\rho := \rho + p_k$; $\gamma' := \gamma' + p_k(\rho - C_k)$;
   if $\gamma' > \gamma$ then begin $\gamma := \gamma'$; $S := S'$ end
   **end**

**end.**

Note that in steps 2 and 3, $\rho = P(S')$ and $\gamma' = \Gamma(S')$. Note also that except for
the $O(n \log(n))$ sort in Step 1, the algorithm runs in linear time. See Queyranne [52] for details.

We now relate the lower bound obtained after adding all (relevant) parallel cuts, to the lower bound obtained by Van de Velde [56]. We may write the precedence constrained scheduling problem as follows.

$$Z_{opt} = \min \sum_{j \in N} w_j C_j$$

$$\text{st. } C_j - C_i \geq p_j, \quad \text{all nonredundant } (i, j)$$

$$C \in Q$$

where $Q$ is the set of all feasible schedules.

Let $B$ be the arc-node incidence matrix of $G = (N, A(N))$. Consider the following linear program:

$$LB_P = \min wc$$

$$\text{st. } BC \geq b$$

$$p \cdot C(S) \geq g(S), \quad \text{all } S \subseteq N$$

where $b$ is an appropriate right-hand-side, so (3.8) is just (3.5) written in matrix form. As (3.9) is a relaxation of (3.6) (cf. Queyranne [52]), problem (3.7)–(3.9) is a relaxation of problem (3.4)–(3.6). So $LB_P \leq Z_{opt}$. In effect, $LB_P$ is precisely the lower bound obtained by the above cutting plane method after all parallel cuts are exhausted.

Van de Velde's lower bound is obtained by relaxing precedence constraints (3.5) with Lagrangian multipliers $\lambda_{ij} \geq 0$. For any vector $\lambda \geq 0$ of such Lagrangian multipliers, the subproblem in this approach is the following unconstrained scheduling problem

$$\min \{ wc + \lambda (b - BC) : C \in Q \} = \min \{(w - \lambda B)C : C \in Q \} - \lambda b,$$

which is solved by Smith's rule in $O(n \log n)$ computational steps [54]. The best possible lower bound $LB_R$, which can be obtained using this approach, is the optimum value of
Chapter 3. Single Machine Scheduling: Polyhedral Computations

the following problem:

\[
LB_R = \max_{\lambda \geq 0} \left\{ \min \{ wC + \lambda(b - BC) : C \in Q \} \right\}
\]

(3.11)

\[
= \max_{\lambda \geq 0} \left\{ \min \{ wC + \lambda(b - BC) : C \in \text{conv } Q \} \right\}
\]

(3.12)

\[
= \max_{\lambda \geq 0} \left\{ \min \{ wC + \lambda(b - BC) : p \ast C(S) \geq g(S), \forall S \subseteq N \} \right\}
\]

(3.13)

\[
= \min \{ wC : BC \geq b, p \ast C(S) \geq g(S), \forall S \subseteq N \}
\]

(3.14)

where (3.12) follows since the objective function in (3.11) is linear in \(C\); (3.13) follows from Queyranne [52]; (3.14) follows from the Geoffrion’s theory of Lagrangian Relaxation [24]. Thus, we have shown the following result:

**Theorem 3.1** The optimal lower bound in Van de Velde’s Lagrangian Relaxation approach is equal to the lower bound \(LB_P\) obtained by the cutting plane approach using only the parallel cuts.

Note that Van de Velde does not solve (3.11) exactly, but only approximately using an ascent method for improving the Lagrangian multipliers. Therefore, his lower bound is generally weaker than \(LB_P\).

The second type of cuts is a subclass of series cuts, the simple series cuts, where one of the subsets, \(S_1\) or \(S_2\) in \(S_1 \rightarrow S_2\), is a singleton. So, simple series cuts are of the form

\[
-p(S_2)C_u + p \ast C(S_2) \geq g(S_2), \quad \text{where } \{u\} \rightarrow S_2
\]

(3.15)

or

\[
-p \ast C(S_1) + p(S_1)C_v \geq g'(S_1 \cup \{v\}), \quad \text{where } S_1 \rightarrow \{v\}
\]

(3.16)

For simplicity, we denote the simple series inequalities by \(\Gamma(S_1, S_2) \leq 0\), where \(S_1\) (resp., \(S_2\)) is singleton if it is the cut (3.15) (resp., (3.16).), and \(\Gamma(S_1, S_2)\) denotes the amount by which the cut is violated.
We now develop an $O(n^2)$ separation procedure for the series cuts. To do so, we rewrite the above inequalities in the following equivalent forms: for (3.15),

$$
\sum_{j \in S_2} p_j(C_j - C_u) \geq g(S_2), \quad \text{where } \{u\} \rightarrow S_2,
$$

(3.17)

and, for (3.16),

$$
\sum_{i \in S_1} p_i(t_v - t_i) \geq g(S_1), \quad \text{where } S_1 \rightarrow \{v\}.
$$

(3.18)

where $t_i = C_i - p_i$ is the start time of job $i \in N$.

Inequality (3.17) can be viewed as a parallel inequality with respect to redefined "completion times" $C_j - C_u$ and restricted to those jobs $j$ that are successors of given job $u$. For inequality (3.18), one may imagine that all jobs in $S_1$ start backward from time $t_v$, and thus have "completion times" $t_v - t_i$. Therefore, it is also in the form of a parallel cut, restricted to the set of jobs that are predecessors of job $v$. For convenience, we call $\{u\} \rightarrow S_2$ a fan-out structure, and $S_1 \rightarrow \{v\}$ a fan-in structure.

A simple extension of the Parallel Cut Separation leads to a separation procedure for the simple series cuts. For every $u = \pi(1), \pi(2), \ldots, \pi(n - 2)$, apply the Parallel Cut Separation to the set of jobs that are successors of $u$, using $C_j - C_u$ as completion times. This produces a fan-out structure that maximizes the violation of (3.17). A similar procedure applies to (3.18). In the end, a most violated simple series cut is identified. Thus, finding a most violated simple series inequality of either fan-in or fan-out structure requires $O(n^2)$ computational steps. This Simple Series Cut Separation algorithm is described in details on the following page (Recall that $\bar{A}(N)$ denotes the transitive closure of $A(N)$).
Simple Series Cut Separation:
INPUT: Integer \( n \); processing time vector \( p \); completion time vector \( C \).
RESULT: Find disjoint subsets \( S_1 \) and \( S_2 \), such that \( S_1 \rightarrow S_2 \),
  one of them is a singleton set, and \( \Gamma(S_1,S_2) \) is maximized.

\[
\begin{align*}
\text{begin} \\
1. & \quad \text{Sort the job set as } N = (\pi(1), \ldots, \pi(n)) \\
   & \quad \text{in nondecreasing order of the completion times } C_j; \\
   & \quad \text{and as } N = (\pi'(1), \ldots, \pi'(n)) \\
   & \quad \text{in nonincreasing order of the start times } t_j; \\
2. & \quad \text{[ FORWARD SEARCH ]} \\
   & \quad \text{Let } S' := \emptyset; \rho := 0; \gamma' := 0; \gamma_1 := 0; S_2 := \emptyset; \\
3. & \quad \text{For } h := 1 \text{ to } n - 2 \text{ do} \\
   & \quad \quad \text{For } j := h + 1 \text{ to } n \text{ do} \\
   & \quad \quad \quad \text{if } (\pi(h), \pi(j)) \in \tilde{A}(N) \text{ then} \\
   & \quad \quad \quad \quad \text{begin} \\
   & \quad \quad \quad \quad \quad \quad i := \pi(h); k := \pi(j); S' := S' \cup \{k\}; \\
   & \quad \quad \quad \quad \quad \quad \rho := \rho + p_k; \gamma' := \gamma' + p_k(\rho - C_k + C_i); \\
   & \quad \quad \quad \quad \quad \quad \text{if } \gamma' > \gamma_1 \text{ then begin } \gamma_1 := \gamma'; S_2 := S', u := i \text{ end} \\
   & \quad \quad \quad \quad \text{end}; \\
4. & \quad \text{[ BACKWARD SEARCH ]} \\
   & \quad \text{Let } S' := \emptyset; \rho := 0; \gamma' := 0; \gamma_2 := 0; S_1 := \emptyset; \\
5. & \quad \text{For } h := n \text{ step } -1 \text{ to } 2 \text{ do} \\
   & \quad \quad \text{For } j := h - 1 \text{ step } -1 \text{ to } 1 \text{ do} \\
   & \quad \quad \quad \text{if } (\pi(j), \pi(h)) \in \tilde{A}(N) \text{ then} \\
   & \quad \quad \quad \quad \text{begin} \\
   & \quad \quad \quad \quad \quad \quad i := \pi'(h); k := \pi'(j); S' := S' \cup \{k\}; \\
   & \quad \quad \quad \quad \quad \quad \rho := \rho + p_k; \gamma' := \gamma' + p_k(\rho - t_i + t_k); \\
   & \quad \quad \quad \quad \quad \quad \text{if } \gamma' > \gamma_2 \text{ then begin } \gamma_2 := \gamma'; S_1 := S', v := i \text{ end} \\
   & \quad \quad \quad \quad \text{end} \\
6. & \quad \text{Retrieve inequality } \Gamma(S_1, S_2) \leq 0 \text{ from} \\
   & \quad (3) \text{ if } \gamma_1 > \gamma_2 \text{ and from } (5) \text{ otherwise.} \\
\end{align*}
\]

end.
3.2.3 Heuristics for the Upper Bounds

Cutting plane algorithms often allow the easy derivation of good feasible solutions. This is the case for the precedence-constrained scheduling problem studied here. The following heuristic procedures are used in our algorithm.

The first heuristic is a procedure for obtaining an initial feasible sequence. Recall that this initial sequence is used to establish a tighter LP formulation than the original one. The heuristic is constructive, starting with the whole job set $S = N$ and the given precedence graph $G$. Then, repeat the following process until the set $S$ is empty: remove from $S$ an initial job $i$ (relative to current graph $G$) with largest ratio $w_i/p_i$; update $S \leftarrow S \setminus \{i\}$ and $G$. The sequence in which the jobs have been removed is, by construction, a feasible sequence.

This simple heuristic has proven to be effective when the precedence graph is sparse. In particular, it yields an optimal schedule when $A(N)$ is empty (Smith [54]). There are other heuristics for obtaining feasible schedules, such as Morton and Dharan’s Tree Optimal heuristic [39]. Sophisticated heuristics may not be very useful in our algorithm. We briefly tested the Tree Optimal heuristic solution against the feasible schedules induced by the LP solutions at stage 2 on several 100-job problem instances. We found that the cutting plane procedure consistently generated better feasible schedules, and generally used less CPU time, than the Tree Optimal heuristic.

The second heuristic derives a feasible sequence from an LP solution by employing a so-called priority rule. At each iteration, that is, when a cut is added, an optimal solution $C$ to the current LP relaxation is obtained. Since all precedence constraints are included in the initial LP formulation, the solution $C$ necessarily induces a feasible sequence in which job $i$ is processed before job $j$ whenever $C_i \leq C_j$.

We then try to improve the feasible sequence produced by the above heuristic. This
involves 1-OPT type interchanges. First, we do a backward search: for \( i = n - 1, \ldots, 2, 1 \), find a set \( S \) of jobs immediately following, but not constrained by, job \( i \) and with largest ratio \( w(S)/p(S) \); if the largest ratio \( w(S)/p(S) \) is larger than \( w_i/p_i \), then update the sequence by interchanging \( i \) and \( S \). We then perform a similar forward search, trying to exchange job \( j \) with a set of jobs immediately preceding it. We call this procedure One-Pass 1-OPT. A Full 1-OPT is to repeat One-Pass 1-OPT until no further improvement can be made.

### 3.2.4 The Cutting Plane Procedure

The whole cutting plane procedure is summarized as follows.

**Step 0:** Read data. Start the timing subroutine.

**Step 1:** Compute the transitive reduction and transitive closure of the precedence digraph. (The former is used for setting up an initial formulation, and the latter for the simple series cut separation.)

**Step 2:** Set up the initial LP formulation and solve this LP problem.

**Step 3:**

(a) Generate a heuristic feasible schedule from the current LP solution; store the schedule if its cost is the best found so far; go to Step 5 if UB=LB, where UB is the cost of the best schedule found so far, and LB is the cost of the current LP solution.

(b) Apply the Parallel Cut Separation.

Go to (d) if a violated parallel cut is found;

(c) Apply the Simple Series Cut Separation.

Go to Step 4 if no violated series cut is found;

(d) Add the violated cut and reoptimize the extended LP to update the LP solution and its cost LB; Go to (a).

**Step 4:** Apply the Full 1-OPT heuristic to the best feasible schedule for final improvement;

**Step 5:** Stop the timing subroutine; Produce a solution report; Halt.

Note that **Step 3** is an iterative phase, the most time-consuming part of the algorithm.
The algorithm, written in FORTRAN, uses the XMP Linear Programming Library as the LP optimizer in Steps 2 and 3. To facilitate the use of the subroutines in XMP, an equivalent *start time* formulation is used. Since start times $t_j = C_j - p_j$, for all $j$, the formulation replaces all trivial constraints $C_j \geq p_j$ by the nonnegativity constraints on all $t_j$ variables. The solution report, however, is presented in terms of completion times for comparison with other solution methods. At the present, the algorithm is coded for problems of size up to 400 jobs. Extending the problem size requires changing only a few statements.

### 3.3 Computational Experience

In this section, we report our computational results on a fairly large sample of randomly generated problems. We evaluate the algorithm based on the computation time, and the *percent gap* $r = 100 (UB - LB)/LB$. For example, if a feasible schedule $C$ with value $UB$ satisfies $r \leq 0.3$, then it is guaranteed, even without knowing the optimal value $Z^opt$, that the schedule cannot deviate from the optimal schedule by more than 0.3 percent, since

$$(UB - Z^{opt})/Z^{opt} \leq (UB - LB)/LB \leq 0.003.$$  

To illustrate how the cutting plane procedure works, we begin with two simple numerical examples.
EXAMPLE 1. Consider Potts' 10-job example [50].

<table>
<thead>
<tr>
<th>Job i</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_i )</td>
<td>6</td>
<td>9</td>
<td>1</td>
<td>3</td>
<td>9</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>( w_i )</td>
<td>2</td>
<td>5</td>
<td>9</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>9</td>
<td>3</td>
<td>8</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 3.1: Data for the 10-job example

The precedence relations can be represented by the following nonredundant arc set

\[
A(N) = \{(1, 6), (1, 7), (2, 4), (2, 5), (3, 8), (5, 9), (6, 10), (7, 9), (8, 10)\}.
\]

After solving the initial LP program, which includes all constraints of types (3.1) and (3.2), an LP solution of value 1526 is obtained. In addition, the heuristic gives a feasible schedule of value 1530. Table 3.2 summarizes the execution of the algorithm, where \( i \) denotes the iteration index in Step 3, and the second column represents the cut type, "P" for a parallel cut and "SS" for a simple series cut.

<table>
<thead>
<tr>
<th>( i )</th>
<th>Cut type</th>
<th>Violated facet cut</th>
<th>LB</th>
<th>UB</th>
<th>% Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I</td>
<td>( 6C_1 + 9C_2 + C_3 + 3C_4 + 9C_5 + 7C_7 + 6C_9 \geq 987 )</td>
<td>1526</td>
<td>1530</td>
<td>4.</td>
</tr>
<tr>
<td>2</td>
<td>P</td>
<td>( 9C_3 + C_4 + 3C_4 \geq 130 )</td>
<td>1526.7</td>
<td>1530.0</td>
<td>3.3</td>
</tr>
<tr>
<td>3</td>
<td>SS</td>
<td>( -6C_1 - 9C_2 - 9C_5 - 7C_7 + 31C_9 \geq 543 )</td>
<td>1527.2</td>
<td>1530.0</td>
<td>2.8</td>
</tr>
<tr>
<td>4</td>
<td>SS</td>
<td>( -6C_5 - 8C_6 - 7C_8 + 18C_{10} \geq 143 )</td>
<td>1527.5</td>
<td>1530.0</td>
<td>2.5</td>
</tr>
<tr>
<td>5</td>
<td>SS</td>
<td>( -6C_1 - 8C_5 - 7C_7 + 22C_9 \geq 291 )</td>
<td>1528.0</td>
<td>1530.0</td>
<td>2.</td>
</tr>
<tr>
<td>6</td>
<td>SS</td>
<td>( -5C_6 - 7C_8 + 12C_{10} \geq 59 )</td>
<td>1530.0</td>
<td>1530.0</td>
<td>0.</td>
</tr>
</tbody>
</table>

Table 3.2: Cutting planes and the corresponding solutions

Observe that after adding two parallel cuts and four series cuts, the problem is solved to optimality without any branch and bound. Note also that Potts' lower bound, obtained after sixteen iterations, is 1516, weaker than our initial LP value 1526.
EXAMPLE 2. Consider Wolsey’s 30-job example [57].

<table>
<thead>
<tr>
<th>Job i</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
</tr>
<tr>
<td>(p_i)</td>
<td>57</td>
<td>45</td>
<td>50</td>
<td>76</td>
<td>73</td>
<td>82</td>
<td>51</td>
<td>19</td>
<td>82</td>
<td>94</td>
<td>98</td>
<td>15</td>
<td>52</td>
<td>78</td>
<td>75</td>
</tr>
<tr>
<td>(w_i)</td>
<td>4</td>
<td>44</td>
<td>89</td>
<td>80</td>
<td>74</td>
<td>43</td>
<td>9</td>
<td>32</td>
<td>2</td>
<td>43</td>
<td>56</td>
<td>27</td>
<td>45</td>
<td>68</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>3</td>
<td>7</td>
<td>1</td>
<td>6</td>
<td>8</td>
<td>6</td>
<td>4</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>2</td>
<td>5</td>
<td>9</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.3: Data for the 30-job problem

The nonredundant precedence relations can be represented by the following arc set.

\[
A(N) = \{ (1, 2), (2, 6), (2, 7), (3, 8), (3, 9), (3, 13), (4, 7), (4, 9), (5, 8), (5, 19), (6, 9), (6, 12), (7, 8), (7, 10), (7, 13), (8, 11), (9, 18), (9, 25), (10, 12), (11, 15), (11, 16), (11, 17), (12, 17), (12, 25), (13, 16), (13, 17), (13, 19), (14, 20), (14, 21), (14, 23), (15, 18), (15, 20), (17, 18), (17, 22), (18, 21), (18, 23), (19, 20), (19, 21), (19, 28), (21, 27), (21, 29), (21, 30), (22, 26), (23, 24), (24, 28), (24, 30), (25, 28), (26, 28), (26, 29), (26, 30) \}

Table 3.4 gives a summary of the solution report for each iteration. The third column represents the cumulative computation time (IDM 386/25). Note that the initial lower bound (LB) is 56,495.18. After five parallel cuts were added, no additional violated parallel cut is found, and the current LB of 119,329.04 is the best possible lower bound \(LBR\) that can be obtained by Van de Velde’s Lagrangian relaxation approach. The algorithm exhausted all parallel and simple series cuts at iteration 31, with a total of 6 parallel cuts and 25 series cuts. At this point, \(LB = 121,031.9\) and \(UB = 121,757\) with the percent gap below 0.6 percent. The cumulative computation time is 12.74 seconds.
### Chapter 3. Single Machine Scheduling: Polyhedral Computations

#### Table 3.4: Solutions from the cutting plane procedure

<table>
<thead>
<tr>
<th>Iter</th>
<th>Cut type</th>
<th>Cumul.time</th>
<th>LB</th>
<th>UB</th>
<th>% Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>1.37</td>
<td>56495.1757</td>
<td>123568</td>
<td>118.72310</td>
</tr>
<tr>
<td>1</td>
<td>P</td>
<td>1.53</td>
<td>71642.9524</td>
<td>123568</td>
<td>72.47754</td>
</tr>
<tr>
<td>2</td>
<td>P</td>
<td>1.81</td>
<td>103696.0370</td>
<td>121799</td>
<td>17.45772</td>
</tr>
<tr>
<td>3</td>
<td>P</td>
<td>2.52</td>
<td>116799.2382</td>
<td>121799</td>
<td>4.28065</td>
</tr>
<tr>
<td>4</td>
<td>P</td>
<td>2.85</td>
<td>119231.7078</td>
<td>121799</td>
<td>2.15320</td>
</tr>
<tr>
<td>5</td>
<td>P</td>
<td>3.02</td>
<td>119329.0365</td>
<td>121799</td>
<td>2.06988</td>
</tr>
<tr>
<td>6</td>
<td>SS</td>
<td>3.35</td>
<td>119422.0401</td>
<td>121799</td>
<td>1.99039</td>
</tr>
<tr>
<td>7</td>
<td>SS</td>
<td>3.68</td>
<td>119611.9931</td>
<td>121799</td>
<td>1.82842</td>
</tr>
<tr>
<td>8</td>
<td>SS</td>
<td>3.95</td>
<td>119623.1333</td>
<td>121799</td>
<td>1.81893</td>
</tr>
<tr>
<td>9</td>
<td>SS</td>
<td>4.28</td>
<td>119703.7625</td>
<td>121799</td>
<td>1.75035</td>
</tr>
<tr>
<td>10</td>
<td>SS</td>
<td>4.66</td>
<td>119867.9137</td>
<td>121799</td>
<td>1.61101</td>
</tr>
<tr>
<td>11</td>
<td>SS</td>
<td>4.99</td>
<td>119959.1578</td>
<td>121799</td>
<td>1.53372</td>
</tr>
<tr>
<td>12</td>
<td>SS</td>
<td>5.32</td>
<td>120062.2191</td>
<td>121799</td>
<td>1.44657</td>
</tr>
<tr>
<td>13</td>
<td>SS</td>
<td>5.65</td>
<td>120364.2442</td>
<td>121799</td>
<td>1.19201</td>
</tr>
<tr>
<td>14</td>
<td>P</td>
<td>5.87</td>
<td>120425.5471</td>
<td>121799</td>
<td>1.14050</td>
</tr>
<tr>
<td>15</td>
<td>SS</td>
<td>6.31</td>
<td>120638.8663</td>
<td>121799</td>
<td>0.96166</td>
</tr>
<tr>
<td>16</td>
<td>SS</td>
<td>6.64</td>
<td>120674.8199</td>
<td>121799</td>
<td>0.93158</td>
</tr>
<tr>
<td>17</td>
<td>SS</td>
<td>6.97</td>
<td>120811.7984</td>
<td>121799</td>
<td>0.81714</td>
</tr>
<tr>
<td>18</td>
<td>SS</td>
<td>7.30</td>
<td>120848.5473</td>
<td>121799</td>
<td>0.78648</td>
</tr>
<tr>
<td>19</td>
<td>SS</td>
<td>7.69</td>
<td>120881.8021</td>
<td>121757</td>
<td>0.72401</td>
</tr>
<tr>
<td>20</td>
<td>SS</td>
<td>8.07</td>
<td>120904.6679</td>
<td>121757</td>
<td>0.70496</td>
</tr>
<tr>
<td>21</td>
<td>SS</td>
<td>8.40</td>
<td>120914.2967</td>
<td>121757</td>
<td>0.69694</td>
</tr>
<tr>
<td>22</td>
<td>SS</td>
<td>8.78</td>
<td>120932.7068</td>
<td>121757</td>
<td>0.68161</td>
</tr>
<tr>
<td>23</td>
<td>SS</td>
<td>9.11</td>
<td>120941.9816</td>
<td>121757</td>
<td>0.67389</td>
</tr>
<tr>
<td>24</td>
<td>SS</td>
<td>9.50</td>
<td>120955.3635</td>
<td>121757</td>
<td>0.66275</td>
</tr>
<tr>
<td>25</td>
<td>SS</td>
<td>9.88</td>
<td>120981.0002</td>
<td>121757</td>
<td>0.64142</td>
</tr>
<tr>
<td>26</td>
<td>SS</td>
<td>10.27</td>
<td>121001.3894</td>
<td>121757</td>
<td>0.62446</td>
</tr>
<tr>
<td>27</td>
<td>SS</td>
<td>10.60</td>
<td>121017.5399</td>
<td>121757</td>
<td>0.61104</td>
</tr>
<tr>
<td>28</td>
<td>SS</td>
<td>10.98</td>
<td>121026.4292</td>
<td>121757</td>
<td>0.60365</td>
</tr>
<tr>
<td>29</td>
<td>SS</td>
<td>11.53</td>
<td>121031.6281</td>
<td>121757</td>
<td>0.59932</td>
</tr>
<tr>
<td>30</td>
<td>SS</td>
<td>11.92</td>
<td>121031.7969</td>
<td>121757</td>
<td>0.59918</td>
</tr>
<tr>
<td>31</td>
<td>SS</td>
<td>12.30</td>
<td>121031.8770</td>
<td>121757</td>
<td>0.59912</td>
</tr>
</tbody>
</table>
Table 3.5: The optimal schedule and the relevant solutions

<table>
<thead>
<tr>
<th>Job</th>
<th>Schedule (optimal)</th>
<th>Schedule (at iter 31)</th>
<th>LP Solution (final)</th>
<th>LP Solution (at iter 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>57</td>
<td>57</td>
<td>57.0000</td>
<td>57.0000</td>
</tr>
<tr>
<td>3</td>
<td>107</td>
<td>107</td>
<td>107.0000</td>
<td>107.0000</td>
</tr>
<tr>
<td>2</td>
<td>152</td>
<td>152</td>
<td>234.1205</td>
<td>382.1086</td>
</tr>
<tr>
<td>6</td>
<td>234</td>
<td>234</td>
<td>443.1205</td>
<td>527.1086</td>
</tr>
<tr>
<td>4</td>
<td>310</td>
<td>310</td>
<td>310.1205</td>
<td>382.1086</td>
</tr>
<tr>
<td>7</td>
<td>361</td>
<td>361</td>
<td>361.1205</td>
<td>433.1086</td>
</tr>
<tr>
<td>13</td>
<td>413</td>
<td>413</td>
<td>413.1205</td>
<td>485.1086</td>
</tr>
<tr>
<td>10</td>
<td>507</td>
<td>599</td>
<td>537.1205</td>
<td>527.1086</td>
</tr>
<tr>
<td>12</td>
<td>522</td>
<td>614</td>
<td>552.1205</td>
<td>542.1086</td>
</tr>
<tr>
<td>5</td>
<td>595</td>
<td>486</td>
<td>434.1205</td>
<td>433.1086</td>
</tr>
<tr>
<td>8</td>
<td>614</td>
<td>505</td>
<td>453.1205</td>
<td>452.1086</td>
</tr>
<tr>
<td>11</td>
<td>712</td>
<td>712</td>
<td>623.4538</td>
<td>550.1086</td>
</tr>
<tr>
<td>16</td>
<td>716</td>
<td>716</td>
<td>627.4538</td>
<td>554.1086</td>
</tr>
<tr>
<td>17</td>
<td>760</td>
<td>760</td>
<td>757.1067</td>
<td>594.1086</td>
</tr>
<tr>
<td>22</td>
<td>769</td>
<td>769</td>
<td>766.1067</td>
<td>603.1086</td>
</tr>
<tr>
<td>26</td>
<td>825</td>
<td>825</td>
<td>822.1067</td>
<td>659.1086</td>
</tr>
<tr>
<td>9</td>
<td>907</td>
<td>907</td>
<td>963.5766</td>
<td>1092.1319</td>
</tr>
<tr>
<td>25</td>
<td>950</td>
<td>950</td>
<td>1006.3766</td>
<td>1135.1319</td>
</tr>
<tr>
<td>14</td>
<td>1028</td>
<td>1028</td>
<td>1041.5766</td>
<td>1181.1319</td>
</tr>
<tr>
<td>15</td>
<td>1103</td>
<td>1103</td>
<td>1116.5766</td>
<td>1092.1319</td>
</tr>
<tr>
<td>18</td>
<td>1192</td>
<td>1192</td>
<td>1205.5766</td>
<td>1181.1319</td>
</tr>
<tr>
<td>23</td>
<td>1224</td>
<td>1224</td>
<td>1237.5766</td>
<td>1213.1319</td>
</tr>
<tr>
<td>24</td>
<td>1226</td>
<td>1226</td>
<td>1239.5766</td>
<td>1215.1319</td>
</tr>
<tr>
<td>19</td>
<td>1306</td>
<td>1306</td>
<td>1260.1559</td>
<td>1181.1319</td>
</tr>
<tr>
<td>21</td>
<td>1349</td>
<td>1349</td>
<td>1323.5428</td>
<td>1224.1319</td>
</tr>
<tr>
<td>29</td>
<td>1417</td>
<td>1417</td>
<td>1391.5428</td>
<td>1292.1319</td>
</tr>
<tr>
<td>28</td>
<td>1462</td>
<td>1462</td>
<td>1365.8683</td>
<td>1260.1319</td>
</tr>
<tr>
<td>27</td>
<td>1489</td>
<td>1489</td>
<td>1489.0000</td>
<td>1489.0000</td>
</tr>
<tr>
<td>30</td>
<td>1552</td>
<td>1552</td>
<td>1552.0000</td>
<td>1552.0000</td>
</tr>
<tr>
<td>20</td>
<td>1626</td>
<td>1626</td>
<td>1626.0000</td>
<td>1626.0000</td>
</tr>
</tbody>
</table>
At the final upper bound improvement stage, we first applied a Full 1-OPT but failed to improve the upper bound. Then we tried to interchange a 2-job subsequence and its adjacent subsequence (cf. Table 3.5, the interchange of subsequences 5,8 and 10,12), and obtained a feasible schedule with the objective value 121,559, and 113,214 with respect to start times. This has been shown to be optimal by Wolsey [57]. Relative to 121,559, the percent gaps of Van de Velde’s lower bound and our lower bound are about 1.9% and 0.47%, respectively. The optimal schedule as well as relevant heuristic and LP solutions are given in Table 3.5. Note that the completion times for the first two jobs and the last three jobs in Table 3.5 are identical for all columns. This may suggest a possible decomposition method for further improving the algorithm.

We now test the algorithm on a sample of randomly generated problems of size n, ranging from n = 30 to 160 jobs. For each job i, an integer job processing time \( p_i \) was obtained from the uniform distribution \([1,100] \), and an integer weight \( w_i \) from the uniform distribution \([1,10] \). The precedence graph \( G \) was induced by specifying the probability \( P \) with which each arc \((i,j)\) with \( i < j \) was included. For each \( n \), we generated two problems for each of the \( P \) values .001, .02, .04, .06, .08, .10, .15, .20, .30 and .50. In total, this produces 280 test problems.

Table 3.6 presents a summary of the computational results. For each \( n \), the second column (ACT) and the third column (MCT) provide respectively the average and maximum computation time, in seconds, over 20 problems. The fourth column (OPT) represents the number of problems (out of 20) which were solved to proven optimality (i.e., when \( LB = UB \)). In total, 47 out of 280 problems were provably solved to optimality. The fifth column (AGAP) gives the average percent gap. The sixth column (MGAP) gives the maximum percent gaps and the last column gives the value of \( P \) for the problem giving this value MGAP.
Chapter 3. Single Machine Scheduling: Polyhedral Computations

Table 3.6: A summary of the results for 280 problems

<table>
<thead>
<tr>
<th>n</th>
<th>ACT</th>
<th>MCT</th>
<th># OPT</th>
<th>AGAP</th>
<th>MGAP</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>6.94</td>
<td>14.06</td>
<td>10</td>
<td>.16</td>
<td>.95</td>
<td>.10</td>
</tr>
<tr>
<td>40</td>
<td>19.50</td>
<td>75.79</td>
<td>7</td>
<td>.10</td>
<td>.74</td>
<td>.30</td>
</tr>
<tr>
<td>50</td>
<td>36.08</td>
<td>72.61</td>
<td>4</td>
<td>.11</td>
<td>.62</td>
<td>.15</td>
</tr>
<tr>
<td>60</td>
<td>65.00</td>
<td>133.19</td>
<td>4</td>
<td>.18</td>
<td>.54</td>
<td>.15</td>
</tr>
<tr>
<td>70</td>
<td>109.46</td>
<td>214.05</td>
<td>2</td>
<td>.20</td>
<td>.61</td>
<td>.15</td>
</tr>
<tr>
<td>80</td>
<td>129.68</td>
<td>246.56</td>
<td>2</td>
<td>.29</td>
<td>.53</td>
<td>.10</td>
</tr>
<tr>
<td>90</td>
<td>225.16</td>
<td>562.22</td>
<td>3</td>
<td>.23</td>
<td>.95</td>
<td>.10</td>
</tr>
<tr>
<td>100</td>
<td>290.33</td>
<td>530.80</td>
<td>2</td>
<td>.21</td>
<td>.51</td>
<td>.20</td>
</tr>
<tr>
<td>110</td>
<td>414.27</td>
<td>1614.70</td>
<td>3</td>
<td>.21</td>
<td>.88</td>
<td>.10</td>
</tr>
<tr>
<td>120</td>
<td>466.18</td>
<td>1449.15</td>
<td>2</td>
<td>.35</td>
<td>.96</td>
<td>.06</td>
</tr>
<tr>
<td>130</td>
<td>605.82</td>
<td>1888.78</td>
<td>2</td>
<td>.19</td>
<td>.49</td>
<td>.15</td>
</tr>
<tr>
<td>140</td>
<td>680.39</td>
<td>1769.81</td>
<td>2</td>
<td>.30</td>
<td>.87</td>
<td>.08</td>
</tr>
<tr>
<td>150</td>
<td>851.16</td>
<td>2074.26</td>
<td>2</td>
<td>.30</td>
<td>.67</td>
<td>.08</td>
</tr>
<tr>
<td>160</td>
<td>1137.32</td>
<td>3513.58</td>
<td>2</td>
<td>.29</td>
<td>.66</td>
<td>.15</td>
</tr>
</tbody>
</table>

Table 3.7 gives a comparison of our computational results with those of Van de Velde (1990). For each value of $P$, Van de Velde tested his Lagrangian relaxation method on a sample of 45 problems (5 problems each for every $n = 20, 30, \ldots, 100$). The resulting average percent gaps $AGAP^V$ are given in the second column of Table 3.7. We used a sample of 28 problems (2 problems each for every $n = 30, 40, \ldots, 160$). The third and fourth columns represent the average percent gaps and the maximum percent gaps, respectively, obtained by our cutting plane procedure. Clearly, the cutting plane approach outperforms the Lagrangian approach for all $P$ values in terms of the resulting percent gaps. (If Van de Velde's sample of problems was used, we expect even better results, because problems with $n$ less than 60 tend to yield smaller gaps for our approach, see Table 3.6.) These results are due to: (a) a much tighter bound from the cutting plane procedure, see Theorem 3.1; and (b) a more effective heuristic solution method. The last column provides the number of problems (out of 28) with the feasible solutions proven to be optimal.
Table 3.7: Computational results with respect to $P$ values

Potts [50] observed that problems with $P$ between $.04$ to $.1$ are more difficult for his approach. For our cutting plane procedure, the problem difficulty may depend on both the gap and the computation time. Our computational results in Table 3.7 show that the percent gaps are not very sensitive to $P$. Problems with $P$ between $.06$ to $.3$ have slightly larger percent gaps (up to $0.96\%$). On the other hand, more computation time is usually required for problems with $P$ values around $.02$. In terms of order strength, that is, the ratio of the number of arcs in the transitive closure to the maximum number $n(n - 1)/2$ of possible arcs, larger percent gaps tend to occur for problems with order strength between $.4$ and $.8$.

In Section 3.4, we present the detailed computational results for 280 problems. Computation Times (CT), in seconds, are given in column 7. In column 4, Ord.Sth. denotes the order strength of the precedence graph. Column 5 represents the number of arcs in the transitive reduction of the digraph. Column 6 indicates the number of cuts used in Step 3 of the cutting plane procedure. Note that for almost all of the instances with $P = 0.001$, only one cut has been used. This explains that our heuristic procedure in the initial formulation finds a very good basis when the precedence graph is very
sparse. (From the computational experiments, we found that those problems are most time-consuming if no heuristic is used in the initial formulation.)

This computational study shows that the cutting plane approach based on facet defining cuts is effective for solving the precedence constrained job sequencing problems. Because of the LP formulation with \( n \) variables and tight lower bounds obtained, the algorithm has potential for solving large-scale problems to optimality with the aid of a Branch and Bound or Branch and Cut procedure. To further improve the algorithm, the following are among the possible improvements:

- a better heuristic for feasible schedules, especially for the initial formulation;
- separation routines for more complex facet classes (see Chapter 2); and
- a Branch and Cut procedure at the final stage.
### 3.4 Computational Results for 280 problems

<table>
<thead>
<tr>
<th>#Jobs</th>
<th>Name</th>
<th>P</th>
<th>Ord. Sth.</th>
<th># Arcs</th>
<th># Cuts</th>
<th>CT</th>
<th>% Gap</th>
<th>Statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>1</td>
<td>.001</td>
<td>.00460</td>
<td>2</td>
<td>1</td>
<td>2.36</td>
<td>.00000</td>
<td>AGAP</td>
</tr>
<tr>
<td>2</td>
<td>.001</td>
<td>.00000</td>
<td>0</td>
<td>1</td>
<td>2.42</td>
<td>.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>.020</td>
<td>.03678</td>
<td>11</td>
<td>2</td>
<td>3.13</td>
<td>.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>.020</td>
<td>.03448</td>
<td>9</td>
<td>1</td>
<td>2.36</td>
<td>.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>.040</td>
<td>.07356</td>
<td>22</td>
<td>2</td>
<td>3.57</td>
<td>.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>.040</td>
<td>.03218</td>
<td>10</td>
<td>2</td>
<td>2.97</td>
<td>.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>.060</td>
<td>.06667</td>
<td>23</td>
<td>4</td>
<td>3.74</td>
<td>.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>.060</td>
<td>.06207</td>
<td>21</td>
<td>7</td>
<td>4.95</td>
<td>.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>.080</td>
<td>.11954</td>
<td>29</td>
<td>6</td>
<td>4.62</td>
<td>.21058</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>.080</td>
<td>.14483</td>
<td>32</td>
<td>22</td>
<td>8.34</td>
<td>.03627</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>.100</td>
<td>.17931</td>
<td>34</td>
<td>24</td>
<td>9.01</td>
<td>.32752</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>.100</td>
<td>.30575</td>
<td>42</td>
<td>26</td>
<td>10.82</td>
<td>.95170</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>.150</td>
<td>.38621</td>
<td>50</td>
<td>24</td>
<td>10.16</td>
<td>.34220</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>.150</td>
<td>.48736</td>
<td>54</td>
<td>24</td>
<td>11.09</td>
<td>.26927</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>.200</td>
<td>.57701</td>
<td>52</td>
<td>16</td>
<td>6.26</td>
<td>.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>.200</td>
<td>.55632</td>
<td>45</td>
<td>18</td>
<td>9.40</td>
<td>.08065</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>.300</td>
<td>.62299</td>
<td>59</td>
<td>29</td>
<td>14.06</td>
<td>.41234</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>.300</td>
<td>.69425</td>
<td>54</td>
<td>20</td>
<td>9.72</td>
<td>.42745</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>.500</td>
<td>.88046</td>
<td>46</td>
<td>23</td>
<td>10.88</td>
<td>.12030</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>.500</td>
<td>.91035</td>
<td>45</td>
<td>20</td>
<td>8.90</td>
<td>.00000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| 40    | 1    | .001 | .00000 | 0     | 1     | 3.90  | .00000 | AGAP      |
| 2     | .001 | .00128 | 1     | 1     | 3.90  | .00000 |           |
| 3     | .020 | .03590 | 23    | 9     | 6.63  | .00000 |           |
| 4     | .020 | .01795 | 12    | 1     | 3.90  | .00000 |           |
| 5     | .040 | .05615 | 34    | 14    | 11.53 | .00462 |           |
| 6     | .040 | .04231 | 27    | 10    | 9.01  | .00000 |           |
| 7     | .060 | .18718 | 55    | 33    | 17.42 | .04461 |           |
| 8     | .060 | .14331 | 32    | 17    | 17.30 | .90125 |           |
| 9     | .080 | .12308 | 54    | 45    | 24.00 | .00000 |           |
| 10    | .080 | .24359 | 52    | 36    | 19.28 | .02760 |           |
| 11    | .100 | .30641 | 61    | 33    | 18.89 | .07638 |           |
| 12    | .100 | .21410 | 66    | 28    | 16.53 | .23652 |           |
| 13    | .150 | .42180 | 73    | 26    | 15.60 | .08634 |           |
| 14    | .150 | .53590 | 72    | 35    | 22.13 | .14889 |           |
| 15    | .200 | .63974 | 74    | 36    | 21.36 | .09571 |           |
| 16    | .200 | .56923 | 83    | 28    | 18.51 | .20624 |           |
| 17    | .300 | .78718 | 80    | 26    | 16.87 | .02686 |           |
| 18    | .300 | .79487 | 74    | 27    | 36.69 | .74286 |           |
| 19    | .500 | .90256 | 62    | 55    | 75.79 | .12927 |           |
| 20    | .500 | .93333 | 66    | 37    | 28.67 | .00000 |           |

Table 3.8: Computational results for \( n = 30, 40 \)
### Chapter 3. Single Machine Scheduling: Polyhedral Computations

#### Table 3.9: Computational results for $n = 50, 60$

<table>
<thead>
<tr>
<th># Jobs</th>
<th>Name</th>
<th>P</th>
<th>Ord.Sth.</th>
<th># Arcs</th>
<th># Cuts</th>
<th>CT</th>
<th>% Gap</th>
<th>Statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td></td>
<td>0.001</td>
<td>0.0000</td>
<td>0</td>
<td>1</td>
<td>5.71</td>
<td>0.0000</td>
<td>AGAP 0.1069</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.001</td>
<td>0.0082</td>
<td>1</td>
<td>1</td>
<td>5.93</td>
<td>0.0000</td>
<td>ACT 36.076</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.020</td>
<td>0.0212</td>
<td>20</td>
<td>1</td>
<td>6.26</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.020</td>
<td>0.0342</td>
<td>28</td>
<td>5</td>
<td>10.10</td>
<td>0.00646</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>0.040</td>
<td>0.0742</td>
<td>50</td>
<td>56</td>
<td>43.72</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>0.040</td>
<td>0.1240</td>
<td>62</td>
<td>43</td>
<td>29.44</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>0.060</td>
<td>0.1306</td>
<td>67</td>
<td>56</td>
<td>41.80</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>0.060</td>
<td>0.1379</td>
<td>65</td>
<td>56</td>
<td>41.80</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>0.080</td>
<td>0.2563</td>
<td>80</td>
<td>42</td>
<td>30.75</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>0.080</td>
<td>0.2253</td>
<td>76</td>
<td>49</td>
<td>36.74</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>0.100</td>
<td>0.2808</td>
<td>84</td>
<td>39</td>
<td>28.23</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>0.100</td>
<td>0.4008</td>
<td>74</td>
<td>47</td>
<td>36.47</td>
<td>0.15972</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>0.150</td>
<td>0.5485</td>
<td>108</td>
<td>51</td>
<td>42.62</td>
<td>0.25105</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>0.150</td>
<td>0.6293</td>
<td>83</td>
<td>58</td>
<td>51.63</td>
<td>0.62363</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>0.200</td>
<td>0.7371</td>
<td>87</td>
<td>55</td>
<td>49.10</td>
<td>0.29168</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>0.200</td>
<td>0.7061</td>
<td>99</td>
<td>48</td>
<td>44.10</td>
<td>0.25035</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>0.300</td>
<td>0.8391</td>
<td>79</td>
<td>30</td>
<td>26.20</td>
<td>0.01573</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td>0.300</td>
<td>0.8347</td>
<td>90</td>
<td>58</td>
<td>63.93</td>
<td>0.11547</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td>0.500</td>
<td>0.9551</td>
<td>67</td>
<td>44</td>
<td>72.61</td>
<td>0.02199</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>0.500</td>
<td>0.9502</td>
<td>75</td>
<td>48</td>
<td>52.89</td>
<td>0.00095</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td></td>
<td>0.001</td>
<td>0.0113</td>
<td>2</td>
<td>1</td>
<td>8.96</td>
<td>0.0000</td>
<td>AGAP 0.1754</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.001</td>
<td>0.0169</td>
<td>3</td>
<td>1</td>
<td>10.16</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.020</td>
<td>0.0649</td>
<td>46</td>
<td>26</td>
<td>36.47</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.020</td>
<td>0.0226</td>
<td>33</td>
<td>40</td>
<td>64.98</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>0.040</td>
<td>0.0632</td>
<td>64</td>
<td>66</td>
<td>63.39</td>
<td>0.00242</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>0.040</td>
<td>0.0836</td>
<td>70</td>
<td>73</td>
<td>69.70</td>
<td>0.02342</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>0.060</td>
<td>0.1920</td>
<td>88</td>
<td>55</td>
<td>50.69</td>
<td>0.11333</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>0.060</td>
<td>0.2271</td>
<td>96</td>
<td>72</td>
<td>69.04</td>
<td>0.47417</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>0.080</td>
<td>0.3209</td>
<td>94</td>
<td>67</td>
<td>65.36</td>
<td>0.14805</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>0.080</td>
<td>0.3180</td>
<td>107</td>
<td>52</td>
<td>53.28</td>
<td>0.16912</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>0.100</td>
<td>0.4937</td>
<td>118</td>
<td>63</td>
<td>69.32</td>
<td>0.36744</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>0.100</td>
<td>0.4435</td>
<td>113</td>
<td>54</td>
<td>62.67</td>
<td>0.24633</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>0.150</td>
<td>0.6158</td>
<td>119</td>
<td>48</td>
<td>56.03</td>
<td>0.53784</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>0.150</td>
<td>0.6249</td>
<td>120</td>
<td>69</td>
<td>86.34</td>
<td>0.54094</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>0.200</td>
<td>0.7107</td>
<td>122</td>
<td>61</td>
<td>81.84</td>
<td>0.15449</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>0.200</td>
<td>0.6966</td>
<td>111</td>
<td>84</td>
<td>112.76</td>
<td>0.23352</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>0.300</td>
<td>0.8474</td>
<td>106</td>
<td>60</td>
<td>81.94</td>
<td>0.10690</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td>0.300</td>
<td>0.8254</td>
<td>116</td>
<td>78</td>
<td>133.19</td>
<td>0.29882</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td>0.500</td>
<td>0.9423</td>
<td>99</td>
<td>60</td>
<td>79.64</td>
<td>0.01473</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>0.500</td>
<td>0.9491</td>
<td>89</td>
<td>35</td>
<td>44.16</td>
<td>0.08674</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.9: Computational results for $n = 50, 60$
<table>
<thead>
<tr>
<th>#Jobs</th>
<th>Name</th>
<th>P</th>
<th>Ord.Sth.</th>
<th># Arcs</th>
<th># Cuts</th>
<th>CT</th>
<th>% Gap</th>
<th>Statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>1</td>
<td>.001</td>
<td>.00041</td>
<td>1</td>
<td>1</td>
<td>12.36</td>
<td>.00000</td>
<td>AGAP</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>.001</td>
<td>.00041</td>
<td>1</td>
<td>1</td>
<td>12.03</td>
<td>.00000</td>
<td>.1951</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>.020</td>
<td>.03664</td>
<td>43</td>
<td>57</td>
<td>108.31</td>
<td>.00975</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>.020</td>
<td>.02650</td>
<td>39</td>
<td>25</td>
<td>105.08</td>
<td>.46689</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>.040</td>
<td>.11511</td>
<td>87</td>
<td>103</td>
<td>139.84</td>
<td>.00377</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>.040</td>
<td>.08861</td>
<td>94</td>
<td>81</td>
<td>106.22</td>
<td>.10148</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>.080</td>
<td>.17143</td>
<td>112</td>
<td>86</td>
<td>108.64</td>
<td>.01781</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>.060</td>
<td>.22485</td>
<td>119</td>
<td>76</td>
<td>94.86</td>
<td>.09505</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>.060</td>
<td>.34576</td>
<td>147</td>
<td>65</td>
<td>84.80</td>
<td>.18851</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>.080</td>
<td>.31387</td>
<td>139</td>
<td>58</td>
<td>68.76</td>
<td>.34954</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>.100</td>
<td>.42774</td>
<td>163</td>
<td>79</td>
<td>108.86</td>
<td>.19217</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>.100</td>
<td>.56522</td>
<td>138</td>
<td>70</td>
<td>98.21</td>
<td>.20274</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>.150</td>
<td>.64306</td>
<td>138</td>
<td>106</td>
<td>169.00</td>
<td>.61276</td>
<td></td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>.150</td>
<td>.60373</td>
<td>148</td>
<td>66</td>
<td>82.71</td>
<td>.23170</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>.200</td>
<td>.82650</td>
<td>132</td>
<td>82</td>
<td>138.85</td>
<td>.34946</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>.200</td>
<td>.76563</td>
<td>145</td>
<td>42</td>
<td>97.98</td>
<td>.40806</td>
<td></td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>.300</td>
<td>.89151</td>
<td>139</td>
<td>78</td>
<td>131.76</td>
<td>.27388</td>
<td></td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>.300</td>
<td>.87992</td>
<td>140</td>
<td>75</td>
<td>196.91</td>
<td>.18546</td>
<td></td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>.500</td>
<td>.95694</td>
<td>111</td>
<td>57</td>
<td>214.05</td>
<td>.09142</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>.500</td>
<td>.96398</td>
<td>105</td>
<td>40</td>
<td>109.91</td>
<td>.12117</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>1</td>
<td>.001</td>
<td>.00158</td>
<td>5</td>
<td>1</td>
<td>17.31</td>
<td>.00000</td>
<td>AGAP</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>.001</td>
<td>.00063</td>
<td>2</td>
<td>1</td>
<td>16.09</td>
<td>.00000</td>
<td>.2870</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>.020</td>
<td>.04620</td>
<td>75</td>
<td>46</td>
<td>110.13</td>
<td>.10601</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>.020</td>
<td>.04620</td>
<td>60</td>
<td>19</td>
<td>47.79</td>
<td>.03868</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>.040</td>
<td>.06709</td>
<td>107</td>
<td>120</td>
<td>221.19</td>
<td>.00306</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>.040</td>
<td>.17690</td>
<td>125</td>
<td>95</td>
<td>132.75</td>
<td>.17918</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>.060</td>
<td>.28196</td>
<td>168</td>
<td>113</td>
<td>175.32</td>
<td>.22464</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>.060</td>
<td>.22722</td>
<td>160</td>
<td>64</td>
<td>117.37</td>
<td>.36282</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>.080</td>
<td>.46171</td>
<td>170</td>
<td>71</td>
<td>109.24</td>
<td>.25996</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>.080</td>
<td>.34240</td>
<td>170</td>
<td>65</td>
<td>98.75</td>
<td>.71447</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>.100</td>
<td>.36266</td>
<td>183</td>
<td>61</td>
<td>99.08</td>
<td>.15841</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>.100</td>
<td>.40772</td>
<td>185</td>
<td>78</td>
<td>126.77</td>
<td>.53486</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>.150</td>
<td>.72468</td>
<td>174</td>
<td>83</td>
<td>157.80</td>
<td>.45760</td>
<td></td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>.150</td>
<td>.70949</td>
<td>168</td>
<td>84</td>
<td>165.54</td>
<td>.41877</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>.200</td>
<td>.78798</td>
<td>162</td>
<td>83</td>
<td>154.67</td>
<td>.39464</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>.200</td>
<td>.72089</td>
<td>206</td>
<td>76</td>
<td>146.54</td>
<td>.38601</td>
<td></td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>.300</td>
<td>.91994</td>
<td>132</td>
<td>19</td>
<td>70.74</td>
<td>.75583</td>
<td></td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>.300</td>
<td>.85886</td>
<td>158</td>
<td>83</td>
<td>246.56</td>
<td>.43060</td>
<td></td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>.500</td>
<td>.95285</td>
<td>122</td>
<td>39</td>
<td>171.69</td>
<td>.23316</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>.500</td>
<td>.95601</td>
<td>136</td>
<td>82</td>
<td>208.22</td>
<td>.08160</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.10: Computational results for \( n = 70, 80 \)
<table>
<thead>
<tr>
<th>#Jobs</th>
<th>Name</th>
<th>P Ord.Sth.</th>
<th># Arcs</th>
<th># Cuts</th>
<th>CT</th>
<th>% Gap</th>
<th>Statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td></td>
<td>0.001</td>
<td>0.0050</td>
<td>2</td>
<td>1</td>
<td>22.36</td>
<td>.00000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.020</td>
<td>0.0371</td>
<td>72</td>
<td>140</td>
<td>562.22</td>
<td>.00000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.020</td>
<td>0.03845</td>
<td>87</td>
<td>94</td>
<td>310.38</td>
<td>.01671</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.040</td>
<td>0.11261</td>
<td>156</td>
<td>107</td>
<td>192.23</td>
<td>.08653</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.040</td>
<td>0.10786</td>
<td>140</td>
<td>102</td>
<td>206.74</td>
<td>.10700</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.060</td>
<td>0.24145</td>
<td>177</td>
<td>92</td>
<td>162.97</td>
<td>.11381</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.060</td>
<td>0.25368</td>
<td>171</td>
<td>97</td>
<td>167.64</td>
<td>.26182</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.080</td>
<td>0.36654</td>
<td>205</td>
<td>70</td>
<td>131.55</td>
<td>.47914</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.080</td>
<td>0.41973</td>
<td>182</td>
<td>92</td>
<td>208.00</td>
<td>.54811</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.100</td>
<td>0.50737</td>
<td>200</td>
<td>78</td>
<td>175.93</td>
<td>.95225</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.100</td>
<td>0.56105</td>
<td>207</td>
<td>74</td>
<td>161.43</td>
<td>.35829</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.150</td>
<td>0.72584</td>
<td>216</td>
<td>108</td>
<td>298.19</td>
<td>.59628</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.150</td>
<td>0.70212</td>
<td>205</td>
<td>98</td>
<td>215.64</td>
<td>.32440</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.200</td>
<td>0.81248</td>
<td>202</td>
<td>103</td>
<td>244.26</td>
<td>.18701</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.200</td>
<td>0.76405</td>
<td>207</td>
<td>81</td>
<td>221.79</td>
<td>.16728</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.300</td>
<td>0.87965</td>
<td>169</td>
<td>96</td>
<td>265.67</td>
<td>.13002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.300</td>
<td>0.87590</td>
<td>194</td>
<td>114</td>
<td>321.92</td>
<td>.20084</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.500</td>
<td>0.97129</td>
<td>129</td>
<td>73</td>
<td>284.40</td>
<td>.00510</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.500</td>
<td>0.96404</td>
<td>145</td>
<td>82</td>
<td>329.00</td>
<td>.02184</td>
</tr>
</tbody>
</table>

| 100   |      | 0.001     | 0.0081 | 4      | 1  | 24.71 | .00000    | AGAP .2095 |
|       |      | 0.001     | 0.00101| 5      | 1  | 25.65 | .00000    | ACT 290.33 |
|       |      | 0.020     | 0.04525| 99     | 60 | 199.71| .05779    |            |
|       |      | 0.020     | 0.03030| 89     | 99 | 530.80| .01623    |            |
|       |      | 0.040     | 0.10101| 162    | 148| 389.54| .08459    |            |
|       |      | 0.040     | 0.09677| 170    | 122| 269.68| .05457    |            |
|       |      | 0.060     | 0.23899| 213    | 106| 222.06| .17924    |            |
|       |      | 0.060     | 0.30929| 209    | 129| 265.18| .06981    |            |
|       |      | 0.080     | 0.43495| 218    | 105| 218.50| .30606    |            |
|       |      | 0.080     | 0.45394| 228    | 110| 243.71| .25238    |            |
|       |      | 0.100     | 0.57737| 239    | 126| 418.37| .30853    |            |
|       |      | 0.100     | 0.50202| 233    | 109| 250.13| .37136    |            |
|       |      | 0.150     | 0.73960| 209    | 126| 372.78| .43368    |            |
|       |      | 0.150     | 0.75414| 218    | 141| 449.40| .48349    |            |
|       |      | 0.200     | 0.83778| 200    | 90 | 320.55| .19209    |            |
|       |      | 0.200     | 0.79859| 217    | 126| 381.19| .51074    |            |
|       |      | 0.300     | 0.91515| 188    | 67 | 376.02| .27127    |            |
|       |      | 0.300     | 0.92222| 178    | 87 | 461.76| .18564    |            |
|       |      | 0.500     | 0.96606| 164    | 56 | 232.22| .12527    |            |
|       |      | 0.500     | 0.97030| 157    | 53 | 154.68| .28807    |            |

Table 3.11: Computational results for \( n = 90, 100 \)
<table>
<thead>
<tr>
<th>#Jobs</th>
<th>Name</th>
<th>P</th>
<th>Ord.Sth.</th>
<th># Arcs</th>
<th># Cuts</th>
<th>CT</th>
<th>% Gap</th>
<th>Statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>1</td>
<td>.001</td>
<td>.00050</td>
<td>3</td>
<td>1</td>
<td>30.43</td>
<td>.00000</td>
<td>AGAP .2101</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>.001</td>
<td>.00100</td>
<td>6</td>
<td>1</td>
<td>30.05</td>
<td>.00000</td>
<td>ACT .414.27</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>.020</td>
<td>.04087</td>
<td>110</td>
<td>317</td>
<td>1614.70</td>
<td>.00000</td>
<td>ACT .414.27</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>.040</td>
<td>.05888</td>
<td>129</td>
<td>222</td>
<td>813.66</td>
<td>.00948</td>
<td>ACT .414.27</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>.040</td>
<td>.26539</td>
<td>201</td>
<td>113</td>
<td>289.51</td>
<td>.07890</td>
<td>.2101</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>.040</td>
<td>.17831</td>
<td>205</td>
<td>102</td>
<td>245.90</td>
<td>.5274</td>
<td>.2101</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>.060</td>
<td>.31793</td>
<td>271</td>
<td>94</td>
<td>233.10</td>
<td>.21554</td>
<td>.2101</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>.060</td>
<td>.32127</td>
<td>266</td>
<td>114</td>
<td>288.90</td>
<td>.45806</td>
<td>.2101</td>
</tr>
<tr>
<td>120</td>
<td>10</td>
<td>.080</td>
<td>.48791</td>
<td>276</td>
<td>97</td>
<td>256.06</td>
<td>.24626</td>
<td>.2101</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>.100</td>
<td>.66806</td>
<td>241</td>
<td>156</td>
<td>459.94</td>
<td>.23973</td>
<td>.2101</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>.100</td>
<td>.58415</td>
<td>294</td>
<td>108</td>
<td>348.67</td>
<td>.88693</td>
<td>.2101</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>.150</td>
<td>.75113</td>
<td>268</td>
<td>109</td>
<td>308.46</td>
<td>.24279</td>
<td>.2101</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>.150</td>
<td>.75663</td>
<td>256</td>
<td>109</td>
<td>488.46</td>
<td>.29757</td>
<td>.2101</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>.200</td>
<td>.84304</td>
<td>232</td>
<td>116</td>
<td>456.71</td>
<td>.40698</td>
<td>.2101</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>.200</td>
<td>.84871</td>
<td>237</td>
<td>132</td>
<td>397.77</td>
<td>.21936</td>
<td>.2101</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>.300</td>
<td>.90142</td>
<td>223</td>
<td>34</td>
<td>166.75</td>
<td>.41074</td>
<td>.2101</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>.300</td>
<td>.92410</td>
<td>206</td>
<td>115</td>
<td>540.19</td>
<td>.04542</td>
<td>.2101</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>.500</td>
<td>.97448</td>
<td>173</td>
<td>86</td>
<td>570.29</td>
<td>.13094</td>
<td>.2101</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>.500</td>
<td>.98931</td>
<td>162</td>
<td>91</td>
<td>500.76</td>
<td>.06043</td>
<td>.2101</td>
</tr>
</tbody>
</table>

Table 3.12: Computational results for \( n = 110,120 \)
<table>
<thead>
<tr>
<th>#Jobs</th>
<th>Name</th>
<th>P</th>
<th>Ord.Sth.</th>
<th># Arcs</th>
<th># Cuts</th>
<th>CT</th>
<th>% Gap</th>
<th>Statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>130</td>
<td>1</td>
<td>.001</td>
<td>.00143</td>
<td>12</td>
<td>45</td>
<td>264.53</td>
<td>.00000</td>
<td>AGAP</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>.001</td>
<td>.00119</td>
<td>10</td>
<td>13</td>
<td>114.96</td>
<td>.00000</td>
<td>.1888</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>.020</td>
<td>.04568</td>
<td>168</td>
<td>296</td>
<td>1726.70</td>
<td>.03171</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>.020</td>
<td>.04770</td>
<td>153</td>
<td>312</td>
<td>1888.78</td>
<td>.05270</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>.040</td>
<td>.17197</td>
<td>291</td>
<td>145</td>
<td>445.51</td>
<td>.11493</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>.040</td>
<td>.13787</td>
<td>274</td>
<td>118</td>
<td>390.14</td>
<td>.04043</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>.060</td>
<td>.41145</td>
<td>321</td>
<td>68</td>
<td>291.05</td>
<td>.44764</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>.060</td>
<td>.42922</td>
<td>302</td>
<td>76</td>
<td>300.77</td>
<td>.34043</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>.080</td>
<td>.52009</td>
<td>344</td>
<td>103</td>
<td>332.69</td>
<td>.25293</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>.080</td>
<td>.56947</td>
<td>301</td>
<td>129</td>
<td>445.23</td>
<td>.25293</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>.100</td>
<td>.68515</td>
<td>322</td>
<td>130</td>
<td>527.62</td>
<td>.3697</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>.100</td>
<td>.64782</td>
<td>329</td>
<td>103</td>
<td>471.09</td>
<td>.37895</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>.150</td>
<td>.75826</td>
<td>332</td>
<td>149</td>
<td>627.36</td>
<td>.49034</td>
<td></td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>.150</td>
<td>.79165</td>
<td>322</td>
<td>115</td>
<td>407.88</td>
<td>.17745</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>.200</td>
<td>.87871</td>
<td>265</td>
<td>115</td>
<td>589.68</td>
<td>.15636</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>.200</td>
<td>.86094</td>
<td>292</td>
<td>110</td>
<td>524.70</td>
<td>.11167</td>
<td></td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>.300</td>
<td>.94252</td>
<td>231</td>
<td>123</td>
<td>1051.00</td>
<td>.12904</td>
<td></td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>.300</td>
<td>.90912</td>
<td>288</td>
<td>170</td>
<td>773.95</td>
<td>.17247</td>
<td></td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>.500</td>
<td>.97665</td>
<td>194</td>
<td>77</td>
<td>485.10</td>
<td>.07439</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>.500</td>
<td>.96023</td>
<td>219</td>
<td>76</td>
<td>457.75</td>
<td>.19034</td>
<td></td>
</tr>
<tr>
<td>140</td>
<td>1</td>
<td>.001</td>
<td>.00113</td>
<td>11</td>
<td>1</td>
<td>64.15</td>
<td>.00000</td>
<td>AGAP</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>.001</td>
<td>.00082</td>
<td>8</td>
<td>124</td>
<td>1180.68</td>
<td>.00000</td>
<td>.2055</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>.020</td>
<td>.05704</td>
<td>181</td>
<td>304</td>
<td>1769.81</td>
<td>.02924</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>.020</td>
<td>.06495</td>
<td>199</td>
<td>259</td>
<td>1342.71</td>
<td>.07154</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>.040</td>
<td>.22354</td>
<td>332</td>
<td>150</td>
<td>571.00</td>
<td>.28460</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>.040</td>
<td>.19168</td>
<td>297</td>
<td>139</td>
<td>581.77</td>
<td>.14007</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>.060</td>
<td>.46218</td>
<td>326</td>
<td>187</td>
<td>647.36</td>
<td>.31147</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>.060</td>
<td>.46578</td>
<td>321</td>
<td>134</td>
<td>525.03</td>
<td>.31678</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>.080</td>
<td>.54820</td>
<td>354</td>
<td>153</td>
<td>643.45</td>
<td>.40014</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>.080</td>
<td>.52795</td>
<td>368</td>
<td>149</td>
<td>601.05</td>
<td>.87223</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>.100</td>
<td>.69239</td>
<td>372</td>
<td>113</td>
<td>567.71</td>
<td>.36579</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>.100</td>
<td>.68612</td>
<td>325</td>
<td>152</td>
<td>775.53</td>
<td>.58323</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>.150</td>
<td>.77359</td>
<td>354</td>
<td>113</td>
<td>527.67</td>
<td>.09663</td>
<td></td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>.150</td>
<td>.82867</td>
<td>304</td>
<td>40</td>
<td>217.95</td>
<td>.79355</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>.200</td>
<td>.84327</td>
<td>347</td>
<td>154</td>
<td>858.60</td>
<td>.14770</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>.200</td>
<td>.86218</td>
<td>312</td>
<td>32</td>
<td>257.88</td>
<td>.86032</td>
<td></td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>.300</td>
<td>.94132</td>
<td>278</td>
<td>162</td>
<td>1052.65</td>
<td>.11478</td>
<td></td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>.300</td>
<td>.93731</td>
<td>291</td>
<td>59</td>
<td>361.58</td>
<td>.15913</td>
<td></td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>.500</td>
<td>.97718</td>
<td>225</td>
<td>87</td>
<td>593.25</td>
<td>.13079</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>.500</td>
<td>.97811</td>
<td>218</td>
<td>55</td>
<td>468.07</td>
<td>.22498</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.13: Computational results for $n = 130, 140$
<table>
<thead>
<tr>
<th>#Jobs</th>
<th>Name</th>
<th>P</th>
<th>Ord.Sth.</th>
<th># Arcs</th>
<th># Cuts</th>
<th>CT</th>
<th>% Gap</th>
<th>Statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>150</td>
<td>1</td>
<td>.001</td>
<td>.00107</td>
<td>12</td>
<td>119</td>
<td>1112.90</td>
<td>.00000</td>
<td>AGAP .3038</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>.001</td>
<td>.00125</td>
<td>14</td>
<td>125</td>
<td>1194.52</td>
<td>.00000</td>
<td>ACT 851.16</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>.020</td>
<td>.05145</td>
<td>223</td>
<td>344</td>
<td>2074.26</td>
<td>.00365</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>.020</td>
<td>.05414</td>
<td>219</td>
<td>268</td>
<td>1613.16</td>
<td>.00618</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>.040</td>
<td>.20546</td>
<td>354</td>
<td>151</td>
<td>585.62</td>
<td>.34732</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>.040</td>
<td>.22407</td>
<td>326</td>
<td>156</td>
<td>537.89</td>
<td>.14711</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>.060</td>
<td>.48054</td>
<td>388</td>
<td>157</td>
<td>671.85</td>
<td>.49276</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>.060</td>
<td>.41682</td>
<td>422</td>
<td>117</td>
<td>583.81</td>
<td>.20510</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>.080</td>
<td>.53127</td>
<td>437</td>
<td>154</td>
<td>692.94</td>
<td>.66706</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>.080</td>
<td>.61494</td>
<td>405</td>
<td>156</td>
<td>847.50</td>
<td>.63095</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>.100</td>
<td>.69942</td>
<td>388</td>
<td>105</td>
<td>653.78</td>
<td>.71995</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>.100</td>
<td>.69110</td>
<td>389</td>
<td>124</td>
<td>682.55</td>
<td>.21691</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>.150</td>
<td>.80143</td>
<td>370</td>
<td>81</td>
<td>443.47</td>
<td>.38478</td>
<td></td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>.150</td>
<td>.83400</td>
<td>338</td>
<td>151</td>
<td>934.89</td>
<td>.22556</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>.200</td>
<td>.88143</td>
<td>326</td>
<td>209</td>
<td>1729.00</td>
<td>.34991</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>.200</td>
<td>.88608</td>
<td>330</td>
<td>60</td>
<td>449.67</td>
<td>.58260</td>
<td></td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>.300</td>
<td>.93709</td>
<td>279</td>
<td>51</td>
<td>430.12</td>
<td>.42886</td>
<td></td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>.300</td>
<td>.94022</td>
<td>280</td>
<td>116</td>
<td>1051.93</td>
<td>.15777</td>
<td></td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>.500</td>
<td>.97888</td>
<td>236</td>
<td>58</td>
<td>296.93</td>
<td>.14932</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>.500</td>
<td>.98201</td>
<td>231</td>
<td>46</td>
<td>436.44</td>
<td>.35984</td>
<td></td>
</tr>
</tbody>
</table>

| 160   | 1    | .001 | .00087   | 11     | 1      | 112.16  | .00000 | AGAP .2889 |
|       | 2    | .001 | .00126   | 15     | 129    | 2280.40 | .00000 | ACT 1137.32 |
|       | 3    | .020 | .06022   | 233    | 289    | 2005.44 | .01548 |
|       | 4    | .020 | .05299   | 220    | 402    | 3513.58 | .01715 |
|       | 5    | .040 | .26832   | 386    | 160    | 633.40  | .20351 |
|       | 6    | .040 | .24772   | 369    | 150    | 591.05  | .24819 |
|       | 7    | .060 | .35855   | 474    | 142    | 625.27  | .34809 |
|       | 8    | .060 | .48884   | 429    | 112    | 738.20  | .54318 |
|       | 9    | .080 | .58530   | 426    | 111    | 623.88  | .39843 |
|       | 10   | .080 | .61580   | 423    | 152    | 921.66  | .42387 |
|       | 11   | .100 | .71069   | 394    | 176    | 950.65  | .72379 |
|       | 12   | .100 | .73538   | 396    | 107    | 698.43  | .43946 |
|       | 13   | .150 | .81942   | 406    | 210    | 1544.83 | .66236 |
|       | 14   | .150 | .83661   | 392    | 166    | 1100.38 | .34659 |
|       | 15   | .200 | .89285   | 357    | 75     | 628.79  | .41888 |
|       | 16   | .200 | .89222   | 345    | 191    | 1288.22 | .24187 |
|       | 17   | .300 | .93789   | 327    | 158    | 1277.35 | .19612 |
|       | 18   | .300 | .93931   | 325    | 86     | 1132.02 | .21530 |
|       | 19   | .500 | .98608   | 223    | 56     | 384.31  | .17023 |
|       | 20   | .500 | .98019   | 251    | 113    | 1694.40 | .16632 |

Table 3.14: Computational results for \( n = 150, 160 \)
Chapter 4

Hamiltonian Path and Symmetric Travelling Salesman Problem

This chapter introduces a Hamiltonian Path (HP) approach for studying symmetric travelling salesman polytopes (STSPs). By studying HP polytopes, we develop general clique-lifting results, all based on simple conditions, for deriving large new classes of STSP facets. These results apply to all known non-trivial STSP facets, and generalize clique-lifting results of Maurras [38], Grötschel and Padberg [27] and Naddef and Rinaldi [41].

4.1 Introduction

Let $K(V') = (V', E(V'))$ be a complete undirected graph, where the node set $V'$ represents a set of cities. A feasible solution to the Symmetric Travelling Salesman (STS) problem is a Hamiltonian cycle $C$ on $V'$, uniquely determined by its corresponding 0-1 incidence vector $y \in \mathbb{R}^{E(V')}$ (where $y_e = 1$ if edge $e$ is used in $C$ and zero otherwise). This chapter deals with the STS polytope $STSP(V')$, the convex hull of incidence vectors $y$ of all feasible solutions to the STS problem.

The facial structure of STS polytopes (STSPs) has been the object of considerable research over the past two decades. We refer to Grötschel and Padberg [28] for an excellent survey (up to 1985). Since then, there have been many advances in studying STS polytopes and in solving large-scale STS problems. Large classes of STSP valid inequalities were proposed by Boyd and Cunningham [7,8], Cornuéjols, et al. [13], Fleischmann [19,20,21], Naddef and Rinaldi [41,42,44], Naddef [40]. Some of these inequalities,
such as clique-tree [29], chain (introduced in [47], and proven to be facet-defining independently by S. Boyd and M. Hartmann, cf. [8]), ladder [8], crown and path-tree [42,43] inequalities are known to be facet-defining. The study of STSP facets is also of practical importance: Padberg and Rinaldi [49] have successfully demonstrated the use of facet-defining inequalities, such as clique-tree inequalities, in solving large-scale, real-world STS problems (with over 2000 cities) to optimality. For small STS polytopes, Naddef and Rinaldi [43] present a list of inequalities defining 133,707 distinct facets of $STSP(V')$ when $|V'| = 8$, including nonnegativity constraints, sub-tour elimination, ladder, chain, crown, and path inequalities. Boyd and Cunningham [8] have proven that part of this list entirely describes $STSP(V')$ with $|V'| \leq 7$. Christof et al [15] use a computer code to derive a complete description of $STSP(V')$ with $|V'| = 8$, where in addition to Naddef and Rinaldi's list, three classes of new facets are discovered. For the case $|V'| = 9$, we introduce a new facet in Figure 4.3, illustrating the incompleteness of all the above known facet classes for $|V'| \geq 9$.

This chapter uses a projective approach to the polyhedral study of symmetric travelling salesman problems. Consider the node set $V \equiv V' \setminus \{h\}$, where $h \in V'$ is any given projection node. Let $HP(V)$ denote the Hamiltonian Path Polytope (HPP), i.e., the convex hull of incidence vectors $x$ of all Hamiltonian paths (with free endnodes) on $V$. Using a one-to-one correspondence between Hamiltonian cycles on $V'$ and Hamiltonian paths on $V \equiv V' \setminus \{h\}$, we show that $HP(V)$ is a projection of $STSP(V')$ onto the subspace $R^{E(V)}$ of $R^{E(V')}$. More importantly, $HP(V)$ exhibits three fundamental properties for this projective approach: (a) $HP(V)$ is near-full dimensional, with a single implicit equation $\sum_{e \in E(V)} x_e = |V| - 1$; (b) a move from one vertex $x \in HP(V)$ to another one may involve only one interchange of components in $x$, whereas such a move involves at least two interchanges in $y \in STSP(V')$ (the so-called 2-opt interchange); (c) polyhedral
equivalence: the polyhedral structures of $HP(V)$ and $STSP(V')$ are isomorphic. Properties (a) and (b) greatly facilitate the indirect proofs of any facetial results for $HP(V)$, and property (c) implies immediately that the corresponding facetial results also hold for $STSP(V')$. In contrast, additional hard work is required for transferring facetial results, derived from monotone [28] or graphical relaxations [13], to corresponding results for $STSP(V')$.

We study extensions of STSP facets by clique-lifting. For ATS polytopes, Balas and Fischetti [6] prove a general clique-lifting result that applies to all nontrivial facets of ATS polytopes. For STS polytopes, it is unknown whether such a powerful lifting result holds. Maurras (1975), Grötschel and Padberg (1979), and Naddef and Rinaldi (1989) present various sufficient conditions for clique-lifting of STSP facets. The early lifting results employs simple conditions (verifiable by inspection or in linear time) but their applications seem restricted. Naddef and Rinaldi’s clique-lifting result applies to large classes of known STSP facets, but the condition seems difficult to verify in general (we do not know whether there exists a polynomial-time algorithm to verify their condition). In this paper, we propose a rather general clique-lifting result (with at least two new nodes added) that applies to any nontrivial STSP facet w.r.t. any given node. (In the sequel, “w.r.t.” stands for “with respect to”.) We also show, by the projective approach, that Naddef and Rinaldi’s sufficient condition for node-cloning (i.e., clique lifting with exactly one new node added) is in fact necessary. As the condition is difficult to verify, we propose three simplified node-cloning results, all based on conditions that can be verified either by inspection or by a procedure of time complexity linear in the number of edges. The first result seems applicable to all known nontrivial facets (such as clique-tree, chain, ladder, crown, path and path tree, etc.) w.r.t. any node. The second result generalizes the clique-lifting results in Maurras [38], Grötschel and Padberg [27]. The third result shows that any node can be cloned in any facet-defining rank inequality (i.e., inequality
ay \leq a_0 \text{ with zero-one coefficients).}

The rest of the chapter is organized as follows. Section 4.2 gives basic terminology and notation. Section 4.3 introduces the projective approach and the polyhedral equivalence of \( HP(V) \) and \( STSP(V') \), as well as other related results. In Section 4.4, we propose a clique-lifting result (with clique size greater than two) for all nontrivial STSP facets and prove that Naddef and Rinaldi's clique-lifting condition is necessary. In Section 4.5, we present our simple sufficient conditions for node-cloning. Section 4.6 offers concluding remarks.

4.2 Definition and Notation

Our notation and terminology mostly follow those in [6],[28],[41].

(1). For any node set \( V \) with \( n \geq |V| \geq 3 \), let \( K(V) = (V, E(V)) \) denote the complete undirected graph with node set \( V \). Note that \( |E(V)| = n(n-1)/2 \). For any subsets \( S_1 \) and \( S_2 \) of \( V \), let

\[
E(S_1 : S_2) = \{(i,j) \in E(V) : i \in S_1, j \in S_2\}.
\]

When \( S = S_1 = S_2 \), denote \( E(S : S) \) by \( E(S) \), i.e., the collection of all edges in \( E(V) \) with both endnodes in \( S \). For any \( S \subseteq V \), denote the complete graph on node set \( S \) by \( K(S) = (S, E(S)) \). For any node \( v \in V \), let \( \delta(v) = E(\{v\} : V \setminus \{v\}) \) denote the star of \( v \), i.e., the set of all edges in \( E(V) \) incident to \( v \).

(2). A partial graph \( (S, \tilde{E}) \) is a subgraph of \( K(S) \) with \( \tilde{E} \subseteq E(S) \). The incidence vector of partial graph \( (V, \tilde{E}) \) of \( K(S) \) is the vector \( \chi \in R^{E(S)} \) such that \( \chi_e = 1 \) if \( e \in \tilde{E} \) and \( \chi_e = 0 \) otherwise. For any vector \( \psi \in R^{E(S)} \) and \( \tilde{E} \subseteq E(S) \), let

\[
\psi(\tilde{E}) = \sum_{e \in \tilde{E}} \psi_e. \tag{4.1}
\]
A Hamiltonian cycle $C$ on $S$ is a connected partial graph $(S, \tilde{E})$ such that

$$|\tilde{E} \cap \delta(v)| = 2, \quad \text{for all } v \in S.$$  

Without risking confusion, a Hamiltonian cycle $C$ with edge set

$$\tilde{E} = \{(v_\sigma(i), v_\sigma(i+1)) : i = 1, \ldots, s - 1\} \cup \{(v_\sigma(s), v_\sigma(1))\}$$

on node set $S = \{v_1, \ldots, v_s\} \subseteq V$ may be represented by edge set $\tilde{E}$ alone or, equivalently, by the circular permutation $(v_\sigma(1)\ldots v_\sigma(s))$ of the nodes.

A Hamiltonian path $P$ on $S = \{v_1, \ldots, v_s\} \subseteq V$ is a connected partial graph $(S, \hat{E})$ such that for some distinct nodes $i, j \in S$, $(S, \hat{E} \cup \{i, j\})$ defines a Hamiltonian cycle on $S$. Note that the endnodes of $P$ are not specified in advance, i.e., we consider here paths with free endnodes. Hamiltonian path $P$ may be represented by its edge set $\{(v_i, v_{i+1}) : i = 1, \ldots, s - 1\}$ or, equivalently, by its corresponding node permutation $(v_1\ldots v_s)$ with endnodes $v_1$ and $v_s$. Note that if $|S| = 1$, the first form is an empty set, whereas the second form reduces to a single node $(v_1)$. When a path is written in the form $(u\ldots v p \ldots q)$, we always allow $u = v$ or $p = q$ or both.

For any node set $S$, the STS polytope $\text{STSP}(S)$ and HP polytope $\text{HP}(S)$ are defined, respectively, as follows.

$$\text{STSP}(S) = \text{conv}\{y^C \in R^{E(S)} : C \text{ is a Hamiltonian cycle on } S\},$$

$$\text{HP}(S) = \text{conv}\{x^P \in R^{E(S)} : P \text{ is a Hamiltonian path on } S\}.$$  

(3). To any inequality $cx \leq c_0$ with $c \geq 0$ and $x \in R^{E(S)}$, we associate its support graph:

$$G_c(S) = (S, E_c), \quad \text{where } E_c = \{e \in E(S) : c_e > 0\}.$$  

The complement graph $\overline{G}_c(S) = (S, \overline{E}_c)$, where $\overline{E}_c = E(S) \setminus E_c$, is called the zero-graph w.r.t. $cx \leq c_0$. 
(4). Let \( ay \leq a_0 \) and \( cx \leq c_0 \) be two valid inequalities for \( STSP(S) \) and \( HP(S) \), respectively. We say that \( C \) is an \( a\)-tight cycle (resp., \( P \) is a \( c\)-tight path) on \( S \) if \( C \) is a Hamiltonian cycle on \( S \) and \( ay^C = a_0 \) (resp., if \( P \) is a Hamiltonian path on \( S \) and \( cx^P = c_0 \)).

4.3 Polyhedral Equivalence: HPP v.s. STSP

The STS polytope (STSP) is far from being full dimensional because every Hamiltonian cycle satisfies a degree constraint equation at each node. In contrast, the HP formulation relaxes all degree constraints. This observation suggests that \( HP(V) \) be more nearly full dimensional. It leads to the development of a projective approach to the study of travelling salesman problems. A similar projective approach has been used for the equipartition polytope by Conforti, Rao and Sassano [12].

Let \( V' \) be a node set with \( |V'| = n+1 \), and for some fixed node \( h \in V' \), let \( V = V' \setminus \{h\} \).

Consider two complete undirected graphs:

\[ K(V) = (V, E(V)) \quad \text{and} \quad K(V') = (V', E(V')). \]

That is, \( K(V) \) is obtained from \( K(V') \) by eliminating node \( h \in V' \) and all incident edges. Let \( STSP(V') \) be the STS polytope defined on \( K(V') \), and \( HP(V) \) be the HP polytope defined on \( K(V) \). Observe that \( P \) is a Hamiltonian path on \( V \) with endnodes \( u \) and \( v \) iff \( C = P \cup \{(h, u), (h, v)\} \) is a Hamiltonian cycle on \( V' \). Thus, the incidence vector \( x^P \in R^{E(V)} \) is a projection of the incidence vector \( y^C \in R^{E(V')} \) onto subspace \( R^{E(V)} \) of \( R^{E(V')} \). It follows that the polytope \( HP(V) \) is a projection of \( STSP(V') \). In what follows, we assume that all incidence vectors \( y \) on \( K(V') \) are indexed as: \( y = (y_1, \ldots, y_m, y_{m+1}, \ldots, y_{m'})^t \), where the first \( m = n(n - 1)/2 \) elements correspond to the edge set \( E(V) \).
Chapter 4. Hamiltonian Path and Symmetric Travelling Salesman Problem

Let $A_n$ be a node-edge incidence matrix of $K(V)$. Observe that the incidence vectors $y^C \in R^{E(V')}$ of all Hamiltonian cycles $C$ on $V'$ and their projections $x^P \in R^{E(V)}$ satisfy the following identity:

$$y^C_{uv} \equiv \begin{cases} x^P_{uv}, & (u, v) \in E(V); \\ 2 - \sum_{e \in E(V)} (A_n)_{ve} x^P_e, & (u, v) \in E(h : V). \end{cases} \quad (4.2)$$

Clearly, the above equation defines an affine transformation from $HP(V)$ to $STSP(V')$. The following result shows that this transformation is affine-rank preserving.

**Lemma 4.1** Let $B_y = [y^1, \ldots, y^l]$ be any matrix such that all column vectors $y^j$ are incidence vectors of Hamiltonian cycles on $K(V')$. Let $B_x = [x^1, \ldots, x^l]$, where $x^j$ is the projection of $y^j$ into $R^{E(V)}$, $j = 1, \ldots, l$. Then the number of affinely independent column vectors in $B_y$ is equal to the number of affinely independent column vectors in $B_x$.

**Proof:** From equation (4.2), it follows

$$\text{aff.rank} \left( B_y \right) = \text{rank} \left( \begin{pmatrix} 1^t \\ B_y \end{pmatrix} \right) \geq \text{rank} \left( \begin{pmatrix} 1^t \\ B_x \end{pmatrix} \right)$$

$$\geq \text{rank} \left( \begin{pmatrix} 1 & 0^t \\ 0 & I \\ 2 & -A_n \end{pmatrix} \begin{pmatrix} 1^t \\ B_x \end{pmatrix} \right) = \text{rank} \left( \begin{pmatrix} 1^t \\ B_y \end{pmatrix} \right) = \text{aff.rank} \left( B_y \right),$$

where "aff.rank" denotes the affine rank of the column set, "rank" denotes the linear rank, $0$ (resp., $1$) is a vector of zeroes (resp., ones) and $I$ is an identity matrix, all of conformable dimension. Hence, equality must hold throughout. 

The dimension of $HP(V)$ could now be derived from that of $STSP(V')$ using Lemma 4.1. In fact, the direct derivation of $\dim(HP(V))$ given below is very simple. Thus, we obtain a much simpler proof than any known proof for $\dim(STSP(V'))$ (see [28], section 4.1, for two other proofs).
Chapter 4. Hamiltonian Path and Symmetric Travelling Salesman Problem

Theorem 4.2 The dimension of $HP(V)$ is:

$$\dim(HP(V)) = \frac{n(n-1)}{2} - 1, \quad \text{where } n = |V| \geq 3. \quad (4.3)$$

Proof. Since every Hamiltonian path $P$ on $V$ must satisfy implicit equation $x^P(E(V)) = n - 1$ for $HP(V)$, we have

$$\dim(HP(V)) \leq \frac{n(n-1)}{2} - 1. \quad (4.4)$$

Let $fx = f_0$ be any equation satisfied by all incidence vectors $x \in HP(V)$. For any distinct $e_1, e_2 \in E(V)$, there exists a Hamiltonian cycle $C$ in $K(V)$ containing $e_1$ and $e_2$. Since $P_1 = C \setminus e_1$ and $P_2 = C \setminus e_2$ are two Hamiltonian paths in $K(V)$,

$$0 = f^{P_1} - f^{P_2} = f_{e_2} - f_{e_1}.$$ 

This shows that $f_e$ is constant for all $e \in E(V)$, and equation $fx = f_0$ must be a multiple of implicit equation $x(E(V)) = n - 1$. \hfill \Box

Remark 4.3 By Theorem 4.2, two valid inequalities $cx \leq c_0$ and $c'x \leq c'_0$ for $HP(V)$ are equivalent iff there exist two scalars $\alpha > 0$ and $\beta$ such that $c'_0 = \alpha c_0 + (n-1)\beta$ and $c'_e = \alpha c_e + \beta$ for all $e \in E(V)$. This equivalence greatly simplifies indirect proofs of facetial results for $HP(V)$. \hfill \Box

Corollary 4.4 The dimension of $STSP(V')$ is:

$$\dim(STSP(V')) = \frac{n(n+1)}{2} - (n+1). \quad (4.5)$$

Proof. From Lemma 4.1 and Theorem 4.2, it follows that

$$\dim(STSP(V')) = \dim(HP(V)) = \frac{n(n-1)}{2} - 1 = \frac{n(n+1)}{2} - (n+1). \quad \Box$$
Remark 4.5 D. Naddef points out that such dimensionality results are equivalent to characterizations of the constant $TSP$ (see [25], section 2). We therefore also obtain as a corollary a simple proof of the theorem of Berenguer (theorem 1 in Gilmore et al. [25]). □

Closed form representations for inequalities $ay \leq a_0$ are used in the figures hereafter. For each inequality with nonnegative coefficients, a collection of node subsets with positive weights (numbers) are given. Weights are indicated next to the corresponding sets and all set weights of value one are omitted for simplicity. Then, each coefficient $a_{ij}$ equals the total weight of all the subsets which contain both nodes $i$ and $j$.

![Figure 4.1: Equivalent envelope inequalities](image)

(a) An Envelope  
(b) $h$-normal form  
(c) $h'$-normal form

Figure 4.1 presents such representations. For instance, in Fig. 4.1(a), $a_0 = 9$, $a_{dh} = 2$, $a_{dh'} = a_{dh''} = 1$ and $a_{hd'} = a_{hh'} = a_{dd'} = 0$, etc.

Now, we turn to relationships between valid inequalities for $STSP(V')$ and those for $HP(V)$. Recall that $V' \setminus V = \{h\}$. Consider two inequalities $ay \leq a_0$ and $cx \leq c_0$, defined on $STSP(V')$ and $HP(V)$, respectively, and satisfying:

$$a_0 = c_0; \quad a_e = c_e \geq 0, \quad \forall e \in E(V); \quad \text{and} \quad a_e = 0, \quad \forall e \in E(h : V).$$

Then, $ay \leq a_0$ is said to have an isolated node $h$ (cf. Fig. 4.1(b)). Inequality $cx \leq c_0$ is the $h$-restriction of $ay \leq a_0$, and conversely, $ay \leq a_0$ is an extension of $cx \leq c_0$.

By Lemma 4.1 and Theorem 4.2, we have immediately:
Theorem 4.6 (Polyhedral Equivalence) For any integer \( k \) such that \( 0 \leq k \leq n(n-1)/2 - 1 \), the inequality \( ay \leq a_0 \) with isolated node \( h \) defines a \( k \)-dimensional face of \( STSP(V') \) iff its \( h \)-restriction \( cx \leq c_0 \) defines a \( k \)-dimensional face of \( HP(V) \).

The above theorem, along with Remark 4.3, provides a projective approach to the study of STS polytopes. This theorem also suggests the following unique representation of an STSP valid inequality w.r.t. a given node \( h \in V' \).

\( h \)-normal form. An inequality \( ay \leq a_0 \) for \( STSP(V') \) is in normal form w.r.t. a given node \( h \in V' \) (\( h \)-normal form, for short) if

(C1) \( a_e = 0 \) for all \( e \in E(\{h\} : V' \setminus \{h\}) \);

(C2) \( a \geq 0 \) and \( a_e = 0 \) for some \( e \in E(V' \setminus \{h\}) \);

(C3) \( \min\{a_e > 0 : e \in E(V')\} = 1. \)

Inequality \( cx \leq c_0 \) for \( HP(V) \) is in normal form if its extension \( ay \leq a_0 \) for \( STSP(V \cup \{h\}) \), where \( h \not\in V \), is in \( h \)-normal form. See Figure 4.1 for examples.

Remark 4.7 Balas and Fischetti [6] use a similar notion, \( h \)-canonical form, to uniquely represent ATSP inequalities. We use this notion here for STSP inequalities with a different scaling in (C3). Such forms can also be obtained by applications of Remark 4.2 of Grötschel and Padberg [27].

Any valid inequality \( ay \leq a_0 \) for \( STSP(V') \) admits a unique \( h \)-normal form \( a'y \leq a'_0 \) w.r.t. any given node \( h \in V' \). This can be obtained by the following \( h \)-normalization procedure.

\( h \)-Normalization procedure

**Input:** A valid inequality \( ay \leq a_0 \) for \( STSP(V') \), where \(|V'| = n + 1\), and any given projection node \( h \in V' \).
Output: An equivalent valid inequality $a'y \leq a'_0$ for $STSP(V')$ in $h$-normal form.

Step 1: $h$-Projection. Adding to $ay \leq a_0$ the following linear combination of degree constraints:

$$
\sum_{v \in V' \setminus \{h\}} -a_{vh}y(\delta(v)) = \sum_{v \in V' \setminus \{h\}} -2a_{vh}
$$

yields $\bar{a}y \leq \bar{a}_0$. Note that after the projection, $\bar{a}_e = 0$ for all $e \in \delta(h)$.

Step 2: Shifting. Adding to $\bar{a}y \leq \bar{a}_0$ the implicit equation $\lambda y(E(V' \setminus \{h\})) = \lambda(n - 1)$, for $STSP(V')$, where

$$
\lambda = -\min\{\bar{a}_e : e \in E(V' \setminus \{h\})\},
$$

yields $\bar{a}y \leq \bar{a}_0$. Note that after shifting, $\bar{a}_0 = \bar{a}_0 + \lambda(n - 1)$, $\bar{a} \geq 0$, and $\bar{a}_e = \bar{a}_e + \lambda$ if $e \in E(V' \setminus \{h\})$ and zero otherwise. Note also that for some $e \in E(V' \setminus \{h\})$, $\bar{a}_e = 0$.

Step 3: Scaling. The scaling operation is defined by

$$(a', a'_0) \doteq (\bar{a}, \bar{a}_0)/\gamma,$$

where $\gamma = \min\{\bar{a}_e > 0 : e \in E(V')\}$.

For instance, Fig. 4.1(a) represents inequality $ay \leq a_0$. After the normalization procedures w.r.t. nodes $h$ and $h'$, we obtain its $h$-normal form in Fig. 4.1(b) and $h'$-normal form in Fig. 4.1(c), respectively. Observe that in this example, the $h$-normal form in Fig. 4.1(b) is obtained by simply complementing set $\{h, d\}$: adding to $ay \leq a_0$ the following implicit equations for $STSP(V')$:

$$
-y(\delta(v)) = -2, \forall v \in \{h, d\}; \quad y(\delta(v)) = 2, \forall v \in V' \setminus \{h, d\}.
$$

(4.6)
To obtain the $h'$-normal form in Fig. 4.1(c), we need two successive set complementations.

$h$-projected inequality, or $h$-projection. Let $a'y \leq a'_0$ be the $h$-normal form of $ay \leq a_0$ for $STSP(V')$, and let $cx \leq c_0$ be the $h$-restriction of $a'y \leq a'_0$. Then $cx \leq c_0$ for $HP(V' \setminus \{h\})$ is an $h$-projected inequality or $h$-projection of $ay \leq a_0$.

Two important observations immediately follow from the above procedure. First, the normalization procedure requires $O(E(V'))$ computational steps. Thus, we obtain a linear time algorithm to check whether or not two valid inequalities define the same face (facet) of $STSP(V')$. Second, the coefficients of the $h$-normal form $a'y \leq a'_0$ satisfy: $a' \geq 0$, $a'_e = 0$ for all $e \in \delta(h)$ and $a'_e = 0$ for some $e \in E(V' \setminus \{h\})$. These edges correspond to linearly independent columns of the node-edge incidence matrix of $K(V')$.

By Lemma 6.3 in [29], it follows that $a'y \leq a'_0$ is in support-reduced form, and therefore $a'y \leq a'_0$ is facet-defining for the monotone STS polytope $MTSP(V')$ if it defines a nontrivial facet of $STSP(V')$.

Remark 4.8 A similar $h$-projection technique was also introduced independently by Araque [2] in his polyhedral study of vehicle routing problems (VRP) with single depot and variable fleet size. Consider a VRP with one vehicle and a single depot node $h$. Then, the coefficients of $\bar{a}$ in Step 1 of the $h$-Normalization procedure satisfy $-\bar{a}_{ij} = a_{hi} + a_{kj} - a_{ij} \equiv s_{ij}$ for all $(i, j) \in E(V' \setminus \{h\})$, where $s_{ij}$, as observed by Araque [2], is exactly the saving defined by Clarke and Wright [11].

From results in [28] and the polyhedral equivalence, we obtain simple classes of facet-defining inequalities for $HP(V)$.

Proposition 4.9 Let $\delta(v) \doteq E(\{v\} : V \setminus \{v\})$ for all $v \in V$, and let $n \doteq |V|$. The following is a system of facet-defining inequalities for $HP(V)$, where $n \geq 4$, no two of which are equivalent.
(a) $x_e \geq 0, \quad \forall e \in E(V)$;
(b) $x(\delta(v)) \leq 2, \quad \forall v \in V$;
(c) $x(E(W)) \leq |W| - 1, \quad \forall W \subseteq V$ and $2 \leq |W| \leq n - 1$.

Note that for $|W| = n - 1$, (c) can be restated as the lower degree constraint $x(\delta(v)) \geq 1, \{v\} = V \setminus W$ for $HP(V)$. Accordingly, constraints (b) are called the upper degree constraints. Note also that, for all $v \in V$, the degree constraint $x(\delta(v)) \geq 1$ (resp., $x(\delta(v)) \leq 2$) is equivalent to $x_{vh} \leq 1$ (resp., $x_{vh} \geq 0$) for $STSP(V \cup \{h\})$, where $h \notin V$. For $W = \{i, j\}$, (c) is equivalent to the simple upper bound constraint $x_{ij} \leq 1$ for $HP(V)$. Constraints (b) and (c) are in normal form. The normal form for the nonnegativity constraints (a) is

$$x(E(V) \setminus e) \leq n - 1, \quad \forall e \in E(V).$$

In accordance with the definition of STSP trivial facets, we also define HPP trivial facets.

**STSP Trivial Facets.** Inequality $ay \leq a_0$ defines a trivial facet of $STSP(V')$ if it is equivalent to $y_e \geq 0$ for some $e \in E(V')$. (cf. [26,29])

**HPP Trivial Facets.** Inequality $cx \leq c_0$ defines a trivial facet of $HP(V)$ if it is equivalent either to $x_e \geq 0$ (with normal form: $x(E(V) \setminus \{e\}) \leq |V| - 1$) for some $e \in E(V')$ or to $x(\delta(v)) \leq 2$ for some $v \in V$. (cf. Proposition 4.9(a),(b).)

The next lemma exhibits an important property of HPP nontrivial facet-defining inequality. This property will be frequently (and sometimes implicitly) used in the proofs of our polyhedral results.

**Lemma 4.10** Let $cx \leq c_0$ be in normal form and define a nontrivial facet of $HP(V)$. Then
(1) for every edge $e \in E(V)$, there exists a c-tight path $P$ on $V$ such that $e \in P$; and
(2) for every node $v \in V$, there exists a c-tight path $P$ on $V$ such that $v$ is one of its endnodes.

Proof: (1) follows since otherwise $cx \leq c_0$ is equivalent to $x_e \geq 0$ for some $e \in E(V)$.
(2) holds since otherwise $cx \leq c_0$ is equivalent to $x(\delta(v)) \leq 2$ for some $v \in V$. •

4.4 Clique-Lifting

In this section, we discuss general clique-lifting results for extending known STSP facets. First, we review related notions and results. Then, we show that any nontrivial STSP facet can be extended to a lifted facet by replacing any node with a clique of size greater than two. Finally, we prove a necessary and sufficient condition for node-cloning (i.e., clique-lifting of size two), extending Naddef and Rinaldi’s zero-lifting result in [41].

Definition 4.A (c-Connectedness, adapted from Naddef and Rinaldi [41]).
Nodes $u, v \in V$ are c-adjacent w.r.t. the valid inequality $cx \leq c_0$ for $HP(V)$ if there exists a c-tight path $P$ on $V$ such that $u$ and $v$ are the endnodes of $P$. A subset $U \subset V$ of nodes is c-connected if for any pair of nodes $p, q \in U$, there exists a c-adjacent node sequence $u_1, u_2, ..., u_l \in U$ with $u_1 = p$ and $u_l = q$ such that $u_i$ and $u_{i+1}$ are c-adjacent for every $i = 1, ..., l - 1$.

This c-connectedness condition is equivalent to the Partition Condition, (PC) for short, as shown below.

Lemma 4.11 (Partition Condition) Let $ay \leq a_0$ be a valid inequality for $STSP(V')$ and $cx \leq c_0$ be its $h$-projection for $HP(V)$, where $V = V' \setminus \{h\}$. Then the following statements are equivalent.
(1) $V$ is c-connected.
(2) For every proper partition \( \{U, U'\} \) of \( V \), there exists a \( c \)-tight path \( (u...u') \) on \( V \) such that \( u \in U \) and \( u' \in U' \).

(3) \((PC)\): For every proper partition \( \{U, U'\} \) of \( V \), there exists an \( a \)-tight cycle \( C \) on \( V' \) containing a 3-node subchain \( (uhu') \) with \( u \in U \) and \( u' \in U' \).

Proof: If there exists a proper partition \( \{U, U'\} \) violating statement (2), \( V \) is clearly not \( c \)-connected. Conversely, let \( (u'...v') \) be a \( c \)-tight path on \( V \) and initially let \( U = \{u', v'\} \).

Observe that \( U \) is \( c \)-connected. Then, repeat the following procedure.

(A) Using statement (2), there exists a \( c \)-tight path \( (u...v) \) on \( V \) for some \( u \in U \) and \( v \in U' = V \setminus U \). Hence, \( U \cup \{v\} \) is \( c \)-connected. Replace \( U \) by \( U \cup \{v\} \). Repeat (A) until \( U = V \).

This shows that statements (1) and (2) are equivalent. Finally, note that by polyhedral equivalence, statements (2) and (3) are equivalent. \( \square \)

The \( c \)-connectedness property plays an important role in deriving our main lifting results. A simple sufficient condition for this property to hold w.r.t. a nontrivial facet-defining HPP inequality \( cx \leq c_0 \) is given below.

**Lemma 4.12 (Sufficient Condition for \( c \)-Connectedness)** Let \( cx \leq c_0 \) define a nontrivial facet of \( HP(V) \) with \( c \geq 0 \), and let \( \overline{G}_c(V) \) be its zero-graph (cf. (3) in Section 2). Suppose that the set \( U \subseteq V \) induces a connected subgraph \( \overline{G}_c(U) \) of \( \overline{G}_c(V) \). Then \( U \) is \( c \)-connected.

Proof: Since \( cx \leq c_0 \) is nontrivial, for any \( (u, v) \) with \( c_{uv} = 0 \), there exists a \( c \)-tight path \( (p...uv...q) \) on \( V \) containing \( (u, v) \). Thus, \( (u...pq...v) \) is also a \( c \)-tight path, showing \( u \) and \( v \) are \( c \)-adjacent. Further, since \( \overline{G}_c(U) \) is connected, there exists a \( c \)-adjacent node sequence connecting any pair of nodes \( u, v \in U \). Hence, \( U \) is \( c \)-connected. \( \square \)

**Definition 4.B (Clones, Balas and Fischetti [6])**.
A pair of nodes $h$ and $t$ in $V'$ are called clones, or form a clone pair, w.r.t. $ay \leq a_0$ if

(C1) $a_{hv} = a_{tv}$ for all $v \in V' \setminus \{h, t\}$, and

(C2) $a_{ht} = \max\{a_{uh} + a_{vh} - a_{uv} : u, v \in V' \setminus \{h, t\}, u \neq v\}.$

An inequality $ay \leq a_0$ for $STSP(V')$ is called primitive (or simple [41]) if there exists no clone pair. In Figure 4.2, (a) represents a primitive inequality, where in (b), $h'$ and $s'$ (resp., $h$ and $s$) form a clone pair.

**Definition 4.4** (Clique-Lifting, Zero-Lifting and Node-Cloning).

Let $ay \leq a_0$ be a valid inequality for $STSP(V')$. For any given $h \in V'$, define clique weight

$$\xi = \max\{a_{ih} + a_{jh} - a_{ij} : i, j \in V' \setminus \{h\}, i \neq j\}. $$

Consider any node set $S$ such that $S \cap V' = \emptyset$. We say that the inequality $a^*y^* \leq a_0^*$ for $STSP(V' \cup S)$ is obtained by clique-lifting of $ay \leq a_0$ w.r.t. to node $h$ if $a_0^* = a_0 + \xi|S|$, and

$$a_{uv}^* = \begin{cases} 
    a_{uv}, & u \in V', v \in V'; \\
    a_{hv}, & u \in S, v \in V' \setminus \{h\}; \\
    \xi, & u \in S \cup \{h\}, v \in S \cup \{h\}. 
\end{cases} \quad (4.7)$$

It is a zero-lifting if $\xi = 0$. It is called node-cloning if $|S| = 1$. The facet-defining inequality $ay \leq a_0$ is clique-liftable (resp., node-clonable) w.r.t. $h$ if the lifted inequality $a^*y^* \leq a_0^*$ is facet-defining for any nonempty $S$ (resp., singleton set $S$).

Clique-Lifting can be viewed as replacing $h$ by the clique ($\{h\} \cup S, E(\{h\} \cup S)$) with clique weight $\xi$. Fig. 4.2(b) represents the lifted inequality obtained by two successive node-cloning of the inequality in Fig. 4.2(a) w.r.t. nodes $h'$ and $h$, respectively.

Clique-Lifting, equivalent to $|S|$ successive node-clonings on $h$, yields a valid STSP inequality, as shown by the following two lemmata.
Chapter 4. Hamiltonian Path and Symmetric Travelling Salesman Problem

Lemma 4.13 (Property of a Node-Cloned Inequality) Let $a_y \leq a_0$ be an inequality for $STSP(V')$, and let nodes $h, t$ be a clone pair w.r.t. $a_y \leq a_0$. If the inequality $a^* y^* \leq a_0^*$ for $STSP(V' \cup \{s\})$ is obtained by node-cloning on $h$, then $a^*_hs = a^*_ts = a_{ht}$.

Proof: Since $h$ and $t$ form a clone pair, node-cloning of $a_y \leq a_0$ w.r.t. $h$ yields

\[
a^*_hs = \max\{a_{ih} + a_{jh} - a_{ij} : i, j \in V' \setminus \{h\}, i \neq j\} = \max\{a_{ht}, \max\{a_{ht} + a_{hv} - a_{tv} : v \in V' \setminus \{h, t\}\}\} = a_{ht}.
\]

Similarly, we show that $a^*_ts = a_{ht}$.

Lemma 4.14 (Validity of Clique-Lifting) Let $a_y \leq a_0$ be any valid inequality for $STSP(V')$, and let the inequality $a^* y^* \leq a_0^*$ be obtained by clique-lifting of $a_y \leq a_0$ w.r.t. any given node $h$. Then $a^* y^* \leq a_0^*$ is valid for $STSP(V' \cup S)$.

Proof: Consider first $S = \{s\}$. If there exists an $a^*$-tight cycle $C^* = (usv...r)$ on $V' \cup \{s\}$ such that $a^*(C^*) > a_0^*$, then $C = (uv...r)$ is a Hamiltonian cycle on $V'$ with

\[
a(C) = a^*(C^*) - a_{us} - a_{vs} + a_{uv} = a^*(C^*) - a_{uh} - a_{vh} + a_{uv} > a_0^* - a^*_{hs} = a_0,
\]

a contradiction. Hence, node-cloning yields a valid inequality. Using induction on $|S|$, and Lemma 4.13, the proof is complete.
Node-cloning or clique-lifting are equivalent to zero-lifting of an STSP inequality by $\geq b_0$ in tight triangular form. Recall the definition of this form below.

**Definition 4.D (TT-Form, Naddef and Rinaldi [41]).**

Inequality $by \geq b_0$ is a tight triangular inequality, or in tight triangular form (TT-inequality, or TT-form, for short) if

1. (T1) for all distinct nodes $u, v, w \in V'$, the triangular inequality $b_{uv} \leq b_{uw} + b_{wv}$ holds; and
2. (T2) for any $w \in V'$, there exists a pair of nodes $u, v \in V' \setminus \{w\}$ such that $b_{uv} = b_{uw} + b_{wv}$.

Note that any STSP inequality has a unique (up to positive scaling) TT-representation.

**Proposition 4.15 (Characterizations of Primitive Inequalities)** Let $ay \leq a_0$ be any valid inequality for $STSP(V')$. For any node $h \in V'$, let $a^h y \leq a^h_0$ be the $h$-normal form of $ay \leq a_0$ and $G_{a^h}(V')$ be its support graph. Then the following statements are equivalent:

1. (a) $ay \leq a_0$ is a primitive inequality.
2. (b) There exists no clone pair w.r.t. $ay \leq a_0$.
3. (c) The support graph $G_{a^h}(V')$ contains exactly one isolated node $h$ for any $h \in V'$.
4. (d) The coefficients of the TT-form $by \geq b_0$ of $ay \leq a_0$ are all positive.

**Proof:** (a) $\iff$ (b) by definition. To prove (b) $\iff$ (c), observe that by the uniqueness property of $h$-normal form, there exists a clone pair $h, t \in V'$ w.r.t. $ay \leq a_0$ iff both $h$ and $t$ are isolated nodes in $G_{a^h}(V')$. The latter is equivalent to $b_{ht} = 0$, proving (c) $\iff$ (d).

From this proposition, it follows that node-cloning of $ay \leq a_0$ w.r.t. node $h$ is equivalent to zero-lifting of its TT-form $by \geq b_0$ w.r.t. $h$ (or, equivalently, node-cloning of $-by \leq -b_0$ with weight $\xi \equiv 0$).
We now present the first main result of this section.

**Theorem 4.16 (Clique-Lifting with \(|S| \geq 2\)**) Let \(ay \leq a^0\) be any nontrivial facet-defining inequality for \(STSP(V')\). Then, inequality \(a^*y^* \leq a^*_0\) obtained by clique-lifting of \(ay \leq a^0\) w.r.t. any node \(h\) is facet-defining for \(STSP(V' \cup S)\), whenever \(|S| \geq 2\).

**Proof.** By Lemma 4.14, \(a^*y^* \leq a^*_0\) is valid. Let \(V = V' \setminus \{h\}\).

Let \(cx \leq c^0\) and \(c^*x^* \leq c^*_0\) be the \(h\)-projections of \(ay \leq a^0\) and \(a^*y^* \leq a^*_0\), respectively. By Lemma 4.13, Lemma 4.14 and Proposition 4.15, \(c^*x^* \leq c^*_0\) is a valid inequality for \(HP(V \cup S)\) with \(c^*_e = c^0; c^*_e = c_e\) for all \(e \in E(V)\) and zero otherwise. Let \(f_x^* \leq f_0\) be a facet-defining inequality for \(HP(V \cup S)\) that dominates \(c^*x^* \leq c^*_0\).

(a) Since \(cx \leq c^0\) is nontrivial, for any distinct nodes \(i, j \in S\) and \(v \in V\), let \((u...v)\) be a \(c\)-tight path on \(V\) and \((ij...k)\) be a Hamiltonian path on \(S\). The Hamiltonian cycle \(C = (ij...ku...v)\) on \(V \cup S\) satisfies \(c^*(C) = c^*_0\). Observe that \(C \setminus (i, j)\) and \(C \setminus (i, v)\) are \(c^*\)-tight paths, and hence \(f_{iv} = f_{ij}\). Repeating (a) shows that there exists \(\beta \geq 0\) such that \(f_e = \beta\) for all \(e \in E(S) \cup E(S : V)\).

(b) For any \(c\)-tight path \(P\) on \(V\), there exists an extended \(c^*\)-tight path \(P^*\) on \(V \cup S\) containing \(P\). Using (a), it follows that \(f_0 = f(P^*) = f(P) + \beta |S|\) for all \(c\)-tight path \(P\) on \(V\).

Since \(cx \leq c^0\) is facet-defining, there exist two scalars \(\alpha \geq 0\) and \(\gamma\) such that \(f_e = \alpha c_e + \gamma\) for all \(e \in E(V)\). Next, since \(cx \leq c^0\) is in normal form, there exists \(e = (p, q) \in E(V)\) such that \(c_e = 0\). Let \((u...pq...v)\) be a \(c\)-tight path on \(V\) and let \((ij...k)\) be a Hamiltonian path on \(S\). Comparing the two extended \(c^*\)-tight paths \((ij...ku...pq...v)\) and \((q...vij...ku...p)\) on \(V \cup S\) yields \(\gamma = f_{pq} = f_{vi} = \beta\). By Remark 4.3, \(f x^* \leq f_0\) is equivalent to \(c^*x^* \leq c^*_0\).

The proof is complete by polyhedral equivalence. \(\square\)
Chapter 4. Hamiltonian Path and Symmetric Travelling Salesman Problem

For any nontrivial facet-defining inequality $ay \leq a_0$ for $STSP(V')$ in $h$-normal form, the above lifting result immediately implies that it also induces a facet of $STSP(V' \cup S)$ for all $|S| \geq 2$.

We now turn to node-cloning. The following general node-cloning result is due to Naddef and Rinaldi [41], and an alternate (simpler) proof is provided below for completeness.

**Proposition 4.17 (Sufficient Condition for Node-Cloning)** Let $ay \leq a_0$ define a nontrivial facet for $STSP(V')$, and let $cx \leq c_0$ be its $h$-projected inequality for $HP(V)$, where $V = V' \setminus \{h\}$. If the condition $PC$ stated in Lemma 4.11 is satisfied, then $a^*y^* \leq a_0^*$, obtained by node-cloning of $ay \leq a_0$ w.r.t. $h$, is facet-defining for $STSP(V' \cup \{s\})$.

**Proof:** Note first that $V$ is $c$-connected by Lemma 4.11. Now, let $c^*x^* \leq c_0^*$ be an $h$-projected inequality of $a^*y^* \leq a_0^*$ and $fx^* \leq f_0$ be a facet-defining inequality for $HP(V \cup \{s\})$ that dominates $c^*x^* \leq c_0^*$. If nodes $u, v \in V$ are $c$-adjacent, then there exists a $c$-tight path $(u...v)$ on $V$. Comparing the $c^*$-tight paths $P_1 = (u...vs)$ and $P_2 = (su...v)$ yields $f_{us} = f_{vs}$. Since $V$ is $c$-connected, we deduce that there exists some scalar $\beta$ such that $\beta = f_{us}$ for all $v \in V$. Next, since $cx \leq c_0$ defines a facet of $HP(V)$ and every $c$-tight path $P = (u...v)$ on $V$ satisfies:

$$f(P) = f(P \cup (v, s)) - f_{us} = f_0 - \beta,$$

it follows that for some scalars $\alpha \geq 0$ and $\gamma$, we have $f_e = \alpha c_e + \gamma$ for all $e \in E(V)$. The rest of the proof is similar to the last part of the proof of Theorem 4.16: since $cx \leq c_0$ is in normal form, there exists an edge $(p, q) \in E(V)$ with $c_{pq} = 0$. Let $(u...pq...v)$ be a $c$-tight path on $V$. Comparing the $c^*$-tight paths $(u...pq...vs)$ and $(q...vsu...p)$ on $V \cup \{s\}$ yields $\gamma = f_{pq} = f_{us} = \beta$. Hence, $c^*x^* \leq c_0^*$ is equivalent to $fx^* \leq f_0$. The proof is complete by polyhedral equivalence. □
The next result implies that if node $h$ is a clone to some other node, then $ay \leq a_0$ is node-clonable w.r.t. node $h$.

**Corollary 4.18 (Lifting of Clone Pairs)** Let $ay \leq a_0$ define a nontrivial facet of $STSP(V')$ with $a \geq 0$, and suppose that nodes $h$ and $t$ are clones w.r.t. $ay \leq a_0$. Then the inequality $a^*y^* \leq a_0^*$ for $STSP(V' \cup \{s\})$ obtained by node-cloning of $ay \leq a_0$ w.r.t. $h$ (or $t$) is facet-defining.

**Proof:** Let $cx \leq c_0$ be the $h$-projected inequality for $HP(V)$ w.r.t. $ay \leq a_0$, where $V \cong V' \setminus \{h\}$. Observe that the support graph $G_c(V)$ contains an isolated node $t$, and thus its complement, the zero-graph $\overline{G_c}(V)$ is connected. By **Lemma 4.12**, $V$ is $c$-connected. The proof is complete by **Proposition 4.17**. □

As an application of **Corollary 4.18**, consider Theorem 4.12 (i) in Grötschel and Padberg [27]. As any two nodes in their set $Z$ form a clone-pair, their result follows directly from **Corollary 4.18**.

The next theorem shows that Naddef and Rinaldi's node-cloning condition in **Proposition 4.17** is also necessary.

**Theorem 4.19 (Necessary and Sufficient Condition for Node-Cloning)** Let $ay \leq a_0$ be any nontrivial facet-defining inequality for $STSP(V')$. Let $a^*y^* \leq a_0^*$ be an inequality for $STSP(V' \cup \{s\})$ obtained by node-cloning of $ay \leq a_0$ w.r.t. node $h$. Then $a^*y^* \leq a_0^*$ is facet-defining if and only if the following Partition Condition holds:

**(PC)** for every proper partition $\{U, U'\}$ of the node set $V' \setminus \{h\}$, there exists an $a$-tight cycle $C$ on $V'$ containing a 3-node subchain $(uhu')$ with $u \in U$ and $u' \in U'$.

**Proof:** Let $cx \leq c_0$ and $c^*x^* \leq c_0^*$ be the $h$-projected inequalities of $ay \leq a_0$ and $a^*y^* \leq a_0^*$, respectively. Let $V \cong V' \setminus \{h\}$.

** Sufficiency** trivially follows from **Proposition 4.17**.
Necessity. Suppose that (PC) does not hold. Then by Lemma 4.11(2), there exists a proper partition \( \{U, U'\} \) of \( V \) such that no \( c \)-tight path on \( V \) exists with one endnode in \( U \) and the other in \( U' \).

Consider any \( c^* \)-tight path \( P^* \) on \( V \cup \{s\} \). Note that \( P^* \) has one of the following forms: 
\((su...v)\); or \((u...psq...v)\). Observe that \( c(P^* \cap E(V)) = c_0 \). For the former form, either \( u, v \in U \) or \( u, v \in U' \); and for the latter, either \( u, v, p, q \in U \) or \( u, v, p, q \in U' \). (Otherwise \( P^* \cap E(V) \) can be extended to a \( c \)-tight path on \( V \) with one endnode in \( U \) and the other in \( U' \), contradicting the supposition above.) In either case, the incidence vector \( x^* \) of \( P^* \) satisfies the equality \( 2x^*(E(U \cup \{s\}) + x^*(E(U : U')) = |U| \). It follows that the incidence vectors of all \( c^* \)-tight paths \( P^* \) satisfy this equation, which is linearly independent of two other linearly independent equations: \( c^*x^* = c_0^* \) and \( x^*(E(V \cup \{s\})) = |V| \) (Note that by Proposition 4.15(c), \( c^*_e = 0 \) for all \( e \in E(\{s\} : V) \) since \( h, s \) is a clone pair w.r.t. \( a^*y^* \leq a_0^* \)). This implies that \( c^*x^* \leq c_0^* \), and thus \( a^*y^* \leq a_0^* \) (by polyhedral equivalence), is not facet-defining.

---

As already observed by Naddef and Rinadi [41], such node-cloning operations are repeatedly applicable. Large classes of STSP facets can be obtained in this fashion. This observation is justified by the following theorem.

**Theorem 4.20 (Repeated Node-Cloning and Clique-Lifting)** Let \( ay \leq a_0 \) be any nontrivial facet-defining inequality for \( STSP(V') \). Suppose that:

(a) \( ay \leq a_0 \) is node-clonable w.r.t. node \( h \in V' \); and

(b) the inequality \( a^*y^* \leq a_0^* \) for \( STSP(V^*) \), obtained by clique-lifting of \( ay \leq a_0 \) w.r.t. any node \( t \in V' \) where \( V^* = V' \cup S \), is facet-defining.

Then \( a^*y^* \leq a_0^* \) is node-clonable w.r.t. any node in \( \{h, t\} \cup S \).

**Proof:** Since any pair of nodes in \( \{t\} \cup S \) forms a clone pair, by Corollary 4.18, \( a^*y^* \leq a_0^* \)
is node-clonable w.r.t. any node in \{t\} \cup S. If t = h, then we are done. Else we show that \(a^*y^* \leq a_0^*\) is node-clonable w.r.t. \(h\), by verifying condition (PC).

Consider any proper partition \(\{U, U'\}\) of \(V^* \setminus \{h\}\). Define a partition \(\{Q, Q'\}\) of \(V' \setminus \{h\}\) corresponding to the following two possible cases:

Case (1). If neither \(U\) nor \(U'\) is contained in \(S\), define \(Q = U \setminus S\) and \(Q' = U' \setminus S\).

Case (2). W.l.o.g. assume that \(S \subseteq U\), and define \(Q = \{t\}\) and \(Q' = V' \setminus \{h, t\}\).

By (a),(b) and Theorem 4.19, there exists an \(a\)-tight cycle \(C\) on \(V'\) containing subchain \((uhu')\) with \(u \in Q\) and \(u' \in Q'\) (condition (PC)). In either case, we obtain an \(a^*\)-tight cycle \(C^*\) on \(V^*\) by replacing node \(t\) in \(C\) with a Hamiltonian path \((t...s)\) on \(\{t\} \cup S\) so that this \(C^*\) contains \((uh'u')\) with \(u \in U\) and \(u' \in U'\). (Note that for Case (2), \(u = t\).)

Hence condition (PC) holds w.r.t. node \(h\). \(\square\)

The condition of the above theorem is weaker than the condition of Theorem 4.9 in Naddef and Rinadi [41], where it is required that every node in \(V'\) be node-clonable. Consider a proper subset \(U \subset V'\), and suppose that a nontrivial facet-defining inequality \(ay \leq a_0\) for \(STSP(V')\) is node-clonable w.r.t. every node in \(U\). Let \(a^*y^* \leq a_0^*\) for \(STSP(V^*)\) be obtained by repeated clique-lifting of \(ay \leq a_0\) w.r.t. every node \(v \in V' \setminus U\), such that each such node \(v\) is replaced by a clique of size greater than two. By Theorem 4.16, \(a^*y^* \leq a_0^*\) is facet-defining. Then Theorem 4.20 implies that \(a^*y^* \leq a_0^*\) is node-clonable w.r.t. any node in \(V^*\), but Theorem 4.9 does not.

4.5 Simple Conditions for Node-Cloning

As condition (PC) may be difficult to verify, we present in this section three simple sufficient conditions for node-cloning. The first one may be checked in linear time and seems to apply to all known STSP facets. The second condition, which is verified by
visual inspection, generalizes clique-lifting results in Maurras [38], and Grötschel and Padberg [27]. The third condition allows cloning any node in a facet-defining inequality with 0-1 coefficients (a so-called rank inequality). We do not know of any STSP facet that cannot be lifted, based on these simple conditions.

Recall that the zero-graph $G_c$ is the complement of the support graph $G^c$ associated with an HPP inequality $cx \leq c_0$.

Let $ay \leq a_0$ be any inequality for $STSP(V')$.

**Definition 4.4 (Condition $B(h, d; u)$).** The inequality $ay \leq a_0$ is said to satisfy condition $B(h, d; u)$ with $h, d \in V'$ and scalar $u \geq 1$ if $ay \leq a_0$ is in $h$-normal form with $h$-restriction denoted by $cx \leq c_0$, and if there exists a partition $\{h, d, U, U'\}$ of $V'$ such that:

**B1.** $U = \{v \in V' \setminus \{h, d\} : c_{dv} > 0\}$ and $U' = \{v \in V' \setminus \{h, d\} : c_{dv} = 0\}$;

**B2.** if $U \neq \emptyset$, then $c_e \leq u$ for all $e \in E(d : U)$ and $c_e \geq u$ for all $e \in E(U)$.

Note that $ay \leq a_0$ in $h$-normal form ensures the existence of a partition $\{h, d, U, U'\}$ satisfying (B1) and $U' \neq \emptyset$, because $a_e = c_e = 0$ for some $e \in E(V' \setminus \{h\})$.

Let $ay \leq a_0$ be a star inequality [21], and let $a'y \leq a'_0$ be its $h$-normal form w.r.t. any node $h$. Then $a'y \leq a'_0$ satisfies condition $B(h, d; u)$ for some properly chosen $d$. (The $h$-normal forms of the star inequalities are easily obtained by appropriately complementing circles and spikes, cf. equation (4.6) and Figure 4.1.) For instance, Fig. 4.1(a) represents the smallest non-comb star inequality. It is easily verified that the inequality in Fig. 4.1(b) (resp., Fig. 4.1(c)) satisfies condition $B(h, d; 2)$ (resp., $B(h', d'; 1)$). Thus, the following lifting result applies to all facet-defining star inequalities w.r.t. any node.

Other known facet-defining inequalities, such as clique-tree, ladder, chain, and crown inequalities, etc., can also be shown to satisfy condition $B(h, d; u)$ w.r.t. any node $h$. In fact, we do not even know of any STSP facet-defining inequality whose $h$-normal form
violates condition \( B(h, d; \omega) \).

The following theorem deals with node-cloning of an STSP facet-defining inequality in \( h \)-normal form w.r.t. the isolated node \( h \). This lifting is a special case of zero-lifting.

**Theorem 4.21 (1-Node Zero-Lifting Theorem)** Let \( ay \leq a_0 \) be a nontrivial facet defining inequality for \( STSP(V') \) satisfying condition \( B(h, d; \omega) \). Then the inequality \( a^*y^* \leq a_0^* \) for \( STSP(V' \cup \{s\}) \), obtained by node-cloning of \( ay \leq a_0 \) w.r.t. \( h \), is facet-defining.

**Proof:** Let \( V \equiv V' \setminus \{h\} \), let \( cx \leq c_0 \) denote the \( h \)-restriction of \( ay \leq a_0 \). Let \( \{U, U'\} \) be a partition of \( V' \setminus \{h, d\} \) as required in condition \( B(h, d; s) \).

First, using (B1), \( U' \cup \{d\} \) induces a connected subgraph of the zero-graph \( \overline{G_c}(V) \), and thus by **Lemma 4.12**, \( U' \cup \{d\} \) is \( c \)-connected. Next, since \( cx \leq c_0 \) is nontrivial, for any \( w \in U \), there exists a \( c \)-tight path on \( V \) of the form \( P \equiv (w...t) \). If \( t \in U' \cup \{d\} \), then \( w \) is \( c \)-adjacent to some node in \( U' \cup \{d\} \). Otherwise \( t \in U \) and \( P \) has the form \( (w...udv...t) \). This implies that \( u \in U \) (since otherwise \( c_{ud} = 0 \) by (B1) and Hamiltonian path \( P' \equiv (u...wt...vd) \) satisfies \( c(P') > c_0 \)), and therefore by (B2), we have \( c_{ut} \geq c_{ud} \). Hence, \( (w...ut...vd) \) is a \( c \)-tight path on \( V \) and \( w \) is \( c \)-adjacent to \( d \). It follows that \( V \) is \( c \)-connected. The proof is complete by **Proposition 4.17** and polyhedral equivalence.

\( \Box \)

In some applications, it is more convenient to apply node-cloning to an STSP facet-defining inequality, not in \( h \)-normal form. In the sequel, we discuss sufficient conditions for such node-clonings. These conditions can be checked directly by visual inspection, using closed form representations.

**Definition 4.F (\( w \)-regular node).** Node \( h \in V' \) is \( \omega \)-regular w.r.t. \( ay \leq a_0 \) if \( \omega > 0 \) and there exists a proper partition \( \{h, Q, Q'\} \) of \( V' \) such that \( a_{hv} = \omega \) for all \( v \in Q \) and \( a_{hv} = 0 \) for all \( v \in Q' \).
Theorem 4.22 (ω-Regular Node-Cloning) Let $ay \leq a_0$ define any nontrivial facet of $STSP(V')$ with $a \geq 0$. Suppose that node $h$ is $\omega$-regular w.r.t. the partition $\{h, Q, Q'\}$. Let $V = V' \setminus \{h\}$. Suppose also that one of the following conditions is satisfied.

(i) $Q = \{d\}$; there exists a node $v' \in Q'$ such that $a_{dv'} = 0$ and $a_{dv} \leq \omega$ for all $v \in Q \setminus \{v'\}$.

(ii) For all $e \in E(Q) \neq \emptyset$, $a_e \geq \omega$; there exists a node $d \in Q$ such that $a_{dv} \leq \omega$ for all $v \in V \setminus \{d\}$.

(iii) There exists an edge $e = (i, j) \in E(Q)$ such that $a_e = 0$ and $a_e \leq \omega$ for all $e \in E(Q) \cup E(Q : Q')$.

Then the inequality $a^*y^* \leq a_0^*$, obtained by node-cloning of $ay \leq a_0$ w.r.t. node $h$, defines a facet of $STSP(V' \cup \{s\})$.

Proof: Let $a'y \leq a_0'$ be the $h$-normal form of $ay \leq a_0$ and $cx \leq c_0$ be the $h$-restriction of $a'y \leq a_0'$. Note that $a^*y^* \leq a_0^*$ is equivalent to the inequality obtained by node-cloning of $a'y' \leq a_0'$ w.r.t. $h$. We thus need only prove that $a'y' \leq a_0'$ is node-clonable w.r.t. $h$.

Note also that $V = Q \cup Q'$.

Case (i). Note first that clique weight $\xi = a_{hs}^* = a_{hd} + a_{hv'} - a_{dv'} = \omega$. Define $U = \{v \in V \setminus \{d\} : c_{dv} > 0\}$ and $U' = \{v \in V \setminus \{d\} : c_{dv} = 0\}$. Then $a'y' \leq a_0'$ satisfies condition $B(h, d; \omega)$.

Case (ii). Note first that clique weight $\xi = a_{hs}^* = a_{hd} + a_{hv} - a_{dv} = \omega$ for any $v \in Q \setminus \{d\}$. Observe that, in this case, $a'y' \leq a_0'$ may be obtained by complementing set $Q \cup \{h\}$, cf. equation (4.6). Thus,

$$a'_e = \begin{cases} 
  a_e - \omega, & e \in E(Q \cup \{h\}); \\
  a_e + \omega, & e \in E(Q'); \\
  a_e & \text{otherwise.}
\end{cases} \quad (4.8)$$

Also observe that (ii) implies $a_{dv} = \omega$, and thus $c_{dv} = 0$, for all $v \in Q \setminus \{d\}$. Further by (ii), we have that $c_{dv} \leq \omega$ for all $v \in Q'$, and $c_e \geq \omega$ for all $e \in E(Q')$. Hence, condition
Chapter 4. Hamiltonian Path and Symmetric Travelling Salesman Problem

$B(h, d; \omega)$ also holds w.r.t. $a'y' \leq a'_t$ and a partition $\{h, d, U, U'\}$, where $U \subseteq Q'$.

The proof for Cases (i) and (ii) is complete by Theorem 4.21.

Case (iii). Note that clique weight $\xi = a^*_h = a_h + a_{h^j} - a_{ij} = 2\omega$. Also observe that:

(O1) $c_e \leq \omega$ for all $e \in E(Q)$; and

(O2) $\omega \leq c_e \leq 2\omega$ for all $e \in E(Q : Q')$, and $c_e \geq 2\omega$ for all $e \in E(Q')$.

(1) We show first that $Q$ is $c$-connected. It suffices to show that for any proper partition $\{Z', Z''\}$ of $Q$, there exists a $c$-tight path on $V$ with one endnode in $Z'$ and the other in $Z''$. Fix $w \in Q'$ and let $P = (w...v)$ be a $c$-tight path on $V$. Consider the following two possible cases.

(1.a) If $v \in Q'$, then by (O2), $c_{wv} \geq 2\omega$. Therefore $P$ contains no edge $e^*$ with $c_{e^*} < 2\omega$ (since otherwise $c(P \cup \{(w, v)\} \setminus \{e^*\}) > c_0$). Therefore, by (O1), $P$ contains no edge in $E(Q)$, and $P$ has the general form $(w'...z'p...qz''...w'')$, where $\{w', w''\} = \{w, v\}$, $z' \in Z'$, $z'' \in Z''$ and $p, q \in Q'$. It follows from (O1) and (O2) that $c_{w'p} + c_{qw''} \geq 2\omega \geq c_{z'p} + c_{qz''}$. Thus $(z'...w'p...qw''...z'')$ is a $c$-tight path with $z' \in Z'$ and $z'' \in Z''$.

(1.b) If $v \in Q$, assume w.l.o.g. $v \in Z'$. Then $P$ has the general form $(w...z''u...v)$ for some $z'' \in Z''$. By (O1) and (O2), we have $c_{wu} \geq c_{z'u}$ for either $u \in Q$ or $u \in Q'$. Thus, $(z''...wu...v)$ is a $c$-tight path with $v \in Z'$ and $z'' \in Z''$.

(1.a) and (1.b) shows that $Q$ is $c$-connected.

(2) We now show that every node $w \in Q'$ is $c$-adjacent to some node in $Q$. In doing so, we consider a $c$-tight path on $V$: $P = (w...v)$ as in (1) above. If $v \in Q$, we are done for node $w$. Otherwise, $v \in Q'$, and $P$ has the general form $(w...pz...v)$ for some $z \in Q$. As in (1.a), $P$ contains no edge in $E(Q)$, and thus $p \in Q'$. Hence, by (O1) and (O2), we have $c_{pv} \geq c_{pz}$, implying that $(w...pv...z)$ is a $c$-tight path on $V$.

From (1) and (2), it follows that $V$ is $c$-connected. The proof of Case (iii) is complete by Proposition 4.17 and by polyhedral equivalence. □
We first observe that Maurras’ clique-lifting result (Prop.1, pp.188, [38]) is a special case of Theorem 4.22(ii) (and also a special case of Theorem 4.24 below). Next, Theorem 4.22 (i) and (ii) generalize Theorem 4.12 (ii) in Grötschel and Padberg [27] (Using $Z = \{h\}$). Finally, observe that Theorem 4.22 (iii) implies Theorem 5.10 in [27].

As examples, we apply $\omega$-regular node-cloning to two new facets in Fig. 4.2(a) and Fig. 4.3(a). (These inequalities were proven by the authors to be facet-defining, using the projective approach). Applying Theorem 4.22 (ii) and (iii) successively to nodes $h$ and $h'$ in Fig. 4.2(a), we obtain a new STSP facet in Fig. 4.2(b). Applying Theorem 4.22 (i) to node $h$ in Fig. 4.3(a) yields a new facet of $STSP(V')$ with $|V'| = 10$ in Fig. 4.3(b). These new facets cannot be generated by the clique-lifting results of Maurras, or Grötschel and Padberg.

Now, we make use of the above result to derive our third main lifting result, rank inequality lifting. We first show an intermediate result.

**Proposition 4.23 (Zero-Lifting for 0-1-Inequalities in $h$-normal form)** Let $ay \leq a_0$ define a nontrivial facet of $STSP(V')$ in $h$-normal form. Assume that $a_e \in \{0, 1\}$ for all $e \in E(V)$, where $V = V' \setminus \{h\}$. Then the inequality $a^*y^* \leq a_0^*$, obtained by node-cloning (zero-lifting) of $ay \leq a_0$ w.r.t. $h$, defines a facet of $STSP(V' \cup \{s\})$. 

![Inequalities](image-url)
Chapter 4. Hamiltonian Path and Symmetric Travelling Salesman Problem

Proof: Let \( cx \leq c_0 \) be its \( h \)-restriction on \( HP(V) \). By Proposition 4.17, it suffices to show that \( V \) is \( c \)-connected.

Suppose that \( V \) is not \( c \)-connected. Then by Lemma 4.11(2), there exists a proper partition \( \{U, U'\} \) of \( V \) such that there exists no \( c \)-tight path on \( V \) with one endnode in \( U \) and the other in \( U' \). If \( c_{uw'} = 0 \) for some \((u, u') \in E(U : U')\), then let \((p...uu'...q)\) be a \( c \)-tight path containing \((u, u')\) (since \( cx \leq c_0 \) is nontrivial). As \( c_{pq} \geq 0 = c_{uw'} \), \((u...pq...u')\) is another \( c \)-tight path, contradicting the supposition. Therefore, we must have \( c_e = 1 \) for all \( e \in E(U : U') \). Assume w.l.o.g. that \( |U| \leq |U'| \). Since \( cx \leq c_0 \) is nontrivial, for any \( u \in U \), there exists a \( c \)-tight path \( P \simeq (u...v) \) on \( V \). If \( v \in U' \), a contradiction is immediate. So consider \( v \in U \). Since \( |U| \leq |U'| \), \( P \) contains at least one edge in \( E(U') \). Thus, \( P \) has the general form \((u...pq...v)\) with \( p, q \in E(U') \). A contradiction is produced by defining a \( c \)-tight path \((u...pv...q)\) (since \( c_{pv} = 1 \geq c_{pq} \)). This shows \( V \) is \( c \)-connected.

Recall (pp.282, [28]) that an STSP rank inequality is a valid STSP inequality \( ay \leq a_0 \) with \( a_e \in \{0, 1\} \) for all \( e \).

Theorem 4.24 (Node-Cloning for Rank Inequalities) Let \( ay \leq a_0 \) be a facet-defining rank inequality for \( STSP(V') \). Then the inequality \( a^*y^* \leq a_0^* \), obtained by node-cloning of \( ay \leq a_0 \) w.r.t any node \( h \), is facet-defining for \( STSP(V' \cup \{s\}) \).

Proof: Let \( V \simeq V' \setminus \{h\} \), \( Z \setminus \{v \in V : (h, v) \in E'\} \) and \( W \simeq V \setminus Z \). If \( Z = \emptyset \), the result holds immediately by Proposition 4.23. If \( W = \emptyset \), we add the degree constraint \(-y(\delta(h)) = -2\), and thus Proposition 4.23 also applies. If both of the above sets are nonempty, then node \( h \) is 1-regular and the result follows by Theorem 4.22 (by (iii) if \( a_e = 0 \) for some \( e \in E(Q) \); and by (i) or (ii) otherwise).

With the above result and Theorem 4.20, one may obtain large new classes of
STSP facets from rank inequalities. For instance, these liftings can repeatedly apply to rank inequalities of Fig. 4.2(a), the Petersen graph, primitive chains, clique-trees, etc. Thus, to show that a general clique-tree is facet-defining, it suffices to consider the corresponding primitive clique-tree. Another consequence is that lifted rank inequalities are again clique-liftable w.r.t. any node.

In conclusion, we remark that the Hamiltonian path approach facilitates the STSP polyhedral study. By exploiting the near-full dimensional property of $HP(V)$ and the polyhedral equivalence between $HP(V)$ and $STSP(V')$, we obtain simpler polyhedral proofs for STSP facetial results, as well as stronger lifting results for extending STSP facets than the results from the other approaches. This approach also applies to:

- simple composition methods for extending STSP facets.
- the polyhedral study of asymmetric travelling salesman problem. See Chapter 5.
- the polyhedral study of other related problems, such as vehicle routing problems [2],[14].
Chapter 5

Symmetric Inequalities for Asymmetric Travelling Salesman Polytopes

In this Chapter, the Hamiltonian Path approach developed in Chapter 4 extends to the polyhedral study of Asymmetric Travelling Salesman (ATS) polytopes. By a Tree Composition technique, we obtain a large new class of symmetric facet-defining inequalities for ATS polytopes.

5.1 Introduction

Let \( D(V') = (V', A(V')) \) be a complete directed graph, where the node set \( V' \) represents a set of cities. A feasible solution to the Asymmetric Travelling Salesman (ATS) problem is a Hamiltonian cycle \( C \) on \( V' \), uniquely determined by its corresponding 0-1 incidence vector \( y \in R^{A(V')} \) (where \( y_e = 1 \) if edge \( e \) is used in \( C \) and zero otherwise). This chapter studies the \( ATS \) polytope \( ATSP(V') \), the convex hull of incidence vectors \( y \) of all feasible solutions to the ATS problem.

Over the past decade, substantial knowledge has been gained on the facial structure of ATS polytopes. Grötschel and Padberg [28] introduced several classes of valid inequalities, such as the \( D^+_k, D^-_k, C3, C2 \) inequalities, and proved some of these to be facet-defining. Fischetti [17] later proved that all these, as well as the directed version of Comb inequalities, are facet defining, except for the 6-node comb. Balas [4], and Balas and Fischetti [5,6] used relaxation methods and a general node lifting technique to derive large new classes of facet-defining inequalities, called \( CAT, FDA \) and \( SD \) inequalities. Very recently, Fischetti [18] has shown by an inductive proof technique that the directed
version of the Clique Tree Inequalities, is also facet-defining for ATS polytopes.

Grötschel and Padberg [28] addressed some interesting relationships between STS and ATS polytopes. To any given valid STSP inequality

\[ \sum \{a_{ij}y_{ij} : i, j \in V', i < j\} \leq a_0', \quad (5.1) \]

which we write in short \( a'y' \leq a_0' \), corresponds a unique directed version

\[ \sum \{a_{ij}y_{ij} : i, j \in V', i \neq j\} \leq a_0, \quad (5.2) \]

in short \( ay \leq a_0 \), for the corresponding ATS polytope, where \( a_0 = a_0' \), and \( a_{ij} = a_{ji} = a'_{ij} \) for all \( i, j \in V', i < j \). Thus, we say that an ATSP inequality \( ay \leq a_0 \) is a symmetric inequality if it is the directed version of some STSP inequality \( a'y' \leq a_0' \). Conversely, any symmetric ATSP inequality \( ay \leq a_0 \) yields an undirected version \( a'y' \leq a_0' \) for the corresponding STS polytope. It is easy to observe that an inequality \( a'y' \leq a_0' \) is an STSP valid inequality if and only if its directed version \( ay \leq a_0 \) is an ATSP valid inequality. An important question raised by Grötschel and Padberg [28] is “whether or not the directed version \( ay \leq a_0 \) of an STSP facet-defining inequality \( a'y' \leq a_0' \) is also ATSP facet-defining.” Conversely, as observed in Fischetti [18], the undirected versions of all symmetric ATSP facet-defining inequalities induce facets of STS polytopes. The study of symmetric ATSP facet-defining inequalities thus has some theoretical importance. To date, the largest known class of symmetric ATSP facet-defining inequalities is the directed version of the famous Clique Tree class, which subsumes all previously known symmetric ATSP facet-defining inequalities, see Fischetti [18].

There are three major motivations for the present work. First, and as mentioned above, in addition to its theoretical interest, the study of symmetric ATSP inequalities is of practical importance: we expect that the cutting plane-based solution procedures that have been so successful in solving large scale STS problems will be adapted to ATS
problems. In this respect, we feel that the symmetric ATSP facet-defining inequalities deserve further study, in particular because STSP separation heuristics also apply to ATSP symmetric facets. Second, large classes of valid inequalities, such as Naddef and Rinaldi's Regular Path Trees, Boyd and Cunningham's Ladders, and Padberg and Hong's Chains, etc., were shown to be STSP facet-defining in recent years [42]. It is of interest to know when the directed versions of these large classes are also ATSP facet defining. Finally, known classes of STSP facet-defining inequalities were extended by various composition methods, see Naddef and Rinaldi [44]. A composition approach is also introduced here for ATS polytopes and is used to generate large new classes of symmetric ATSP facet-defining inequalities.

Two major difficulties arise when directly studying ATS polytopes: (i) these are quite far from being full dimensional polytopes, due to the degree constraints; and (ii) Hamiltonian cycles and circuits are rather cumbersome objects to work with. These difficulties complicate the (indirect) proofs that stated valid inequalities are facet-defining. In an indirect proof that valid inequality $cx \leq c_0$ is facet-defining for a given polyhedron $\mathcal{P}$, one aims at showing that any valid inequality $fx \leq f_0$ that dominates $cx \leq c_0$ (i.e., such that $fx = f_0$ whenever $cx = c_0$, for all $x \in \mathcal{P}$) must be equivalent to $cx \leq c_0$. This amounts to showing that $fx \leq f_0$ can be obtained as a linear combination of implicit equations for $\mathcal{P}$ (equations satisfied by all $x \in \mathcal{P}$) and a positive multiple of $cx \leq c_0$. If there are several linearly independent implicit equations for $\mathcal{P}$, as is the case for STS and ATS polytopes, the indirect proofs tend to be difficult. In addition, the cumbersomeness of Hamiltonian cycles or circuits manifests itself when trying to determine the structure of such dominating inequalities $fx \leq f_0$. The components of $f$ are usually determined by comparing c-tight solutions (Hamiltonian cycles or circuits whose incidence vector $x$ satisfies $cx = c_0$), which differ from each other in as few components as possible. Indeed,
two distinct Hamiltonian circuits differ by at least six edges (the so-called 3-opt interchange). The resulting conditions on the components of vector $f$ then involve at least six components and are thus quite unwieldy.

Various relaxation methods were proposed to overcome these difficulties, most notably (i) the Monotone ATSP approach (see Grötschel and Padberg [28] for a recent survey), and Fischetti [17]) and Graphical ATSP approach introduced by Chopra and Rinaldi [10]. The corresponding objects, edge-subsets of Hamiltonian cycles/circuits, and tours, respectively, are much more manageable than Hamiltonian cycles/circuits. Unfortunately, it is not easy to convert MATSP and GATSP facetial results into corresponding ATSP results.

The Hamiltonian Path approach, introduced in Chapter 4, can be extended to the polyhedral study of ATS polytopes. This approach uses an Asymmetric Hamiltonian Path (AHP) formulation and thus also relaxes the degree constraints. For any arbitrary node $h \in V'$, let $V = V' \setminus \{h\}$. Let $A$ (resp., $A'$) denote the edge set of the complete digraph on node set $V$ (resp., $V'$). The Asymmetric Hamiltonian Path polytope $AHP(V) \subseteq \mathbb{R}^A$ is the convex hull of incidence vectors of all Hamiltonian paths (with free endpoints) on node set $V$. We show that polytope $AHP(V)$ is a projection of $ATSP(V')$ onto subspace $\mathbb{R}^A$ of $\mathbb{R}^{A'}$. This projection has two fundamental properties: $AHP(V)$ has full dimension minus one (with a single implicit equation $\sum_e x_e = |V| - 1$); and the polyhedral structures of $AHP(V)$ and $ATSP(V')$ are equivalent. We are led to prefer the AHP approach for studying ATS polytopes to the MATSP or GATSP relaxations, for three main reasons which follow mostly from the above two properties. First, the (indirect) proofs for the AHP polyhedral results are almost as easy as in the full dimensional case, because the single implicit equation is easily dealt with. Second, all polyhedral results for $AHP(V)$ trivially extend to $ATSP(V')$, due to their polyhedral equivalence. Finally, we found that Hamiltonian paths are much easier to work with than Hamiltonian circuits, in particular
when comparing solutions in indirect proofs, as two distinct Hamiltonian paths may differ in a single pair of edges.

The present paper is organized as follows. In Section 5.2, we introduce definitions and notation. In Section 5.3, we establish the main properties of \( ATSP(V') \) and its projection \( AHP(V) \), and then develop a projective approach for the ATS problem. In Section 5.4, we show that some known classes of STSP facet-defining inequalities, such as Path, Wheelbarrow, Envelope, Ladder and Chain Inequalities, induce symmetric ATSP and AHP facet-defining inequalities. The derivations are based on the projective approach, along with a Symmetric Dominance Lemma. In Section 5.5, we propose a Tree Composition of symmetric inequalities. This composition generates a large class of symmetric facet-defining inequalities for ATS polytopes, called \( ATSP \) Tree Inequalities, which generalize the directed versions of non-spanning Clique Tree Inequalities and Regular Path Tree Inequalities. Furthermore, the proposed Tree Composition may also be used to derive a large new class of STSP facet-defining Tree Inequalities. Figures 5.1- 5.5 herein present examples of new ATSP facets. The proofs of the main results in Sections 5.4 and 5.5 are provided in Sections 5.6 and 5.7, respectively.

5.2 Definitions and Notation

Most of the notation and terminology below follow from [5] and [28].

Let \( D(V) = (V, A) \) denote the complete (loop-free) digraph on \( n = |V| \) nodes, where

\[
A = \{(i, j) : i \in V, j \in V \setminus \{i\}\}.
\]

Hereafter, we assume \( n \geq 3 \). Digraph \( D(V) \) contains \( m = |A| = n(n - 1) \) edges. For any two subsets \( V_1, V_2 \) of \( V \), define

\[
(V_1 : V_2) = \{(i, j) \in A : i \in V_1, j \in V_2\}.
\]
Chapter 5. Symmetric Inequalities for Asymmetric Travelling Salesman Polytopes

If $V_1 = V_2$, $(V_1 : V_1)$ is denoted by $A(V_1)$. Thus, $A = A(V)$. A singleton set $\{v\}$ is denoted by $v$ whenever convenient without causing confusion. Let $\delta^+(v) = (v : V)$ and $\delta^-(v) = (V : v)$ denote the out-star and in-star of node $v$, respectively.

For any complete undirected graph $(V, E(V))$ and any subsets $V_1, V_2 \subseteq V$, the edge sets $E(V_1 : V_2) = \{(i, j) \in E(V) : i \in V_1, j \in V_2\} = E(V_2 : V_1)$ and $E(V_1) = E(V_1 : V_1)$ are similarly defined.

A partial graph $(S, \tilde{A})$ of $D(V)$ is a digraph with $S \subseteq V$ and $\tilde{A} \subseteq A$. The incidence vector of the partial graph $(V, \tilde{A})$ of $D(V)$ is the vector $y \in \mathbb{R}^A$ such that $y_e = 1$ if $e \in \tilde{A}$ and $y_e = 0$ otherwise. For any vector $\chi \in \mathbb{R}^A$ and $\tilde{A} \subseteq A$, let

$$\chi(\tilde{A}) = \sum_{e \in \tilde{A}} \chi_e.$$  \hspace{1cm} (5.5)

For simplicity, let

$$\chi(S : T) = \chi((S : T)) \text{ and } \chi(S) = \chi(A(S)), \forall S, T \subseteq V.$$  \hspace{1cm} (5.6)

For any $S \subseteq V$, a Hamiltonian circuit $C$ on $S$ is a connected partial graph $(S, \tilde{A})$ such that

$$|\tilde{A} \cap \delta^+(v)| = |\tilde{A} \cap \delta^-(v)| = 1 \text{ for all } v \in S.$$  

By definition, the incidence vector $y^C$ of any Hamiltonian circuit $C$ on $V$ thus satisfies the following degree constraints:

$$y^C(\delta^+(v)) = y^C(\delta^-(v)) = 1, \text{ for all } v \in V.$$  

Without risking confusion, a Hamiltonian circuit $C$ with edge set

$$\tilde{A} = \{(v_s, v_1)\} \cup \{(v_i, v_{i+1}) : i = 1, \ldots, s - 1\}$$

on node set $S = \{v_1, \ldots, v_s\} \subseteq V$ may be represented by edge set $\tilde{A}$ alone or, equivalently, by the circular permutation $(v_1 \ldots v_s)$ of the nodes in $S$. 

Chapter 5. Symmetric Inequalities for Asymmetric Travelling Salesman Polytopes 103

A Hamiltonian path $P$ on $S = \{v_1, ..., v_s\} \subseteq V$ is a connected partial graph $(S, A)$ such that for some distinct $i, j \in S$, $(S, \hat{A} \cup \{i, j\})$ defines a Hamiltonian circuit on $S$. Note that the endnodes of $P$ are not specified in advance, i.e., we consider here $P$ with free endnodes. Hamiltonian path $P$ may be represented as the edge set $\{(v_i, v_{i+1}) : i = 1, ..., s - 1\}$ or, equivalently, as its corresponding node permutation $(v_1, ..., v_s)$. Note that if $|S| = 1$, the first form is an empty set, whereas the second form reduces to a single node $(v_1)$. Note also that whenever a subchain of a Path is written in the form $u...u$, we allow the subchain to be a singleton, i.e., $u = v$.

For any node set $S$, the ATS polytope $ATSP(S)$ (resp., AHP polytope $AHP(S)$) is the convex hull of the incidence vectors of all Hamiltonian circuits (resp., Hamiltonian paths) on the complete (loop-free) digraph $(S, A(S))$:

$$ATSP(S) = \text{conv}\{y^C \in R^{A(S)} : C \text{ is a Hamiltonian circuit on } S\},$$

$$AHP(S) = \text{conv}\{x^P \in R^{A(S)} : P \text{ is a Hamiltonian path on } S\}.$$

Recall that $STSP(S)$ (resp., $HP(S)$) is the convex hull of incidence vectors of the undirected Hamiltonian circuits (resp., undirected Hamiltonian paths) in a complete undirected graph $K(S) = (S, E)$. Note that the number of variables defining polytope $STSP(S)$ (resp., $HP(S)$) is half of the number of variables defining its asymmetric counterpart $ATSP(S)$ (resp., $AHP(S)$).

Let $a_y \leq a_0$ and $c_x \leq c_0$ be any valid inequalities for $ATSP(S)$ and $AHP(S)$, respectively. An $a$-tight circuit $C$ (resp., a $c$-tight path $P$) on $S$ is a Hamiltonian circuit (resp., a Hamiltonian path) on $S$ such that $a_y^C = a_0$ (resp., $c_x^P = c_0$).
5.3 The ATS Polytope and its Projection

For any fixed node \( h \in V' \), let \( V = V' \setminus \{h\} \) and \( n = |V| \). Consider two complete (loop-free) digraphs:

\[
D(V) = (V, A) \quad \text{and} \quad D(V') = (V', A'),
\]

where \( A' = A \cup \{(v, h), (h, v) : v \in V\} \). That is, \( D(V) \) is obtained from \( D(V') \) by eliminating node \( h \in V' \) and all incident edges. Let \( ATSP(V') \) be the ATS polytope defined on \( D(V') \), and \( AHP(V) \) be the AHP polytope defined on \( D(V) \). Observe that \( P \) is a Hamiltonian path on \( V \) from node \( u \) to node \( v \) iff \( C = P \cup \{(h, u), (v, h)\} \) is a Hamiltonian circuit on \( V' \). Thus, incidence vector \( x^P \in R^A \) is a projection of incidence vector \( y^C \in R^{A'} \) onto subspace \( R^A \) of \( R^{A'} \). It follows that polytope \( AHP(V) \) is a projection of \( ATSP(V') \). Moreover, there is a one-to-one correspondence between the vertices of \( ATSP(V') \) and those of \( AHP(V) \). This correspondence will be shown to be affine-rank preserving.

Associate two \( n \times n(n-1) \) matrices, \( A^- \) and \( A^+ \), to complete digraph \( D(V) \) as follows.

For all \( v \in V \) and \( e \in A \), let \( A^-_{ve} = 1 \) if \( e \in (V : v) \); i.e., if \( v \) is the head of edge \( e \); and zero otherwise. Similarly, let \( A^+_{ve} = 1 \) if \( e \in (v : V) \), i.e., if \( v \) is the tail of edge \( e \); and zero otherwise. Thus, \( A^- \) (resp., \( A^+ \)) is the tail (resp., head) node-edge incidence matrix of \( D(V) \). By definition, the incidence vector \( y^C \in R^{A'} \) of any Hamiltonian circuit \( C \) on \( V' \) and its projection \( x^P \in R^A \) satisfy the following identity:

\[
y^C_{uv} = \begin{cases} 
x^P_{uv}, & (u, v) \in A; \\
1 - \sum_{e \in A} A^-_{ve} x^P_e, & (u, v) \in (h : V); \\
1 - \sum_{e \in A} A^+_{ve} x^P_e, & (u, v) \in (V : h).
\end{cases}
\]

(5.7)

Throughout the paper, we assume, unless otherwise noted, that

- \( x \in R^A \) is a projection of \( y \in R^{A'} \); and
Chapter 5. Symmetric Inequalities for Asymmetric Travelling Salesman Polytopes

- \( y = (\gamma_1, \ldots, \gamma_m, y_{hv_1}, \ldots, y_{hv_n}, y_{v_1}, \ldots, y_{v_n})^t \), where \( m = n(n-1) \), \( e_l \in A \) for all \( l = 1, \ldots, m \), and \( v_l \in V \) for all \( l = 1, \ldots, n \).

Lemma 5.1 Let \( B_y = [y^1, \ldots, y^l] \) be any matrix such that all columns \( y^j \) are incidence vectors of Hamiltonian circuits on \( D(V) \). Let matrix \( B_x = [x^1, \ldots, x^l] \), where \( x^j \) is the projection of \( y^j \) onto \( R^A \), \( j = 1, \ldots, l \). Then, the number of affinely independent columns in \( B_y \) is equal to the number of affinely independent columns in \( B_x \).

Proof. From (5.7), it follows

\[
\begin{aligned}
\text{rank} \begin{pmatrix}
1^t \\
B_y \\
\end{pmatrix} & \geq \text{rank} \begin{pmatrix}
1^t \\
B_x \\
\end{pmatrix} \\
& \geq \text{rank} \begin{pmatrix}
1 & 0^t \\
0 & I \\
1 & -A^- \\
1 & -A^+ \\
\end{pmatrix}
\begin{pmatrix}
1^t \\
B_x \\
\end{pmatrix}
\end{aligned}
\]

where \( 0 \) (resp., \( 1 \)) is a vector of zeroes (resp., ones) and \( I \) is an identity matrix, all of conformable dimension. Hence, equality must hold throughout.

This completes the proof. \( \square \)

Theorem 5.2 The dimension of \( AHP(V) \) is

\[
\dim(AHP(V)) = n(n - 1) - 1,
\]

where \( n = |V| \geq 2 \).

Proof. For \( n = 2, 3, 4 \), equation (5.8) follows from enumerating all possible corresponding Hamiltonian paths. Let \( n \geq 5 \). First, observe that for any Hamiltonian path \( P \) on \( V \), we have \( \sum_{e \in A} x^P_e = n - 1 \). Therefore,

\[
\dim(AHP(V)) \leq n(n - 1) - 1.
\]
Next, let $f x = f_0$ be any equality satisfied by the incidence vectors $x$ of all Hamiltonian paths on $V$. Let $e$ and $e'$ be any two edges in $A(V)$ with distinct endnodes. Then, there exists a Hamiltonian circuit $C$ on $V$ containing both $e$ and $e'$. Comparing Hamiltonian paths $C \setminus e$ and $C \setminus e'$ yields:

$$0 = f x^{C \setminus e} - f x^{C \setminus e'} = f_{e'} - f_e.$$ 

As $n \geq 5$, repeating the above argument to all pairs of non-adjacent edges implies that $f x \leq f_0$ is a multiple of $x(V) = n - 1$. 

The dimension of the ATS polytope is described in Grötschel and Padberg [28]. The following is an alternative and simple derivation of this result.

**Corollary 5.3** The dimension of $ATSP(V')$ is

$$\dim(ATSP(V')) = n(n + 1) - 2(n + 1) + 1, \quad (5.9)$$

where $|V'| = n + 1 \geq 3$.

**Proof.** Follows directly from Lemma 5.1 and Theorem 5.2 by noting that

$$n(n - 1) - 1 = n(n + 1) - 2(n + 1) + 1. \quad \square$$

Now, we turn to relationships between valid inequalities for $ATSP(V')$ and for $AHP(V)$. Consider any two inequalities $a y \leq a_0$ and $c x \leq c_0$, defined on $ATSP(V')$ and $AHP(V)$, respectively, and satisfying:

$$a_0 = c_0; \quad a_e = c_e \geq 0, \quad \forall e \in A; \quad \text{and} \quad a_e = 0, \quad \forall e \in (h : V) \cup (V : h).$$

We then say that inequality $a y \leq a_0$ has an isolated node, that $a y \leq a_0$ is an extension of $c x \leq c_0$, and that $c x \leq c_0$ is a restriction of, or a projected inequality for, $a y \leq a_0$ w.r.t. node $h$. By Lemma 5.1 and Theorem 5.2, we have immediately:
Chapter 5. Symmetric Inequalities for Asymmetric Travelling Salesman Polytopes 107

Theorem 5.4 (Polyhedral Equivalence) For any $k$ such that $0 \leq k \leq n(n - 1) - 1$, inequality $ay \leq a_0$ with an isolated node defines a $k$-dimensional face of $ATSP(V')$ iff its restriction $cx \leq c_0$ defines a $k$-dimensional face of $AHP(V)$.

The above result may also be derived by an application of Remark 4.2 of [27]. By appropriate scaling, the inequality $ay \leq a_0$ with an isolated node is in $h$-canonical form and its corresponding restriction $cx \leq c_0$ is a $AHP$ canonical inequality.

$h$-canonical form (Balas and Fischetti [5])

An inequality $ay \leq a_0$ for $ATSP(V')$ is in canonical form with respect to a given node $h \in V'$ ($h$-canonical form, for short) if

(C1) $a_vh = a_hv = 0$ for all $v \in V = V' \setminus h$;

(C2) $a \geq 0$ and $a_e = 0$ for some $e \in A = A(V)$;

(C3) the coefficients $a_e$ are relatively prime integers.

AHP Canonical Inequality Inequality $cx \leq c_0$ for $AHP(V)$ is a canonical inequality, or in canonical form, if the coefficients $c_e$ satisfy conditions (C2) and (C3) above.

It is shown in [5] that any valid inequality for $ATSP(V')$ has a unique $h$-canonical form. This uniqueness property also follows directly from Theorems 5.2 and 5.4 by observing that conditions (C2) and (C3) may be imposed on the restriction $cx \leq c_0$ by using the unique implicit equation $x(V) = n - 1$ for $AHP(V)$, followed by an appropriate positive scaling.

Following the results in [28] and the polyhedral equivalence, trivial facets of ATSP and AHP polytopes are defined as follows:

Trivial Facets Inequality $ay \leq a_0$ (resp., $cx \leq c_0$) defines a trivial facet of $ATSP(V')$ (resp., $AHP(V)$) if it is equivalent to $y_e \geq 0$ (resp., $x_e \geq 0$) for some $e \in A'$ (resp., $e \in A(V)$).
From now on, we will work with symmetric inequalities, in canonical form for $AHP(V)$, and in $h$-canonical form for $ASTP(V')$, i.e., with an isolated node $h$ readily identified, for the following reasons. First, the projections of these inequalities $ay \leq a_0$ with respect to isolated node $h$ are in fact the restrictions $cx \leq c_0$ of $ay \leq a_0$. Consequently, the “structural” definitions of inequalities, such as Comb, Clique Tree, etc., $ay \leq a_0$ for $ATSP(V')$ directly apply to $cx \leq c_0$ for $AHP(V' \setminus h)$ as well. Next, it follows immediately from Theorem 5.4 that $ay \leq a_0$ defines a facet of $ATSP(V')$ iff $cx \leq c_0$ defines a facet of $AHP(V' \setminus h)$. So, no additional work is needed to recover one inequality from the other. Finally, the ATSP facet composition results are relatively easier to obtain, as is the case for 2-SUM GTSP facet compositions of Naddef and Rinaldi [44]. The proposed ATSP facet composition in Section 5.4 requires that all composing inequalities be in $h$-canonical forms. The facet-defining inequalities given as examples in Figures hereafter are for ATS polytopes and are all in $h$-canonical form, unless otherwise noted.

5.4 Symmetric Facet-defining Inequalities

In this section, we study three classes of symmetric inequalities: Regular Star, Chain and Ladder Inequalities. Of particular interest are these inequalities in “nice” $h$-canonical form, i.e., with an isolated node. Their undirected versions are known to be facet-defining for STS polytopes. For STS polytopes, the inequalities in the first class was introduced as the Path Inequalities, along with variations, the Wheelbarrow and Bicycle Inequalities, by Cornuéjols et al. [13]. They were called Regular Star Inequalities by Fleischmann [21] and shown to be STSP facet-defining by Naddef and Rinaldi [42]. The second class was introduced by Padberg and Hong [47] and shown to be STSP facet-defining by M. Hartmann and by S. Boyd (cf. Naddef and Rinaldi [41]). The Ladder Inequality was introduced and proven to be STSP facet-defining by Boyd and Cunningham [8]. We will
show by projection that the Path, Wheelbarrow, Envelope, Ladder and Chain inequalities induce facets of ATS polytopes as well.

A closed form representation for inequalities $ay \leq a_0$ is used in the Figures hereafter, where a collection of node subsets with positive weights (numbers) are given. Then, each coefficient $a_{ij}$ equals the total weight of all the subsets which contain both nodes $i$ and $j$. All set weights of value one are omitted for simplicity. Whenever necessary, nodes are labeled with letters or italic numbers. A hollow circle $g$ denotes an optional node, i.e., node $g$ may or may not be present. If the right hand side of an inequality with node $g$ is different from that without $g$, then the former is also indicated in parentheses. Figure 5.1 gives such a representation. Some of the resulting coefficients in the represented ATSP inequality are

$$a_{b_1b_3} = 2, \quad a_{b_1b_4} = 4, \quad a_{b_3b_5} = 4, \quad a_{b_1b_4} = 0, \quad \text{etc.}$$

and the right hand side $a_0 = 44$ if node $g$ is absent, $a_0 = 48$ if present.

Figure 5.1: An ATSP Regular Star Inequality

Balas and Fischetti [6] recently proposed a powerful node lifting result for ATSP facets,
called Clique-Lifting. This result implies that a class of ATSP inequalities are facet-defining if all the primitive inequalities in this class are facet-defining (cf. Fischetti [17] for the definition of a primitive inequality). Therefore, it suffices to consider ATSP inequalities in primitive form. In the sequel, all inequalities are introduced in primitive form.

We now introduce primitive Star Inequalities for ATSP(V'). Figure 5.1 presents an example. Given are proper subsets of V' as follows: N nested circles, $A_1 \supset A_2 \supset ... \supset A_N$; and an odd number $s \geq 3$ of disjoint spikes, $B^1, B^2, ..., B^s$, where, for all $i$, $B^i$ consists of nodes $b^i_j, j = 1, ..., n_i$ such that $b^i_n \in A_N$ and $b^i_1 \in V' \setminus A_1$. Moreover, assume that for every $i$, inequality $j < l$ implies that there is no $q$ such that $b^i_q \in A_q$ and $b^i_q \notin A_q$. Let $M(v)$ denote the number of circles $A_q$ containing node $v$. With each spike $B^i$ associate a spike weight $N_i$ such that $N_i \geq M(b^i_{j+1}) - M(b^i_j)$ for all $1 \leq j \leq n_i - 1$ (cf. Fleischmann [21] for details). Then, a primitive Star Inequality $a_1y < a_0$ for ATSP(V') is defined by

$$
\sum_{q=1}^{N} y(A_q) + \sum_{i=1}^{s} N_i y(B^i) \leq \sum_{q=1}^{N} |A_q| + \sum_{i=1}^{s} N_i(|B^i| - 1) - \frac{(s + 1)N}{2}. \tag{5.10}
$$

Let $a'y' \leq a'_0$ be the undirected version of inequality (5.10) for STSP(V'). By adding to $a'y' \leq a'_0$ proper multiples of the following valid equations (obtained from the degree constraints):

$$
2y'(S) + y'(\delta(S)) = 2|S|, \quad \text{for all } S \in \{A_1, ..., A_N, B^1, ..., B^s\},
$$

we obtain

$$
\sum_{q=1}^{N} y'(\delta(A_q)) + \sum_{i=1}^{s} N_i y'(\delta(B^i)) \geq 2 \sum_{q=1}^{N} N_q + (s + 1)N, \tag{5.11}
$$

which is the Star Inequality for STSP(V') in Fleischmann's form [21].

When $N_i = M(b^i_{j+1}) - M(b^i_j)$ for all $i$ and $j$, inequality (5.10) represents a Regular Star Inequality. If $N = 1$ and all spike weights $N_i = 1$, inequality (5.10) reduces to
a Comb Inequality [28]. Hence, the Regular Star class properly generalizes the Comb class. Figure 5.1 presents a Regular Star Inequality with four circles and five spikes, i.e., a 5-Star (equivalently, a Path Inequality with five “paths”), in $h$-canonical form. It reduces to a 3-Star if the spikes in the dash-box are removed. By definition, the spike weights for $B_i^1$, $B_i^2$ and the leftmost spike in the dash-box are 2, 4 and 2, respectively. Recall that all other set weights are equal to one. A Path Inequality includes both nodes $h$ and $g$, whereas a Wheelbarrow Inequality (resp., Bicycle Inequality) excludes node $g$ (resp., excludes both nodes $h$ and $g$). If node $g$ exists, then the right hand side needs to be properly adjusted according to (5.10).

If a Regular Star is not a Bicycle Inequality, its restriction $cz \leq c_0$ is in canonical form for $AHP(V)$, and reads:

$$
\sum_{q=1}^{N} x(A_q) + \sum_{i=1}^{s} N_i x(B_i^i) \leq \sum_{q=1}^{N} |A_q| + \sum_{i=1}^{s} N_i (|B_i^i| - 1) - \frac{(s+1)N}{2}. \tag{5.12}
$$

**Theorem 5.5** Primitive Path and Wheelbarrow Inequalities (resp., their restrictions) are facet-defining for $ATSP(V')$ (resp., $AHP(V)$).

The proof of Theorem 5.5 is given in Section 6.

The ATSP Bicycle Inequalities do not have such nice $h$-canonical forms. The smallest non-comb Bicycle Inequality, also called the Envelope Inequality by Boyd and Cunningham, is given in Fig. 5.2(a). An $h$-canonical form of the Envelope Inequality is presented in Fig. 5.2(b). Fig. 5.2(c) represents the Ladder Inequality. Recall that node $g$ is optional. Fig. 5.2(d) is its $h$-canonical form. Note that these $h$-canonical forms are obtained by replacing all non-empty subsets containing node $h$ with their complements. In Fig. 5.2(b) and (d), *italic* numbers are used to label the nodes for convenience in proving the next theorem. (The other numbers are set weights.) If node $g$ exists, an alternative simple $h$-canonical form for Fig. 5.2(c) is Fig. 5.2(c) itself by exchanging nodes $h$ and $g$. We choose
the form in Fig. 5.2(c), because it is convenient for showing that the Ladder Inequality is facet-defining for $ATSP(V')$ for $|V'| = 8, 9$.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig5_2.png}
\caption{Symmetric facets and their projections}
\end{figure}

\textbf{Theorem 5.6} The following statements are true:

1. The Envelope Inequality defines a facet of $ATSP(V')$, where $|V'| = 7$.

2. The Ladder Inequality defines a facet of $ATSP(V')$, where $|V'| \geq 8$.

A proof of \textbf{Theorem 5.6} is contained in Section 5. We remark that the Envelope Inequality is one of the two smallest non-trivial, non-subtour elimination, primitive symmetric facet-defining inequalities for $ATSP(V')$ with $|V'| = 7$. (Otherwise, the undirected version of some primitive symmetric ATSP facet defining inequality would induce a facet of $STSP(V')$ with $|V'| = 7$, contradicting a theorem by Boyd and Cunningham [8].) The other one is a 7-node Wheelbarrow Inequality.
Chapter 5. Symmetric Inequalities for Asymmetric Travelling Salesman Polytopes

Note that Clique Lifting may then be applied to any node of these ATSP facet-defining Envelope and Ladder Inequalities. Other extensions of the above results to general Bicycle and Ladder Inequalities (without isolated nodes) are left for the interested reader, using proofs similar to those of Theorems 5.5 and 5.6.

We now consider primitive ATSP Chain Inequalities with an isolated node \( h \). Given are any positive integers \( p \) and \( k \) such that \( k - p \) is a positive even number. Let \( R \) and \( S_i \) for \( i = 0, \ldots, k \), be proper subsets of \( V' \) satisfying

(i) \( S_i \cap S_j = \emptyset \) for all \( i, j = 1, \ldots, k \) \((i \neq j)\);
(ii) \( S_i = \{s_i\} \) for all \( i = 1, \ldots, p \);
(iii) \( S_i \setminus S_0 = \{b_i^1\} \) and \( S_0 \cap S_i = \{b_i^2\} \) for all \( i = p + 1, \ldots, k \);
(iv) \( R = \{r_1, \ldots, r_p\} \subset S_0 \) and \( R \cap S_i = \emptyset \) for all \( i = 1, \ldots, k \); and
(v) \( V' \setminus \bigcup_{i=0}^{k} S_i = \{h\} \).

Let \( T = \{s_1, \ldots, s_p\} \).

![Figure 5.3: A 9-node ATSP Chain Inequality](image)

From the general Chain Inequality (see [47] for definition), we obtain a primitive
Chain Inequality for $ATSP(V')$:

$$\sum_{i=0}^{k} y(S_i) + y(R: T) + y(T: R) \leq \left| S_0 \right| + \frac{1}{2}(k + p - 2).$$  \hspace{1cm} (5.13)

For $p = 1$, this reduces to a Comb Inequality. If $k - p > 0$ is odd, then it can be shown that the inequality is still valid but not facet-defining, see Padberg and Hong [47]. The smallest non-comb Chain Inequality in $h$-canonical form is presented in Figure 5.3.

**Theorem 5.7** Primitive Chain Inequalities defined in (5.13) with an isolated node (resp., their restrictions) are facet-defining for $ATSP(V')$ (resp., $AHP(V)$).

This theorem is also proved in Section 5.

**General** Chain Inequalities are then shown to be ATSP facet-defining by Clique-Lifting on any node.

The proofs of **Theorems 5.5-5.7** are based on the following two auxiliary results. The first one follows from the known STSP facetial results and the polyhedral equivalence of $STSP(V')$ and $HP(V)$ in Chapter 1.

**Proposition 5.8** The undirected versions of the symmetric inequalities $ay \leq a_0$ and their projections $cx \leq c_0$ stated in **Theorems 5.5-5.7** define facets of $STSP(V')$ and $HP(V)$, respectively.

**Lemma 5.9 (Symmetric Dominance Lemma)** Let $cx \leq c_0$ be a symmetric inequality for $AHP(V)$ whose undirected version $c'x' \leq c'_0$ defines a facet of $HP(V)$. If there exists a symmetric facet-defining inequality $fx \leq f_0$ for $AHP(V)$ which dominates $cx \leq c_0$, then $cx \leq c_0$ defines a facet of $AHP(V)$.

**Proof.** If $fx \leq f_0$ is symmetric, then it has an undirected version $f'x' \leq f'_0$ for $HP(V)$. Observe that $f'x' \leq f'_0$ also dominates the HP facet-defining inequality $c'x' \leq c'_0$, and
therefore, they are equivalent. Because the equality set of $HP(V)$ reduces to $x'(E(V)) = n - 1$, there exist scalars $\alpha > 0$ and $\beta$ such that $f'_e = \alpha c'_e + \beta$ for all $e \in E(V)$ and $f'_0 = \alpha c'_0 + (n - 1)\beta$. It follows that $f_e = \alpha c_e + \beta$ for all $e \in A(V)$. Thus, $cx \leq c_0$ is equivalent to $fx \leq f_0$, hence facet-defining.

Thus, the proofs in Section 5 amount to showing that for each restriction $cx \leq c_0$ of the proposed $h$-canonical ATSP inequalities, there exists a symmetric AHP facet-defining inequality that dominates $cx \leq c_0$. The result then immediately translates into ATSP term by the polyhedral equivalence of Theorem 5.4.

5.5 Composing Symmetric Inequalities

In this section, we propose a Tree Composition of ATSP symmetric inequalities. Regular Star, Ladder and Chain Inequalities, each with an isolated node, are possible building blocks for the composition. Tree Composition may be used to generate a large new class of ATSP facet-defining inequalities, called ATSP Tree Inequalities, for ATS polytopes, generalizing the directed versions of Regular Path Tree, and thus all non-spanning Clique Tree, Inequalities with an isolated node.

The application of Tree Composition also extends to facets of STS polytopes. The composition is similar in spirit to Naddef and Rinaldi’s 2-SUM composition. However, the conditions required here for each composing inequality are different and much easier to apply, mostly by visual inspection. Furthermore, large classes of STSP facet-defining Tree Inequalities resulting from 2-SUM composition, such as Regular Path Trees, may also be obtained by the present Tree Composition. Consequently, this Tree Composition may be used to obtain large new classes of tree-like facets of STS polytopes as well.

Definition 5.A We say that an inequality $ay \leq a_0$ for $ATSP(V')$ satisfies Condition $S(h, S, Z; d; \omega; \eta)$ if there exists a proper partition \{h, S, Z\} of $V'$, with $|S| \geq 2$; a node
Chapter 5. Symmetric Inequalities for Asymmetric Travelling Salesman Polytopes 116

d ∈ S, called a tip; and two positive scalars ω and η, such that the following conditions hold:

1. \( a_e \geq 0 \), \( \forall e \in A(V') \);

2. \( a_e = 0 \), \( \forall e \in (h : S \cup Z) \cup (S \cup Z : h) \cup (d : Z) \cup (Z : d) \);

3. if \(|S| \geq 3\), then \( a_e \geq \eta \omega \), \( \forall e \in A(S \setminus d) \); and

4. \( a_{vd} = a_{dv} = \omega \), \( \forall v \in S \setminus d \).

An inequality \( ay \leq a_0 \) for \( ATSP(V') \) is said to be an \( ATSP \) Tree Inequality with respect to a subset \( S \) of \( V' \), called a bud, if the following two conditions hold:

(S1) \( ay \leq a_0 \) is a symmetric valid inequality for \( ATSP(V') \); and

(S2) \( ay \leq a_0 \) satisfies Condition \( S(h, S, Z; d; \omega; 1) \).

An ATSP Tree Inequality \( ay \leq a_0 \) with respect to bud \( S \) is said to be a composable ATSP Tree Inequality if, in addition to (S2), it satisfies the following three conditions:

(S1') \( ay \leq a_0 \) defines a non-trivial symmetric facet of \( ATSP(V') \);

(S3) for every \( v \in S \setminus d \), there exists an \( a \)-tight Hamiltonian circuit \( C \) on \( V' \), called an \( S \)-circuit, of the form \( (z,...uv,...qdh) \) with \( a_{uv} = 0 \); and

(S4) there exists a Hamiltonian circuit \( C_d \) on \( V' \setminus d \), called a \( U \)-circuit, such that \( a(C_d) = a_0 \).

Conditions (S3) and (S4) are sometimes hard to verify and may be removed by strengthening (S2):

(S2') There exist two disjoint subsets \( S \) and \( S' \) of \( V' \) such that \( ay \leq a_0 \) satisfies both Conditions \( S(h, S, Z; d; \omega; 2) \) and \( S(h, S', Z'; d'; \omega'; 2) \).

Proposition 5.10 (S1'),(S2') \( \Rightarrow \) (S3),(S4) w.r.t. both \( (S,d) \) and \( (S',d') \).
A proof is provided in Section 6.

By Proposition 5.10, conditions (S1’) and (S2’) are sufficient for recognizing composable ATSP Tree Inequalities by inspection. Path and Wheelbarrow Inequalities are composable ATSP Tree Inequalities with respect to any spike as a bud. Clique Tree, Ladder and Chain Inequalities, each with an isolated node, are composable ATSP Tree Inequalities with respect to any pendant tooth as a bud. For some cases, however, the weaker conditions (S1’), (S3) and (S4) may apply, while (S2’) fails. For instance, the Chain Inequality in Figure 5.3 is also a composable ATSP Tree Inequality relative to bud \( S = \{s_1, r_1, r_2\} \) with tip \( d = s_1 \) and \( \omega = 1 \), in which case condition (S2’) fails but (S2) holds, since \( a_{r_1 r_2} = 1 < 2\omega \). However, (S3) and (S4) hold: for \( v = r_1 \) and \( v = r_2 \), the \( S \)-circuits are \( (b_1^1 b_2^2 b_1^1 b_1^1 v s_2 r_2 d h) \) and \( (b_1^1 b_2^2 b_1^1 v s_2 r_1 d h) \), respectively. The \( U \)-circuit is \( C_d = (b_1^1 b_2^2 r_1 s_2 r_2 b_3^2 h) \).

Let \( a_1 y^1 \leq a_0^1 \) (resp., \( a_2 y^2 \leq a_0^2 \)) be an ATSP Tree Inequality for \( ATSP(V'_1) \) (resp., \( ATSP(V'_2) \)) with respect to bud \( S_1 \) (resp., bud \( S_2 \)). Assume that \( V'_1 \cap V'_2 = \{d_1, h_1\} = \{d_2, h_2\} \), \( d = d_1 = d_2 \) and \( h = h_1 = h_2 \), where \( d_1 \in S_1 \) and \( d_2 \in S_2 \) are tips. Assume also w.l.o.g. that \( \omega = \omega_1 = \omega_2 \) (by scaling). Define \( V' = V'_1 \cup V'_2 \), \( S = S_1 \cup S_2 \) and construct inequality \( ay \leq y_0 \) for \( ATSP(V') \) by the following Tree Composition:

\[
\begin{align*}
a_e = \begin{cases} 
  a_i^e, & e \in A(V'_i \setminus h_i), \quad i = 1, 2; \\
  \omega, & e \in A(S) \setminus \{A(S_1) \cup A(S_2)\}; \\
  0, & \text{otherwise}; 
\end{cases}
\end{align*}
\]

(5.14)

and \( a_0 = a_0^1 + a_0^2 \). This composition is illustrated in Figure 5.4. After composition, two buds containing their corresponding tips \( d_1 \) and \( d_2 \) are combined into a single bud \( S \) in Fig. 5.4(3), and \( d_1 \) and \( d_2 \) (resp., nodes \( h_1 \) and \( h_2 \)), are merged into a single tip \( d \) (resp., node \( h \)).
Chapter 5. Symmetric Inequalities for Asymmetric Travelling Salesman Polytopes

118

Chapter 5. Symmetric Inequalities for Asymmetric Travelling Salesman Polytopes

118

Theorem 5.11 (Tree Composition Theorem) Let $a_1^1 y^1 \leq a_0^1$ and $a_2^2 y^2 \leq a_0^2$ be ATSP Tree Inequalities for ATSP($V'_1$) and ATSP($V'_2$) w.r.t. buds $S_1$ and $S_2$, respectively, and let $a y \leq a_0$ be obtained by the Tree Composition (5.14). Then

(i) $a y \leq a_0$ is a valid inequality for ATSP($V'$), where $V' = V'_1 \cup V'_2$.

(ii) if, furthermore, the composing inequalities are composable ATSP Tree Inequalities w.r.t. buds $S_1$ and $S_2$, then the following statements hold:

(A) The composed inequality $a y \leq a_0$ is facet-defining for ATSP($V'$).

(B) The composed inequality $a y \leq a_0$ is a composable ATSP Tree Inequality for ATSP($V'$) with respect to the composed bud $S = S_1 \cup S_2$.

(C) If $a^1 y^1 \leq a_0^1$ is a composable Tree for ATSP($V'_1$) with respect to another bud $S'$ with tip $d'$, satisfying $S' \cap S_1 = \emptyset$, then, the composed inequality $a y \leq a_0$ is a composable ATSP
Chapter 5. Symmetric Inequalities for Asymmetric Travelling Salesman Polytopes 119

Tree Inequality for ATSP(V') with respect to bud S'.

The proof is given in Section 6.

Statements (B) and (C) ensure that the Tree Composition procedure can be repeatedly applied so as to construct a large class of tree-like facet-defining inequalities for ATSP polytopes. For instance, Theorem 5.11 implies that Fig. 5.4(3) represents an ATSP Tree Inequality with respect to any one of buds S, S' and S'', along with a corresponding tip (S by Statement (B); the other cases by Statement (C)). Therefore, the resulting inequality can be further used for Tree composition with respect to any bud. By repeating Tree Compositions, a large class of tree-like facet-defining inequalities may then be constructed for ATS polytopes.

Finally, we show how the Tree Composition generates the ATSP version of a subclass of the Path Trees (See Figure 5.5), which were introduced as generalizations of the well-known clique trees in [44] and proven to be STSP facet-defining in [42].
Write inequality (5.12) in the following equivalent form:

\[
\sum_{q=1}^{N} y(A_q) + \sum_{i=1}^{s} N_i y(B^i) \leq \sum_{q=1}^{N} |A_q| + \sum_{i=1}^{s} N_i(|B^i| - 1) - \frac{1}{2} \sum_{q=1}^{N} (k_q + 1),
\]

where \(k_q\) denotes the number of spikes intersected by circle \(q\). By (S1') and (S2'), a Regular Star with an isolated node (i.e., Path or Wheelbarrow) is a composable ATSP Tree Inequality with respect to any spike as a bud. For \(i = 1, 2\), let \(a^i y^i \leq a^i_0\) be such a Regular Star Inequality with an isolated node \(h_i\) in form (5.15) for \(ATSP(V_i)\). Then, the Tree Composition can be applied to these inequalities and the resulting inequality \(a y \leq a_0\) can be easily shown to be in form (5.15) as well. It then follows by repeating the
Tree Composition that the Path Tree Inequalities in form (5.15) are ATSP facet-defining as well.

Figure 5.5 illustrates how a Path Tree facet-defining inequality for the ATS polytopes may be built by repeated Tree Compositions. Fig. 5.5(1) represents a Path Tree Inequality after one application of the Tree Composition on the spikes with weight two. After two other applications of the composition on the spikes with weight one, a larger Path Tree facet-defining Inequality for the ATS polytope is obtained as shown in Fig. 5.5(4). It also follows that the undirected version of this inequality defines a facet of the corresponding STS polytope.

5.6 Proofs of the Main Results in Section 5.4

We show in this section that the proposed classes of symmetric inequalities $cx \leq c_0$ in section 4 define facets of AHP polytopes. For each of the following proofs, we implicitly assume that $fx \leq f_0$ are facet-defining inequalities that dominate $cx \leq c_0$. By Proposition 5.8 and the Symmetric Dominance Lemma, it suffices to show that their dominating facet-defining inequalities $fx \leq f_0$ are symmetric.

We will use the following notation in the sequel. For a subchain $P = (v_1 \ldots v_l)$ of any Hamiltonian path on $V$, define $\overrightarrow{P} \doteq (v_l \ldots v_1)$ to be the same subchain in reverse order. Given any inequality $fx \leq f_0$ for $AHP(V)$, define the $f$-length of $P$ by

$$f(P) = f(v_1 \ldots v_l) \doteq \sum_{i=1}^{l-1} f_{v_i,v_{i+1}}, \quad 1 \leq l \leq n. \quad (5.16)$$

Note that $P$ may be a single edge, or even a single node; in the latter case $f(P) \doteq 0$. Path $P$ is $f$-symmetric if $f(P) = f(\overrightarrow{P})$. If $p = f(P)$, define $\overrightarrow{p} = f(\overrightarrow{P})$. By definition, for any symmetric inequality $cx \leq c_0$, all edges $e \in A$ are $c$-symmetric.
Chapter 5. Symmetric Inequalities for Asymmetric Travelling Salesman Polytopes

Proof of Theorem 5.5: Define

\[ \alpha = f(b_i b_i') \quad \overline{\alpha} = f(b_i b_i') \quad \beta = f(b_i n_i) \quad \overline{\beta} = f(b_i' n_i) \]

Define also \( P_l = (b_i' \ldots b_i') \) for all \( l = 1, \ldots, s \).

Given in Figure 5.6 are six \( c \)-tight paths on \( V \) (cf. Figure 5.1, disregarding \( h \) and inequality (5.12)), where the dashed box contains an even (possibly zero) number of spikes and a fixed Hamiltonian chain on these spikes. The node \( g \) in Figure 5.6 or in the following given paths and chains may or may not be present.

(i) Comparing path \( P = P(a) \) with path \( P' = P(a) \cup (b_i', b_i') \setminus (b_i b_i') \), i.e.,

\[ 0 = f x^P - f x^{P'} = f_{b_i' b_i'} - f_{b_i' b_i'} \]

yields \( f_{b_i' b_i'} = \overline{\alpha} \). Replacing \( i, j, k \) in \( P(a) \) by \( k, i, j \), we obtain as above that \( f_{b_i' b_i'} = \overline{\alpha} \).

Similarly, we show by reversing these paths that

\[ f_{b_i' b_i'} = f_{b_i' b_i'} = f_{b_i' b_i'} = \alpha. \quad (5.17) \]
Chapter 5. Symmetric Inequalities for Asymmetric Travelling Salesman Polytopes

(ii) Let $P(l)$ be obtained from $P(b)$ by replacing $(b^i_l, b^j_l)$ and $(b^j_l...b^i_l)$ with $(b^i_l, b^j_l)$ and $(b^j_l...b^i_l)$, respectively, for $l = 2, ..., n_j - 1$. Let $P(n_j)$ be obtained from $P(d)$ by replacing $(b^i_l, b^j_l)$ and $P_j$ with $(b^i_l, b^j_l)$ and $\overleftarrow{P}_j$, respectively. By (i), and by comparing $P(n_j)$ with $P(d)$, $P(l)$ with $P(b)$ for $l = 2, ..., n_j - 1$, it follows that the following $n_j - 1$ linearly independent equations hold:

$$\sum_{t=1}^{l-1} f_{b^i_t b^j_{t+1}} = \sum_{t=1}^{l-1} f_{b^j_t b^i_{t+1}}, \quad l = 2, ..., n_j.$$

Therefore, edges $(b^i_l, b^j_{l+1})$ are $f$-symmetric for all $1 \leq t \leq n_j - 1$ and $1 \leq j \leq s$, and $f(P_l) = f(\overleftarrow{P}_l)$ for all $1 \leq l \leq s$.

(iii) Let $\gamma = f_{b^n_{n-1} b^j_{n-1}} = f_{b^n_{n-1} b^j_{n-1}}$, due to (ii). From path $P(e)$, we construct:

$$P \doteq (b^k_{n-k} ... b^i_l b^i_{n_j} b^j_{n_j} ... b^j_1),$$

and from path $P(c)$:

$$P' \doteq (b^i_l ... b^j_{n_j} b^k_{n-k} ... b^i_l).$$

By (i) and (ii), and by comparing $P$ with $P(e)$ and $P'$ with $P(c)$, respectively, it follows

$$\alpha + \beta = \overleftarrow{\alpha} + \overleftarrow{\beta}, \quad \alpha + \gamma = \overleftarrow{\alpha} + \overleftarrow{\beta}.$$

Further, comparing $\overleftarrow{P}$ with $\overleftarrow{P}(c)$ yields $\overleftarrow{\alpha} + \gamma = \alpha + \overleftarrow{\beta}$. From these three equations, we obtain $\alpha = \overleftarrow{\alpha}$ and $\beta = \overleftarrow{\beta}$. Moreover, if node $g$ exists, i.e., for Path Inequalities, define

$$P \doteq P(d) \cup \overleftarrow{P} \cup (b^i_{n_j}, b^j_{n_j}) \setminus \{P \cup (b^i_{n_j}, g)\}.$$

By (i), (ii), (iii), and by comparing this $P$ with $P(d)$ and $\overleftarrow{P}$ with $\overleftarrow{P}(d)$ respectively, it follows

$$f_{b^n_{n-1} g} = f_{g b^n_{n-1}} = \beta, \quad \forall 1 \leq i \leq s.$$

Since the above hold for any $i, j, k$, it follows that $A(\{b^1_i, ..., b^k_i\}) \cup A(\{g, b^n_{n_j}, ..., b^n_{n_j}\})$ contains only $f$-symmetric edges.
(iv) For any \( j \) such that \( n_j \geq 3 \), construct the following c-tight path:

\[
P' = (b'_1 \ldots b'_n g \ldots b''_{n_k} \ldots b''_{n_j} b''_{i+1} \ldots b''_j b''_{i+1} \ldots b''_i \ldots b''_1) \.
\]

Observe that by (ii) and (iii), all edges in \( P' \) are \( f \)-symmetric except edge \((b''_{i+1}, b''_{i})\), therefore comparing \( P' \) with \( \overrightarrow{P} \) yields that \((b''_{i+1}, b''_{i})\) is also \( f \)-symmetric for \( 1 \leq l \leq n_j - 2 \).

Further, if \( n_j \geq 4 \), from \( P(d) \), construct the following c-tight path:

\[
P = (b'_1 \ldots b'_n g \ldots b''_{n_k} \ldots b''_{i+1} b''_i b''_{i+1} \ldots b''_j b''_{i+1} \ldots b''_1) \.
\]

Observe that all edges in \( P \) have been shown to be \( f \)-symmetric except edge \((b'_i, b'_{i+1})\). Therefore, comparing \( P \) with \( \overrightarrow{P} \) implies that \((b'_i, b'_{i+1})\) is \( f \)-symmetric for all \( 1 \leq t < l < n_j \). It follows that all edges in \( A(\{b'_i, \ldots, b'_n\}) \) are \( f \)-symmetric for all \( 1 \leq j \leq s \).

(v) For any pair of nodes \( b'_i \) and \( b'_{i+1} \) such that \( 1 \leq t \leq n_i - 1 \), \( 1 \leq l \leq n_j - 1 \), \( i \neq j \) and \( M(b'_i) \leq M(b'_{i+1}) \) (cf. page 110 for the definition of \( M(*) \)), define c-tight paths: for \( t \geq 2 \), \( P(t, l + 1) \doteq P(f) \); otherwise \( t = 1 \), we have

\[
P(t, l + 1) = (b'_i \ldots b'_n g \ldots b''_{n_k} \ldots b''_{i+1} b''_i b''_{i+1} \ldots b''_j g \ldots b''_1 \ldots b''_1) \, .
\]

Observe that \( P(t, l + 1) \) contains exactly one edge \((b''_{i+1}, b''_{i})\) not shown to be \( f \)-symmetric, therefore comparing \( P(t, l+1) \) with \( \overrightarrow{P} (t, l+1) \) yields \((b''_{i+1}, b''_{i})\) as \( f \)-symmetric.

For all \( b'_i \) and \( b'_{i+1} \) such that \( M(b'_i) > M(b'_{i+1}) \), exchange \( i \) and \( j \), and then repeat (v).

If node \( g \) exists, define c-tight paths

\[
P_j(t) \doteq (b'_i \ldots b'_{n_k} \ldots b''_{n_k} \ldots b''_{i+1} g \ldots b''_j \ldots b''_{i+1}, \quad 1 \leq t \leq n_j - 1) \, .
\]

Observe also that for any \( 1 \leq t \leq n_j - 1 \), the edges in \( P_j(t) \) are shown to be \( f \)-symmetric except for the edge \((b'_i, g)\). By comparing \( P_j(t) \) with \( \overrightarrow{P}_j(t) \), it follows that edges \((b'_i, g)\) are \( f \)-symmetric for all \( t \) and \( j \).
From (iii), (iv) and (v), it follows that \( fx \leq f_0 \) is a symmetric inequality. The proof is complete. \( \square \)

**Proof of Theorem 5.6:** This proof is conducted by successive augmentations of a set \( E_f \) of \( f \)-symmetric edges until \( E_f = A(V) \). The set \( E_f \) has the property that \( e \in E_f \) implies \( \overleftarrow{e} \in E_f \). For any subset \( E' \subseteq A(V) \), define the following operation

\[
E' \oplus E_f : \text{ replace } E_f \text{ with } E_f \cup \{ e \in A(V) : \text{ either } e \text{ or } \overleftarrow{e} \text{ is in } E' \}.
\]

The proof consists of the following three phases.

(i) An initial set \( E_f \subseteq A(V) \) of \( f \)-symmetric edges with the same \( f \)-length is constructed.

(ii) A sequence of pairwise \( c \)-tight path comparisons is conducted and \( E_f \) is augmented at each step. We denote this by \( \"P, P' \implies something\" \), meaning that comparing \( c \)-tight paths \( P \) and \( P' \) implies \( \"something\" \). Moreover, the implication may implicitly use the previous results and obvious symmetries. The \( f \)-length of the underlined part in \( P \) has been shown to be equal to that in \( P' \).

(iii) Select any edge \( e \in A(V) \setminus E_f \) and construct a \( c \)-tight path \( P \) on \( V \) such that \( e \in P \) and \( P \setminus e \subseteq E_f \). Thus, \( P, \overleftarrow{P} \implies \{ e \} \oplus E_f \). Repeating (iii) until \( E_f = A(V) \).

**Envelope:** Let \( \alpha = f_{12}, \beta = f_{34}, \overleftarrow{\beta} = f_{43}, \gamma = f_{24}, \overleftarrow{\gamma} = f_{32}, \xi = f_{56} \) and \( \overleftarrow{\xi} = f_{56} \).

(i) Let \( C^1 = \langle 124356 \rangle \) and \( C^2 = \langle 214356 \rangle \).

\[
C^1 \setminus (1, 2), C^1 \setminus (3, 5) \implies f_{12} = f_{35}; C^2 \setminus (2, 1), C^2 \setminus (3, 5) \implies f_{21} = f_{35}.
\]

Hence, \( f_{21} = f_{12} = \alpha \) and let \( E_f = \{(1, 2), (2, 1)\} \).

Observe that for any \( e \in E_1 = \{(3, 5)\} \cup \{(1), (4, 6)\} \), there exists \( e' \in E_f \) and a Hamiltonian cycle \( C \) on \( V \) such that \( e, e' \in C \) and \( cx^C = c_0 + 1 \). (For \( e = (3, 5) \), let \( e' = (1, 2) \) and \( C = C^1 \) above; for \( e = (1, 4) \), let \( e' = (2, 1) \) and \( C = C^2 \) above; for \( e = (1, 6) \), let \( e' = (2, 1) \) and \( \langle 216534 \rangle \).) Thus, comparing \( C \setminus e \) and \( C \setminus e' \), and then
Chapter 5. Symmetric Inequalities for Asymmetric Travelling Salesman Polytopes

\( \overrightarrow{C} \setminus e \) and \( \overrightarrow{C} \setminus e' \) implies that \( e \) is \( f \)-symmetric. Hence, set \( E_f = \{e\} \oplus E_f \). It follows by repeating the above that \( E_f = E_1 \oplus E_f \). Moreover, \( f_e = \alpha \) for all \( e \in E_f \).

(ii) \((4356\ 21), (12\ 4356) \implies f_{62} = f_{24} = \gamma. \)

\((2156\ 43), (156243) \implies f_{64} = 2\gamma - \alpha, \) due to the above and (i).

Reversing the above two pairs of paths, we have

\( f_{26} = f_{42} = \frac{\gamma}{2} \); and \( f_{46} = 2\frac{\gamma}{2} - \alpha, \) respectively.

Using the above results, we have the following implications:

\( (126435), (534621) \implies \frac{\gamma}{2} + 2\gamma - \alpha + \frac{\beta}{2} = \frac{\gamma}{2} + 2\gamma - \alpha + \beta; \) and

\( (561243), (342165) \implies \xi + \gamma + \frac{\beta}{2} = \frac{\xi}{2} + \gamma + \beta. \)

It follows that \( \xi = \frac{\xi}{2} \). By symmetry, \( \beta = \frac{\beta}{2} \). Hence, \( \gamma = \frac{\gamma}{2} \) and

\( \{(3,4), (5,6), (4,6), (2,4), (2,6)\} \oplus E_f. \)

(iii) For edges \((1,3),(1,5),(2,3),(2,5),(3,6),(4,5)\), construct, respectively, the corresponding desired \( e \)-tight paths

\( (213465), (215643), (123465), (125643), (124365), (345621). \)

\( \square \)

**Ladder:** Let \( \alpha = f_{47} \) and \( E_f = \emptyset. \)

(i) Let \( C^1 = \langle 7216534 \rangle \) and \( C^2 = \langle 7432156 \rangle \).

\( C^1 \setminus (7,2), C^1 \setminus (4,7) \implies f_{72} = f_{47} = \alpha; \) it follows \( f_{72} = f_{67} = \alpha \) by exchanging nodes 3,4 with 5,6.

\( C^2 \setminus (7,4), C^2 \setminus (6,7) \implies f_{74} = f_{67}. \) Hence, \( f_{74} = f_{47} = f_{67} = f_{76} = \alpha \) and \( \{(4,7), (6,7)\} \oplus E_f. \)

Observe that for any \( e \in E_1 \equiv \{(7,2)\} \cup \{(1,3,4,5,6)\} \), there exists \( e' \in E_f \) and a Hamiltonian cycle \( C \) on \( V \) such that \( e, e' \in C \) and \( \sigma_C^C = c_0 + 1 \) (either \( C^1 \) or \( C^2 \) above).
Chapter 5. Symmetric Inequalities for Asymmetric Travelling Salesman Polytopes

Thus, comparing $C \setminus e$ and $C \setminus e'$, $\overrightarrow{C} \setminus e$ and $\overrightarrow{C} \setminus e'$ implies that $e$ is $f$-symmetric. Hence, $\{e\} \oplus E_f$. Repeating the above yields $E_1 \oplus E_f$. Moreover, $f_e = \alpha$ for all $e \in E_f$.

(ii) $(7432156), (7432165) \Rightarrow \{(5,6), (3,4)\} \oplus E_f$.

$(1274356), (1276534) \Rightarrow \{(3,5)\} \oplus E_f$.

$(7213465), (7215643) \Rightarrow \{(4,6)\} \oplus E_f$.

$(7213465), (5643127) \Rightarrow \{(1,2)\} \oplus E_f$.

$(7432156), (6512347) \Rightarrow \{(2,3), (2,5)\} \oplus E_f$.

(iii) For edges $(1,7),(2,4),(2,6),(3,6),(3,7),(4,5),(5,7)$, construct, respectively, the corresponding desired $c$-tight paths

$(5643217), (1243567), (1265347), (7436521), (1256437), (1235647), (1234657)$.

This completes the proof for $|V'| = 8$. If node $g$ exists, revise the above proof as follows:

- for every path or cycle used in (i), (ii) and (iii) above, replace subchains $(12)$ and $(21)$ by $(g12)$ and $(21g)$, respectively;
- in (i), replace $E_1$ by $\{(7,2), (1,g)\} \cup \{\{g\}, \{3,4,5,6\}\}$;
- in (iii), replace $(1,7)$ by $(g,7)$, and add: “for edges $(2,g), (1,3), (1,4), (1,5), (1,6), (1,7)$, construct, respectively, the following desired $c$-tight paths

$(12g43567), (7213465g), (7214356g), (7215643g), (7216534g), (g5643217)$.”

This shows that the result holds for the case $|V'| = 9$. Finally, apply Clique-Lifting on the isolated node $g$ to complete the proof.

Proof of Theorem 5.7: Let $cx \leq c_0$ be the restriction of inequality (5.13), and let $fx \leq f_0$ be a facet-defining inequality for $AHP(V)$ that dominates $cx \leq c_0$. Note
that by definition, for any \( t, q \in \{1, \ldots, p\} \), there exists a "zig-zag" Hamiltonian path \( \hat{P} = (r_q \ldots s_t) \) on \( R \cup T \) such that \( c(\hat{P}) = 2p - 1 \) (cf. Fig. 5.7(1)). Let \( (b^k_{n_k} \ldots b^k_1) \) and \( (b^k_1 \ldots b^k_{n_k}) \) denote two fixed subchains \( (r_q \ldots s_t) \) and \( (s_t \ldots r_q) \), respectively. Let \( n_i = 2 \) for all \( i \). Thus, following (i), (ii), (iii) and (v) in the proof of Theorem 5.5, where \( n_i = 2 \) for all \( i \), and the subchains \( \hat{P} \) and \( P \) are replaced by \( (r_q \ldots s_t) \) and \( (s_t \ldots r_q) \), respectively, we show that all edges in \( A \setminus A(R \cup T) \) are \( f \)-symmetric. Construct the four \( c \)-tight paths on \( V \), see Figure 5.7.

For any \( 1 < q < p \), we obtain from path \( P(1) \) a \( c \)-tight path

\[
P' = P(1) \cup (b_2^1 r_q g b_2^2 b_1^1) \setminus (b_1^1 b_2^2 g r_q s_q) .
\]

Thus, comparing \( P(1) \) with \( P' \), and \( \overleftarrow{P} \) with \( \overleftarrow{P'} \), respectively, yields \( f_{s_t r_q} = f_{r_q s_t} \).

For any \( 1 < q < t < p \), we obtain from path \( P(2) \) a \( c \)-tight path

\[
P'' = P(2) \cup \{(r_t, b_2^1), (s_t, r_t)\} \setminus \{(s_q, r_t), (r_t, s_t)\}.
\]

Figure 5.7: \( c \)-tight paths for the Chain Inequality
Chapter 5. Symmetric Inequalities for Asymmetric Travelling Salesman Polytopes

As above, comparing $P(2)$ with $P''$, and $\overrightarrow{P}(2)$ with $\overrightarrow{P''}$, respectively, yields $f_{sqt}=f_{rstq}$.

Finally, observe that all edges in paths $P(3)$ and $P(4)$ have been shown to be $f$-symmetric except edges $(r_q,r_t)$ and $(s_q,s_t)$. Thus, comparing $P(3)$ with $\overrightarrow{P}(3)$, and $P(4)$ with $\overrightarrow{P}(4)$, respectively, implies that edges $(r_q,r_t)$ and $(s_q,s_t)$ are $f$-symmetric. It follows that $fx \leq fo$ is symmetric. 

5.7 Proofs of the Main Results in Section 5.5

Proof of Proposition 5.10: Let $d \in S$ and $u = d' \in S'$ be the tips. For any $v \in S \setminus d$, we have $a_{uv} = 0$ by (S2'). By (S1'), construct an $a$-tight circuit $C$ containing $(u,v)$. Thus, $C$ contains a subchain $(uv...qd)$. First, observe that $a_{qd} = \omega$; otherwise replacing the subchain in $C$ with $(uq...vd)$ yields a circuit with $a$-length at least $a_0 + \omega$. So, $C$ is in the form of either $(v...h...qdp...u)$ or $(v...qdp...h...u)$. If $a_{dp} = 0$, then construct the desired $S$-circuit by moving $h$ in $C$ to the position between $d$ and $p$; else, $a_{dp} = \omega$, and moving $d$ in $C$ to the position immediately before $h$ yields the desired $S$-circuit, since $a_{qd} + a_{dp} = 2\omega \leq a_{pq}$ (due to (S2')) and $ay \leq a_0$ is valid for $ATSP(V')$. This shows that (S3) holds.

To show that (S4) also holds, we first claim that there exists an $a$-tight circuit $C$ on $V'$ of the form $(u...pdq)$ with either (i) $p,q \in Z \cup h$ or (ii) $p,q \in S$. To prove this claim, let $n = |V' \setminus h|$ and let $cx \leq c_0$ be the restriction of $ay \leq a_0$ w.r.t. node $h$. By (S1') and polyhedral equivalence, there are $n(n-1)-1$ affinely independent incidence vectors $y$ of $a$-tight circuits on $V'$ (resp., incidence vectors $x$ of $c$-tight paths on $V' \setminus h$). If the claim is false, then these incidence vectors $y$ satisfy equations $y(d:S \setminus d)+y(S \setminus d:d)=1$, $ay=a_0$ and $y(V' \setminus h)=n-1$. Observe that the above three equations are in "$h$-canonical form", and their restrictions $x(d:S \setminus d)+x(S \setminus d:d)=1$, $cx=c_0$ and $x(V' \setminus h)=n-1$ are three linearly independent equations on $AHP(V' \setminus h)$. (Note that $c_e \geq 2\omega$ for all $e \in A(S \setminus d)$,
due to (S2').) Further, these equations are satisfied by \( \dim(AHP(V' \setminus h)) = n(n - 1) - 1 \) affinely independent incidence vectors \( x \) above, a contradiction. Hence, the claim holds.

Construct \( C_d = (u...pq) \). Note that \( a_{pd} = a_{dq} = 0 \) for case (i); \( a_{pq} \geq 2\omega = a_{pd} + a_{dq} \) for case (ii). Therefore, \( C_d \) is the desired \( U \)-circuit.

To prove the Tree Composition Theorem, we need the following four auxiliary results.

Let \( ay \leq a_0 \) be a composable ATSP Tree Inequality with respect to a bud \( S \) for \( ATSP(V') \) and let \( \{h, S, Z\} \) be the corresponding partition. Define \( V = V' \setminus h \). Then, the restriction \( cx \leq c_0 \) of \( ay \leq a_0 \) w.r.t. \( h \) is called a composable Tree Component for \( AHP(V) \). Define the following special paths w.r.t. \( cx \leq c_0 \) and tip \( d \):

**Z-path:** A \( c \)-tight path \( (t...vd...s) \) w.r.t. \( v \in Z \) such that
\[
c_e = 0, \forall e \in (\{t, v\} : \{d, s\}).
\]

**S-path:** A \( c \)-tight path \( (z...uv...qd) \) w.r.t. \( v \in S \setminus d \) such that
\[
c_{zu} = c_{uv} = 0 \text{ and } c_e \geq \omega \text{ for all } e \text{ used in subchain } (v...qd).
\]

**U-path:** A Hamiltonian path \( P_d \) on \( V \setminus d \) such that
\[
c(P_d) = c_0 \text{ and both endnodes in } Z.
\]

**Lemma 5.12** Let \( cx \leq c_0 \) be a composable Tree Component for \( AHP(V) \). Then,

1. for every \( v \in Z \), there exists a Z-path \( (t...vd...s) \);
2. for every \( v \in S \setminus d \), there exists an S-path \( (z...uv...qd) \); and
3. there exists \( U \)-path \( P_d \) on \( V \setminus d \).

**Proof.** Note first that by polyhedral equivalence and (S1), \( cx \leq c_0 \) defines a non-trivial facet of \( AHP(V) \).
Chapter 5. Symmetric Inequalities for Asymmetric Travelling Salesman Polytopes 131

(1) For every $v \in Z$, there exists a $c$-tight path containing $(v, d)$ of the form $(t...vd...s)$. If there is an edge $e \in \{\{t, v\}, \{d, s\}\}$ with $c_e > 0$, say $e = (v, s)$, w.l.o.g., then a Hamiltonian path $P = (t...vs...d)$ yields $cx^P > c_0$, a contradiction.

(2) For every $v \in S \setminus d$, by (S3), there exists an $a$-tight $S$-circuit $(z...uv...qd)$ with $a_{uv} = 0$. Remove $h$ from this circuit to obtain an $S$-path $(z...uv...qd)$. (Note that $c_{uv} = 0$ holds by definition and $c_{zv} = 0$ follows since $(u...zv...qd)$ is also a $c$-tight path. The last condition follows by contradiction: if for some $e$ in subchain $(v...qd)$, $c_e < \omega$, replacing $e$ with $(d, v)$ and then connecting an endnode of the resulting subchain with node $u$ of $(u...z)$ yields a Hamiltonian path $P$ with $c(P) > c_0$.)

(3) By (S4), there exists a $U$-circuit $C_d$. Then, $U$-path $P_d$ is obtained by removing $h$ from $C_d$. If one of the endnodes in $P_d$ is in $S$, then connecting this endnode with $d$ yields a Hamiltonian path $P$ on $V$ with $c(P) = c_0 + \omega$, a contradiction. □

Lemma 5.13 (3-Path Lemma) Let $cx \leq c_0$ and $fx \leq f_0$ be two valid inequalities for $AHP(V)$. Assume the former inequality is symmetric and dominated by the latter. Let $U_1, U_2, U_3$ be a partition of $V$ and let $P_i = (u_i...v_i)$ be a Hamiltonian path on $U_i$, $i = 1, 2, 3$. Define

$$E_p = \{P_i, \overline{P_i} : i = 1, 2, 3\}, \quad E_0 = \bigcup_{i \neq j} (\{u_i, v_i\} : \{u_j, v_j\}).$$

If $c_e = 0$ for all $e \in E_0$ and $\sum_{i=1}^{3} c(P_i) = c_0$, then

(i) $f(P_i) = f(\overline{P_i})$ for $i = 1, 2, 3$; and

(ii) $f_e$ is a constant for all $e \in E_0$.

Proof. (i) From $P_1, P_2, P_3$, we obtain two Hamiltonian circuits on $V$:

$$C \doteq (u_1...v_1u_2...v_2u_3...v_3), \quad C' \doteq (u_1...v_1v_2...u_2u_3...v_3), \quad (5.18)$$
Chapter 5. Symmetric Inequalities for Asymmetric Travelling Salesman Polytopes

such that \( c(C) = c(C') = c_0 \). Comparing \( c \)-tight paths \( C \setminus (v_3, u_1) \) and \( C \setminus (v_1, u_2) \) yields \( f_{v_3u_1} = f_{v_3u_2} \); comparing \( C' \setminus (v_3, u_1) \) with \( C' \setminus (v_1, v_2) \) yields

\[
f_{v_1u_2} = f_{v_3u_1} = f_{v_1u_2}.
\] (5.19)

Further, comparing

\[
P = (u_3...v_3u_1...v_1u_2...v_2) \quad \text{and} \quad P' = (u_3...v_3u_1...v_1v_2...u_2)
\]

yields

\[
0 = f(P) - f(P') = f_{v_1u_2} + f(P_2) - f_{v_1u_2} - f(P_2) = f(P_2) - f(P_2).
\]

Similarly, we show that \( P_1 \) and \( P_3 \) are \( f \)-symmetric. Hence, (i) holds.

(ii) Observe that for any \( e_0 \in E_0 \), there exists a Hamiltonian circuit \( C \subseteq E_p \cup E_0 \) on \( V \) such that \( e_0 \in C \) and \( c(C) = c_0 \). W.l.o.g., assume that \( C \) has form (5.18) and \( e_0 = (v_1, u_2) \). Comparing the \( c \)-tight paths \( C \setminus (v_1, u_2) \), \( C \setminus (v_2, u_3) \) and \( C \setminus (v_3, u_1) \) yields

\[
f_{v_1u_2} = f_{v_3u_3} = f_{v_3u_1} = f(C) - f_0.
\]

Similarly, using \( \overline{C} \), we obtain

\[
f_{u_3v_2} = f_{u_3v_2} = f_{u_1v_3} = f(\overline{C}) - f_0.
\]

Further, by (i) and \( f(C \setminus (v_1, u_2)) = f(\overline{C} \setminus (u_2, v_1)) \), we have

\[
0 = \sum_{i=1}^{3} f(P_i) + 2f_{v_1u_2} - \sum_{i=1}^{3} f(\overline{P}_i) - 2f_{u_3v_1} = 2f_{v_1u_2} - 2f_{u_2v_1}.
\]

Hence, all edges \( e_0 \in E_0 \) are \( f \)-symmetric.

Since all elements in \( E_0 \cup E_p \) are shown to be \( f \)-symmetric, repeating (5.19) for all permutations of indices of \( u_1, u_2, u_3 \) and exchanging the roles of the \( u \) and \( v \) nodes imply that (ii) holds. The proof is complete. \( \square \)
Lemma 5.14 The composed inequality \( ay \leq a_0 \) for ATSP(V') defined in (5.14) is valid for ATSP(V').

Proof. It suffices to show that its undirected version \( a'y \leq a_0 \) is valid for STSP(V').

Let \( S_1' = S_1 \setminus d_1 \) and \( S_2' = S_2 \setminus d_2 \). An edge \( e \) is called an \( \omega \)-edge if \( e \in E(d : S_1' \cup S_2') \cup E(S_1' : S_2') \). Let \( C^* \) be any solution to

\[
\max\{a'y^C : C \text{ is a Hamiltonian cycle}\} = a_0^*.
\]

Let \( C = C^* \).

(i) [\( \omega \)-Edge Reduction] Let \( E(V_1' : V_2 \setminus d_2) \) be the set of all crossing edges, and let \( E(S_1' : S_2') \) be the set of all \( \omega \)-crossing edges.

For any Hamiltonian cycle \( C_0 \) on \( V' \), let \( N_\omega(C_0) = |C_0 \cap E(S_1' : S_2')| \) denote the number of \( \omega \)-crossing edges used in \( C_0 \). If there exists a pair \((u, v)\) and \((s, t)\) of \( \omega \)-crossing edges in \( C \) such that some subchain connecting \((u, v)\) and \((s, t)\) in \( C \) contains an odd number of crossing edges, then transform \( C \) to Hamiltonian cycle \( C' \) as follows:

\[
C \doteq (d...uv...st...r) \rightarrow C' \doteq (d...us...vt...r).
\]

Since \( a'_us + a'_vt \geq 2\omega \), we have \( a_0^* = a'(C) \leq a'(C') = a_0^* \), and \( N_\omega(C') = N_\omega(C) - 2 \). Let \( C = C' \). Observe that such a reduction is always possible whenever \(|N_\omega(C)| > 2\). Hence, we repeat (i) until \( N_\omega(C) \leq 2 \).

(ii) Exchange sets \( S_1' \) and \( S_2' \) in (i) and then apply (i).

(iii) First, observe that \( C \) contains at most one edge in \( E_\omega = E(S_1' : S_2') \). Otherwise, \( C \) contains exactly two edges \((u, v), (t, s) \in E_\omega \). Further, by (i) and (ii), \( C \) contains the subchain \((pdq)\) with \( a'_{pd} = a'_{dq} = 0 \), where \( d \) is the common tip. Replacing subchains \((uv)\) and \((pdq)\) in \( C \) with \((udv)\) and \((pq)\), respectively, we obtain a Hamiltonian cycle \( C' \) on \( V' \) such that \( a'(C') \geq a'(C) + \omega > a_0^* \), a contradiction.
Chapter 5. Symmetric Inequalities for Asymmetric Travelling Salesman Polytopes

Now, let \( P_1 = C \cap E(V'_1) \) and \( P_2 = C \cap E(V'_2) \). Note that \( P_1 \) and \( P_2 \) are subsets of some Hamiltonian cycles \( C_1 \) on \( V'_1 \) and \( C_2 \) on \( V'_2 \), respectively. If \( C \) contains no edges in \( E_w \), then

\[
a'(C^*) = a'(C) = a'(P_1) + a'(P_2) \leq a'(C_1) + a'(C_2) \leq a_0^1 + a_0^2 = a_0.
\]

Otherwise, \( C \) contains exactly one edge \((u, v) \in E_w \) with \( u \in S'_1 \) and \( v \in S'_2 \), and a subchain \((pdq)\). Let \( P'_1 = P_1 \cup (u, d) \) and \( P'_2 = P_2 \cup (d, v) \). If \( p, q \in V'_1 \) (resp., \( p, q \in V'_2 \)), then, \( P'_1 \) (resp., \( P'_2 \)) is a subset of some Hamiltonian cycle on \( V'_2 \) (resp., \( V'_1 \)). Else, w.l.o.g. assume \( q \in V'_1 \) and \( p \in V'_2 \), and observe also that either \( P'_1 \) is a subset of some Hamiltonian cycle on \( V'_1 \) or \( P'_2 \) is on \( V'_2 \), since otherwise both \( P'_1 \) and \( P'_2 \) contain some sub-cycles covering \( d \), and so does \( P_1 \cup P_2 \cup (u, v) \subset C \), which contradicts that \( C \) is a Hamiltonian cycle on \( V \). If \( P'_1 \) is contained in some Hamiltonian cycle \( C'_1 \) on \( V'_1 \), then

\[
a'(C^*) = a'(C) = a'(P'_1) + a'(P'_2) \leq a'(C'_1) + a'(C_2) \leq a_0^1 + a_0^2 = a_0;
\]

else the above holds by exchanging indices 1 and 2. \( \Box \)

Let inequality \( ay \leq a_0 \) for \( ATSP(V') \) be obtained by the Tree Composition (5.14) of two ATSP Tree Inequalities \( a^1y^1 \leq a_0^1 \) and \( a^2y^2 \leq a_0^2 \), relative to buds \( S_1 \) and \( S_2 \), respectively, and let \( \{h_1, S_1, Z_1\} \) and \( \{h_2, S_2, Z_2\} \) be the corresponding partitions. Define \( V_i = V'_i \setminus h_i, i = 1, 2 \), and \( V = V' \setminus h \). Recall that \( V_1 \cap V_2 = \{d\} = \{d_1\} = \{d_2\} \), where \( d \) is the common tip. The restrictions \( cx \leq c_0, c^1x^1 \leq c_0^1 \) and \( c^2x^2 \leq c_0^2 \) of \( ay \leq a_0 \), \( a^1y^1 \leq a_0^1 \) and \( a^2y^2 \leq a_0^2 \), w.r.t. \( h \), \( h_1 \) and \( h_2 \), are called Tree Components, for \( AHP(V) \), \( AHP(V_1) \) and \( AHP(V_2) \), respectively. Then, we say that \( cx \leq c_0 \) is obtained by \( AHP \) Tree Composition of two Tree Components \( c^1x^1 \leq c_0^1 \) and \( c^2x^2 \leq c_0^2 \). Recall that the restriction of a composable ATSP Tree Inequality is a composable Tree Component. See Figure 5.4 for an example of the AHP Tree Composition, disregarding nodes \( h_1, h_2, h \).
Theorem 5.15  Inequality $cx \leq c_0$ obtained by AHP Tree Composition of two composable Tree Components $c^1x^1 \leq c_0^1$ and $c^2x^2 \leq c_0^2$ defines a facet of $AHP(V)$.

Proof: The validity of $cx \leq c_0$ follows by Lemma 5.14 and the polyhedral equivalence. Let $fx \leq f_0$ be a facet-defining inequality that dominates $cx \leq c_0$.

Let $\{V_1, V_2 \setminus d_2\}$ be the partition of $V$, and by Lemma 5.12(3), let $P_2 = (u_2...v_2)$ be a $U$-path on $V_2 \setminus d_2$ with $c(P_2) = c_0^2$ and $u_2, v_2 \in Z_2$.

(i) For every $v_1 \in Z_1$, by Lemma 5.12(1), we construct a $Z$-path $(t_1...v_1d...s_1)$ on $V_1$, and let $P_1 = (t_1...v_1)$ and $P_3 = (d...s_1)$; for every $v_1 \in S_1 \setminus d_1$, by Lemma 5.12(2), we construct an $S$-path $(z_1...u_1v_1...q_1d)$ on $V_1$, and let $P_1 = (z_1...u_1)$ and $P_3 = (v_1...q_1d)$. Note that for all $v \in V_1 \setminus d_1$, we obtain $P_1$, $P_2$ and $P_3$ satisfying the conditions in Lemma 5.13, and therefore we have $f_e = f_{ud}$ for all $e \in (\{u_2, v_2\} : V_1) \cup (V_1 : \{u_2, v_2\})$ and $f(P_2) = f(P_0)$. This further implies that for any $c^1$-tight path $P'$ on $V_1$, there exists a $c$-tight path $P$ on $V$ containing $P'$ and $P_2$ such that

$$f_0 = fx^P = f(P') + f(P_2) + f_{ud}.$$ 

Since $c^1x^1 \leq c_0^1$ is facet-defining for $AHP(V_1)$ and the above equality holds for all $c^1$-tight paths $P'$, there exist real numbers $\alpha_1 > 0$ and $\beta_1$ such that $f_e = \alpha_1 c_e + \beta_1$ for all $e \in A(V_1)$.

By partitioning $V$ into $\{V_2, V_1 \setminus d_1\}$, we show by symmetry that for some real numbers $\alpha_2 > 0$ and $\beta_2$, $f_e = \alpha_2 c_e + \beta_2$ holds for all $e \in A(V_2)$.

(ii) For any $v_2 \in Z_2$, construct a $Z$ path $(t_2...v_2d...s_2)$ on $V_2$. As in (i), for every $v_1 \in Z_1$, construct a $Z$-path $(t_1...v_1d...s_1)$ on $V_1$; and for every $v_1 \in S_1 \setminus d$, we construct an $S$-path $(z_1...u_1v_1...q_1d)$ on $V_1$. As required by the conditions for Lemma 5.13, we obtain for the former case

$$P_1 \equiv (t_1...v_1), \quad P_2 \equiv (t_2...v_2), \quad P_3 \equiv (q_1...d...s_2),$$

where $\equiv$ denotes equality up to permutation of the vertices.
Chapter 5. Symmetric Inequalities for Asymmetric Travelling Salesman Polytopes

and for the latter case,

\[ P_1 \doteq (z_1...u_1), \quad P_2 \doteq (t_2...v_2), \quad P_3 \doteq (v_1...q_1d...s_2). \]

Further, since \( c_{z_1v_1} = c_{v_2s_2} = 0 \), by (i) and Lemma 5.13, we have

\[ f_{v_2v_1} = f_{u_1v_2} = f_{v_2s_2} = f_{z_1v_1} = \beta_1 = \beta_2 = \beta. \]

By symmetry, it follows \( f_e = \beta \) for all \( e \in A(V) \setminus \{A(V_1) \cup A(V_2) \cup A(S)\} \).

(iii) For any \( v_i \in S_i \setminus d \), construct an \( S \)-path \((z_i...u_iv_i...q_id)\), \( i = 1,2 \). Recall that \((v_i...q_id)\) uses only edges \( e \) with \( c_e \geq \omega \) and \( c_{q_id} = \omega \) by definition. Possibly \( q_i = v_i \). Comparing the following two \( c \)-tight paths:

\[ P' \doteq (z_1...u_1z_2...u_2v_2...q_2d...s_2), \quad P'' \doteq (z_1...u_1z_2...u_2d...q_2v_2...s_2) \]

yields \( 0 = f(P') - f(P'') = \beta + f_{d_{q1}} - \beta - f_{v_2v_1} = \alpha_1 \omega + \beta - f_{v_2v_1} \). Further, comparing \( \overrightarrow{P'} \) and \( \overrightarrow{P''} \) yields \( f_{v_1v_2} = \alpha_1 \omega + \beta \), due to (i) and (ii).

By exchanging \( V_1 \) and \( V_2 \), we show also \( f_{v_1v_2} = f_{v_2v_1} = \alpha_2 \omega + \beta \). It follows that \( \alpha = \alpha_1 = \alpha_2 \),

\[ f_e = \alpha \omega + \beta, \quad e \in (S_1,S_2) \cup (S_2,S_1). \]

(i), (ii) and (iii) imply that \( f_e = \alpha c_e + \beta \) for all \( e \in A(V) \).

Now, we are in a position to prove the Tree Composition Theorem.

Proof of the Tree Composition Theorem: Part (i) of the theorem is proved in Lemma 5.14. For part (ii), Statement (A) follows from Theorem 5.15, by polyhedral equivalence.

For statements (B) and (C), (S2) holds by inspection. We need to examine conditions (S3) and (S4) for each case.
For (B), Condition (S3) follows by constructing, for every \( v_1 \in S_1 \setminus d \), \( S \)-circuit 
\[(u_2 \ldots v_2 z_1 \ldots u_1 v_1 \ldots q_1 d h)\] 
on \( V' \), where \((u_2 \ldots v_2)\) is a \( U \)-path on \( V_2 \setminus d \) and \((z_1 \ldots u_1 v_1 \ldots q_1 d)\) is an \( S \)-path on \( V_1 \). Similarly, we obtain a requisite \( S \)-circuit for every \( v_2 \in S_2 \setminus d \).

(S4) holds by exhibiting \( U \)-circuit \((u_1 \ldots v_1 u_2 \ldots v_2 h)\) on \( V' \setminus d \), where \((u_i \ldots v_i)\) is a \( U \)-path on \( V_i \setminus d_1, i = 1, 2 \).

For (C), Condition (S3) follows by constructing, for every \( v_1 \in S' \setminus d' \), \( S \)-circuit 
\[(u_2 \ldots v_2 z_1 \ldots u_1 v_1 \ldots q_1 d'h)\] 
on \( V' \), where \((u_2 \ldots v_2)\) is a \( U \)-path on \( V_2 \setminus d \) and \((z_1 \ldots u_1 v_1 \ldots q_1 d')\) is an \( S \)-path on \( V_1 \).

(S4) holds by exhibiting \( U \)-circuit \((u_1 \ldots v_1 u_2 \ldots v_2 h)\) on \( V' \setminus d' \), where \((u_1 \ldots v_1)\) (resp., \((u_2 \ldots v_2)\)) is a \( U \)-path on \( V_1 \setminus d' \) (resp., \( V_2 \setminus d \)).

This completes the proof of the theorem. \( \square \)
Bibliography


138


Bibliography


143


Appendix A

Generating the Sample of 280 scheduling problems

This appendix contains all relevant information to generate the 280 scheduling problems used in the computational experiments in Chapter 3.

A problem generator, coded in FORTRAN, is presented. The problem generator uses the portable random number generators in [9]. To randomly generate an instance of the scheduling problem, one needs to specify five input parameters: the number $n$ of jobs, the maximum processing time ($I P M A X = 100$ in this study), the maximum job weight ($I W M A X = 10$ in this study), the $P$-value, i.e., the probability of including each arc $(i, j)$ with $i < j$ (before the transitive closure is taken), and the seed number for the random generator. The output consists of three parts: (1) the number of jobs, and the number of arcs in the transitive reduction of the precedence graph; (2) the integer job processing times and weights; and (3) the arcs $e$, represented by their head nodes ($T O(e)$) and tail nodes ($F R O M(e)$), in the transitive reduction.

Tables A.1 and A.2 provide the seed numbers used for generating the 280 problems. These numbers are produced as follows. For each $n$, one arbitrarily chooses a seed number for the first problem in the group of 20 problems. Then, the random number produced immediately after the completion of the previous instance is used as the seed number for the next problem. This process continues till all instances in the group are produced. Thus, the 280 instances can be easily generated in 14 runs of the problem generator, using a DO loop with relevant parameters in it (note that the $P$ values vary within each group).
C PROGRAM FOR GENERATING THE SINGLE-MACHINE PRECEDENCE CONSTRAINED
C JOB SEQUENCING PROBLEMS. (USING MICROSOFT FORTRAN 5.0)
C PRECEDENCE MATRIX, PROCESSING TIMES AND JOB WEIGHTS.
COMMON NP(200,200),P(200),W(200)
INTEGER FROM(3000),TO(3000)
C INPUT
C VARIABLE FORMAT DEFINITION
-----------------------------------------------
C N    I5     NO. OF JOBS
C IPMAX I5     MAXIMUM PROCESSING TIME
C IWMAX I5     MAXIMUM JOB WEIGHT
C PROB F5.3     P VALUE FOR THE PRECEDENCE GRAPH
C NSEED I10    SEED FOR THE RANDOM NUMBER GENERATOR
C
C THE FOLLOWING 5 STATEMENTS MAY BE CHANGED IF
C A DIFFERENT FORTRAN COMPILER IS USED.
WRITE(*,*) 'Input data file name?'
OPEN(UNIT=23,FILE=' ')
OPEN(UNIT=24,FILE='dg.log')
WRITE(*,*) 'Output data file name?'
OPEN(UNIT=25,FILE=' ')
C
READ(23,100) N,IPMAX,IWMAX,PROB,NSEED
100 FORMAT(315,F5.3,I10)
NPOS=0
NAME=NAME+1
DO 33 I=1,N
DO 33 J=1,N
33 NP(I,J)=0
DO 10 I=1,N
 P(I) = IVUNIF(NSEED,IPMAX)*1.0
 W(I) = IVUNIF(NSEED,IWMAX)*1.0
 IA=I+1
 DO 20 J=IA,N
  PP=UNIF(NSEED)
  IF(PP.GE.PROB) GOTO 20
  NP(I,J)=1
  NP(J,I)=1
  NPOS=NPOS+1
20 CONTINUE
10 CONTINUE
Appendix A. Generating the Sample of 280 scheduling problems

C FIND THE TRANSITIVE CLOSURE AND REDUCTION OF THE PRECEDENCE DIGRAPH.
CALL TRANSCR(N,NPOSA)

C

C COMPUTE THE ORDER STRENGTH R
NPOSTO=NPOS+NPOSA
R=(NPOSTO*1.0)/(N*(N-1)*1.0/2.0)
WRITE(24,101) N,NC,IPMAX,IWMAX,PROB,R,NSEED
101 FORMAT(4I5,2X,F5.3,2X,F7.6,2X,F7.6,2X,I10)
N1=N-1
NC=0
DO 61 J=1,N1
   JA=J+1
   DO 62 I=JA,N
      IF(NP(I,J).LT.1) GOTO 62
      NC=NC+1
      FROM(NC)=J
      TO(NC)=I
62 CONTINUE
61 CONTINUE

C NC IS THE NUMBER OF ARCS IN THE TRANSITIVE CLOSURE
WRITE(25,63) N,NC
63 FORMAT(3I5)

DO 140 I=1,N
   WRITE(25,150) P(I),W(I)
150 FORMAT(2F10.2)
140 CONTINUE

DO 143 I=1,NC
   WRITE(25,153) FROM(I),TO(I)
153 FORMAT(2I5)
143 CONTINUE

STOP
END
SUBROUTINE TRANSCL(NOP,NPSA)
COMMON NP(200,200),P(200),W(200)

C
C FIND THE TRANSITIVE CLOSURE AND REDUCTION
C OF THE GIVEN DIGRAPH REPRESENTED BY NP(I,J).
C
C INPUT
C ----- 
C THE PRECEDENCE GRAPH NP
C
C OUTPUT
C -------
C THE UPPER TRIANGULAR OF NP: THE TRANSITIVE CLOSURE
C THE LOWER TRIANGULAR OF NP: THE TRANSITIVE REDUCTION
C NPSA: NO. OF ADDITIONAL ARCS USED IN CONSTRUCTING THE
C TRANSITIVE CLOSURE
C
NPSA=0

C NT=NOP-2
DO 10 III=1,NT
  I = NT-III+1
  I1=I+1
  I2=I+2
DO 20 J=I2,NOP
  J0=J-1
  DO 30 K=I1,J0
    IF (NP(I,K).EQ.0 .OR. NP(K,J).EQ. 0) GOTO 30
    IF (NP(I,J).NE.1) NPSA=NPSA+1
    NP(I,J)=1
    NP(J,I)=0
  30 CONTINUE
20 CONTINUE
10 CONTINUE
RETURN
END
FUNCTION UNIF(IX)
C
C PORTABLE RANDOM NUMBER GENERATOR IMPLEMENTING THE RECURSION:
C IX = 16807 * IX MOD (2**31 - 1)
C USING ONLY 32 BITS, INCLUDING SIGN.
C
CC REFERENCE: "A GUIDE TO SIMULATION", P. BRATLEY,
C B.L. FOX AND L.E. SCHRAGE, 1983,
C Springer-Verlag New York Inc.
C
C SOME COMPILERS REQUIRE THE DECLARATION:
C INTEGER*4 IX, K1
C
C INPUT:
C IX = INTEGER GREATER THAN 0 AND LESS THAN 2147483647
C
C OUTPUTS:
C IX = NEW PSEUDORANDOM VALUE,
C UNIF = A UNIFORM FRACTION BETWEEN 0 AND 1
C
K1 = IX/127773
IX = 16807*(IX - K1*127773) - K1 * 2836
IF ( IX .LT. 0 ) IX = IX + 2147483647
UNIF = IX*4.656612875E-10
RETURN
END
FUNCTION IVUNIF(IX,N)
C
C PORTABLE CODE TO GENERATE AN INTEGER UNIFORM OVER
C THE INTERVAL 1,N USING INVERSION
C
CC REFERENCE: "A GUIDE TO SIMULATION", P. BRATLEY,
CC B.L. FOX AND L.E. SCHRAGE, 1983,
CC Springer-Verlag New York Inc.
C
C INPUTS:
C IX = RANDOM NUMBER SEED
C N = THE UPPER BOUND OF THE UNIFORM INTEGER
C M = LARGEST VALUE FOR IX + 1
C
C AUXILIARY ROUTINE:
C UNIF
C
C OUTPUTS:
C IX = NEW PSEUDORANDOM VALUE,
C IVUNIF = A UNIFORM INTEGER BETWEEN 1 AND N
C
C THE RANDOM NUMBER GENERATOR IS ASSUMED TO USE THIS MODULUS
DATA M/2147483647/
C
JUNK = UNIF(IX)
C
WE WANT INTEGER PART OF IX/(M/N) USING INFINITE PRECISION.
C FORTRAN TRUNCATION UNDERSTATES M/N, THUS, FORTRAN MAY
C OVERSTATES WHEN IT COMPUTES IX/(M/N). SO DECREASE JTRY
C UNTIL JTRY <= IX/(M/N), I.E. M/IX <= N/JTRY.
C
JTRY = IX/(M/N)
IF(JTRY .LE. 0) GOTO 500
GOTO 200
100 JTRY=JTRY-1
Appendix A. Generating the Sample of 280 scheduling problems

C NOW CHECK IF M/IX <= N/JTRY
C
200 I = M
   J = IX
   K = N
   L = JTRY
C
C MORE GENERALLY, IS I/J <= K/L, WHERE I>=J, K>=L
C IF THE INTEGER PARTS DIFFER THE QUESTION IS ANSWERED.
C
300 IJI = I/J
   KLI = K/L
   IF( IJI .LT. KLI ) GOTO 500
   IF( IJI .GT. KLI ) GOTO 100
C
C MUST LOOK AT THE REMAINDERS; IS IJR/J <= KLR/L
C
   IJR = I - IJI * J
   IF ( IJR .LE. 0 ) GOTO 500
   KLR = K - KLI * L
   IF ( KLR .LE. 0 ) GOTO 100
C
C STILL NOT RESOLVED; REPHRASE IN STANDARD FORM,
C I.E. IS L/KLR <= J/IJR?
C
   I = L
   K = J
   J = KLR
   L = IJR
   GOTO 300
C
C NOW ADD 1 TO PUT IN THE RANGE 1,N
C
500 IVUNIF = JTRY + 1
   RETURN
END
### Appendix A. Generating the Sample of 280 Scheduling Problems

<table>
<thead>
<tr>
<th>n=30</th>
<th>n=40</th>
<th>n=50</th>
<th>n=60</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.01</td>
<td>8913445</td>
<td>47920113</td>
</tr>
<tr>
<td>2</td>
<td>0.01</td>
<td>755493537</td>
<td>1617675166</td>
</tr>
<tr>
<td>3</td>
<td>0.02</td>
<td>1809917341</td>
<td>1175781217</td>
</tr>
<tr>
<td>4</td>
<td>0.02</td>
<td>1725092278</td>
<td>1058321011</td>
</tr>
<tr>
<td>5</td>
<td>0.04</td>
<td>1822638204</td>
<td>1046809820</td>
</tr>
<tr>
<td>6</td>
<td>0.04</td>
<td>1180075628</td>
<td>1112508140</td>
</tr>
<tr>
<td>7</td>
<td>0.06</td>
<td>150617066</td>
<td>1562075571</td>
</tr>
<tr>
<td>8</td>
<td>0.06</td>
<td>1548235839</td>
<td>1175781217</td>
</tr>
<tr>
<td>9</td>
<td>0.08</td>
<td>1157536406</td>
<td>234794403</td>
</tr>
<tr>
<td>10</td>
<td>.100</td>
<td>94545423</td>
<td>1482029273</td>
</tr>
<tr>
<td>11</td>
<td>0.100</td>
<td>1103566408</td>
<td>1192173437</td>
</tr>
<tr>
<td>12</td>
<td>0.150</td>
<td>206050178</td>
<td>2115447001</td>
</tr>
<tr>
<td>13</td>
<td>0.150</td>
<td>1221144650</td>
<td>697947435</td>
</tr>
<tr>
<td>14</td>
<td>0.200</td>
<td>581095035</td>
<td>439147435</td>
</tr>
<tr>
<td>15</td>
<td>0.200</td>
<td>417497068</td>
<td>234303362</td>
</tr>
<tr>
<td>16</td>
<td>0.260</td>
<td>758772556</td>
<td>1679599250</td>
</tr>
<tr>
<td>17</td>
<td>0.300</td>
<td>2069131884</td>
<td>1762170024</td>
</tr>
<tr>
<td>18</td>
<td>0.500</td>
<td>1424549194</td>
<td>1310705738</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>n=70</th>
<th>n=80</th>
<th>n=90</th>
<th>n=100</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.01</td>
<td>76815623</td>
<td>4375178</td>
</tr>
<tr>
<td>2</td>
<td>0.01</td>
<td>1201897528</td>
<td>2111547401</td>
</tr>
<tr>
<td>3</td>
<td>0.02</td>
<td>1630262163</td>
<td>697923001</td>
</tr>
<tr>
<td>4</td>
<td>0.02</td>
<td>1221144650</td>
<td>697947435</td>
</tr>
<tr>
<td>5</td>
<td>0.04</td>
<td>18421799</td>
<td>439147435</td>
</tr>
<tr>
<td>6</td>
<td>0.04</td>
<td>417497068</td>
<td>234303362</td>
</tr>
<tr>
<td>7</td>
<td>0.06</td>
<td>758772556</td>
<td>1679599250</td>
</tr>
<tr>
<td>8</td>
<td>0.06</td>
<td>107891396</td>
<td>1669397446</td>
</tr>
<tr>
<td>9</td>
<td>0.08</td>
<td>829080974</td>
<td>1558045213</td>
</tr>
<tr>
<td>10</td>
<td>0.08</td>
<td>1807075147</td>
<td>1105086510</td>
</tr>
<tr>
<td>11</td>
<td>0.10</td>
<td>345975380</td>
<td>985796663</td>
</tr>
<tr>
<td>12</td>
<td>0.10</td>
<td>999548725</td>
<td>1837490202</td>
</tr>
<tr>
<td>13</td>
<td>0.15</td>
<td>698081069</td>
<td>1344833772</td>
</tr>
<tr>
<td>14</td>
<td>0.15</td>
<td>599774400</td>
<td>2128144389</td>
</tr>
<tr>
<td>15</td>
<td>0.20</td>
<td>776266564</td>
<td>967140344</td>
</tr>
<tr>
<td>16</td>
<td>0.20</td>
<td>2084721901</td>
<td>514443436</td>
</tr>
<tr>
<td>17</td>
<td>0.30</td>
<td>984818336</td>
<td>2013815749</td>
</tr>
<tr>
<td>18</td>
<td>0.30</td>
<td>1482079848</td>
<td>1460738242</td>
</tr>
<tr>
<td>19</td>
<td>0.500</td>
<td>1692157679</td>
<td>1748668138</td>
</tr>
<tr>
<td>20</td>
<td>0.500</td>
<td>338192698</td>
<td>1739693055</td>
</tr>
</tbody>
</table>

Table A.1: Seeds for the instances with $n = 30, 40, 50, 60, 70, 80, 90, 100$
### Appendix A. Generating the Sample of 280 scheduling problems

Table A.2: Seeds for the instances with \( n = 110, 120, 130, 140, 150, 160 \)

<table>
<thead>
<tr>
<th>n</th>
<th>n=110</th>
<th>n=120</th>
<th>n=130</th>
<th>n=140</th>
<th>n=150</th>
<th>n=160</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.001</td>
<td>578</td>
<td>327819</td>
<td>7456673</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>.001</td>
<td>872858166</td>
<td>1896871566</td>
<td>2090383671</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>.020</td>
<td>1966911261</td>
<td>1879863869</td>
<td>613433542</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>.020</td>
<td>2035263084</td>
<td>1751592867</td>
<td>2002449486</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>.040</td>
<td>1077304999</td>
<td>927231622</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>.040</td>
<td>1367175068</td>
<td>1907798662</td>
<td>818258585</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>.060</td>
<td>1810714895</td>
<td>1896871566</td>
<td>1970034241</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>.060</td>
<td>1350354151</td>
<td>1351713937</td>
<td>1274437629</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>.080</td>
<td>1145775358</td>
<td>1788610971</td>
<td>1473953588</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>.080</td>
<td>1992218419</td>
<td>1633349205</td>
<td>2013606980</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>.100</td>
<td>120830678</td>
<td>1207063493</td>
<td>731735072</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>.100</td>
<td>115769795</td>
<td>302181295</td>
<td>1371935740</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>.150</td>
<td>392329467</td>
<td>278166407</td>
<td>830555894</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>.150</td>
<td>1248974721</td>
<td>1730677699</td>
<td>1335044189</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>.200</td>
<td>1161785585</td>
<td>922885587</td>
<td>259379126</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>.200</td>
<td>147857977</td>
<td>1942371325</td>
<td>1562102629</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>.300</td>
<td>559322792</td>
<td>615429360</td>
<td>1301809084</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>.300</td>
<td>1142323614</td>
<td>2101567237</td>
<td>1795967001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>.500</td>
<td>1497262303</td>
<td>1172431752</td>
<td>2131368145</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>.500</td>
<td>81858226</td>
<td>134821562</td>
<td>1849154095</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>