Linear programming formulation of the set partitioning problem

Moustapha Diaby

Operations and Information Management,
University of Connecticut,
Storrs, CT 06268, USA
E-mail: moustapha.diaby@business.uconn.edu

Abstract: In this paper, we present a linear programming (LP) model of the set partitioning problem (SPP). The number of variables and the number of constraints of the proposed model are bounded by (third-degree) polynomial functions of the number of non-zero entries of the SPP input matrix, respectively. Hence, the model provides a new affirmative resolution to the all-important ‘P vs. NP’ question. We use a transportation problem-based reformulation that we develop, and a path-based modelling approach similar to that used in Diaby (2007) to formulate the proposed LP model. The approach is illustrated with a numerical example.

Keywords: SPP; set partitioning problem; LP; linear programming; computational complexity; combinatorial optimisation.


Biographical notes: Moustapha Diaby is a Professor of Production and Operations Management at the University of Connecticut. He received a PhD in Management Science/Operations Research, MS in Industrial Engineering and BS in Chemical Engineering from the State University of New York at Buffalo. His teaching and research interests are in the areas of mathematical programming, manufacturing systems modelling and analysis, and supply chain and logistics management. His publications have appeared in top-tier journals such as European Journal of Operational Research, Int. J. Production Research, Journal of the Operational Research Society, Management Science, Operations Research, etc. He also serves has served as a reviewer and/or ad hoc editorial team member for many of these journals.

1 Introduction

One of the three most widely applied combinatorial optimisation problems (along with the travelling salesmen problem (see Samanlioglu et al., 2007) and the set covering problem (see Kinney et al., 2007)) is the set partitioning problem (SPP) (Balas and Padberg, 1976). The sheer volume of industrial applications of the problem that have been described in the literature precludes an exhaustive listing in a single journal paper. The richest source of applications in the recent literature – as well as the ‘older’ literature (see Balas and
Padberg, 1976) – has been the transportation industry, dealing with problems of dispatching vehicles (Desaulniers et al., 2003; Westphal and Krumke, 2008), determining vehicle fleet mixes (Lee et al., 2008), routing and scheduling of vehicles (Alvarenga et al., 2007; Baldacci et al., 2008; Freling et al., 2003; Hong et al., 2009; Ileri et al., 2006; Jepsen et al., 2008; Kliewer et al., 2006) and scheduling of transportation system crews (Medard and Sawhney, 2007; Mesquita and Paias, 2008) among many others. Recent applications outside the transportation industry include problems of cellular manufacturing system design (Mahdavi et al., 2006), communication system network design (Oliveira et al., 2005 (‘clique’ form); Tombus and Bilgic, 2004), computer hardware and software designs (Osei-Bryson and Joseph, 2006; Thomadsen and Larsen, 2007), design of balanced student teams (Desrosiers et al., 2005), design of maintenance schedules for production machines (Grigoriev et al., 2006), facility location (Berger et al., 2007), image compression (Jyotheswar and Mahapatra, 2007), molten iron allocation in the steel industry (Tang et al., 2007), supply chain design (Chiang and Russell, 2004; Sadler and Gervet, 2008; Sindhuchao et al., 2005; Teo and Shu, 2004), ‘surgical theater’ planning involving room and surgical team assignments in hospitals (Fei et al., 2008) and workforce scheduling (Beliën and Demeulemeester, 2007; Everborn and Ronqvist, 2004; Eveborn et al., 2006).

Because of the extremely broad range of applicability (as highlighted above), the SPP can be interpreted from a wide variety of perspectives. The perspective we adopt in this paper is that of a manufacturing/operations context that can be described as follows. There is a set \( \Gamma = \{1, \ldots, \gamma\} \) of processors, and a set \( \Theta = \{1, \ldots, \theta\} \) of tasks that must be performed using those processors. A fixed cost, \( c_p(p \in \Gamma) \), is incurred if processor \( p \) is used. The problem is to select a subset of the processors to be used so that each task can be performed on exactly one of the chosen processors, and the total cost of the chosen processors is minimised.

Let \( o \) denote the \( \gamma \times \theta \) input matrix for the SPP, with the \((p, t)\)th entry (denoted \( o_{pt} \)) being a 0/1 binary indicator that is equal to ‘1’ iff task \( t \in \Theta \) can be performed on processor \( p \in \Gamma \). Let \( u_p \) be a 0/1 binary variable that is equal to ‘1’ iff \( p \in \Gamma \) is used. Then, the classical integer programming (IP) formulation of the SPP is as follows:

\[
\text{Problem 1 (Problem SPP):} \\
\text{minimise:} \\
\xi (u) := \sum_{p \in \Gamma} c_p u_p \\
\text{s.t.} \\
\sum_{p \in \Gamma} o_{pt} u_p = 1; \quad t \in \Theta \\
u_p \in \{0, 1\}; \quad p \in \Gamma
\]

where

\[
o_{pt} = \begin{cases} 
1 & \text{if task } t \text{ can be performed on processor } p \\
0 & \text{otherwise}
\end{cases}
\]

\[
u_p = \begin{cases} 
1 & \text{if processor } p \text{ is used} \\
0 & \text{otherwise}
\end{cases}
\]

\[
c_p = \text{cost of using processor } p
\]
Problem SPP was one of the first problems to be classified as NP-Complete (Karp 1972). Hence, research directed at developing solution methods has focused on heuristics and efficient enumerative search procedures. Reviews can be found in Balas and Padberg (1976), Boschetti et al. (2008) and Joseph (2002). The linear programming (LP) relaxation tends to yield good (‘tight’) lower bounds. However, solving it can pose a challenge due to the high degree of degeneracy (Barahona and Anbil, 2002). Hence, heuristic procedures aimed at solving the LP relaxation have been developed (Barahona and Anbil, 2000, 2002; Boschetti et al., 2008; Cavalcante et al., 2008; Chan and Yano, 1992; Conforti et al., 2007; Fisher and Kedia, 1990; Klabjan, 2004; Lucena, 2005). Meta-heuristic approaches (Alvarenga et al., 2007; Lee et al., 2008) and heuristics based on reformulations (Ali and Han, 1998; Ali and Thiagarajan, 1989; El-Darzi and Mitra, 1992, 1995; Lewis et al., 2008; Sherali and Lee, 1996) have also been developed for the overall problem. The exact procedures that have been proposed have been, for the most part, enumerative search procedures (Balas and Padberg, 1976; Boschetti et al., 2008; Chan and Yano, 1992; Fisher and Kedia, 1990; Harche and Thompson, 1994; Hoffman and Padberg, 1993; Joseph, 2002; Linderoth et al., 2001; Marsten, 1974; Marsten and Shepardson, 1981). Cutting plane methods have also been developed, which are reviewed in Balas and Padberg (1976).

Tremendous successes at solving very-large, industrial-scale problems have been achieved using some of the procedures above. However, these fall short with respect to the fundamental issue of the tractability of the problem (see Garey and Johnson, 1979). This paper represents a significant addition to the existing literature in that sense. It presents a first LP model of the SPP. The number of variables and the number of constraints of the proposed model are bounded by (third-degree) polynomial functions of the number of non-zero entries of the SPP input matrix, respectively. Hence, in particular, beyond the scope of the SPP per se, the model provides a new affirmative resolution to the all-important ‘P vs. NP’ question. We use a transportation problem (TP)-based reformulation that we develop, and a path-based modelling approach similar to that used in Diaby (2007) to formulate the proposed LP model. The approach is illustrated with a numerical example.

The plan of the paper is as follows. We discuss the TP-based reformulation in Section 2. The path-based reformulation of the constraints set of the TP-based model is discussed in Section 3. The overall LP model is discussed in Section 4. Conclusions are discussed in Section 5.

Notation 1: The following notation will be used throughout the rest of this paper:

1. The set of real numbers is denoted by \( \mathbb{R} \).

2. For two column vectors \( \mathbf{a} \) and \( \mathbf{b} \), \( \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix} = (\mathbf{a}^T, \mathbf{b}^T)^T \) will be written as \( (\mathbf{a}, \mathbf{b})' \) (where \( (\cdot)^T \) denotes the transpose of \( (\cdot) \)), except for where that causes ambiguity.

3. The \( i \)th component of a column vector \( \mathbf{a} \) is denoted \( a_i \).

4. The notation ‘\( 0 \)’ denotes a column vector of comfortable size that has every entry equal to 0.

5. The notation ‘\( 1 \)’ denotes a column vector of comfortable size that has every entry equal to 1.

6. The convex hull of \( (\cdot) \) is denoted \( \text{Conv}(\cdot) \).

7. The set of extreme points of \( (\cdot) \) is denoted \( \text{Ext}(\cdot) \).
The notation \( \forall \langle i_1 \in A_1 : B_1 ; \ldots ; i_p \in A_p : B_p \rangle , \langle C_1 ; \ldots ; C_q \rangle \) stands for \( \forall i_1 \in A_1 : B_1 , \ldots , \forall i_p \in A_p : B_p \), each statement \( C_j (j = 1, \ldots , q) \) holds true. Where that does not cause ambiguity, the brackets (one or both sets) will be omitted.

The symbol ‘϶’ stands for ‘such that’.

The notation \( \exists \langle i_1 \in A_1 ; \ldots ; i_p \in A_p \rangle , \langle B_1 ; \ldots ; B_q \rangle \) stands for ‘There exists at least one object from each \( A_r (r = 1, \ldots , p) \), such that each expression \( B_s (s = 1, \ldots , q) \) holds true.’ Where that does not cause ambiguity, the brackets (one or both sets) will be omitted.

2 Transportation problem-based reformulation

Unlike previous network flow-based models that have been proposed (see Ali and Han, 1998; Ali and Thiagarajan, 1989; El-Darzi and Mitra, 1992, 1995, for example), our TP-based reformulation of Problem SPP does not depend on the presence of any special structures in the SPP input matrix. Hence, our TP-based reformulation is applicable to any instance of the SPP, and does not require any searches of the SPP input matrix for structures. In this section, we present the model and illustrate it with a numerical example.

Assumptions 1: We assume without loss of generality (w.l.o.g.) that a ‘dummy’ task, indexed as \( \theta + 1 \), has been added to the set of tasks, and that the SPP input matrix has been expanded accordingly, with \( o_{p, \theta + 1} = 1 \) for all \( p \in \Gamma \).

Notation 2 (SPP parameters):

1. \( \gamma \) : number of processors
2. \( \theta \) : number of tasks
3. \( \Gamma := \{1, \ldots , \gamma \} \) (set of processors)
4. \( \Theta := \{1, \ldots , \theta \} \) (set of tasks)
5. \( \forall p \in \Gamma , T_p := \{ t \in \Theta : o_{pt} = 1 \} \) (set of tasks that can be performed using processor \( p \))
6. \( \forall p \in \Gamma , \tau_p := |T_p| \)
7. \( \forall p \in \Gamma , K_p := \{1, \ldots , \tau_p \} \)
8. \( n := \sum_{p \in \Gamma} \tau_p \) (number of non-zero entries of the SPP input matrix)
9. \( \forall t \in \Theta , P_t := \{ p \in \Gamma : o_{pt} = 1 \} \) (set of processors that can perform task \( t \))
10. \( \forall t \in \Theta , \pi_t := |P_t| \)
11. \( \forall (p \in \Gamma ; t \in (\Theta \cup \{\theta + 1\})) , v_{pt} \) denotes a non-negative variable that is positive iff task \( t \) is performed using processor \( p \).

Assumptions 2: We assume w.l.o.g. that:

1. The members of \( T_p (p \in \Gamma) \) have been arranged in increasing order of task indices, with \( \alpha_{pk} (k \in K_p) \) as the index of the \( k \)th member; that is, the ordering of \( T_p \) is such that \( \alpha_{pk} < \alpha_{p\ell} \forall k , \ell \in K_p^2 : k < \ell \).
2. There is no restriction on the number of times the ‘dummy’ task, \( \theta + 1 \), can be included in any SPP solution.
3. The feasible set of Problem SPP is non-empty.
Definition 1: For \( p \in \Gamma \), we refer to the members of \( K_p \) as the ‘(processing) slots’ on \( p \).

Our proposed TP-based reformulation is as follows:

Problem 2 (Problem TP):

\[
\text{minimise:} \quad \zeta(v) := \sum_{p \in \Gamma} c_p v_{p,\alpha_{p,1}} \tag{4}
\]

s.t.

\[
\sum_{p \in \Gamma} v_{pt} = 1; \quad t \in \Theta \tag{5}
\]

\[
\sum_{p \in \Gamma} v_{p,\theta+1} = n - \theta \tag{6}
\]

\[
\sum_{t \in T_p \cup \{\theta+1\}} v_{pt} = \tau_p; \quad p \in \Gamma \tag{7}
\]

\[
v_{p,\alpha_{p,k}} - v_{p,\alpha_{p,1}} = 0 \quad p \in \Gamma; \quad k \in K_p \setminus \{1\} \tag{8}
\]

\[
v_{pt} \in \{0, 1\}; \quad p \in \Gamma, \quad t \in T_p; \quad v_{p,\theta+1} \geq 0, \quad p \in \Gamma \tag{9}
\]

Constraints (5) ensure (in light of constraints (9)) that each task can be performed on exactly one of the chosen processors. Constraints (7) stipulate that every slot on a given processor must be assigned. Constraints (8) ensure that the slots are all assigned either to actual tasks (i.e. the processor is used) or to the dummy task (i.e. the processor is not used). Constraints (6) account the total number of processing slots not used. Hence, the objective function (4) correctly accounts the total cost of the processors used.

Definition 2: Let \( W := \{v \in \mathbb{R}^{n+\gamma}: v \text{ satisfies (5)--(9)}\} \). We refer to \( \text{Conv}(W) \) as the ‘TP-based Polytope’.

Theorem 1: The following statements hold true:

(i) There is a one-to-one correspondence between feasible solutions of Problem TP and feasible SPP solutions.

(ii) There is a one-to-one correspondence between feasible solutions of Problem TP and feasible solutions of Problem SPP.

(iii) Problem TP and Problem SPP are equivalent optimisation problems.

A numerical illustration of the TP-based reformulation is shown in Figure 1.

3 Reformulation of the TP-based model constraints set

3.1 Multipartite graph representation

We reformulate constraints (5)--(9) of Problem TP in terms of flows over the multipartite digraph, \( G \), illustrated in Figure 2. In this graph, each node corresponds to a feasible (task, processor, slot) triplet. That is, a triplet \((t, p, k) \in (\Theta, \Gamma, K_p)\) has a corresponding node in the graph iff \( t = \alpha_{p,k} \). On the other hand, there is a node in the graph corresponding to each triplet \((\theta + 1, p, k)(p \in \Gamma, k \in K_p)\). The arcs of the graph are specified through the explicit statements of the forward and backward stars of each of the nodes of the graph.
Figure 1 Illustration of the transportation problem-based reformulation; (a) problem input data and notation and (b) illustration of the transportation tableau form

Number of processors, $\gamma = 8$;

Set of processors, $\Gamma = \{1, 2, 3, 4, 5, 6, 7, 8\}$;

Number of Tasks, $\theta = 6$;

Set of tasks, $\Theta = \{1, 2, 3, 4, 5, 6\}$:

<table>
<thead>
<tr>
<th>Processors</th>
<th>Tasks</th>
<th>Ordering</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p$</td>
<td>$T_p$</td>
<td>$\tau_p$</td>
</tr>
<tr>
<td>1</td>
<td>[1, 3, 5]</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>[2, 6]</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>{2}</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>{4}</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>[1, 2, 6]</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>{1, 3}</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>{1, 4, 5}</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>[2, 5]</td>
<td>2</td>
</tr>
</tbody>
</table>

(a)

(b)

Definition 3:
1. We refer to the set of nodes of Graph G that correspond to a given (processor, slot) pair as a stage of the graph.
2. We refer to the set of nodes of Graph G that correspond to a given task as a level of the graph.
To simplify the exposition, we perform a sequential indexing of the stages, as described below.

Notation 3 (Graph representation formalisms):

1. \( \Omega := \Theta \cup \{\theta + 1\} \).
2. \( S := \{1, \ldots, n\} \) (set of stages of Graph G).
3. \( R := S\setminus\{n\} \).
4. \( \forall p \in \Gamma, f_p := \begin{cases} 1; & \text{for } p = 1 \\ \sum_{j=1}^{p-1} \tau_j + 1; & \text{for } p > 1 \end{cases} \) (index of the stage corresponding to the (processor, slot) pair, \((p, 1)\)).
5. \( \forall p \in \Gamma, \ell_p := \sum_{j=1}^{p} \tau_j \) (index of the stage corresponding to the (processor, slot) pair, \((p, \tau_p)\)).
6. \( \forall r \in S, \xi_r := \arg \max\{f_p : f_p \leq r\} \) (processor index for stage \(r\)).
7. \( \forall r \in S, M_r := [\alpha_G, r - f_{\xi_r} + 1] \) (set of tasks in \(\Theta\) that define nodes at stage \(r\)).
8. \( \forall r \in S, N_r := M_r \cup \{\theta + 1\} = [\alpha_G, r - f_{\xi_r} + 1, \theta + 1] \).
9. \( V := \{(i, r) : r \in S, i \in N_r\} \) (set of nodes of Graph G).
\[ \forall \{ r \in S; t \in (T_\xi \cup \{ \theta + 1 \}) \}, \]
\[
F_r(t) := \begin{cases} 
N_r \setminus \{ t \} & \text{for } r < n; \ r = \ell_\xi; \ t \neq \theta + 1 \\
N_r + 1 & \text{for } r < n; \ r = \ell_\xi; \ t = \theta + 1 \\
M_{r+1} = \{ \alpha_{r,i} - f_{r+2} \} & \text{for } r < n; \ f_{r+2} \leq r < \ell_\xi; \ t \neq \theta + 1 \\
\{ \theta + 1 \} & \text{for } r < n; \ f_{r+2} \leq r < \ell_\xi; \ t = \theta + 1 \\
\emptyset & \text{for } r = n 
\end{cases}
\]
(forward star of node \((t, r)\) of Graph \(G\)).

\[ \forall \{ r \in S; t \in N_r \}, \]
\[
B_r(t) := \begin{cases} 
\{ j \in N_{r-1} : t \in F_{r-1}(j) \} & \text{for } r > 1 \\
\emptyset & \text{for } r = 1 
\end{cases}
\]
(backward star of node \((t, r)\) of Graph \(G\)).

\[ A := \{ (i, r, j) \in (\Omega, R, \Omega) : i \in N_r; \ j \in F_r(i) \} \text{ (set of arcs of Graph } G)\).

The notation and structure of the graph representation are illustrated in Figure 3 for the numerical example shown in Figure 1.

Remark 1:
1 Each stage of Graph \(G\) comprises exactly two nodes of the graph.
2 The maximum number of arcs originating from any stage of Graph \(G\) is four.

Definition 4:
1 We refer to a path of Graph \(G\) that spans the set of stages of the graph as a through-path of the graph.
2 We refer to a through-path of Graph \(G\) that includes each task in \(\Theta\) exactly once as a SPP path of the graph; that is, a set of arcs,
\[
\{(i_1, 1, i_2), (i_2, 2, i_3), \ldots, (i_{n-1}, n-1, i_n)\} \in A^{n-1},
\]
is a SPP path iff \((\forall t \in \Theta, \exists p \in S \ni i_p = t)\), and \((\forall (p, q) \in (S, S \setminus \{ p \}) : (i_p, i_q) \in \Theta^2, i_p \neq i_q)\).

Remark 2: It follows directly from definitions that:
1 There exists a one-to-one correspondence between SPP paths of Graph \(G\) and feasible SPP solutions.
2 There exists a one-to-one correspondence between SPP paths of Graph \(G\) and feasible solutions to Problem SPP.
3 There exists a one-to-one correspondence between SPP paths of Graph \(G\) and feasible solutions to Problem TP.

SPP paths are illustrated in Figure 4 for the numerical example shown in Figure 1. The through-path shown in Figure 4(a) is a SPP path, and corresponds to the SPP solution in which the processors used are Processors 1, 2 and 4. The partial (i.e. non-spanning
with respect to the set of stages) path shown in Figure 4(b) corresponds to SPP solutions in which both Processors 4 and 5 are used. It is easy to verify that there exists no SPP path in the graph that comprises this partial path, which is consistent with the fact that there exists no feasible SPP solution in which both Processors 4 and 5 are used.

Theorem 2: *A given SPP path of Graph G cannot be represented as a convex combination of other SPP paths of Graph G.*

Notation 4: *We denote the set of all SPP paths of Graph G as \( \Delta \); that is,\[
\Delta = \{ ((i_1, 1, i_2), (i_2, 2, i_3), \ldots, (i_{n-1}, n-1, i_n)) \in A^{n-1} : \\
(\forall t \in \Theta, \exists p \in S : i_p = t) ; \\
(\forall (p, q) \in (S, S \setminus \{p\}) : (i_p, i_q) \in \Theta^2, i_p \neq i_q) \}.\]

**Figure 3** Numerical illustration of Graph G; (a) Graph notations and (b) Network flow graph (see online version for colours)

\[
n = \sum_{p \in \Gamma} \tau_p = 17; \\
S = \{1, \ldots, 17\}; \\
R = \{1, \ldots, 16\};
\]

<table>
<thead>
<tr>
<th>Processors</th>
<th>Stages</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p )</td>
<td>( r )</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

\[
r = 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16 \\ 17 \\
\]
3.2 Integer programming reformulation

Assumptions 3: We assume w.l.o.g. that the number of stages of Graph G is greater than 5 (i.e., \( n \geq 6 \)).

Notation 5 (Flow variables):

1. \( \forall ((p, r, s) \in R^3 : r < s < p; (i, j, k, t, u, v) \in (N_r, F_r(i), N_s, F_s(k), N_p, F_p(u))) \),
   \( z_{(irj)(kst)(upv)} \) denotes a non-negative variable that represents the amount of flow in Graph G that propagates from arc \((i, r, j)\) onto arc \((u, p, v)\), via arc \((k, s, t)\).

2. \( \forall ((r, s) \in R^2 : r < s; (i, j, k, t) \in (N_r, F_r(i), N_s, F_s(k))) \), \( y_{(irj)(kst)} \) denotes a non-negative variable that represents the total amount of flow in Graph G that propagates from arc \((i, r, j)\) onto arc \((k, s, t)\).

Our reformulation of \( W \) is as follows:

\[
\sum_{t \in G} \sum_{j \in F_2(i)} \sum_{t \in F_2(j)} \sum_{v \in F_3(t)} z_{(i,1,j)(j,2,t)(t,3,v)} = 1
\]  (10)
**LP formulation of the SPP**

\[ \sum_{v \in B_p(u)} z_{(irj)(kst)(v,p-1,u)} - \sum_{v \in F_p(u)} z_{(irj)(kst)(upv)} = 0 \]

\( p, r, s \in R : r < s < p - 1; \ i \in \Omega; \ j \in F_r(i); \ k \in \Omega; \ t \in F_s(k); \ u \in \Omega \) \hspace{1cm} (11)

\[ \sum_{v \in B_p(u)} z_{(irj)(kst)(v,p-1,u)} - \sum_{v \in F_p(u)} z_{(irj)(kst)(upv)} = 0 \]

\( p, r, s \in R : r + 1 < p < s; \ i \in \Omega; \ j \in F_r(i); \ k \in \Omega; \ t \in F_s(k); \ u \in \Omega \) \hspace{1cm} (12)

\[ \sum_{v \in B_p(u)} z_{(irj)(kst)(v,p-1,u)} - \sum_{v \in F_p(u)} z_{(irj)(kst)(upv)} = 0 \]

\( p, r, s \in R : 1 < p < r < s; \ i \in \Omega; \ j \in F_r(i); \ k \in \Omega; \ t \in F_s(k); \ u \in \Omega \) \hspace{1cm} (13)

\[ \sum_{v \in B_p(u)} z_{(irj)(kst)(v,p-1,u)} - \sum_{v \in F_p(u)} z_{(irj)(kst)(upv)} = 0 \]

\( p, r, s \in R : r < s < p; \ i \in \Omega; \ j \in F_r(i); \ k \in \Omega; \ t \in F_s(k); \ u \in \Omega \) \hspace{1cm} (14)

\[ \sum_{v \in B_p(u)} z_{(irj)(kst)(v,p-1,u)} - \sum_{v \in F_p(u)} z_{(irj)(kst)(upv)} = 0 \]

\( p, r, s \in R : r < s < p; \ i \in \Omega; \ j \in F_r(i); \ k \in \Omega; \ t \in F_s(k); \ u \in \Omega \) \hspace{1cm} (15)

\[ \sum_{v \in B_p(u)} z_{(irj)(kst)(v,p-1,u)} - \sum_{v \in F_p(u)} z_{(irj)(kst)(upv)} = 0 \]

\( p, r, s \in R : r < s < p; \ i \in \Omega; \ j \in F_r(i); \ k \in \Omega; \ t \in F_s(k); \ u \in \Omega \) \hspace{1cm} (16)

\[ \sum_{k \in \Omega \setminus \{i,j,k,t\}} \sum_{j \in Fr(i)} \sum_{k \in Bs+1(i)} y_{(irj)(ksi)} = 0; \ r \in R \setminus \{n-1\}; \ i \in \Omega; \ j \in F_r(i) \hspace{1cm} (17) \]

\[ \sum_{(r,s) \in R^2 : s > r} \sum_{j \in Fr(i)} \sum_{k \in Bs+1(i)} y_{(irj)(kst)} + \sum_{(r,s) \in R^2 : s > r} \sum_{i \in Fr(i)} \sum_{k \in Fr(i)} y_{(irj)(irs)} + \sum_{(r,s) \in R^2 : s > r} \sum_{j \in Bs+1(i)} \sum_{k \in Bs+1(i)} y_{(irj)(iks)} = 0 \hspace{1cm} (18) \]
The propagation of one unit of flow from stage 1 of Graph $G$ is initiated by constraint (10). Constraints (11)–(13) enforce, in light of the structure of Graph $G$, constraints (7) of Problem TP. They stipulate that all flows initiated at stage 1 propagate onward, to stage $n$ of the graph, in a connected and balanced manner. Specifically, constraints (11) stipulate that the total amount of flow from arc $(i, r, j)$ that propagates through arc $(k, s, t)$ and subsequently enters node $(u, p)$ is equal to the amount of flow from the arc $(i, r, j)$ that propagates through arc $(k, s, t)$ and leaves the node. Constraints (12) stipulate that the total amount of flow from arc $(i, r, j)$ that enters node $(u, p)$ to propagate on to arc $(k, s, t)$ is equal to the total amount of flow from arc $(i, r, j)$ that leaves node $(u, p)$ to propagate on to arc $(k, s, t)$. Constraints (13) stipulate that the total amount of flow from arc $(i, r, j)$ that propagates through arc $(k, s, t)$ after having entered node $(u, p)$ is equal to the amount of flow from arc $(i, r, j)$ that propagates through arc $(k, s, t)$ and leaves the node. Constraints (14)–(16) ensure that the flow propagation between any pair of arcs of Graph $G$ is consistently accounted across all the stages of the graph. Constraints (17) require that the total flow on any given arc of Graph $G$ must propagate on to every level of the graph, or be part of a flow propagation that spans the levels of the graph. They enforce the conditions of constraints (5)–(6) of Problem TP. Constraints (18) ensure that the initial flow propagation from any given arc occurs in an ‘unbroken’ fashion. Finally, constraints (19) stipulate (in light of the other constraints) that no part of the flow from arc $(i, r, j)$ of Graph $G$ can propagate back onto level $i$ of the graph for $i \in \Theta$, or onto level $j$ for $j \in \Theta$. Note that the conditions of the “side’’ constraints (8) of Problem TP are implicit in the system (10)–(21), due to the structure of Graph $G$.

Theorem 3:

(i) The number of variables in the system (10)–(19) is $O(n^3)$.

(ii) The number of constraints in the system (10)–(19) is $O(n^3)$.

Definition 5:

1. Let $Q_I := \{(y, z) \in \mathbb{R}^m : (y, z) \text{ satisfies } (10)–(21)\}$, where $m$ is the number of variables in the system (10)–(21). We refer to $\text{Conv}(Q_I)$ as the IP Polytope.

2. We refer to the LP relaxation of $Q_I$ as the LP Polytope, and denote it by $Q_L$; i.e. $Q_L := \{(y, z) \in \mathbb{R}^m : (y, z) \text{ satisfies } (10)–(19), \text{ and } 0 \leq (y, z) \leq 1\}$, where $m$ is the number of variables in the system (10)–(19).

Theorem 4: $(y, z) \in Q_I \iff$ There exists exactly one $n$-tuple, $(i_r \in N_r, r = 1, \ldots, n)$, such that:

(i) $z_{(arb)(csd)(epf)} = \begin{cases} 1 & \text{for } p, r, s \in R : r < s < p; \ (a, b, c, d, e, f) \\ = (i_r, i_{r+1}, i_s, i_{s+1}, i_p, i_{p+1}) & \end{cases}$
\textit{LP formulation of the SPP} \hfill \textcolor{red}{411}

(ii) \(y_{(arb)\text{(csd)}} = \begin{cases} 
1 & \text{for } r, s \in R : r < s; \quad (a, b, c, d) = (i_r, i_{r+1}, i_s, i_{s+1}) \\
0 & \text{otherwise}
\end{cases}\)

(iii) \(\forall t \in \Theta, \exists p \in S : i_p = t\)

(iv) \(\forall (p, q) \in (S, S \setminus \{p\}) : (i_p, i_q) \in \Theta^2, \ i_p \neq i_q.\)

Theorem 5: The following statements hold true:

(i) there exists a one-to-one correspondence between the points of \(Q_I\) and SPP paths of Graph \(G\)

(ii) there exists a one-to-one correspondence between the points of \(Q_I\) and the points of \(W\)

(iii) there exists a one-to-one correspondence between the points of \(Q_I\) and feasible solutions to Problem SPP

(iv) there exists a one-to-one correspondence between the points of \(Q_I\) and feasible SPP solutions.

Definition 6: Let \((y, z) \in Q_I.\) Let \((i_r \in N, \ r = 1, \ldots, n)\) be the \(n\)-tuple satisfying

Theorem 4 for \((y, z)\):

1 we refer to the solution to Problem TP corresponding to \((y, z)\) as the 'assignment corresponding to \((y, z)\)' and denote it by \(M(y, z) := \{(\xi_r, i_r), r = 1, \ldots, n\}\)

2 we refer to \(W(y, z) := \{p \in \Gamma : (\forall k \in K, \ (p, \alpha_{pk}) \in M(y, z))\}\) as the 'SPP solution corresponding to \((y, z)\').

3.3 \textit{LP reformulation}

Our LP reformulation of the \(TP\)-based polytope consists of \(Q_L.\) We show that every point of \(Q_L\) is a convex combination of points of \(Q_I,\) thereby establishing (in light of Remark 2, and Theorems 2 and 5) the one-to-one correspondence between the extreme points of \(Q_L\) and the points of \(Q_I.\) In the following discussion, the first set of results (Lemma 1 — Theorem 10) pertain to flow propagation patterns that are associated with the points of \(Q_L.\) The second set of results (Lemma 3 — Corollary 1) establish convexity properties based on mass/flow conservations.

Lemma 1: Let \((y, z) \in Q_L.\) The following holds true:

\(\forall (r \in R : r \leq n - 3; \ (i_r, i_{r+1}, i_{r+2}, i_{r+3}) \in (\Omega, F_r(i_r), \Omega, F_{r+2}(i_{r+2})))\)

\(y_{(i_r, i_{r+1}, i_{r+2}, i_{r+3})} > 0 \iff \begin{cases} 
(i) \quad i_{r+2} \in F_{r+1}(i_{r+1}); \text{ and} \\
(ii) \quad z_{(i_r, i_{r+1}, i_{r+2}, i_{r+3})} > 0.
\end{cases}\)

Lemma 1 is generalised in Theorem 6. To simplify the presentation, in the remainder of this paper, we focus on the 'support graphs' of points of \(Q_L.\)

Notation 6 ('Support graph' of \((y, z)\)): For \((y, z) \in Q_L: \)

1 The sub-graph of Graph \(G\) induced by the positive components of \((y, z)\) is denoted as:

\(H(y, z) := (V(y, z), \overline{A}(y, z))\)
\[
V(y, z) := \begin{cases} 
(i, 1) \in V : \sum_{j \in F_1(i)} \sum_{t \in F_2(j)} y((i, 1), (j, 2, t)) > 0 \\
(i, r) \in V : 1 < r < n; \sum_{a \in N_1} \sum_{b \in F_1(a)} \sum_{j \in F_2(b)} y((a, 1, b), (j, n-1, i)) > 0 \\
(i, n) \in V : \sum_{a \in N_1} \sum_{b \in F_1(a)} \sum_{j \in B_1(a)} y((a, 1, b), (j, n-1, i)) > 0 
\end{cases}
\]

\[
\bar{A}(y, z) := \begin{cases} 
(i, j) \in A : \sum_{t \in F_2} y((i, 1, j), (j, 2, t)) > 0 \\
(i, r, j) \in A : r > 1; \sum_{a \in N_1} \sum_{b \in F_1(a)} y((a, 1, b), (i, r, j)) > 0 
\end{cases}
\]

The set of arcs of \(H(y, z)\) originating at stage \(r\) of \(H(y, z)\) is denoted \(A_r(y, z)\).

The number of arcs originating at stage \(r\) of Graph \(H(y, z)\) is denoted \(\eta_r(y, z) = |A_r(y, z)|\). For simplicity \(\eta_r(y, z)\) will be, henceforth, written as \(\eta_r\) (unless that causes ambiguity).

The index set associated with \(A_r(y, z)\) is denoted \(\Lambda_1(r) := \{1, 2, \ldots, \eta_r\}\). For simplicity \(\Lambda_1(r)\) will be, henceforth, written as \(\Lambda_1\).

The \(\nu\)th arc in \(A_r(y, z)\) is denoted as \(a_{r, \nu}(y, z)\). For simplicity \(a_{r, \nu}(y, z)\) will be, henceforth, written as \(a_{r, \nu}\).

Where that causes no confusion (and where that is convenient), for \((r, s) \in R^2 : r < s\) and \((\rho, \sigma) \in (\Lambda_r, \Lambda_s)\), \(z((i, r, \rho, j, \sigma), (i, s, \nu, j, \tau))\) will be, henceforth, written as \(z_{(r, \rho), (s, \sigma)}\). Similarly, for \((r, s, t) \in R^3 : r < s < t\) and \((\rho, \sigma, \tau) \in (\Lambda_r, \Lambda_s, \Lambda_t)\), \(z((i, r, \rho, j, \sigma), (i, s, \nu, j, \tau), (i, t, \mu, j, \tau))\) will be, henceforth, written as \(z_{(r, \rho), (s, \sigma), (t, \tau)}\).

\(\forall (r, s) \in R^2 : s \geq r + 2; (\rho, \sigma) \in (\Lambda_r, \Lambda_s)\), the set of arcs at stage \((r + 1)\) of \(H(y, z)\) through which flow propagates from \(a_{r, \rho}\) onto \(a_{s, \sigma}\) is denoted:

\[
I_{(r, \rho), (s, \sigma)}(y, z) := \{\lambda \in \Lambda_{r+1} : z((r, \rho), (s, \sigma), (r, \lambda)) > 0\}.
\]

\(\forall (r, s) \in R^2 : s \geq r + 2; (\rho, \sigma) \in (\Lambda_r, \Lambda_s)\), the set of arcs at stage \((s - 1)\) of \(H(y, z)\) through which flow propagates from \(a_{s, \rho}\) onto \(a_{r, \sigma}\) is denoted:

\[
J_{(r, \rho), (s, \sigma)}(y, z) := \{\mu \in \Lambda_{s-1} : z((r, \rho), (s-1, \mu), (s, \sigma)) > 0\}.
\]

Remark 3: Let \((y, z) \in Q_L\). An arc of Graph \(G\) is included in Graph \(H(y, z)\) iff at least one flow variable associated with the arc (as defined in Notation 5) is positive.
Lemma 2: Let \((y, z) \in Q_L\). The following holds true:

\[ \forall (r, s) \in R^2: s \geq r + 2; (\rho, \sigma) \in (\Lambda_r, \Lambda_s), \]

\[ \begin{cases} (i) \quad y_{(r, \rho)(s, \sigma)} > 0 \iff \mathcal{I}_{(r, \rho)(s, \sigma)}(y, z) \neq \emptyset; \\ (ii) \quad y_{(r, \rho)(s, \sigma)} > 0 \iff \mathcal{J}_{(r, \rho)(s, \sigma)}(y, z) \neq \emptyset; \\ (iii) \quad y_{(r, \rho)(s, \sigma)} = \sum_{\lambda \in \mathcal{I}_{(r, \rho)(s, \sigma)}(y, z)} z_{(r, \rho)(s + 1, \sigma)}(r, \rho) = \sum_{\mu \in \mathcal{J}_{(r, \rho)(s, \sigma)}(y, z)} z_{(r, \rho)(s - 1, \rho)}(\sigma, \sigma) \end{cases} \]

Definition 7 (‘Paths in \((y, z)\)’): Let \((y, z) \in Q_L\). For \((r, s) \in R^2: s \geq r + 2\), we refer to the set of arcs, \(\{a_{r, v}, a_{r+1, v+1}, \ldots, a_{s, v}\}\), of a walk of \(H(y, z)\) as a ‘path in \((y, z)\) from \((r, v)\) to \((s, v)\)’ if \((\forall (g, p, q) \in R^3: r \leq g < p < q \leq s, z_{(g, v)}(p, v)(q, v) > 0)\).

Remark 4: Let \((y, z) \in Q_L\), \((r, s) \in R^2: s > r\), \((v_r, v_s) \in (\Lambda_r, \Lambda_s)\). It follows directly from definitions that distinct arcs at a given stage \(p \in R: r \leq p \leq s\) cannot belong to a same given path in \((y, z)\) from \((r, v_r)\) to \((s, v_s)\).

Notation 7: Let \((y, z) \in Q_L\). \(\forall (r, s) \in R^2: s \geq r + 2; (\rho, \sigma) \in (\Lambda_r, \Lambda_s)\):

1 the set of all paths in \((y, z)\) from \((r, \rho)\) to \((s, \sigma)\) is denoted \(U_{(r, \rho)(s, \sigma)}(y, z)\)
2 the index set associated with \(U_{(r, \rho)(s, \sigma)}(y, z)\) is denoted \(\Phi_{(r, \rho)(s, \sigma)}(y, z) := \{1, 2, \ldots, \psi_{(r, \rho)(s, \sigma)}(y, z)\}\), where \(\psi_{(r, \rho)(s, \sigma)}(y, z) := |U_{(r, \rho)(s, \sigma)}(y, z)|\)
3 The \(k\)th element of \(U_{(r, \rho)(s, \sigma)}(y, z)(k \in \Phi_{(r, \rho)(s, \sigma)}(y, z))\) is denoted \(L_{(r, \rho)(s, \sigma), k}(y, z)\)
4 \(\forall k \in \Phi_{(r, \rho)(s, \sigma)}(y, z)\), the \((s - r + 2)\)-tuple of task indices (i.e., of levels of \(H(y, z)\)) for the nodes upon which arcs in \(L_{(r, \rho)(s, \sigma), k}(y, z)\) are incident is denoted \(T_{(r, \rho)(s, \sigma), k}(y, z) := (b_{r, v_k}, \ldots, b_{r+1, v_{k+1}})\), where the \((p, v_p, k)\)'s \((p = r, \ldots, s)\) index the arcs in \(L_{(r, \rho)(s, \sigma), k}(y, z)\), and \(b_{r+1, v_{k+1}} := e_{r, v_k}\).

Theorem 6: Let \((y, z) \in Q_L\). The following holds true:

\[ \forall (r, s) \in R^2: s \geq r + 2; (\rho, \sigma) \in (\Lambda_r, \Lambda_s) \]

\[ y_{(r, \rho)(s, \sigma)} > 0 \iff \begin{cases} (i) \quad U_{(r, \rho)(s, \sigma)}(y, z) \neq \emptyset \quad \text{and} \\ (ii) \quad \forall p \in R: r < p < s; \ \forall v_p \in \Lambda_{p}; \quad \langle z_{(r, \rho)(p, v_p)}(\sigma, \sigma) > 0 \quad \iff \exists k \in \Phi_{(r, \rho)(s, \sigma)}(y, z) \quad \exists \alpha_p, v_p \in L_{(r, \rho)(s, \sigma), k}(y, z). \end{cases} \]

Theorem 7: Let \((y, z) \in Q_L\). For \((a, b) \in (\Lambda_1, \Lambda_{n-1})\): \(y_{(1, a)(n-1, b)} > 0\), the following statements are true:

\[ \begin{cases} (i) \quad U_{(1, a)(n-1, b)}(y, z) \neq \emptyset, \quad \text{and} \quad \Phi_{(1, a)(n-1, b)}(y, z) \neq \emptyset \\ (ii) \quad \forall k \in \Phi_{(1, a)(n-1, b)}(y, z), \ \Theta \subseteq T_{(1, a)(n-1, b), k}(y, z) \\ (iii) \quad \forall \langle k \in \Phi_{(1, a)(n-1, b)}(y, z); \ \langle p, q \in (S, S \setminus \{p\}) \rangle, \quad \langle (i_p, i_q) \notin T_{(1, a)(n-1, b), k}(y, z) \quad \text{and} \quad (i_p, i_q) \neq (\theta + 1, \theta + 1) \implies i_p \neq i_q \rangle. \end{cases} \]

Definition 8 (‘SPP path in \((y, z)\)’): Let \((y, z) \in Q_L\). \(\forall (v_1, v_{n-1}) \in (\Lambda_1, \Lambda_{n-1})\), a path in \((y, z)\) from \((1, v_1)\) to \((n - 1, v_{n-1})\) is referred to as an ‘SPP path in \((y, z)\) from \((1, v_1)\) to \((n - 1, v_{n-1})\)’. 
Notation 8: Let \((y, z) \in Q_L\). For all \((\alpha, \beta) \in (\Lambda_1, \Lambda_{n-1})\):

1. the set of all paths in \((y, z)\) from \((1, \alpha)\) to \((n-1, \beta)\) is denoted as \(\Pi_{\alpha\beta}(y, z)\)
2. the index set associated with \(\Pi_{\alpha\beta}(y, z)\) is denoted as \(\Psi_{\alpha\beta}(y, z)\)
   := \{1, 2, \ldots, \pi_{\alpha\beta}(y, z)\}
3. the \(k\)th element of \(\Pi_{\alpha\beta}(y, z)\) is denoted as \(P_{\alpha\beta k}(y, z)\).

Remark 5: Let \((y, z) \in Q_L\).

1. \(\Pi_{\alpha\beta}(y, z) = U_{(1, \alpha)(n-1, \beta)}(y, z)\)
2. \(\Psi_{\alpha\beta}(y, z) = \Phi_{(1, \alpha), (n-1, \beta)}(y, z)\)
3. \(\pi_{\alpha\beta}(y, z) = \varphi_{(1, \alpha)(n-1, \beta)}(y, z)\)
4. we assume (w.l.o.g.) that: \(\forall k \in \pi_{\alpha\beta}(y, z), P_{\alpha\beta k}(y, z) = L_{(1, \alpha), (n-1, \beta), k}(y, z)\).

Theorem 8: For \((y, z) \in Q_L\):

(i) every SPP path in \((y, z)\) corresponds to exactly one SPP path of Graph G
(ii) every SPP path in \((y, z)\) corresponds to exactly one extreme point of the TP-based polytope
(iii) every SPP path in \((y, z)\) corresponds to exactly one point of \(W\)
(iv) every SPP path in \((y, z)\) corresponds to exactly one feasible SPP solution
(v) every SPP path in \((y, z)\) corresponds to exactly one point of \(Q_I\).

Theorem 9: Let \((y, z) \in Q_L\). The following hold true:

(i) \(\forall (r, s, t) \in \mathbb{R}^3 : r < s < t, \; \rho \in \Lambda_r, \; \sigma \in \Lambda_s, \; \tau \in \Lambda_t, \; y(r, \rho)(s, \sigma)(t, \tau) > 0 \iff \exists (\alpha \in \Lambda_1; \beta \in \Lambda_{n-1}; \iota \in \Psi_{\alpha\beta}(y, z)) \ni (a_{r, \rho}, a_{s, \sigma}, a_{t, \tau}) \in \mathcal{P}_{\alpha\beta}(y, z)\)

(ii) \(\forall (r, s) \in \mathbb{R}^2 : r < s, \; \rho \in \Lambda_r, \; \sigma \in \Lambda_s, \; \tau \in \Lambda_t, \; z(r, \rho)(s, \sigma) > 0 \iff \exists (\alpha \in \Lambda_1; \beta \in \Lambda_{n-1}; \iota \in \Psi_{\alpha\beta}(y, z)) \ni (a_{r, \rho}, a_{s, \sigma}, a_{t, \tau}) \in \mathcal{P}_{\alpha\beta}(y, z)\)

Theorem 10 (‘Convex independence’ of SPP paths in \((y, z)\)): Let \((y, z) \in Q_L\). A given SPP path in \((y, z)\) cannot be represented as a convex combination of other SPP paths in \((y, z)\).

Lemma 3: The following constraints are valid for \(Q_L\):

(i) \(\forall (r, s, t) \in \mathbb{R}^3 : r < s < t, \; \sum_{\rho \in \Lambda_r} \sum_{\sigma \in \Lambda_s} \sum_{\iota \in \Lambda_t} \sum_{(\alpha, \beta) \in (\Lambda_1, \Lambda_{n-1})} z(r, \rho)(s, \sigma)(t, \iota) = 1\)

(ii) \(\forall (r, s) \in \mathbb{R}^2 : r < s, \; \sum_{\rho \in \Lambda_r} \sum_{\sigma \in \Lambda_s} y(r, \rho)(s, \sigma) = 1\)
LP formulation of the SPP

Definition 9 (‘Weights’ of SPP paths in \((y, z)\)): Let \((y, z) \in Q_L\). For \((\alpha, \beta) \in (\Lambda_1, \Lambda_{n-1})\): 
\[ y_{(1,\alpha)}(n-1,\beta) > 0, \text{ and } k \in \Psi_{\alpha,\beta}(y, z), \]
we refer to the quantity
\[ \omega_{\alpha\beta k}(y, z) := \min_{(r,s,t) \in R^3: r < s < t; (\rho, \sigma, \tau) \in (\Lambda_r, \Lambda_s, \Lambda_t)} \{ z_{(r,\rho)}(s,\sigma)(t,\tau) \} \]
as the ‘weight’ of (SPP path in \((y, z)\)) \(P_{\alpha\beta k}(y, z)\).

Theorem 11: Let \((y, z) \in Q_L\). The following hold true:

(i) \( \forall (r, s) \in R^2: r < s; (\rho, \sigma) \in (\Lambda_r, \Lambda_s) \),
\[ y_{(r,\rho)}(s,\sigma) = \sum_{\alpha \in \Lambda_1} \sum_{\beta \in \Lambda_{n-1}} \sum_{\iota \in \Psi_{\alpha,\beta}(y, z)} \omega_{\alpha\beta\iota}(y, z) \]

(ii) \( \forall (r, s, t) \in R^3: r < s < t; (\rho, \sigma, \tau) \in (\Lambda_r, \Lambda_s, \Lambda_t) \),
\[ z_{(r,\rho)}(s,\sigma)(t,\tau) = \sum_{\alpha \in \Lambda_1} \sum_{\beta \in \Lambda_{n-1}} \sum_{\iota \in \Psi_{\alpha,\beta}(y, z)} \omega_{\alpha\beta\iota}(y, z) \]

Corollary 1: \((y, z) \in Q_L \iff (y, z)\) corresponds to a convex combination of SPP solutions with coefficients equal to the weights of the corresponding SPP paths in \((y, z)\).

Theorem 12: The following holds true:

(i) \( Q_L = \text{Conv}(Q_I) \)
(ii) \( \text{Ext}(Q_L) = Q_I \).

Corollary 2: The following mappings are bijective:

(i) \( B_1 : \text{Ext}(Q_L) \rightarrow W \)
(ii) \( B_2 : \text{Ext}(Q_L) \rightarrow \Delta \).

4 Overall LP model

4.1 Reformulation of the SPP costs

Notation 9 (Graph \(G\) arc ‘costs’): The ‘cost’ associated with \((i, r, j) \in A\) is defined as:
\[ d_{(i, r)} := \begin{cases} 
  c_{\xi} & \text{if } \xi < \gamma - 1; \ r = f_{\xi}; i = \theta + 1 \\
  c_{\xi} & \text{if } \xi = \gamma - 1; \ r = f_{\xi}; \ i = \theta + 1 \\
  c_{\xi} + c_{\xi+1} & \text{if } \xi = \gamma - 1; \ r = f_{\xi}; \ i = \theta + 1 \\
  c_{\xi} & \text{if } \xi = \gamma - 1; \ r = f_{\xi}; \ i = \theta + 1; \ j = \theta + 1 \\
  c_{\xi+1} & \text{if } \xi = \gamma - 1; \ r = f_{\xi}; \ j = \theta + 1 \\
  0 & \text{otherwise} \end{cases} \]
(24)
The arc costs are illustrated in Figure 5 for the numerical example of Figure 1.
Notation 10 (LP objective function ‘costs’): The ‘costs’ associated with the variables of our LP model are as follows: 

\[ \forall (p, r, s) \in \mathbb{R}^3 : p < r < s; (u, v, i, j, k, t) \in (\Omega, F_p(u), \Omega, F_r(i), \Omega, F_s(k)) \],

\[ d_{(upv)(irj)(kst)} := \begin{cases} 
  d_{upv} + d_{irj} + d_{kst} & \text{if } p = 1; r = 2; s = 3 \\
  d_{kst} & \text{if } p = 1; r = 2; s > 3 \\
  0 & \text{otherwise}
\end{cases} \]

Theorem 13: Let:

\[ \vartheta(y, z) := d^T \times z + 0^T \times y \]

\[ = \sum_{(p, r, s) \in \mathbb{R}^3} \sum_{u \in N_p} \sum_{v \in F_p(u)} \sum_{i \in N_r} \sum_{j \in F_r(i)} \sum_{k \in N_s} \sum_{t \in F_s(k)} d_{(upv)(irj)(kst)} z_{(upv)(irj)(kst)} \]

Then, for \((y, z) \in Ext(Q_L)\), \(\vartheta(y, z)\) accurately accounts the cost of the SPP solution corresponding to \((y, z)\).

4.2 Overall linear programme

Our overall LP model is as follows:

Problem 3 (Problem LP):

\[ \min \{ \vartheta(y, z) : (y, z) \in Q_L \} \]

Theorem 14: The following statements are true of basic feasible solutions (BFS) of Problem LP and SPP solutions:

(i) every BFS of Problem LP corresponds to a SPP solution
(ii) every SPP solution corresponds to a BFS of Problem LP
(iii) the mapping of BFS’s of Problem LP onto SPP solutions is surjective.

Corollary 3: Problem LP solves the SPP.

Figure 5 Illustration of the costs associated with arcs of Graph G (see online version for colours)
5  Concluding remarks

We have developed a TP-based reformulation and a first LP model of the SPP. Our proposed LP model has $O(n^3)$ constraints and $O(n^3)$ variables, where $n$ is the number of non-zero entries of the SPP input matrix. Hence, beyond the scope of the SPP per se, the model provides a new affirmative answer to the long-standing and central question of the equality of the computational complexity classes ‘P’ and ‘NP’.

With respect to solving SPP’s of practical sizes, it appears that the TP-based reformulation we have developed can serve as the basis for good heuristic solution procedures. The challenge involved in using of our LP model to solve practical problems stems from the large-scale nature of the model (even though the complexity order of size is of relatively low degree), and its high level of degeneracy. We believe this challenge may be effectively met however, if the special structure of the model (as developed in this paper) can be exploited judiciously enough, using for example, large-scale optimisation techniques. In particular, we believe that the development of procedures for our proposed LP model along the lines of procedures that have been developed for solving the LP relaxation of the standard IP formulation of the SPP could be the subject of fruitful future research.

6  Acknowledgement

The author would like to thank anonymous referees for comments that greatly improved the paper.

References


LP formulation of the SPP


Proofs

Proof of Theorem 1: Trivial. □

Proof of Theorem 2: The theorem follows directly from the fact that every SPP path represents an extreme point of the standard shortest path network flow polytope associated with Graph $G$, $X := \{ x \in [0,1]^A : \sum_{i \in N_1} \sum_{j \in F_1(i)} x_{i,1,j} = 1; \sum_{j \in F_1(i)} x_{irj} - \sum_{j \in B_1(i)} x_{jtr-1,i} = 0 \forall r \in R \setminus \{1\}, \forall i \in N_r \}$ (where $x$ is the vector of flow variables associated with the arcs of Graph $G$) (see Bazaraa et al., 2005). □

Proof of Theorem 3: With respect to the order of complexity, an upper bound (UB$_v$) on the number of $z$-variables can be obtained as follows. The number of distinct $z_{(r,s)(k,t)(u,v)}$'s ($(p,r,s) \in R^3 : r < s < p; (i,j,k,t,u,v) \in (N_r,F_r(i),N_s,F_s(k),N_p,F_p(u))$) is bounded as follows:

$\begin{align*}
\text{index } i & : 2 \text{ possibilities} \\
\text{index } r & : n \text{ possibilities} \\
\text{index } j & : 2 \text{ possibilities} \\
\text{index } k & : 2 \text{ possibilities} \\
\text{index } s & : n \text{ possibilities} \\
\text{index } t & : 2 \text{ possibilities} \\
\text{index } u & : 2 \text{ possibilities} \\
\text{index } p & : n \text{ possibilities} \\
\text{index } v & : 2 \text{ possibilities}
\end{align*}$

$\Rightarrow \text{UB}_z = (2 \times n \times 2)^3 = 2^6 \times n^3 = 64n^3$

Similarly, the number of distinct $y_{(r,s)(k,t)(u,v)}$'s ($(r,s) \in R^3 : r < s; (i,j,k,t) \in (N_r,F_r(i),N_s,F_s(k))$) is bounded by $U B_y = (2 \times n \times 2)^2 = 2^4 \times n^2 = 16n^2$.

Hence, the number of variables is bounded by $U B_v = U B_z + U B_y = 64n^3 + 16n^2$, which is $O(n^3)$.

The complexity order for the constraints (excluding non-negativity and upper bound constraints) can be based on either of the sets of constraints (11), (12) or (13) (since the number of any of the other sets of constraints would be a lower-degree polynomial function of $n$). Using an approach similar to that for the variables, an upper bound, $U B_c$, for the number of these constraints (respectively) is:

$U B_c = n \times n \times n \times 2 \times 2 \times 2 \times 2 \times 2 = 32n^3$

Hence, the complexity order of the number of constraints is $O(n^3)$. □
Proof of Theorem 4: Let \((y, z) \in Q_I\). Then, given constraints (20)–(21):

(a) \(\implies\):

\((a.i)\) Constraint (10) \(\implies\) there exists a unique 4-tuple, \((i_r \in \Omega, r = 1, \ldots, 4)\), such that:

\[ z_{i_1, i_2, i_3, i_4} = 1 \quad \text{(A1)} \]

Condition (i) follows directly from the combination of (A1), and constraints (11)–(13).

\((a.ii)\) Condition (ii) follows from the combination of Condition (i) with constraints (14)–(16), and (18).

\((a.iii)\) Condition (iii) follows from the combination of Conditions (i) and (ii) with constraints (17).

\((a.iv)\) Condition (iv) follows from the combination of Condition (iii) with constraints (19).

(b) \(\iff\): Trivial.

Proof of Theorem 5: Condition (i) follows directly from the combination of Theorem 4 and Definition 4.2. Conditions (ii)–(iv) of the theorem follow directly from the combination of Condition (i) with Remark 2.

Proof of Lemma 1: For \(r \in R\), constraints (15) for \(s = r + 1\) and \(p = r + 2\) can be written as:

\[ y_{(i_r, i_{r+1})(i_{r+2}, r+2, i_{r+3})} = \sum_{k \in \Omega} \sum_{t \in Fr_{r+1}(k)} z_{(i_r, r, i_{r+1})(k, r+1, t)(i_{r+2}, r+2, i_{r+3})} = 0 \quad \forall (i_r, i_{r+1}, i_{r+2}, i_{r+3}) \in (\Omega, Fr(i_r), \Omega, Fr_{r+2}(i_{r+2})) \quad \text{(A2)} \]

Constraints (18), and (14)–(16) \(\implies\)

\[ \forall (i_r, i_{r+1}, i_{r+2}, i_{r+3}, k, t) \in (\Omega, Fr(i_r), \Omega, Fr_{r+2}(i_{r+2}), \Omega, \Omega), \]

\[ \{z_{(i_r, r, i_{r+1})(k, r+1, t)(i_{r+2}, r+2, i_{r+3})} > 0 \implies k = i_{r+1}, \text{ and } t = i_{r+2}\} \quad \text{(A3)} \]

Using (A2), (A3) can be written as:

\[ y_{(i_r, i_{r+1})(i_{r+2}, r+2, i_{r+3})} = z_{(i_r, r, i_{r+1})(i_{r+1}, r+1, i_{r+2})(i_{r+2}, r+2, i_{r+3})} = 0 \quad \forall (i_r, i_{r+1}, i_{r+2}, i_{r+3}) \in (\Omega, Fr(i_r), \Omega, Fr_{r+2}(i_{r+2})) \quad \text{(A4)} \]

Condition (ii) of the equivalence in the lemma follows directly from (A3) to (A4). Condition (i) follows from the combination of (A4) and the fact that by constraints (21),

\[ z_{(i_r, r, i_{r+1})(i_{r+1}, r+1, i_{r+2})(i_{r+2}, r+2, i_{r+3})} \]

is allowed to be positive only if \(i_{r+2} \in Fr_{r+2}(i_{r+2})\).

Proof of Lemma 2: The theorem follows directly from the combination of constraints (15) and constraints (18).
LP formulation of the SPP

Proof of Theorem 6: First, note that it follows directly from the combination of Lemma 1, that the theorem holds true for all \((r, s) \in R^2\) with \(s = r + 2\), and all \((v_r, v_s) \in (\Lambda_r, \Lambda_s)\).

\((a) \implies \):

Assume there exists an integer \(\omega \geq 2\) such that the theorem holds true for all \((r, s) \in R^2\) with \(s = r + \omega\), and all \((v_r, v_s) \in (\Lambda_r, \Lambda_s)\). We will show that the theorem must hold for all \((r, s) \in R^2\) with \(s = r + \omega + 1\), and all \((v_r, v_s) \in (\Lambda_r, \Lambda_s)\).

Let \((p, q) \in R^2\) with \(q = p + \omega + 1\), and \((\alpha, \beta) \in (\Lambda_p, \Lambda_q)\) be such that:

\[ y_{(p, \alpha)}(q, \beta) > 0 \]  \hspace{0.5em} \text{(A5)}

\((a.1)\) Relation (A5) and Lemma 2 \implies

\[ I_{(p, \alpha)(q, \beta)}(y, z) \neq \emptyset \]  \hspace{0.5em} \text{(A6)}

It follows from (A6), Notation 6.8, and constraints (16) that:

\[ \forall \lambda \in I_{(p, \alpha)(q, \beta)}(y, z), \; y_{(p+1, \lambda)}(q, \beta) > 0 \]  \hspace{0.5em} \text{(A7)}

By assumption (since \(q = (p + 1) + \omega\)), (A7) \implies

\((a.1.1)\) \hspace{0.5em} \forall \lambda \in I_{(p, \alpha)(q, \beta)}(y, z), \; U_{(p+1, \lambda)}(q, \beta)(y, z) \neq \emptyset \hspace{0.5em} \text{(A8a)}

\((a.1.2)\) \hspace{0.5em} \forall (\lambda, \mu) \in I_{(p, \alpha)(q, \beta)}(y, z), \; t \in R : p + 1 < t < q; \; \tau \in \Lambda_t, \;

\[ \left\{ \begin{array}{l}
z_{(p+1, \lambda)(\tau)}(q, \beta) > 0 \\
\exists i \in \Phi_{(p+1, \lambda)(q, \beta)}(y, z) \; \ni a_{i, \tau}
\end{array} \right. \in L_{(p+1, \lambda)(q, \beta)}(y, z) \hspace{0.5em} \text{(A8b)}

\((a.2)\) Relation (A5) and Lemma 2 \implies

\[ J_{(p, \alpha)(q, \beta)}(y, z) \neq \emptyset \]  \hspace{0.5em} \text{(A9)}

It follows from (A9), Notation 6.9, and constraints (14) that:

\[ \forall \mu \in J_{(p, \alpha)(q, \beta)}(y, z), \; y_{(p, \alpha)}(q-\mu) > 0 \]  \hspace{0.5em} \text{(A10)}

By assumption (since \((q - 1) = p + \omega\)), (A10) \implies

\((a.2.1)\) \hspace{0.5em} \forall \mu \in J_{(p, \alpha)(q, \beta)}(y, z), \; U_{(p, \alpha)}(q-\mu)(y, z) \neq \emptyset \hspace{0.5em} \text{(A11a)}

\((a.2.2)\) \hspace{0.5em} \forall (\mu, \lambda) \in J_{(p, \alpha)(q, \beta)}(y, z), \; t \in R : p < t < q - 1; \; \tau \in \Lambda_t, \;

\[ \left\{ \begin{array}{l}
z_{(p, \alpha)(t, \tau)}(q-\mu) > 0 \\
\exists k \in \Phi_{(p, \alpha)(q-\mu)}(y, z) \; \ni a_{i, \tau}
\end{array} \right. \in L_{(p, \alpha)(q-\mu)}(y, z) \hspace{0.5em} \text{(A11b)}

\((a.3)\) Constraints (14)–(17) and Lemma 2.iii \implies

\((a.3.1)\) \hspace{0.5em} \forall \alpha \in \Lambda_{q-1}, \; \exists (\lambda, \mu) \in I_{(p, \alpha)(q, \beta)}(y, z), \; i \in \Phi_{(p+1, \lambda)(q, \beta)}(y, z)

\[ \ni \{ a_{q-1, \mu} \in L_{(p+1, \lambda)(q, \beta)}(y, z) \} \hspace{0.5em} \text{(A12a)}

\((a.3.2)\) \hspace{0.5em} \forall \alpha \in \Lambda_{p+1}, \; \exists (\mu, \lambda) \in J_{(p, \alpha)(q, \beta)}(y, z), \; k \in \Phi_{(p, \alpha)(q-\mu)}(y, z)

\[ \ni \{ a_{p+1, \lambda} \in L_{(p, \alpha)(q-\mu)}(y, z) \} \hspace{0.5em} \text{(A12b)}

The combination of (A8a), (A8b), (A11a)–(A12b), constraints (17) and constraints (12) \[\implies \exists \langle \lambda \in I(p,\alpha)(q,\beta) (y, z); \ i \in \Phi(p+1,\lambda)(q,\beta)(y, z); \ \mu \in J(p,\alpha)(q,\beta) (y, z); \ \kappa \in \Phi(p,\alpha)(q-1,\mu)(y, z) \rangle \in \langle \forall \langle t \in R: p < t < q; \ \tau \in \Lambda_t: a_{r,\tau} \in \mathcal{L}(p+1,\lambda)(q,\beta), i (y, z) \rangle; \ z(p,\alpha)(t,\tau)(q,\beta) > 0; \ (\mathcal{L}(p+1,\lambda)(q,\beta), i (y, z) \{aq,\beta\}) \neq \emptyset \rangle \] (A13)

(In words, (A13) says that there must exist paths in \((y, z)\) from \((p+1, \lambda)\) to \((q, \beta)\), and paths in \((y, z)\) from \((p, \alpha)\) to \((q-1, \beta)\) that ‘overlap’ at intermediary stages between \((p+1)\) and \((q-1)\) (inclusive)).

Let \(\lambda \in I(p,\alpha)(q,\beta) (y, z), i \in \Phi(p+1,\lambda)(q,\beta)(y, z), \mu \in J(p,\alpha)(q,\beta) (y, z)\) and \(k \in \Phi(p,\alpha)(q-1,\mu)(y, z)\) be such that they satisfy (A13). Then, it follows directly from definitions that

\[
\mathcal{L}(p,\alpha)(q-1,\mu), k (y, z) \]

is a path in \((y, z)\) from \((p, \alpha)\) to \((q, \beta)\).

Hence, we have that \(U(p,\alpha)(q,\beta)(y, z) \neq \emptyset\).

\(b) \iff: \) Follows directly from definitions and constraints (14)–(16).

Proof of Theorem 7: Condition (i) follows from Theorem 6. Condition (ii) follows from constraints (17). Condition (iii) follows from constraints (19).

Proof of Theorem 8: Condition (i) follows from the combination of Theorem 7 and Definition 4.2. Conditions (ii)–(v) follow from the combination of Condition (i) with Theorem 5, and Remark 2.

Proof of Theorem 9: The theorem follows directly from Theorem 6.

Proof of Theorem 10: The theorem follows directly from the combination of Theorems 2 and 8.

Proof of Lemma 3:

(i) First, note that by constraint (10), Condition (i) of the lemma holds for \((r, s, t) = (1, 2, 3)\).

Now, assume \(1 < r < s < t\). Then, we have:

\[
\sum_{\nu_i \in \Lambda_r} \sum_{\nu_j \in \Lambda_s} \sum_{\nu_k \in \Lambda_t} z(r,\nu_i)(s,\nu_j)(t,\nu_k) = \sum_{\nu_i \in \Lambda_r} \sum_{\nu_j \in \Lambda_s} y(r,\nu_i)(s,\nu_j) \quad \text{(Using (14))}
\]

\[
= \sum_{\nu_i \in \Lambda_r} \sum_{\nu_j \in \Lambda_s} \sum_{\nu_k \in \Lambda_t} z(t,\nu_k)(r,\nu_j)(s,\nu_i) \quad \text{(Using (16))}
\]

\[
= \sum_{\nu_k \in \Lambda_t} \sum_{\nu_i \in \Lambda_r} \sum_{\nu_j \in \Lambda_s} z(t,\nu_k)(r,\nu_j)(s,\nu_i) \quad \text{(Rearranging)}
\]
LP formulation of the SPP

\[ \sum_{v_1 \in \Lambda_1} \sum_{v_2 \in \Lambda_2} \gamma_{(1,v_1),(r,v_2)} \] (Using (15))

\[ \sum_{v_1 \in \Lambda_1} \sum_{v_2 \in \Lambda_2} \sum_{v_3 \in \Lambda_3} \gamma_{(1,v_1),(2,v_2),(3,v_3)} \] (Using (16))

\[ \sum_{v_1 \in \Lambda_1} \sum_{v_2 \in \Lambda_2} \sum_{v_3 \in \Lambda_3} \gamma_{(1,v_1),(2,v_2),(3,v_3)} \] (Rearranging)

\[ \sum_{v_1 \in \Lambda_1} \sum_{v_2 \in \Lambda_2} \sum_{v_3 \in \Lambda_3} \gamma_{(1,v_1),(2,v_2),(3,v_3)} \] (Using (14))

\[ = 1 \] (Using (10) and (18))

(ii) Condition (ii) of the theorem follows directly from the combination of Condition (i) and constraints (14)–(16).

Proof of Theorem 11:

(i) Let \((r,s) \in R^2 : r < s\).

From the combination of constraints (10)–(16), Theorem 10 and Lemma 3, we have:

\[ \sum_{\rho \in \Lambda_r} \sum_{\sigma \in \Lambda_s} \gamma_{(r,\rho),(s,\sigma)} = \sum_{\alpha \in \Lambda_1} \sum_{\beta \in \Lambda_{n-1}} \sum_{\iota \in \Psi(y,z)} \omega_{\alpha\beta\iota}(y,z) = 1 \] (A15)

Using Theorem 9, we have:

\[ \sum_{\alpha \in \Lambda_1} \sum_{\beta \in \Lambda_{n-1}} \sum_{\iota \in \Psi(y,z)} \omega_{\alpha\beta\iota}(y,z) = \sum_{\sigma \in \Lambda_s} \sum_{\alpha \in \Lambda_1} \sum_{\beta \in \Lambda_{n-1}} \sum_{\iota \in \Psi(y,z)} \omega_{\alpha\beta\iota}(y,z) \sigma_{s,\sigma} \in \mathcal{P}_{\alpha\beta\iota}(y,z) \] (A16)

Combining (A15) and (A16), we have:

\[ \sum_{\sigma \in \Lambda_s} \left( \sum_{\rho \in \Lambda_r} \gamma_{(r,\rho),(s,\sigma)} - \sum_{\alpha \in \Lambda_1} \sum_{\beta \in \Lambda_{n-1}} \sum_{\iota \in \Psi(y,z)} \omega_{\alpha\beta\iota}(y,z) \right) = 0 \] (A17)

Using Remark 4 and Theorem 10, (A17) \(\implies\)

\[ \forall \sigma \in \Lambda_s, \sum_{\rho \in \Lambda_r} \gamma_{(r,\rho),(s,\sigma)} - \sum_{\alpha \in \Lambda_1} \sum_{\beta \in \Lambda_{n-1}} \sum_{\iota \in \Psi(y,z)} \omega_{\alpha\beta\iota}(y,z) = 0 \] (A18)

(Intuitively, the total flow on a given arc cannot be made to 'balance' using flows from paths to which the arc does not belong.)
Theorem 9 \(\implies\)
\[
\forall \sigma \in \Lambda_s, \sum_{\alpha \in \Lambda_1} \sum_{\beta \in \Lambda_{n-1}} \sum_{\tau \in \Psi_1(\alpha, \beta) : (\alpha, \beta, \tau) \in P_{\lambda, \mu}} \omega_{\alpha \beta \tau}(y, z)
\]
\[
= \sum_{\rho \in \Lambda_r} \sum_{\alpha \in \Lambda_1} \sum_{\beta \in \Lambda_{n-1}} \sum_{\tau \in \Psi_1(\alpha, \beta) : (\alpha, \beta, \tau) \in P_{\lambda, \mu}} \omega_{\alpha \beta \tau}(y, z)
\]  \hspace{1cm} (A19)

Combining (A18) and (A19) gives:
\[
\forall \sigma \in \Lambda_s, \sum_{\rho \in \Lambda_r} \left( y(r, \rho)(s, \sigma) - \sum_{\alpha \in \Lambda_1} \sum_{\beta \in \Lambda_{n-1}} \sum_{\tau \in \Psi_1(\alpha, \beta) : (\alpha, \beta, \tau) \in P_{\lambda, \mu}} \omega_{\alpha \beta \tau}(y, z) \right) = 0
\]  \hspace{1cm} (A20)

Using Remark 4 and Theorem 10, (A20) \(\implies\)
\[
\forall \{\rho \in \Lambda_r, \sigma \in \Lambda_s\}, y(r, \rho)(s, \sigma) - \sum_{\alpha \in \Lambda_1} \sum_{\beta \in \Lambda_{n-1}} \sum_{\tau \in \Psi_1(\alpha, \beta) : (\alpha, \beta, \tau) \in P_{\lambda, \mu}} \omega_{\alpha \beta \tau}(y, z) = 0
\]  \hspace{1cm} (A21)

hence, condition (i) of the theorem.

(ii) Let \((r, s, t) \in \mathbb{R}^3 : r < s < t, (\rho, \sigma) \in (\Lambda_r, \Lambda_s)\).

Constraints (14) and condition (i) of the theorem \(\implies\)
\[
y(r, \rho)(s, \sigma) = \sum_{\tau \in \Lambda_t} z(r, \rho)(s, \sigma)(t, \tau) = \sum_{\alpha \in \Lambda_1} \sum_{\beta \in \Lambda_{n-1}} \sum_{\tau \in \Psi_1(\alpha, \beta) : (\alpha, \beta, \tau) \in P_{\lambda, \mu}} \omega_{\alpha \beta \tau}(y, z)
\]  \hspace{1cm} (A22)

Using Theorem 9, we have:
\[
\forall \tau \in \Lambda_t,
\]
\[
\sum_{\alpha \in \Lambda_1} \sum_{\beta \in \Lambda_{n-1}} \sum_{\tau \in \Psi_1(\alpha, \beta) : (\alpha, \beta, \tau) \in P_{\lambda, \mu}} \omega_{\alpha \beta \tau}(y, z)
\]
\[
= \sum_{\tau \in \Lambda_t} \sum_{\alpha \in \Lambda_1} \sum_{\beta \in \Lambda_{n-1}} \sum_{\tau \in \Psi_1(\alpha, \beta) : (\alpha, \beta, \tau) \in P_{\lambda, \mu}} \omega_{\alpha \beta \tau}(y, z)
\]  \hspace{1cm} (A23)

Combining (A22) and (A23) gives:
\[
\sum_{\tau \in \Lambda_t} \left( z(r, \rho)(s, \sigma)(t, \tau) - \sum_{\alpha \in \Lambda_1} \sum_{\beta \in \Lambda_{n-1}} \sum_{\tau \in \Psi_1(\alpha, \beta) : (\alpha, \beta, \tau) \in P_{\lambda, \mu}} \omega_{\alpha \beta \tau}(y, z) \right) = 0
\]  \hspace{1cm} (A24)
**LP formulation of the SPP**

Using Remark 4 and Theorem 10, (A24) \[ \implies \]
\[ \forall \tau \in \Lambda_1, \; z_{(r,p)(s,\sigma)(t,\tau)} = \sum_{a \in \Lambda_1} \sum_{\beta \in \Lambda_{a-1}} \sum_{i \in \Psi^{a\beta}_{a\beta}(y,z)} \alpha_{a\beta i}(y,z) \; \in \mathcal{P}_{a\beta i}(y,z) \]
\[ \omega_{a\beta i}(y,z) = 0 \quad (A25) \]

Condition (ii) of the theorem follows from (A25) directly. \[ \square \]

**Proof of Theorem 12**: The theorem follows directly from the combination of Corollary 1, Theorem 10, and Theorem 8. \[ \square \]

**Proof of Theorem 13**: From Theorem 12,
\[ (y, z) \in \text{Ext}(Q_L) \iff (y, z) \in Q_L \]
Now, using Theorem 4, and Definition 6, it can be verified directly that for \((y, z) \in Q_L\),
\[ \theta(y,z) = \sum_{p \in \mathcal{W}(y,z)} c_p. \]
\[ \square \]

**Proof of Theorem 14**: Statements (i) and (ii) of the theorem follow directly from the combination of Theorem 12, Corollary 2, and the correspondence between BFS’s of LP models and extreme points of their associated polyhedra (see Bazaraa et al., 2005, pp.92–101). Statement (iii) follows from the primal degeneracy of Problem LP (see Nemhauser and Wolsey, 1988, p.32). \[ \square \]