Reformulating Linear Programs with Transportation Constraints – with Applications to Workforce Scheduling

Tolga Çezik
Columbia University, New York, New York 10027

Oktay Günlük
IBM Research, Yorktown Heights, New York 10598
Email: oktay@watson.ibm.com

December 11, 2002

Abstract

We study linear programming models that contain transportation constraints in their formulation. Typically, these models have a multi-stage nature and the transportation constraints together with the associated flow variables are used to achieve consistency between consecutive stages. We describe how to reformulate these models by projecting out the flow variables. The reformulation can be more desirable since it has fewer variables and can be solved faster. We apply these ideas to reformulate two well-known workforce staffing and scheduling problems: the shift scheduling problem and the tour scheduling problem. We also present computational results.

1 Introduction

In this paper, we study linear programming (LP) models that contain transportation constraints in their formulation. Models with this structure may arise in different contexts such as workforce scheduling or production planning. Typically, the model has a multi-stage nature and the transportation constraints (together with the associated flow variables) are needed to attain consistency between consecutive stages of the model. Our motivation for this study comes from a workforce scheduling application (see Cezik et al. [9]) where the original model has several stages and the flow variables model the "flow" of workers from one stage to the next.

A generic two-stage LP model with this structure can be formulated as follows:

$$\textbf{(LP)} \qquad \text{Min} \qquad c^1x + c^2y + c^3w \tag{1}$$

Subject to:

$$[x, y, w] \in P \tag{2}$$

$$\sum_{j \in R(i)} v_{ij} = x_i \qquad for\ all\ i \in I \tag{3}$$

$$\sum_{i \in T(j)} v_{ij} = y_j \qquad for\ all\ j \in J \tag{4}$$

$$x, y, v \geq 0. \tag{5}$$

Constraint (2) above contains application specific constraints which we refer to as the *production* constraints. Constraints (3) and (4) are the *transportation* constraints. Variables x and y correspond to *supply and demand* levels in consecutive stages of the model and variable v corresponds to the flow of goods, or personnel from one stage to the next. Sets $I = \{1, 2, \ldots, |I|\}$ and $J = \{1, 2, \ldots, |J|\}$ contain the indices of the x and y variables, respectively. Finally, for every supply variable x_i there is an associated index set $R(i) \subseteq J$ that specifies the accessible demand variables in the next stage. Similarly, $T(j) = \{i \in I \ : \ j \in R(i)\}$ gives the indices of the supply variables that can access y_j. Note that the flow variables do not appear in the objective function and they are only used to guarantee consistency between the two stages.

Our purpose in this paper is to formulate an equivalent problem **(LP′)** by replacing transportation constraints (3) and (4) with a new set of constraints that only involve supply and demand variables. In other words, we project out the flow variables to obtain an equivalent formulation in the space of the remaining variables. The projection clearly depends on the sets $R(i)$ for $i \in I$ and we study some conditions under which it can be described easily. It is straight forward to extend this approach to models with more than two stages by simply applying the projection idea iteratively. The new formulation **(LP′)** can be more desirable since it has fewer variables and can be solved faster. In Section 4 we provide some computational evidence supporting this claim.

The paper is organized as follows: In the remainder of this section, we describe our problem more precisely. In Section 2, we investigate some special cases when the projected formulation **(LP′)** can be described with a small number of additional constraints. In Section 3, we present two well-known workforce scheduling formulations from the literature, namely, the shift scheduling problem and the tour scheduling problem and describe how our results may lead to better formulation. Finally, in Section, 4 we present some computational results.

1.1 Preliminaries

Let T be the set of points (in the space of x, y and v variables) that satisfy constraints (3), (4) and (5). The set of feasible solutions to **(LP)** can now be written as:

$$Q \;=\; \{[x, y, w, v] \in X \;:\; [x, y, w] \in P,\; [x, y, v] \in T\}$$

where X is the space of real numbers of appropriate dimension. Projecting out v variables from Q (see Nemhauser and Wolsey [18]) gives the set:

$$Proj_{[x,y,w]}(Q) \;=\; \{[x, y, w] \in X' \;:\; [x, y, w, v] \in Q \text{ for some } v\}$$

which can be rewritten as:

$$Proj_{[x,y,w]}(Q) \;=\; \{[x, y, w] \in X' \;:\; [x, y, w] \in P,\; [x, y] \in Proj_{[x,y]}(T)\}$$

where X' is the space of real numbers of appropriate dimension. Problem **(LP′)** then becomes:

(LP′) Min $c^1x + c^2y + c^3w$

Subject to:

$$\begin{aligned} [x, y, w] &\in P \\ [x, y] &\in Proj_{[x,y]}(T). \end{aligned} \tag{6}$$

Therefore, we are essentially interested in an explicit description of $Proj_{[x,y]}(T)$ which can be obtained by identifying the conditions (linear inequalities) that x and y should satisfy to guarantee the existence of an associated flow vector v so that $[x, y, v] \in T$.

1.2 The Bipartite Flow Problem

To produce an explicit description of $Proj_{[x,y]}(T)$, we study the following closely related flow problem which can be considered as the feasibility version of the well-known *transportation problem* (see, for example, Ahuja et al. [1]).

Bipartite Flow Feasibility Problem: *Let two disjoint sets of nodes $\mathcal{U}$ and $\mathcal{K}$ be given together with two associated (supply and demand) vectors $s, d \geq 0$, $s \in R^{|\mathcal{U}|}$ and $d \in R^{|\mathcal{K}|}$. Furthermore, for each $u \in \mathcal{U}$ assume that an associated set $R(u) \subseteq \mathcal{K}$ be also given.*

The bipartite flow feasibility problem (BFFP) is to decide if it is possible to route all of the supplies to satisfy all of the demands using the directed edge set $A = \{(u, v) : u \in \mathcal{U}, v \in R(u)\}$.

Clearly $\sum_{u\in\mathcal{U}} s_u = \sum_{v\in\mathcal{K}} d_v$ is a necessary condition for feasibility. To find the sufficiency conditions, we construct a capacitated network $G^+ = (V^+, A^+)$ where $V^+ = \mathcal{U} \cup \mathcal{K} \cup \{o, t\}$ and $A^+ = A \cup \{(o, u) : u \in \mathcal{U}\} \cup \{(v, t) : v \in \mathcal{K}\}$. For $u \in \mathcal{U}$, capacity of (o, u) is set to s_u and for $v \in \mathcal{K}$, capacity of (v, t) is set to d_v. Capacity of the edges in A are set to ∞.

Using the min-cut max-flow theorem (Ford and Fulkerson [12]), BFFP is feasible if and only if G^+ can accommodate $\sum_{u\in\mathcal{U}} s_u$ units of flow from o to t. For a partition $S \cup \bar{S}$ of V^+ with $o \in S$, $t \in \bar{S}$, let $\delta(S) \subset A^+$ be the associated (directed) cut and let $cap(S)$ denote the capacity of $\delta(S)$. Notice that

$$cap(S) = \begin{cases} \infty & if \quad \delta(S) \cap A \neq \emptyset, \\ \displaystyle\sum_{u\in\mathcal{U}\setminus S} s_u + \sum_{v\in\mathcal{K}\cap S} d_v & otherwise. \end{cases}$$

For $\bar{\mathcal{U}} \subseteq \mathcal{U}$, let $R(\bar{\mathcal{U}}) = \bigcup_{u\in\bar{\mathcal{U}}} R(u)$ and call a set $S \subseteq V^+$ *non-trivial* if $\delta(S) \cap A = \emptyset$, or, equivalently, if

$$R(\mathcal{U} \cap S) \subseteq \mathcal{K} \cap S.$$

Therefore, the flow problem has a solution if and only if the cut capacity $\sum_{u\in\mathcal{U}\setminus S} s_u + \sum_{v\in\mathcal{K}\cap S} d_v$ is at least $\sum_{u\in\mathcal{U}} s_u$ or, equivalently,

$$\sum_{v\in\mathcal{K}\cap S} d_v \quad \geq \quad \sum_{u\in\mathcal{U}\cap S} s_u \tag{7}$$

holds for all non-trivial subsets S of V^+. For $\bar{\mathcal{K}} \subseteq \mathcal{K}$, let $T(\bar{\mathcal{K}}) = \{u \in \mathcal{U} : R(u) \subseteq \bar{\mathcal{K}}\}$. Notice that inequality (7) for a given $S \subseteq V^+$ is dominated by another inequality of the same form for some $S' \subseteq V^+$, unless it satisfies the following conditions:

$$\begin{aligned} (i) \quad & R(\mathcal{U} \cap S) = \mathcal{K} \cap S, \text{ and} \\ (ii) \quad & T(\mathcal{K} \cap S) = \mathcal{U} \cap S. \end{aligned}$$

Therefore, BFFP is feasible if and only if total supply equals total demand, and inequality (7) holds for all $S \subseteq V^+$ that satisfy $R(T(\mathcal{K} \cap S)) = \mathcal{K} \cap S$.

We note that this observation also follows from various earlier results in network flows. See, for example, Gale [13]. We also note that the derivation we presented above is very similar to the one presented in Cook et al.[10].

We next relate these observations to **(LP′)** by defining $R(\cdot)$ and $T(\cdot)$ for subsets of the associated index sets.

Proposition 1

$$Proj_{[x,y]}(T) = \Big\{[x,y] \in X'' : \sum_{j\in J} y_j = \sum_{i\in I} x_i, \text{ and,} \quad \sum_{j\in \bar{J}} y_j \geq \sum_{i\in T(\bar{J})} x_i \text{ for all } \bar{J} \subseteq J \text{ such that } R(T(\bar{J})) = \bar{J}\Big\}$$

where X'' is the space of non-negative real numbers of appropriate dimension. ∎

Flow variables v can therefore be deleted from **(LP)** together with constraints (3) and (4) if the above (exponentially many) constraints are included in the formulation. In Section 2 present some special cases which require fewer constraints.

1.3 Relationship to the Matching Problem

Bipartite matching problems are closely related to and can be considered as special cases of network flow problems (see Lawler [17] for examples of reduction techniques that demonstrate this). In particular, transportation problem and bipartite matching problems are very closely related: When all of the entries of the supply and demand vectors in BFFP are 1, BFFP reduces to verifying or falsifying existence of a perfect matching in a bipartite graph.

Based on this observation, we note that Proposition 1 can be considered as a generalization of an earlier result by Balas and Pulleyblank [4] on the perfectly matchable subgraph polytope of a bipartite graph. Given a graph $G = (V, E)$, a subset of the node set $S \subseteq V$ is said to induce a perfectly matchable subgraph of G, if G admits a matching that covers only and all of the nodes in S. In [4], the authors show that the convex hull of the incidence vectors of perfectly matchable node sets of a graph is the intersection of $Proj_{[x,y]}(T)$ with the unit cube of appropriate dimension. We note that the special cases and resulting simple descriptions of $Proj_{[x,y]}(T)$ that we describe in the following section can also be used to produce simple descriptions of the perfectly matchable subgraph polytope for bipartite graphs.

2 Special Cases

In this section we make some assumptions on the input data under which BFFP can be solved easily. We then relate these assumptions and the corresponding results to $Proj_{[x,y]}(T)$. In all cases it is also assumed that the following trivial property holds:

Assumption 0 (A0): *Assume that the input data satisfies* $\sum_{u\in\mathcal{U}} s_u = \sum_{v\in\mathcal{K}} d_v$.

The remaining assumptions are on the structure of the sets $R(u)$ based on a fixed ordering of the nodes in $\mathcal{K} = \{v_1, v_2, \ldots, v_{|\mathcal{K}|}\}$. To simplify notation, from now on, we use the index set $\mathcal{J} = \{1, 2, \ldots, |\mathcal{K}|\}$ in place of $\mathcal{K}$.

2.1 Special Case I

We first study the case when the demand nodes can be ordered in such a way that the sets $R(u)$ form intervals for all $u \in \mathcal{U}$. This assumption models the case when the indices of the demand nodes correspond to time periods and supply nodes have "availability" and "expiration" dates on their supply.

Assumption 1 (A1): *Assume that for all $u \in \mathcal{U}$, $R(u) = \{a_u, a_u + 1, \ldots, b_u\}$ for some $a_u, b_u \in \mathcal{J}$.*

For this case, it is sufficient to consider only a quadratic number of inequalities of the form (7) to solve the feasibility problem.

Lemma 2 *Under Assumptions A0 and A1, BFFP is feasible if and only if*

$$\sum_{j \in \bar{\mathcal{J}}} d_j \geq \sum_{u \in T(\bar{\mathcal{J}})} s_u \tag{8}$$

holds for all $\bar{\mathcal{J}} = \{j_1, j_1 + 1, \ldots, j_2\}$ with $j_1 \leq j_2$; $j_1,\ j_2 \in \mathcal{J}$.

Proof. To see the only if part it suffices to notice inequality (8) is identical to (7) with $S = \{o\} \cup \bar{\mathcal{J}} \cup T(\bar{\mathcal{J}})$.

Let $\kappa = \min_{Q \subseteq V^+}\{cap(Q)\}$. To prove the if part by contradiction, assume that $\kappa < \sum_{u \in \mathcal{U}} s_u$ even though (8) holds for all $j_1,\ j_2 \in \mathcal{J}$. Let $\bar{S} \subseteq V^+$ give a minimum cut and be minimal i.e. (i) $cap(\bar{S}) = \kappa$ and (ii) $|\bar{S}| = \min\{|S| \ :\ S \subseteq V^+,\ cap(S) = \kappa\}$.

Let a be the smallest member of $\bar{S} \cap \mathcal{J}$ and let b the smallest member of $\bar{S} \cap \mathcal{J}$ such that $b{+}1 \notin \bar{S} \cap \mathcal{J}$. More precisely $a = \min\{j : j \in \bar{S} \cap \mathcal{J}\}$ and $b = \max\{j : \{a, a{+}1, \ldots, j\} \subseteq \bar{S} \cap \mathcal{J}\}$. Let $\mathcal{J}' = \{a, a + 1, \ldots, b\}$. If $\bar{S} \cap \mathcal{J} = \mathcal{J}'$, then (8) implies that $cap(\bar{S}) = \kappa \geq \sum_{u \in \mathcal{U}} s_u$, a contradiction. Therefore, we assume that $\bar{S} \cap \mathcal{J} \neq \mathcal{J}'$, and $c = \min\{j > b + 1 \ :\ j \in \bar{S} \cap \mathcal{J}\}$ exists.

Since $\bar{S}$ is minimal, it has to be non-trivial and thus $R(\bar{S} \cap \mathcal{U}) \subseteq \bar{S} \cap \mathcal{J}$. Let $\mathcal{J}'' = \bar{S} \cap \mathcal{J} \setminus \mathcal{J}'$. For any $u \in \mathcal{U} \cap \bar{S}$, since $b + 1 \notin R(u) \subseteq \bar{S} \cap \mathcal{J}$, either $R(u) \subseteq \mathcal{J}'$ or $R(u) \subseteq \mathcal{J}''$ must hold.

Therefore we can partition $\mathcal{U} \cap \bar{S}$ into $\mathcal{U}' = \{u \in \bar{S} \cap \mathcal{U} : R(u) \subseteq \mathcal{J}'\}$ and $\mathcal{U}'' = \{u \in \bar{S} \cap \mathcal{U} : R(u) \subseteq \mathcal{J}''\}$ (see Figure 1).

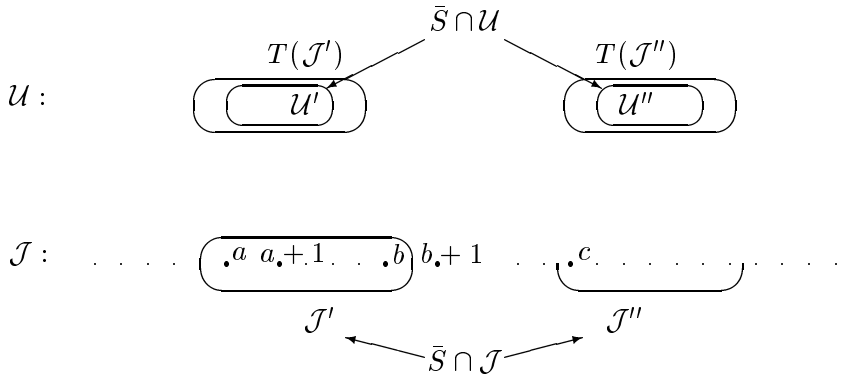

Figure 1: Node sets for the proof of Lemma 2

It is now possible construct a new non-trivial set $\tilde{S} = \{o\} \cup \mathcal{U}'' \cup \mathcal{J}''$ with

$$\begin{aligned}
cap(\tilde{S}) &= \sum_{u \notin \mathcal{U}''} s_u + \sum_{j \in \mathcal{J}''} d_j \\
&= \sum_{u \notin \mathcal{U}'' \cup \mathcal{U}'} s_u + \sum_{u \in \mathcal{U}'} s_u + \sum_{j \in \mathcal{J}'' \cup \mathcal{J}'} d_j - \sum_{j \in \mathcal{J}'} d_j \\
&= cap(\bar{S}) + \sum_{u \in \mathcal{U}'} s_u - \sum_{j \in \mathcal{J}'} d_j \\
&\leq cap(\bar{S}) + \sum_{u \in T(\mathcal{J}')} s_u - \sum_{j \in \mathcal{J}'} d_j \\
&\leq cap(\bar{S})
\end{aligned}$$

where the last inequality follows from the starting assumption that (8) is satisfied for $\mathcal{J}'$. This gives the desired contradiction since $|\tilde{S}| < |\bar{S}|$ even though $\bar{S}$ should be minimal. ∎

Lemma 2 shows that under Assumptions A0 and A1, feasibility of BFFP can be verified (or, falsified) by checking at most $|\mathcal{J}|(|\mathcal{J}| - 1)/2$ inequalities instead of checking all $2^{|\mathcal{J}|}$ inequalities. Also notice that for a given pair $(j_1,\ j_2)$, the associated inequality (8) is implied by another inequality of the same form unless: (i) $j_1 = a_u$ for some $u \in \mathcal{U}$, and, (ii) $j_2 = b_v$ for some $v \in \mathcal{U}$. We next relate these observations to **(LP′)**.

Corollary 3 *If* $R(i) = \{a_i, a_i + 1 \ldots, b_i\} \subseteq J$ *for all* $i \in I$, *then*

$$Proj_{[x,y]}(T) = \Big\{ [x,y] \in X'' : \sum_{j \in J} y_j = \sum_{i \in I} x_i, \text{ and,} \quad \sum_{j \in \bar{J}} y_j \geq \sum_{i \in T(\bar{J})} x_i \text{ for all } \bar{J} = \{a_k, a_k + 1, \ldots, b_l\} \text{ for } k, l \in I \Big\}.$$

■

We note that Corollary 3 leads to an *implicit* formulation for the *shift scheduling problem* in the presence of *extraordinary overlap*. We discuss the shift scheduling problem in detail later in Section 3.1.

2.2 Special Case II

We next study another special case which is obtained by restricting Assumption A1 by requiring that none of the sets $R(\cdot)$ strongly dominates another.

Assumption 2 (A2): *Assume that for all* $u \in \mathcal{U}$, $R(u) = \{a_u, a_u + 1 \ldots, b_u\}$ *for some* $a_u, b_u \in \mathcal{J}$. *Furthermore, assume that for all* $u, v \in \mathcal{U}$ *either* $a_u \geq a_v$ *or* $b_u \leq b_v$ *holds.*

Note that the second part of the assumption imposes a partial order on $\mathcal{U}$ and without loss of generality we can therefore assume that $a_i \leq a_{i+1}$ and $b_i \leq b_{i+1}$ for all $i = 1, 2, \ldots, |\mathcal{U}| - 1$. If $\mathcal{U}$ does not satisfy this property, it is possible to reorder the nodes, first in decreasing order of a_i and then (for nodes with the same a_i,) in decreasing order of b_i. Ties can be broken arbitrarily. The new ordering is unique up to a permutation of $u \in \mathcal{U}$ with identical $R(u)$.

Based on this observation, we note that under Assumptions A2 the maximum flow can be found using a using a greedy algorithm, which in turn implies that the feasibility of the transportation problem can be tested efficiently, see Hoffman [16] and Dietrich [11]. We discuss this result further in Section 2.3. In addition, we can also show that, it is sufficient to consider only a linear number of inequalities of the form (7) to solve the feasibility problem.

Lemma 4 *Under Assumptions A0 and A2, BFFP is feasible if and only if*

$$\sum_{j \in \bar{\mathcal{J}}} d_j \geq \sum_{u \in T(\bar{\mathcal{J}})} s_u \tag{9}$$

holds for $\bar{\mathcal{J}} = \{1, 2, \ldots, j\}$ *and* $\bar{\mathcal{J}} = \{j, j + 1, \ldots, |\mathcal{J}|\}$ *for all* $j \in \mathcal{J}$.

Proof. The only if part is due to Lemma 2. To show the if part by contradiction, we assume that there exists a set $S \subseteq V^+$ with $cap(S) < \sum_{u \in \mathcal{U}} s_u$ even though (9) holds for all $j \in \mathcal{J}$. Using Lemma 2, there has to be a set $\bar{\mathcal{J}} = \{a, a+1, \ldots, b\}$ for some $a, b \in \mathcal{J}$, $a > 1$ and $b < |\mathcal{J}|$ such that $\sum_{j \in \bar{\mathcal{J}}} d_j < \sum_{u \in T(\bar{\mathcal{J}})} s_u$. Note that $T(\bar{\mathcal{J}}) \neq \emptyset$.

Let $\mathcal{J}' = \{1, 2, \ldots, b\}$ and $\mathcal{J}'' = \{a, a+1, \ldots, |\mathcal{J}|\}$ so that $\bar{\mathcal{J}} = \mathcal{J}' \cap \mathcal{J}''$, and note that $T(\mathcal{J}') \cap T(\mathcal{J}'') = T(\bar{\mathcal{J}})$. Furthermore, if there exists a node $u \in \mathcal{U}$ such that $u \notin T(\mathcal{J}') \cup T(\mathcal{J}'')$ then $a_u < a$ and $b_u > b$, Due to Assumption A2, and the fact that $T(\bar{\mathcal{J}}) \neq \emptyset$, this is not possible, and therefore we have $T(\mathcal{J}') \cup T(\mathcal{J}'') = \mathcal{U}$. See Figure 2.

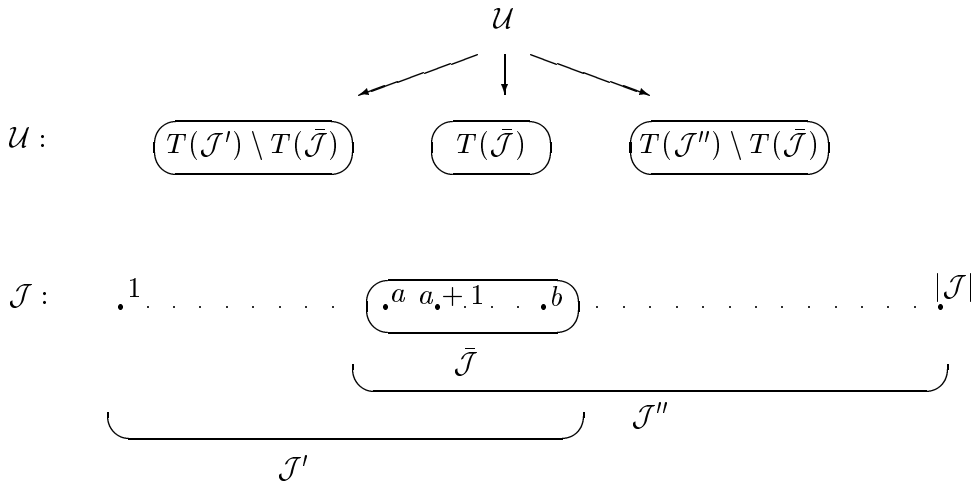

Figure 2: Node sets for the proof of Lemma 4

Next, we combine inequalities (9) associated with $\mathcal{J}'$ and $\mathcal{J}''$:

$$\sum_{j \in \mathcal{J}'} d_j + \sum_{j \in \mathcal{J}''} d_j \geq \sum_{u \in T(\mathcal{J}')} s_u + \sum_{u \in T(\mathcal{J}'')} s_u$$

implying,

$$\sum_{j \in \mathcal{J}} d_j + \sum_{j \in \bar{\mathcal{J}}} d_j \geq \sum_{u \in \mathcal{U}} s_u + \sum_{u \in T(\bar{\mathcal{J}})} s_u$$

which gives the desired contradiction since $\sum_{j \in \mathcal{J}} d_j = \sum_{u \in \mathcal{U}} s_u$ by Assumption A0. ∎

Lemma 4 shows that under Assumptions A0 and A2, BFFP can be solved by checking at most $2|\mathcal{J}| - 1$ inequalities. Also notice that inequality (9) is implied by another inequality of the same form unless: $\bar{\mathcal{J}} = \{1, 2, \ldots, b_u\}$, or, $\bar{\mathcal{J}} = \{a_u, j+1, \ldots, |\mathcal{J}|\}$ for some $u \in \mathcal{U}$. It is

therefore sufficient to consider one inequality for each distinct member of $\{a_u : u \in \mathcal{U}\}$ and one inequality for each distinct member of $\{b_u : u \in \mathcal{U}\}$.

We next relate these observations to **(LP′)**.

Corollary 5 *If $R(i) = \{a_i, a_i + 1 \ldots, b_i\} \subseteq J$ for all $i \in I$, such that $a_i \leq a_{i+1}$, and $b_i \leq b_{i+1}$ for $i < |I|$, then*

$$Proj_{[x,y]}(T) = \Big\{ [x,y] \in X'' : \sum_{j \in J} y_j = \sum_{i \in I} x_i, \text{ and,}$$
$$\sum_{j=1}^{b_k} y_j \geq \sum_{i=1}^{k} x_i \quad \text{for all } k \in I \text{ such that } b_k < b_{k+1}, \text{ and,}$$
$$\sum_{j=a_k}^{|J|} y_j \geq \sum_{i=k}^{|I|} x_i \quad \text{for all } k \in I \text{ such that } a_k > a_{k-1}. \quad \Big\}$$

We also note that Corollary 5 provides a short proof for the validity of a certain formulation of the shift scheduling problem in the absence of extraordinary overlap. We discuss the details in Section 3.1.1.

2.3 Special Case III

The next special case is obtained by generalizing Assumption A2 by allowing a new supply node connected to all of the demand nodes and a new demand node connected to all of the supply nodes. This case essentially models external demand which can absorb surplus production and external supply which can cover for shortages. More precisely:

Assumption 3 (A3): *Assume that for a given special node $u^* \in \mathcal{U}$, $R(u^*) = \mathcal{J}$, and for the remaining nodes $u \in \mathcal{U} \setminus u^*$, $R(u) = \{a_u, a_u + 1 \ldots, b_u, |\mathcal{J}|\}$ for some $a_u, b_u \in \mathcal{J}$. Furthermore, assume that for all $u, v \in \mathcal{U} \setminus \{u^*\}$ either $a_u \geq a_v$ or $b_u \leq b_v$ holds.*

Notice that under Assumptions A0 and A3, BFFP is feasible if and only if

$$\bar{\beta} = \max \Big\{ 0, \sum_{u \in \mathcal{U} \setminus u^*} s_u - d_{|\mathcal{J}|} \Big\}$$

units of flow can be transported from $\bar{\mathcal{U}} = \mathcal{U} \setminus \{u^*\}$ to $\bar{\mathcal{J}} = \mathcal{J} \setminus \{|\mathcal{J}|\}$. Finding the maximum flow in a bipartite graph under Assumption A2 can be done in linear-time by using the Northwest rule as discussed by Hoffman in [16]. Once again, without loss of generality,

we assume that $\mathcal{U} = \{u_1, u_2, \ldots, u_{|\mathcal{U}|} = u^*\}$ satisfies $a_i \leq a_{i+1}$ and $b_i \leq b_{i+1}$ for all $i = 1, 2, \ldots, |\mathcal{U}| - 2$.

Lemma 6 (Hoffman [16]) *Let N denote the incidence matrix of the transportation problem, where N_{ij} is set to 1 if supply node i is connected to the demand node j and it is set to 0 otherwise. If N does not contain $\Gamma = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ as a sub-matrix, then the maximum flow can be found using a greedy (Northwest rule) algorithm.*

We note that finding a maximum flow in a general bipartite network with n nodes takes $O(n^2 log(n))$ time if the network is sparse and $O(n^3)$ time if the network is dense, see Ahuja et al. [2]. We also note that the greedy algorithm can be considered as an efficient specialization of the maximum cardinality matching algorithm for *convex* bipartite graphs presented in Glover [15]. A bipartite graph is called convex if it satisfies Assumption A1.

We can specialize this result to our setting as follows:

Corollary 7 *Under Assumption A2, the incidence matrix can not contain Γ as a sub-matrix and therefore, the maximum flow from $\bar{\mathcal{U}}$ to $\bar{\mathcal{J}}$ can be computed in linear time ($O(|\bar{\mathcal{U}}| + |\bar{\mathcal{J}}|)$) using the following greedy algorithm.*

Algorithm Max-Flow:

STEP 0: **Set** $\bar{s}_i \leftarrow s_{u_i}$, **and** $\bar{d}_j \leftarrow d_j$, **for all** $u_i \in \bar{\mathcal{U}}$ **and** $j \in \bar{\mathcal{J}}$
Set $i \leftarrow 1$, $j \leftarrow a_i$

STEP 1: **Set** $f_{ij} \leftarrow \min\{\bar{s}_i, \bar{d}_j\}$
Update $\bar{s}_i \leftarrow \bar{s}_i - f_{ij}$ **and** $\bar{d}_j \leftarrow \bar{d}_j - f_{ij}$

STEP 2: **IF** $\bar{s}_i = 0$, **THEN** $i \leftarrow i + 1$
IF $\bar{d}_j = 0$, **THEN** $j \leftarrow j + 1$
IF $i > |\bar{\mathcal{U}}|$ **OR** $j > |\bar{\mathcal{J}}|$ **THEN Go To Step 6**

STEP 3: **IF** $j < a_i$ **THEN** $j \leftarrow a_i$

STEP 4: **WHILE** $j > b_i$, **DO**
$i \leftarrow i + 1$
IF $i > |\bar{\mathcal{U}}|$ **THEN Go To Step 6**

STEP 5: **Go To Step 1**

STEP 6: **OUTPUT** $\sum_{i=1}^{|\bar{\mathcal{U}}|} (s_i - \bar{s}_i)$

Even though solving this maximum flow and the associated minimum cut problem is not very difficult, it is not possible to produce a small list of of important cuts that would contain a minimum cut. For the sake of completeness, we next present a linear-time algorithm that finds the minimum cut.

Corollary 8 *Under Assumption A2, a (minimum) cut with capacity equal to the maximum flow value from $\bar{\mathcal{U}}$ to $\bar{\mathcal{J}}$ can be constructed in linear time ($O(|\bar{\mathcal{U}}| + |\bar{\mathcal{J}}|)$) using the following algorithm.*

Algorithm Find-Cut:

STEP 0: **Execute algorithm Max-Flow**
Set $k \leftarrow 0$, $last_i \leftarrow |\bar{\mathcal{U}}|$, $last_j \leftarrow |\bar{\mathcal{J}}|$

STEP 1: **While** $\bar{s}_{last\,i} = 0$ **DO**
$last_i \leftarrow last_i - 1$
IF $last_i = 0$ **THEN Go To STEP 8**

STEP 2: **Set** $k \leftarrow k + 1$, $i'_k \leftarrow last_i$, **and,** $\bar{b}_k \leftarrow b_{i'_k}$

STEP 3: **While** $\bar{d}_{last\,j} = 0$ **DO**
$last_j \leftarrow last_j - 1$
IF $last_j = 0$ **THEN Go To STEP 4**

STEP 4: **Set** $j_k \leftarrow last_j$

STEP 5: **While** $a_{last_i} > j_k$ **DO**
$last_i \leftarrow last_i - 1$
IF $last_i = 0$ **THEN Go To STEP 6**

STEP 6: **Set** $i''_k \leftarrow last_i + 1$, **and,** $\bar{a}_k \leftarrow a_{i''_k}$
Set $\bar{\mathcal{J}}^k \leftarrow \{\bar{a}_k, \bar{a}_k + 1, \ldots, \bar{b}_k\}$ **and** $\bar{\mathcal{U}}^k \leftarrow \{u_{i''_k}, \ldots, u_{i'_k}\}$

STEP 7: **IF** $last_i > 0$ **THEN Go To STEP 1**

STEP 8: **RETURN** $S = \{o\} \cup \bigcup_{t=1}^{k} (\bar{\mathcal{U}}^t \cup \bar{\mathcal{J}}^t)$

Alternate proofs of Corollary 7 and 8 can also be found in an earlier version of this paper [8]. We next relate these observations to BFFP and **(LP$'$)**.

Corollary 9 *Under Assumptions A0 and A3, BFFP can be solved in linear-time. Furthermore, if it is infeasible, a violated inequality of the form (7) can be found in linear time.*

Corollary 10 *If $R(|I|) = J$ and $R(i) = \{a_i, a_i + 1 \ldots, b_i,\ |J|\} \subseteq J$ for all $i \in I \setminus \{|I|\}$, such that $a_i \leq a_{i+1}$, and $b_i \leq b_{i+1}$ for all $i < |I| - 2$, then the separation problem over $Proj_{[x,y]}(T)$ can be solved in linear-time.*

In other words, given a point $p' = [x', y'] \in X''$ with $\sum_{j \in J} y'_j = \sum_{i \in I} x'_i$, if $p' \notin Proj_{[x,y]}(T)$ then a set $\bar{J} \subset J$ that gives a violated valid inequality

$$\sum_{j \in \bar{J}} y_j \quad \geq \quad \sum_{i \in T(\bar{J})} x_i$$

can be identified in linear time.

3 Workforce Scheduling Applications

Workforce staffing and scheduling problems deal with determining the size and schedule of the workforce necessary to satisfy the labor needs of an organization. The objective typically involves minimization of a combination of labor cost and cost of unsatisfied demand. We refer the reader to Grinold and Marshall [14] for a variety of these problems (also see, Burgess and Busby [7]). In this section we describe how to apply our results to two such problems, namely, *the shift scheduling problem* and *the tour scheduling problem*. The shift scheduling problem is solved to determine daily work schedules for the workforce. The tour scheduling problem is solved to combine daily shifts with days-off to determine weekly schedules. The detailed requirements for a daily shift and a weekly tour are usually specified in labor contracts (see [3], for example).

3.1 Daily shift scheduling problem

We next briefly describe the shift scheduling model presented in Bechtold and Jacobs [5] and [6]. The model assumes that a collection of shift types I together with labor requirement d_j for each (discrete) time period $j \in [1, P]$ are given. Here we use $[a, b]$ to denote $\{a, a+1, \ldots, b\}$ for $a, b \in Z$. For each shift type $i \in I$ shift start time s_i, shift end time f_i and a window of possible break periods $R_i = [a_i, b_i]$ is provided. It is required that each worker is assigned to a break period during which time she rests. The objective is to determine the number of workers that should be assigned to each shift type so as to minimize a linear cost function.

Let x_i denote the number of workers assigned to shift type $i \in I$, and w_{ij} denote the number of them assigned to break period $j \in R_i$. Furthermore, let y_j denote the total number of

workers that take a break at period $j \in J = \cup_{i \in I}\{a_i, \ldots, b_i\}$. The shift scheduling problem can be (explicitly) formulated as follows:

ESS-IP: Min $\sum_{i \in I} c_i \; x_i$

Subject to:

$$\sum_{i:j \in [s_i, f_i]} x_i - y_j \geq d_j \qquad for\ all\ j \in [1, P] \tag{10}$$

$$x_i = \sum_{j \in R_i} w_{ij} \qquad for\ all\ i \in I \tag{11}$$

$$y_j = \sum_{i:j \in R_i} w_{ij} \qquad for\ all\ j \in J \tag{12}$$

$$x, y, w \geq 0 \quad and\ integer. \tag{13}$$

In this formulation, constraint (10) requires that there are at least d_j workers available at period j, constraint (11) requires that every worker should be assigned to a break window and constraint (12) counts the number of workers taking a break for each time period.

3.1.1 Without extraordinary overlap

Note that, without loss of generality, we can assume the shifts are indexed in such a way that $a_1 \leq a_2, \ldots, \leq a_{|I|}$. If in addition $b_1 \leq b_2, \ldots, \leq b_{|I|}$ holds, then Corollary 5 implies that ESS-IP can be reformulated (to obtain a so-called *implicit* formulation) as follows:

ISS-IP1: Min $\sum_{i \in I} c_i \; x_i$

Subject to:

$$\sum_{i:j \in [s_i, f_i]} x_i - y_j \geq d_j \qquad for\ all\ j \in [1, P]$$

$$\sum_{j \in J} y_j = \sum_{i \in I} x_i$$

$$\sum_{j=1}^{b_k} y_j \geq \sum_{i=1}^{k} x_i \qquad for\ all\ k \in I$$

$$\sum_{j=a_k}^{|J|} y_j \geq \sum_{i=k}^{|I|} x_i \qquad for\ all\ k \in I$$

$$x, y \geq 0 \quad and\ integer.$$

In Bechtold and Jacobs [5], the authors have presented ISS-IP1 and conjectured that in the absence of *extraordinary overlap*, this is a correct reformulation. In a follow-up paper

Bechtold and Jacobs [6], they have indeed proved that ISS-IP1 is correct. Extraordinary overlap refers to the condition when there are two shift types $i_1, i_2 \in I$ with $a_{i_1} > a_{i_2}$ and $b_{i_1} < b_{i_2}$. Notice that this is precisely what Assumption A2 excludes and therefore Corollary 5 gives a short proof of correctness for ISS-IP1.

3.1.2 With extraordinary overlap

If extraordinary overlap is present, Assumption A2 does not hold and consequently ISS-IP1 is not a correct reformulation. In this case, we can use Corollary 3. Note that Assumption A1 is still satisfied since break windows $R(i)$ consist of contiguous time periods for all $i \in I$.

We next present the resulting formulation:

ISS-IP2:

$$\text{Min} \quad \sum_{i \in I} c_i \, x_i$$

Subject to:

$$\sum_{i:j \in [s_i, f_i]} x_i - y_j \geq d_j \qquad for\ all\ j \in [1, P]$$

$$\sum_{j \in J} y_j = \sum_{i \in I} x_i$$

$$\sum_{j \in \bar{J}} y_j \geq \sum_{i \in T(\bar{J})} x_i \quad for\ all\ \bar{J} = \{a_k, a_k + 1, \ldots, b_l\}\ for\ k, l \in I$$

$$x, y \geq 0 \quad and\ integer.$$

An implicit formulation of the shift scheduling problem in the presence of extraordinary overlap is a new result and it might have practical applications.

3.1.3 More complicated shift schedules

Corollaries 3 and 5 can be applied to more complicated shift scheduling formulations that involve overtime and several break periods with different durations. One example is the formulation presented in Thompson [19]. In this formulation a shift is represented by $[s, b^1, b^2, b^3, f]$ where s and f denote the start and finish time of the shift and b^i denotes the start time of the i'th break period. A shift is considered feasible if it satisfies the following requirements: (i) $f - s \in [l_{min}, l_{max}]$, (ii) $b^1 - s \in [h^1_{min}, h^1_{max}]$, (iii) $b^2 - s \in [h^2_{min}, h^2_{max}]$, and, (iv) $b^3 - b^2 \in [h^3_{min}, h^3_{max}]$, where l_{min}, l_{max} and h^i_{min}, h^i_{max}, for $i = 1, 2, 3$, are input parameters. Note that all of these requirements can be modeled in a network flow framework and As-

sumption A2 holds. Therefore, Corollary 5 provides a proof of correctness for the implicit formulation presented in Thompson [19]. For the sake of completeness, we present a version of this formulation in the Appendix.

3.2 Weekly tour scheduling problem

We next briefly describe the tour scheduling model presented in Cezik et al. [9]. In this model, a weekly tour is considered feasible if it satisfies the following requirements: (i) the tour consists of five feasible daily shift schedules and two days off (possibly with side constraints such as requiring one of the days off to fall on a weekend), and (ii) the start time differential between two consecutive daily shifts is within specified bounds. More precisely, for two consecutive working days d and $d+1$, it is required that $s^d + \beta \geq s^{d+1} \geq s^d - \alpha$ where s^d denotes the starting time of the shift in day d and α and β are two specified parameters. In Cezik et al. [9], the collection of tours are also required to be *cyclic* so that they can be used in consecutive weeks without violating requirement (ii). The formulation presented in Cezik et al. [9] handles this requirement the same way requirement (ii) is handled. To keep the presentation simple, we skip this requirement and note that our approach applies to cyclic tours as well.

The formulation consist of two separate parts linked together by a small number of coupling constraints. The first part contains a separate copy of the daily shift scheduling formulation for every day of the week, which guarantees that solutions of the model correspond to a collection of feasible daily shifts schedules. The second part contains a network flow model that makes sure that the resulting shift schedules can be combined to produce feasible weekly tours. In this formulation, $P = \{1, 2, \ldots, n\}$ denotes the time periods in a day, $\bar{P} = \{1, 2, \ldots, \bar{n}\} \subset P$ denotes the set of periods when a feasible shift may start, and $D = \{1, 2, \ldots, 7\}$ denotes the days of the week where day 6 and 7 correspond to the weekend. Finally, for a shift that starts at period p, $R(p) = \{q \in \bar{P} \,:\, p + \beta \geq q \geq p - \alpha\}$ denotes the set of acceptable shift start times for the next day, where β and α are input parameters.

We next present the integer programming formulation for the weekly tour scheduling problem:

ETS-IP: Min $z = \sum_{d \in D} c_d^1 \, x(d) + c_d^2 \, s(d)$

Subject to:

$$Ax(d) + Gs(d) \geq b(d) \qquad for\ all\ d \in D \tag{14}$$

$$x_{d,p} = w_{d,p,0} + w_{d,p,1} + w_{d,p,2} \qquad for\ all\ \ d \in D,\ p \in \bar{P} \tag{15}$$

$$w_{d,p,i} = k_{d,p,i} + \sum_{q\,:\,p\in R(q)} v_{d-1,q,i,p} \quad \textit{for all } w_{d,p,i} \textit{ with } d \geq 2 \tag{16}$$

$$w_{d,p,i} = h_{d,p,i} + \sum_{q\in R(p)} v_{d,p,i,q} \quad \textit{for all } w_{d,p,i} \textit{ with } d \leq 6 \tag{17}$$

$$l_{d,i} = r_{d-1,i-1} + \sum_{p=1}^{\bar{n}} h_{d-1,p,i-1} \quad \textit{for all } l_{d,i} \textit{ with } d \geq 2 \tag{18}$$

$$l_{d,i} = r_{d,i} + \sum_{p=1}^{\bar{n}} k_{d+1,p,i} \quad \textit{for all } l_{d,i} \textit{ with } d \leq 6 \tag{19}$$

$$s,\ w,\ v,\ h,\ k,\ r,\ l \geq 0 \quad \textit{and } s, w \textit{ integer.}$$

The variables of this formulation are:

Variable name		**Denotes the number of workers that**
$x_{d,p}$	:	start a shift at period $p \in \bar{P}$ on day $d \in D$.
$w_{d,p,i}$	:	have taken i days off prior to starting a shift at period p on day d; defined for all $d \in D$, $p \in \bar{P}$, and $i = 0, 1, 2$ such that $5 \geq d - i \geq 1$
$l_{d,i}$	:	take day d off as their i'th day-off; defined for all $d \in D$ and $i = 1, 2$ such that $5 \geq d - i \geq 0$.
$v_{d,p,i,q}$	:	have taken i days off prior to starting their shift at p on day d and start their shift at period q the next day.
$h_{d,p,i}$	:	have taken i days off prior to starting their shift at period p on day d and take the next day off.
$k_{d,p,i}$	:	have taken the previous day off as their i'th day off and start a shift at the period p on day d.
$r_{d,i}$	:	have taken day d off as their i'th day off and take the next day off as well.

The details of the daily models are hidden in constraints (14) (see the Appendix) where $x(d)$ denotes $[x_{d,1},\ x_{d,2},\ \ldots, x_{d,\bar{n}}]$ and $s(d)$ contains all of the remaining variables of the daily model. Constraints (15) link the daily models with the network model.

The underlying network model has seven stages, one for each day of the week, and in each stage nodes are divided into five groups. The first two groups consist of a single node each and represent workers that take the day off as their first or second day off. The remaining three groups represent workers that work on that day and have previously taken 0, 1 or 2 days off. Each group that represents a working day contains $|\bar{P}|$ nodes, one for each time

period when a feasible shift can start. In this network, edges connect nodes that belong to consecutive days, consistent groups and with an acceptable start time differential as defined by (ii).

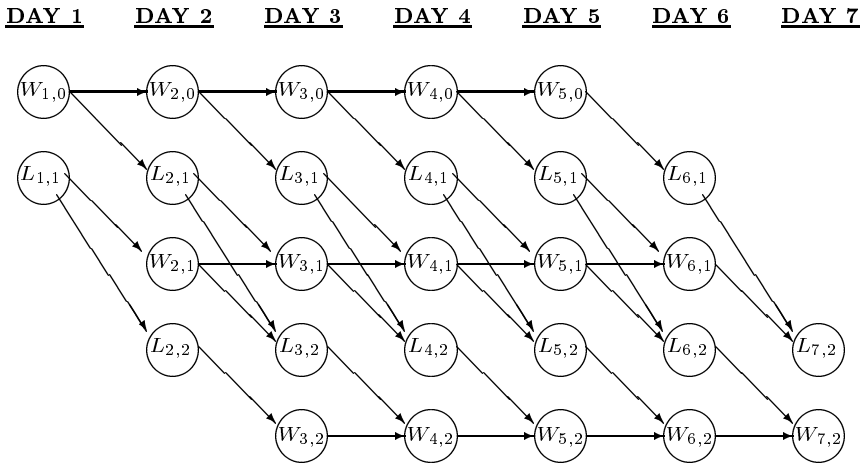

Figure 3: An aggregate view of the weekly network

In Figure 3, we display the corresponding network by contracting each group into a single node. Nodes that represent days-off are labeled L and groups of nodes that represent a working day are labeled W. The first index of a node represents the day of the week, and the second index represents the days off count. In this network, every path from day 1 to day 7 corresponds to a sequence of daily shifts and days off that satisfy requirement (i).

Notice that based on day d and days off count i, edges of this network can be partitioned into groups that induce bipartite graphs. For example, all outgoing edges from nodes $L_{2,1}$ and $W_{2,1}$ go to nodes $W_{3,1}$ and $L_{3,2}$ and all incoming edges to nodes $W_{3,1}$ and $L_{3,2}$ come from nodes $L_{2,1}$ and $W_{2,1}$. This observation leads to two new (implicit) formulations for ETS-IP.

3.2.1 Reformulation based on Assumption A2

The first reformulation is obtained by treating constraints (16) and (17) as "transportation" type constraints and treating the remaining constraints (14), (15), (18), and (19) as "production" type constraints. To eliminate the flow variables $v_{d,p,i,q}$, we construct a bipartite graph $B^1_{[d,i]} = (U^1_{[d,i]}, V^1_{[d,i]}, E^1_{[d,i]})$ for all $d \in \{1, 2, \ldots, 6\}$ and $i \in \{0, 1, 2\}$ that satisfy $5 \geq d - i \geq 1$. For every $p \in \bar{P}$, node set $U^1_{[d,i]}$ contains a node u_p with supply $(w_{d,p,i} - h_{d,p,i})$ and node set $V^1_{[d,i]}$ contains a node v_p with demand $(w_{d+1,p,i} - k_{d+1,p,i})$. The edge set $E^1_{[d,i]}$ contains a

directed edge from $u_p \in U^1_{[d,i]}$ to $v_q \in V^1_{[d,i]}$ if $q \in R(p)$. In this construction, directed edge $(u_p, v_q) \in E^1_{[d,i]}$ corresponds to the flow variable $v_{d,p,i,q}$.

Clearly, all $B^1_{[d,i]}$ satisfy Assumption A2, and consequently we can project out all $v_{d,p,i,q}$ variables from ETS-IP by applying Corollary 5 repeatedly. The resulting (implicit) formulation of the weekly tour scheduling problem is:

ITS-IP1: Min $z = \sum_{d \in D} c^1_d\, x(d) + c^2_d\, s(d)$

Subject to:

$$\textit{inequalities} \quad (14),\ (15),\ (18),\ \textit{and}\ (19)$$

$$\sum_{p \in \bar{P}} w_{d,p,i} - l_{d+1,i+1} - \Big[\sum_{p \in \bar{P}} w_{d+1,p,i} - l_{d,i}\Big] = 0 \quad \textit{for all}\ \ i \textit{ and } d \le 6 \tag{20}$$

$$w_{d,p,i} - h_{d,p,i} \ge 0 \quad \textit{for all}\ \ w_{d,p,i} \ :\ i < 2 \tag{21}$$

$$w_{d,p,i} - k_{d,p,i} \ge 0 \quad \textit{for all}\ \ w_{d,p,i} \ :\ i > 0 \tag{22}$$

$$\sum_{t=1}^{\min\{\bar{n},k+\beta\}} [w_{d+1,t,i} - k_{d+1,t,i}] - \sum_{t=1}^{p} [w_{d,t,i} - h_{d,t,i}] \ge 0 \quad \textit{for all}\ \ w_{d,p,i} \tag{23}$$

$$\sum_{t=\max\{1,k-\alpha\}}^{\bar{n}} [w_{d+1,t,i} - k_{d+1,t,i}] - \sum_{t=p}^{\bar{n}} [w_{d,t,i} - h_{d,t,i}] \ge 0 \quad \textit{for all}\ \ w_{d,p,i} \tag{24}$$

$$s,\ w,\ h,\ k,\ r,\ l \ge 0 \quad \textit{and } s, w \textit{ integer.}$$

In this formulation, constraint (20) requires that total supply should be equal to total demand; constraints (21) and (22) require that supply and demand should be non-negative; and constraints (23) and (24) are from Corollary 5.

3.2.2 Reformulation based on Assumption A3

The second reformulation is obtained by treating constraints (14) and (15) as "production" type constraints and treating all remaining constraints (16), (17), (18), and (19) as "transportation" type constraints. To eliminate all of the the flow variables $h_{d,p,i}$, $k_{d,p,i}$, $r_{d,i}$ and $v_{d,p,i,q}$, we construct a bipartite graph $B^2_{[d,i]} = (U^2_{[d,i]}, V^2_{[d,i]}, E^2_{[d,i]})$ for all $d \in \{1, 2, \ldots, 6\}$ and $i \in \{0, 1, 2\}$ that satisfy $6 \ge d - i \ge 0$. For every $p \in \bar{P}$, if $d - i \ge 1$, node set $U^2_{[d,i]}$ contains a node u_p with supply $w_{d,p,i}$ and if $5 \ge d - i$, node set $V^2_{[d,i]}$ contains a node v_p with demand $w_{d+1,p,i}$. Furthermore, $U^2_{[d,i]}$ contains a node $u_{\bar{n}+1}$ with supply $l_{d,i}$ and $V^2_{[d,i]}$ contains a node

$v_{\bar{n}+1}$ with demand $l_{d+1,i+1}$. The edge set $E^2_{[d,i]}$ contains a directed edge from $u_p \in U^2_{[d,i]}$ to $v_q \in V^2_{[d,i]}$ whenever $p = \bar{n}+1$, or $q = \bar{n}+1$, or $q \in R(p)$.

In this construction, directed edge (u_p, v_q) in $B^2_{[d,i]}$ corresponds to the flow variable $v_{d,p,i,q}$ for $p, q \leq \bar{n}$. Directed edges $(u_{\bar{n}+1}, v_q)$ represent flow variables $k_{d+1,p,i}$ and edges $(u_p, v_{\bar{n}+1})$ represent flow variables $h_{d,p,i}$. Finally, edge $(u_{\bar{n}+1}, v_{\bar{n}+1})$ represents variable $r_{d,i}$.

It is easy to see that all $B^2_{[d,i]}$ satisfy Assumption A3, and therefore variables $h_{d,p,i}$, $k_{d,p,i}$, $r_{d,i}$ and $v_{d,p,i,q}$ can be projected out from ETS-IP by applying Corollary 10 repeatedly. The resulting formulation is:

ITS-IP2: Min $z = \sum_{d \in D} c^1_d\, x(d) + c^2_d\, s(d)$

Subject to:

$$\textit{inequalities} \quad (14), \ \textit{and} \ (15)$$

$$\sum_{p \in \bar{P}} w_{d,p,i} + l_{d,i} - \Big[\sum_{p \in \bar{P}} w_{d+1,p,i} + l_{d+1,i+1} \Big] = 0 \quad \textit{for all} \ \ i \ \textit{and} \ d \leq 6 \tag{25}$$

$$l_{d+1,i+1} + \sum_{q \in Q} w_{d+1,q,i} \geq \sum_{p : R(p) \subseteq Q} w_{d,p,i} \quad \textit{for all} \ i, \ d, \ \textit{and} \ \ Q \subseteq \bar{P} \tag{26}$$

$$w, \ l \geq 0 \quad \textit{and} \ w, \ s \ \textit{integer}.$$

Even though ITS-IP2 has an exponential number of constrains, Corollary 10 implies that these constraints can be separated very efficiently in a *cutting-plane* framework.

4 Computational Experience

In this section we investigate the computational performance of the weekly tour scheduling formulations described in Section 3. For the computational study, we use the same data sets and model parameters used in Cezik et al. [9]. This data was originally obtained from call centers of a large telecommunications company.

To compare the formulations (ETS-IP, ITS-IP1 and ITS-IP2) and some variations on them, we report the problem size together with the computational effort required to solve the linear programming relaxations. We think that this gives a fair basis for comparison since any LP based enumeration method to solve these IP's would require solving many LP's.

This approach also gives a more robust comparison as it is not effected by the heuristic strategies necessary to solve an integer program such as branching, node selection, diving etc. Furthermore, implementing an algorithm to solve the integer programs requires significantly more effort and is beyond the scope of this study.

4.1 Problem Data

The data consists of 15 data sets equally divided into three groups with comparable labor requirements: the first group needs approximately 65 workers to satisfy the staffing requirements without overtime; the second and third group require approximately 200 and 1000 workers respectively. Each data set consists of staffing requirements for every 15 minute period for a 19 hour day (5am-midnight) in a seven day week.

The daily models have a regular shift length of 9 hours plus an extra hour of possible overtime. Every shift has three relief periods of length 15, 30 and 15 minutes respectively. Using the notation presented in the Appendix, the parameters for the daily models are: $l_{min} = 36$, $l_{max} = 40$, $t^1 = 1$, $t^2 = 2$, $t^3 = 1$, $h^1_{min} = 6$, $h^1_{max} = 8$, $h^2_{min} = 18$, $h^2_{max} = 20$, $h^3_{min} = 6$, and $h^3_{max} = 8$.

For the weekly models, any two days can be given off to agents. The shift starting time differential can be at most 90 minutes in consecutive working days, implying $R(p) = \{q \in P : p - 6 \leq q \leq p + 6\}$, and therefore, $|R(p)| = 13$ for most time periods $p \in P$. We note that the size of ETS-IP strongly depends on $|R(p)|$.

The objective is to minimize the labor cost while satisfying the demand without shortage. The cost of overtime is assumed to be twice the regular pay.

4.2 Original Formulation ETS-IP

In the original formulation ETS-IP, we also included constraints of type (16)-(19) to connect day 7 to day 1 in order to produce cyclic schedules. As discussed in Cezik et al. [9], it might be useful to produce cyclic schedules so that the same collection of tours can be used in consecutive weeks within a given season. For the daily models, we used the implicit formulation described in the Appendix. In Table 1, we present the number of variables in this formulation for the instances described in Section 4.1.

As seen in Table 1, the formulation has a total of 9,410 variables under realistic modeling assumptions. It has 4,425 constraints and 142,684 non-zeros. Approximately 15% of the

	7 daily models		Weekly network flow model					
Variable Name	$x,\ y$	z^1, z^2, z^3	w	l	v	h	k	r
No. of variables for sample	584	917	615	12	6383	451	451	7
Total	1491		7919					

Table 1: Number of variables in the original formulation ETS-IP

variables, 75% of the constraints and 90% of the non-zeros come from the daily models. Even though the size of the formulation is not prohibitively large, it is nonetheless large enough to pose a computational challenge for the integer program. Solving the LP relaxation of the formulation takes, on the average, over a minute on a Sun 3000 Enterprise server using Cplex 6.0. Solving the same formulation takes approximately four times longer on a Sun-Sparc 20 using Cplex version 4.0

4.3 New Formulations

We implemented the following reformulations of the original formulation ETS-IP:

ITS-IP1: This formulation is obtained by projecting out v variables from ETS-IP as described in Section 3.2.1.

ITS-IP1 w/Cuts: This formulation is obtained by using constraints (21)-(24) of formulation ITS-IP1 as cutting planes. Initially the formulation only consists of the remaining constraints: (14), (15), (18), (19) and (20). Based on the solution of this relaxation, violated inequalities of type (21)-(24) are added to the formulation in a cutting plane fashion. Until the solution is feasible for the full formulation, the cutting plane algorithm does the following for all (21)-(24): for all d and i, a violated inequality with the smallest p is searched by starting with $p = 1$ and iteratively incrementing p. If a violated inequality is identified for $p = \bar{p}$, it is added to the formulation and the sequential search is restarted with $\bar{p} + 10$ for the same d and i.

ITS-IP2 w/Cuts: This formulation is obtained by projecting out h, k, r and v variables from ETS-IP as described in Section 3.2.2. The initial formulation consists of inequalities: (14), (15), and (25). Based on the solution of this relaxation, violated inequalities of type (26) are added to the formulation in a cutting plane fashion. At every iteration of the

cutting plane algorithm, the most violated inequality is identified for every i and d using the separation algorithm described in Section 2.3. At most one violated inequality is added to the formulation for each i and d.

ITS-IP2 w/Cuts2: This formulation is identical to ITS-IP2 w/Cuts, except some inequalities of type (14) are also treated as cutting planes. More precisely, inequalities (28)-(30) of the daily models are not included in the initial formulation and only the violated ones are added, iteratively, in a cutting plane fashion. Separation algorithm is similar to the sequential approach used for (21)-(24) for ITS-IP1 w/Cuts.

4.4 Computational Results

We solved the LP relaxations of the original formulation ETS-IP together with the four reformulations for all 15 instances. We recorded the CPU time together with the following statistics for the LP's: number of variables, number of constraints and number of non-zero entries in the constraint matrix. For the formulations that use cutting planes, we recorded both initial and final problem size. In Table 2, we report a summary of these statistics. Entries of this table for final problem size and CPU time are obtained by taking the average over the 15 instances. CPU time is measured on a Sun 3000 Enterprise server using Cplex version 6.0 as the LP solver. CPU time is reported in seconds.

	variables	**constraints**		**non-zeros**		**CPU time**
Formulation	—	initial	final	initial	final	—
ETS-IP	9410	4425	-	143,762	-	74.1
ITS-IP1	3027	5180	-	217,590	-	46.0
ITS-IP1 w/Cuts	3027	4114	4211	131,927	138,284	36.9
ITS-IP2 w/Cuts	2118	3188	3264	129,178	132,435	26.9
ITS-IP2 w/Cuts 2	2118	864	1748	27,538	64,898	24.4

Table 2: Computational performance of the weekly formulations

As seen in Table 2, solving formulation ITS-IP1 is significantly faster than solving the original formulation ETS-IP. Note that this speed-up is achieved solely by the reduction in the number of variables even though both the number of constraints and the number of non-zeroes in the formulation increase. When the constraints of ITS-IP1 are used as cutting planes, the LP can

be solved twice as fast and the size of the final formulation is smaller than the original one not only in terms of the number of variables, but also in terms of the number of constraints and the number of non-zeroes. Typically, it takes 10 rounds of cutting plane generation to achieve feasibility.

Formulation ITS-IP2 w/Cuts reduces the solution times further and it produces a smaller LP. We note that the computation time can possibly be improved even further by using a more sophisticated separation algorithm (ex: adding more than one violated inequality for every d and i per iteration). This might in turn reduce the number of necessary cutting plane iterations and improve the computational performance. In our implementation, it typically takes 20 rounds of cutting plane generation to achieve feasibility. Finally, by using constraints of the daily models as cutting planes, formulation ITS-IP2 w/Cuts 2, decreases the size of the final LP drastically and reduces the computation time by another 10%.

We would like to emphasize the difference in the number of non-zeroes between the original formulation ETS-IP and formulation ITS-IP1. This significant increase is not very surprising since inequalities (9) are very dense inequalities. When they are used as cutting planes (i.e. formulation ITS-IP1 w/Cuts), only 10% of these inequalities are needed to achieve feasibility. The resulting total CPU time is 20% less than that of formulation ITS-IP1. A similar comparison can be made between ITS-IP2 w/Cuts and ITS-IP2 w/Cuts 2.

5 Conclusion

In this paper we have studied how to project out flow variables from linear programming models that contain transportation constraints in their formulation. We also described how to apply the projection idea to workforce scheduling problems.

Depending on the problem instance, our reformulations may have significantly fewer variables and therefore can be solved faster. Furthermore, if the constraints of the reformulation are included in a cutting plane fashion, the final formulation can have significantly fewer non-zeros, and therefore require less memory.

We believe that this approach can also be applied to problems other than workforce scheduling, especially in production planning and supply chain management.

Acknowledgments: We would like to thank Gabor Pataki for his comments on an earlier version of paper and pointing to us relevant work on the bipartite matching problem. We would like to thank Beth Munson, Moshe Rosenwein, Rich Wong and Hanan Luss for fruitful discussions. We also thank Sophia Günlük for her comments on an earlier version of this paper.

Research was partially conducted when both authors were at AT&T Labs.

References

[1] R. K. Ahuja, T. L. Magnanti and J. B. Orlin (1993): Network Flows, Prentice Hall.

[2] R. K. Ahuja, J. B. Orlin, C. Stein, and R. E. Tarjan, "Improved Algorithms for Bipartite Network Flow", *SIAM J. Comput.,* 23 (1994), 906–933.

[3] AT&T Labor Contract, "Agreement between AT&T Corporation and Communication Workers of America", (1998).

[4] E. Balas and W. Pulleyblank, "The Perfectly Matchable Subgraph Polytope of a Bipartite Graph", *Networks,* 13 (1983), 495–516.

[5] S. Bechtold and L. Jacobs, "Implicit optimal modeling of flexible break assignments in optimal shift scheduling", *Management Science,* 36 (1990), 1339-1351.

[6] S. Bechtold and L. Jacobs, "The equivalence of general set-covering and implicit integer programming formulations for shift scheduling ", *Naval Research Logistics,* 34 (1996), 223–249.

[7] W. J. Burgess, and R. E. Busby "Personnel Scheduling" in *Handbook of Industrial Engineering*, G. Salvendy (ed.) Wiley, (1992), pp. 2155–2169.

[8] T. Cezik, and O. Gunluk, " Reformulating Linear Programs with Transportation Constraints – with Applications to Workforce Scheduling", *IBM Research report* RC22311 (W0201-093), T.J. Watson Labs, New York, (2001).

[9] T. Cezik, O. Gunluk and H. Luss, "An Integer Programming Model for the Weekly Tour Scheduling Problem", *Naval Research Logistics* 48 (2001), 607–624.

[10] B. Cook, B. Cunningham, B. Pulleyblank and A. Schrijver, *Combinatorial Optimization*, Wiley, New York, (1998).

[11] B. Dietrich, "Monge sequences, antimatroids and the transportation problem with forbidden arcs", *Linear Algebra and its Applications* 139 (1990),133–145.

[12] L. R. Ford, Jr. and D. R. Fulkerson (1962): Flows in Networks, Princeton University Press.

[13] D. Gale, "A Theorem on Flows in Networks", *Pacific Journal of Mathematics* 7 (1957), 1073–1082.

[14] R. Grinold and K. Marshall, Manpower planning models, North-Holland, New York, (1977).

[15] F. Glover, "Maximum matching in a convex bipartite graph", *Naval Research Logist. Quart.,* 14 (1967), 313–316.

[16] A. Hoffman, "On simple linear programming problems", in Convexity, *Proc. Symposia in Pure Mathematics*, volume 7, pages 317 – 327, (1963), American Mathematical Society.

[17] E. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, (1976).

[18] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization*, Wiley, New York, (1988).

[19] G. Thompson, "Improved implicit modeling of the labor shift scheduling problem", *Management Science,* 41 (1995), 595–607.

Appendix: Formulation of the Shift Scheduling Problem

We next present an integer programming formulation for the daily shift scheduling problem (see Thompson [19]). As discussed in Section 3.1, a shift is represented by $[s, b^1, b^2, b^3, f]$ and it is considered feasible if it satisfies requirements $(i) - (iv)$ of Section 3.1. Using n to denote the number of discrete time periods in a working day, t^i to denote the duration of the i'th break, and r_k to denote the staffing requirement for agents in period k, the formulation becomes:

Min $\quad z = f(x, y, z, u)$

Subject to:

$$\sum_{p=1}^{k} x_p - \sum_{p=1}^{k} y_p - \sum_{i=1}^{3} \sum_{p=0}^{t^i-1} z^i_{k-p} + u_k \geq r_k \qquad for\ all\ k \in [1, n] \tag{27}$$

$$\sum_{p=1}^{k+l_{max}} y_p \geq \sum_{p=1}^{k} x_p \geq \sum_{p=1}^{k+l_{min}} y_p \qquad for\ all\ k \in [1, n - l_{min} + 1] \tag{28}$$

$$\sum_{p=1}^{k+h^i_{max}} z^i_p \geq \sum_{p=1}^{k} x_p \geq \sum_{p=1}^{k+h^i_{min}} z^i_p \qquad for\ i = 1, 2\ and\ k \in [1, n - l_{min} + 1] \tag{29}$$

$$\sum_{p=1}^{k+h^3_{max}} z^3_p \geq \sum_{p=1}^{k} z^2_p \geq \sum_{p=1}^{k+h^3_{min}} z^3_p \qquad for\ all\ k\ such\ that\ z^2_k\ is\ defined \tag{30}$$

$$\sum_{p} x_p = \sum_{p} y_p = \sum_{p} z^1_p = \sum_{p} z^2_p = \sum_{p} z^3_p \tag{31}$$

$$x, y, z^i \geq 0, \quad and\ integer.$$

where the variables of the model and their ranges are:

u_p : unsatisfied demand (or, shortage) at period $p \in [1, n]$,
x_p : number of shifts starting at the beginning of period $p \in [1, n - l_{min} + 1]$,
y_p : number of shifts finishing at the end of period $p - 1$, $p \in [l_{min} + 1, n + 1]$,
z^i_p : number of breaks of type i commencing at the beginning of period p, where
$p \in [h^i_{min} + 1, n - l_{min} + h^i_{max} + 1]$ for $i = 1, 2$, and
$p \in [h^2_{min} + h^3_{min} + 1, n - l_{min} + h^2_{max} + h^3_{max} + 1]$ for $i = 3$.

The objective function $f(\cdot)$ is a linear combination of total number of agents, and total overtime hours and total unsatisfied demand.