Pergamon

Computers Math. Applic. Vol. 33, No. 12, pp. 1–13, 1997

0898-1221/97 $17.00 + 0.00

PII: S0898-1221(97)00090-4

The Pickup and Delivery Problem: Faces and Branch-and-Cut Algorithm

K. S. RULAND
Systems & Electronics, Inc.
201 Evans Lane
St. Louis, MO 63121-1126, U.S.A.

E. Y. RODIN
Department of Systems Science and Mathematics
Washington University in St. Louis
One Brookings Drive, St. Louis, MO 63130, U.S.A.

(Received and accepted April 1997)

Abstract—This paper formulates the pickup and delivery problem, also known as the dial-a-ride problem, as an integer program. Its polyhedral structure is explored and four classes of valid inequalities developed. The results of a branch-and-cut algorithm based on these constraints are presented.

Keywords—Integer programming, Branch-and-cut algorithm, Pickup and delivery problem.

1. INTRODUCTION

The Pickup and Delivery Problem (PDP), in its most basic form, consists of a fleet of vehicles and set of customer requests. Each request specifies an origin and destination location. The vehicles must travel through the locations so that each origin is visited before the corresponding destination. This basic structure is common to all variants of the problem.

The PDP is a constrained (multiple) Traveling Salesman Problem (TSP). PDP routes are actually TSP tours under the additional constraint that origins must precede destinations. Because of this, it is likely that successful solution approaches to the TSP can be modified to solve the PDP.

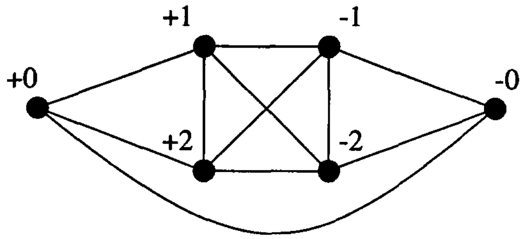
The most successful exact TSP algorithms utilize advanced knowledge of the structure of the TSP polytope. The branch-and-cut algorithm has successfully optimized a 2,392 city problem [1]. In order to apply the branch-and-cut algorithm to the PDP, the polyhedral structure of an integer program model must be examined.

Here the PDP is formulated as an integer program similar to the Dantzig-Fulkerson-Johnson model for the TSP [2]. The polytope defined by this integer program is quite similar to the TSP polytope explored by a number of researchers [3–6].

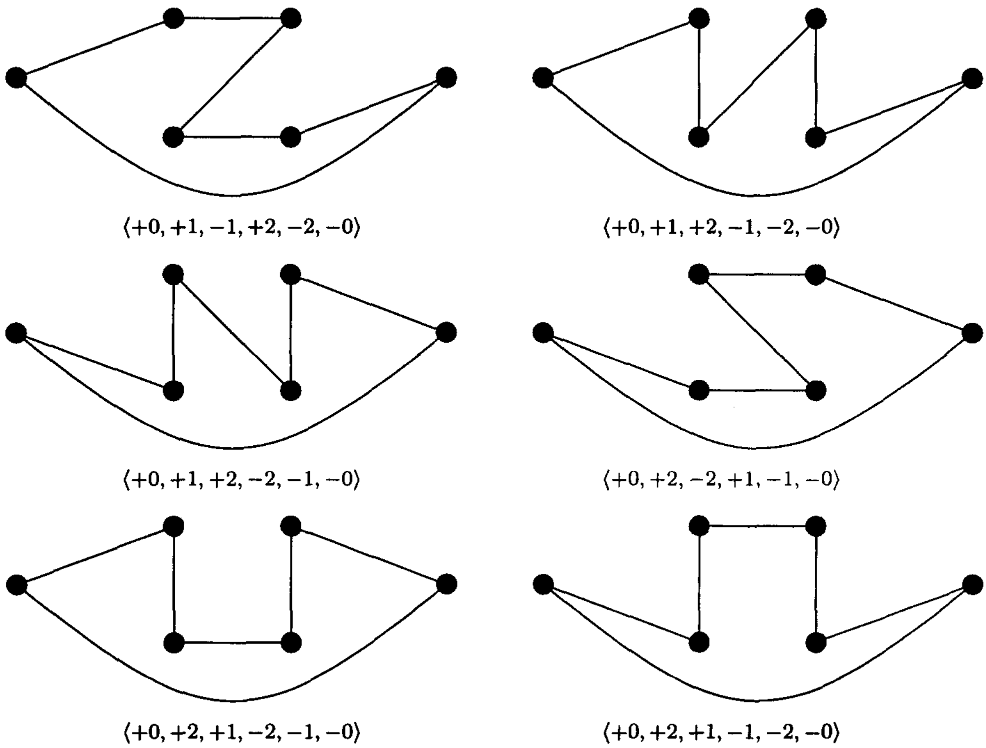
Variants of the basic problem have been approached in a variety of ways including dynamic programming [7,8], nonlinear integer programming [9,10], and column generation [11]. Numerous heuristics have also been examined [12–14]. The recently published survey by Savelsbergh and Sol describes many more approaches in detail [15].

This research was supported by AFOSR Grant F49620-93-1-0012.

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX



(a) Graph of the Pickup and Delivery Problem with two customers.



(b) The six feasible solutions to the two customer problem.

Figure 1. The graph and feasible routes of the Pickup and Delivery Problem with two customers.

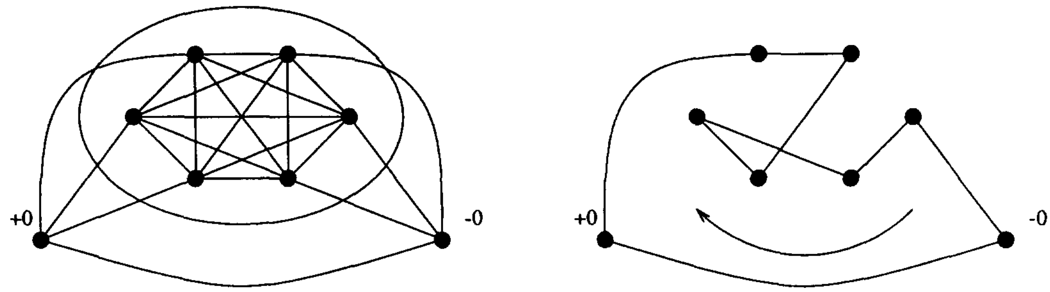
2. THE PICKUP AND DELIVERY PROBLEM

The pickup and delivery problem examined in this paper is given by a single vehicle and a set N of customers. The vehicle has an origin depot, represented by $+0$, and destination depot -0. Associated with each customer i is an origin $+i$, and destination $-i$. The vehicle is to service each customer by picking him up at his origin and delivering him to his destination. The objective is to minimize the total distance traveled by the vehicle in servicing all customers.

The problem is described by a graph $G_N = (V_N, E_N)$ with

$$\begin{aligned} V_N &= \{+0, -0\} \cup \{+i, -i \mid i \in N\}, \\ E_N &= \{(+0, -0)\} \cup \{(+0, +i) \mid i \in N\} \cup \{(-0, -i) \mid i \in N\} \cup E\left(K_{2n}\right), \end{aligned}$$

where $E(K_{2n})$ is the edge set of the complete graph on the customer origins and destinations. See Figure 1a for the graph of the problem on 2 customers and Figure 2a for the graph on n customers. When only the size of the customer set is important, the subscript N is replaced with an integer n. Note $|V_n| = 2n+2$ and $|E_n| = 2n^2+n+1$. Each edge has an associated weight equal to the cost of traveling between the end points. The costs are assumed to be symmetric.



(a) G_n the graph for the Pickup and Delivery Problem with n customers.

(b) A feasible route. The arrow indicates the natural direction.

Figure 2. G_n, the encircled subgraph is the complete graph on the $2n$ customer locations.

Separating the vehicle depot into an origin and destination has two benefits. First, it allows modeling of real-life situations where the depots are not the same physical location. Second, it gives each cycle on the undirected graph an implicit direction—from the origin depot to the destination depot.

A *feasible route* is a collection of edges $R \subset E$ satisfying the following conditions:

(i) $(+0, -0) \in R$,
(ii) $|R \cap \delta(v)| = 2$ for each $v \in V$,
(iii) the graph generated by R is connected, and
(iv) $+i$ is on the path from $+0$ to $-i$ in $R \setminus \{(+0, -0)\}$ for each customer $i \in N$,

where $\delta(v) = \{e \in E \mid e \text{ is incident to } v\}$ is the star of v.

As an example, Figure 1a shows the graph G_2 for the pick up and delivery problem with two customers, while Figure 1b shows the six feasible routes.

Condition (i) gives every feasible route a direction. When the edge $(+0, -0)$ is removed from R, the remaining edges form a Hamiltonian path from $+0$ to -0. This is the natural direction given to any valid PDP route (see Figure 2b).

Conditions (ii) and (iii) are exactly the conditions required of a TSP tour on the graph G_n. R consists of a single Hamiltonian cycle of the vertices V.

Condition (iv) requires that each customer is picked up before he is dropped off. This property of the route is the primary difference between PDP routes and TSP tours and will be referred to as the *precedence condition (for customer i)*.

For G_n, define $\mathcal{R}(G_n)$ to be the collection of all feasible PDP routes and $\mathcal{T}(G_n)$ the collection of all feasible TSP tours. Immediately $\mathcal{R}(G_n) \subset \mathcal{T}(G_n) \subset \mathcal{T}(K_{2n+2})$.

A route in $\mathcal{R}(G_n)$ will be denoted by a sequence of vertices beginning with the origin depot and ending with the destination depot. For $R \in \mathcal{R}(G_n)$,

$$R = \langle +0, (R)_1, (R)_2, \dots, (R)_{2n}, -0 \rangle ,$$

where each $(R)_i$ is a unique customer origin or destination. Figure 1b shows both the graph and sequence for all six feasible routes in G_2.

A *partial route* is a specification of the relative positions of a subset of the vertices. Given $U = \{u_1, u_2, \dots, u_m\} \subset V$ and π a permutation of $\{1, 2 \dots, m\}$, a partial route is represented by

$$\left\langle +0, \dots, u_{\pi(1)}, \dots, u_{\pi(2)}, \dots, u_{\pi(m)}, \dots, -0 \right\rangle .$$

A partial route is *feasible* if no precedence conditions are violated. That is, if for all $i \in N$ with $u_j = +i$ and $u_k = -i$, then u_j appears before u_k, or equivalently, $\pi(j) < \pi(k)$, on the partial route.

A *completion* of a partial route is a route preserving the relative positions of the vertices of the partial route. Every feasible partial route can be completed to a feasible route. A feasible completion can be constructed by inserting the customer origins not in U between $+0$ and $u_{\pi(1)}$ and the destinations not in U between $u_{\pi(m)}$ and -0. There are typically many completions of a feasible partial route.

3. NUMBER OF FEASIBLE ROUTES

THEOREM 3.1. *The number of feasible PDP routes, $k_n = |\mathcal{R}(G_n)|$, is given by the recurrence relationship*

$$k_1 = 1,$$
$$k_n = n(2n-1)k_{n-1},$$

or in closed form,

$$k_n = \frac{(2n)!}{2^n}.$$

PROOF. G_1 has only a single feasible route $\langle +0, +1, -1, -0 \rangle$. To show the inductive equation, suppose $\mathcal{R}(G_{n-1}) = \{R_1, R_2, \dots, R_{k_{n-1}}\}$ is the collection of feasible routes for G_{n-1}. We construct for each route $R \in \mathcal{R}(G_{n-1})$, $n(2n-1)$ different routes in G_n by inserting the n^{th} customer on R. If

$$R = \langle +0, (R)_1, (R)_2, \dots, (R)_{2n-3}, (R)_{2n-2}, -0 \rangle,$$

insert $+n$ on R arbitrarily and insert $-n$ later. That is, $\forall i = 1, \dots, 2n-1$, and $\forall j = i, \dots, 2n-1$, construct $R_{ij} \in \mathcal{R}(G_n)$ as

$$R_{ij} = \langle +0, (R)_1, \dots, (R)_{i-1}, +n, (R)_i, \dots, (R)_{j-1}, -n, (R)_j, \dots, (R)_{2n-2}, -0 \rangle,$$

with the understanding that $(R)_{2n-1}$ represents the final stop at the depot -0, and if $i = j$, $+n$ immediately precedes $-n$. Each R_{ij} constructed in this manner is feasible, so $\{R_{ij}\} \subset \mathcal{R}(G_n)$.

To show the reverse inclusion holds, notice that for any $R \in \mathcal{R}(G_n)$, the removal of $+n$ and $-n$ from the route is a route $R' \in \mathcal{R}(G_{n-1})$, and for appropriate i and j, $R = R'_{ij}$.

In addition, the constructed routes are unique. Suppose R, $R' \in \mathcal{R}(G_{n-1})$, and $1 \le i \le j \le 2n+1$ and $1 \le i' \le j' \le 2n+1$, $R_{ij} = R'_{i'j'}$. Then $(R_{ij})_p = (R'_{i'j'})_p$ for all $p = 1, \dots, 2n$. By the nature of the construction, the removal of $+n$ and $-n$ from R_{ij} and $R'_{i'j'}$ would then yield the same route in $\mathcal{R}(G_{n-1})$. This shows $R = R'$. Now it is quite obvious that $i = i'$ and $j = j'$. Therefore,

$$\begin{aligned} k_n &= \sum_{i=1}^{2n-1} \sum_{j=i}^{2n-1} k_{n-1} \\ &= n(2n-1)k_{n-1}. \end{aligned}$$

We now only need to show the closed form is correct. For $n = 1$, $((2 \cdot 1)!)/2^1 = 1$. Suppose the closed form is valid for some $n \ge 1$,

$$\begin{aligned} k_{n+1} &= (n+1)(2n+1)\frac{(2n)!}{2^n} \\ &= \frac{(2n+2)(2n+1)(2n)!}{2 \cdot 2^n} \\ &= \frac{(2(n+1))!}{2^{n+1}}. \end{aligned}$$ ∎

The number of different traveling salesman tours on K_{2n+1} is known to number $(1/2)(2n)!$. The addition of precedence constraints on the vertices reduces the number of feasible solutions by a factor of $1/(2^{n-1})$.

4. DEFINITION OF THE POLYTOPE

Each feasible route R has an associated characteristic function $x_R\colon E \to \mathbb{R}$ defined by

$$x_R(e) = \begin{cases} 1, & \text{if } e \in R, \\ 0, & \text{otherwise.} \end{cases}$$

The single vehicle, symmetric polytope is the convex hull of all feasible routes in $\mathbb{R}^E$.

$$\text{PDP}\,(G_n) = \text{conv}\left\{x_R \in \mathbb{R}^E \mid R \in \mathcal{R}(G_n)\right\}.$$

The TSP polytope, defined similarly by

$$\text{TSP}(K_n) = \text{conv}\left\{x_T \in \mathbb{R}^E \mid T \in \mathcal{T}(K_n)\right\}$$

has received considerable attention. Numerous families of facets of $\text{TSP}(K_n)$ are known. All these are valid inequalities for PDP.

5. INTEGER PROGRAMMING FORMULATION

The conditions for a collection of edges to define a feasible route are translated into algebraic conditions on characteristic vectors in $\mathbb{R}^E$. Since a route is a collection of edges, if $x \in \mathbb{R}^E$ is the characteristic vector of a route, it is a vertex of the unit cube. That is, $0 \leq x \leq 1$ and is integer.

Conditions (i) and (ii) are equivalent to

$$x((+0,-0)) = 1, \tag{1}$$

$$x(\delta(v)) = 2, \qquad \forall v \in V, \tag{2}$$

respectively.

Conditions (iii) and (iv) are enforced using constraints on cutsets. For $U \subset V$, the cutset $[U\!:\!\bar{U}]$ is the collection of edges connecting U to $\bar{U}$

$$\left[U\!:\!\bar{U}\right] = \left\{(u,u') \in E \mid u \in U \text{ and } u' \in \bar{U}\right\}.$$

Also note,

$$\left[\bar{U}\!:\!\bar{\bar{U}}\right] = \left[\bar{U}\!:\!U\right] = \left[U\!:\!\bar{U}\right].$$

That is, the cutset generated by U and $\bar{U}$ are identical. This ambiguity causes small technical problems when talking about collections of cutsets. In order to uniquely specify a cutset by a vertex set, construct

$$\mathcal{U} = \{U \subset V \mid +0 \in U\}.$$

Now $\forall U,\, U' \in \mathcal{U}$, $U \neq U'$ if and only if $[U\!:\!\bar{U}] \neq [U'\!:\!\bar{U'}]$. In addition, if F is a cutset, then for some $U \in \mathcal{U}$, $F = [U\!:\!\bar{U}]$.

Condition (iii), connectedness, is enforced by the same subtour elimination constraints in the TSP.

THEOREM 5.1. *For all $U \in \mathcal{U}$,*

$$x\left([U:\bar{U}]\right) \geq 2 \tag{3}$$

is a valid inequality.

In addition,

- *if $U = \{+0\}$, $\{+0,-0\}$, or $\overline{\{v\}}$, for some $v \in V$, then $x([U\!:\!\bar{U}]) = 2$ and the constraint is a linear combination of the equality constraints (1) and (2), or*
- *otherwise, the constraint contains an inner point.*

PROOF. The subtour elimination constraints (3) are valid for TSP(K_{2n+2}), in fact, they are facets [16], and since PDP (G_n) $\subset$ TSP(K_{2n+2}), (3) are valid inequalities for PDP (G_n).

To show the second statement, first the degenerate cases are exposed. If $U = \overline{\{v\}}$ for some $v \in V$, then the cutset $[U:\bar{U}] = \delta(v)$, and the subtour elimination constraint is $x(\delta(v)) \geq 2$. Equation (2) requires equality to hold so the subtour elimination constraint is redundant. A similar condition holds if $U = \{+0\}$.

If $U = \{+0, -0\}$,

$$\begin{aligned} 2 \leq x\left(\left[U:\bar{U}\right]\right) &= \sum_{u \in \bar{U}} (x(+0,u) + x(-0,u)) \\ &= x(\delta(+0)) - x(+0,-0) + x(\delta(-0)) - x(+0,-0) \\ &= 2 - 1 + 2 - 1 = 2. \end{aligned}$$

The last equality come from the equality constraints (1) and (2).

Finally, for the second case, construct an inner point—a feasible route satisfying (3) with strict inequality. Since $+0 \in U$, and $U \neq \{+0\}$, there must be another vertex $a \in U$ with $a \neq -0$. And because $\bar{U} \neq \{v\}$, there are two vertices $b, c \in \bar{U}$. Since $a \neq -0$, either b need not precede a, c need not precede a (or both). Therefore, at least one of the two partial routes

$$\langle +0, \ldots, b, \ldots, a, \ldots, c, \ldots, -0 \rangle$$
$$\langle +0, \ldots, c, \ldots, a, \ldots, b, \ldots, -0 \rangle$$

can be completed to a feasible route with $[U:\bar{U}] \geq 4$. The two partial routes represent traversing the route in Figure 3 in either direction. ∎

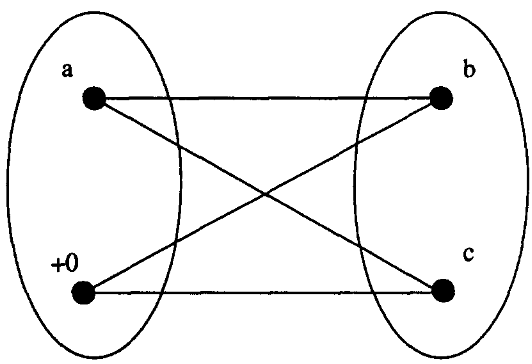


Figure 3. If $a \neq -0$, then one of the two directions of this route is feasible and is an inner point for the subtour elimination constraint.

By simple manipulation of the degree equations (1) and the identity

$$x(V) = x(U) + x\left(\left[U:\bar{U}\right]\right) + x\left(\bar{U}\right),$$

the subtour elimination constraint (3) is equivalent to

$$x(U) \leq |U| - 1.$$

A stronger version of the subtour elimination constraint enforces the precedence condition required by Condition (iv).

THEOREM 5.2. *Let $U \in \mathfrak{U}$ satisfy for some $i \in N$*

$$+0, -i \in U \text{ and } -0, +i \in \bar{U},$$

then

$$x\left(\left[\,U\!:\!\bar{U}\,\right]\right) \geq 4 \tag{4}$$

is a valid inequality for PDP (G_n).

And, for such U, either

- *if $U = \{+0, -i\}$ or $\overline{\{-0, +i\}}$, then $x([\,U\!:\!\bar{U}\,]) = 4$ and is equivalent to a linear combination of (1) and (2), or*
- *the constraint (4) contains an inner point.*

PROOF. We show that if R is a collection of edges satisfying Conditions (i)–(iii) and $x([\,U\!:\!\bar{U}\,]) < 4$ for some U satisfying the hypotheses, then R fails Condition (iv).

Because R satisfies Conditions (ii) and (iii), it is a valid TSP tour and must satisfy the subtour elimination constraint $x([\,U\!:\!\bar{U}\,]) \geq 2$. Because TSP tours are biconnected, every cutset must be even, hence, $x([\,U\!:\!\bar{U}\,]) = 2$. Therefore there can be only one edge (u, v) other than $(+0, -0)$ in the cutset $[\,U\!:\!\bar{U}\,]$, as shown in Figure 4. This means

$$R = \langle +0, \ldots, -i, \ldots, u, v, \ldots, +i, \ldots, -0 \rangle$$

and $-i$ is on the route from $+0$ to $+i$, and the precedence condition for customer i is violated.

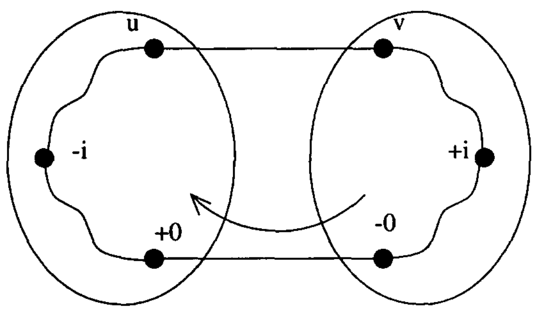


Figure 4. A violated precedence constraint.

To show the second statement, proceed as in the proof for the subtour elimination constraints by showing the degenerate cases, then construct a feasible route for the other. In the first case, $U = \{+0, -i\}$. Because $(+0, -i) \notin E$, $[\,U\!:\!\bar{U}\,] = \delta(+0) \cup \delta(-i)$, and the precedence constraint reduces to

$$x(\delta(+0)) + x(\delta(-i)) = 4.$$

A similar argument shows the precedence constraint with $\bar{U} = \{+i, -0\}$ is also redundant.

Suppose $+0, -1 \in U$ and $-0, +1 \in \bar{U}$. If U does not result in a redundant case, then there is a vertex $a \in U$, $a \neq +0, -1$, and $b \in \bar{U}$, $b \neq -0, +1$.

- Then one of the following partial routes

$$\begin{aligned}
&\langle +0, +1, -1, b, a, \ldots, -0 \rangle, \\
&\langle +0, +1, a, b, -1, \ldots, -0 \rangle, \\
&\langle +0, b, a, +1, -1, \ldots, -0 \rangle,
\end{aligned}$$

as shown in Figure 5, can be completed to a feasible route satisfying $x([U : \bar{U}]) > 4$. ∎

As with the subtour elimination constraints, the precedence constraint (4) is equivalent to

$$x(U) \leq |U| - 2.$$

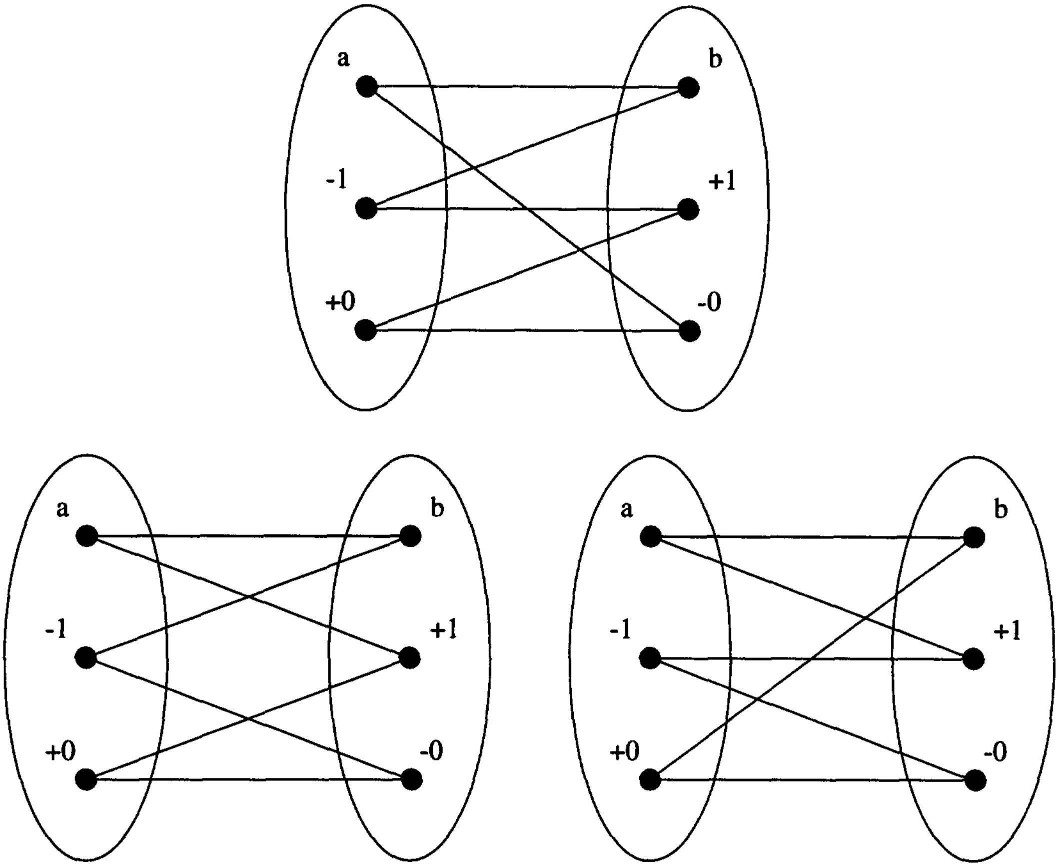


Figure 5. One of these three routes is feasible and an inner point of the precedence constraint.

The node balance equations and subtour elimination constraints fully define the TSP polytope on the integer lattice

$$\mathrm{TSP}(K_{2n+2}) = \left\{x \in \mathbb{Z}^E \mid x(\delta(v)) = 2, x\left(\left[U : \bar{U}\right]\right) \geq 2\right\}.$$

Similarly, the PDP problem is completely defined by

$$\begin{aligned} \mathrm{PDP}\,(G_n) = \big\{x \in \mathbb{Z}^E \mid \ & x(\delta(v)) = 2, \\ & x(+0, -0) = 1, \\ & x\left(\left[U : \bar{U}\right]\right) \geq 2, \\ & x\left(\left[U : \bar{U}\right]\right) \geq 4\big\}. \end{aligned} \tag{5}$$

The precedence constraints are strong enough in themselves to eliminate all TSP tours that are infeasible PDP routes.

6. FACIAL STRUCTURE OF $\mathrm{PDP}\,(G_n)$

This section describes the known facial structure of the pickup and delivery polytope. In addition to showing the subtour and precedence constraints are faces, two other classes of valid constraints are described.

6.1. Subtour Elimination and Precedence Constraints

In order to facilitate discussion about subtour elimination and precedence constraints, partition $\mathcal{U}$ into four families.

$$\begin{aligned}
\mathcal{U}_1 &= \left\{\overline{\{v\}} \mid v \in V, v \neq +0\right\} \cup \{\{+0\}\{+0,-0\}\},\\
\mathcal{U}_2 &= \{\{+0,-i\} \mid i \in N\} \cup \left\{\overline{\{-0,+i\}} \mid i \in N\right\},\\
\mathcal{U}_3 &= \{U \in \mathcal{U} \mid \exists i \in N, \{+0,-i\} \subset U \text{ and } \{-0,+i\} \subset \bar{U}\} \setminus \mathcal{U}_2,\\
\mathcal{U}_4 &= \mathcal{U} \setminus (\mathcal{U}_1 \cup \mathcal{U}_2 \cup \mathcal{U}_3).
\end{aligned}$$

Now to restate the results of Theorems 5.1 and 5.2 more concisely.

THEOREM 6.1. *With $\mathcal{U}$ partitioned as above,*

- *if $U \in \mathcal{U}_1$, $x([U:\bar{U}]) = 2$ and is redundant;*
- *if $U \in \mathcal{U}_2$, $x([U:\bar{U}]) = 4$ and is redundant;*
- *if $U \in \mathcal{U}_3$, $x([U:\bar{U}]) \geq 4$ defines a proper face;*
- *if $U \in \mathcal{U}_4$, $x([U:\bar{U}]) \geq 2$ defines a proper face.*

PROOF. Most of this is a rewording of the results of Theorems 5.1 and 5.2. All that needs to be shown is in the third and fourth cases, the inequality actually defines a face. This is relatively easy to show by constructing a feasible route satisfying the constraint with equality.

Consider $U \in \mathcal{U}_3$, fix i so that $\{+0,-i\} \subset U$ and $\{-0,+i\} \subset \bar{U}$. A feasible route with $x([U:\bar{U}]) = 4$ can be constructed by originating at $+0$, stopping at all customer origins in U, traveling to $+i \in \bar{U}$, stopping at all customer origins in $\bar{U}$. Now that all customers are picked up, drop off customer i at $-i \in U$, continue dropping off all customers with destinations in U, finally drop off all customers with destinations in $\bar{U}$ and terminate at -0. This route traverses the cutset $[U:\bar{U}]$ exactly four times: immediately before the stop at $+i$, immediately before the stop $-i$, back to drop off customers in $\bar{U}$ and finally the edge $(-0,+0)$.

For $U \in \mathcal{U}_4$, the most difficult part is understanding what conditions are required of U. First, U is not a one point set, and $\bar{U}$ is not a one point set. U is not $+0$ and a destination (-0 or $-i$ for some i), and $\bar{U}$ is not -0 and an origin. And most importantly, $-0 \in U$ or, for all $i \in N$, $+i \in U$ or $-i \in \bar{U}$.

Now proceed by cases. If $-0 \in U$, begin the route at $+0 \in U$. First stop at all origins in U, then all origins in $\bar{U}$. Then stop at all destinations in $\bar{U}$, crosses over to U and stop at all destinations before ending at -0. This route crosses the cutset $[U:\bar{U}]$ exactly twice.

If $-0 \in \bar{U}$, then because $U \notin \mathcal{U}_3$, $\forall i \in N$, either $+i \in U$ or $-i \in \bar{U}$ (or both). We construct a route similar to the other case. Begin at $+0 \in U$, stop at all origins in U, then all destinations in U, travel to $\bar{U}$ and stop at all origins, then all destinations. Finally, the route stops at the destination depot $-0 \in \bar{U}$ and returns to $+0$. If $-i \in U$, then $+i$ must also be in U and its precedence condition is satisfied. This route is feasible and crosses the cutset $[U:\bar{U}]$ twice.

6.2. Generalized Order Constraints

The family of order constraints combine the precedence condition for different customers.

THEOREM 6.2. (GENERALIZED ORDER CONSTRAINTS). *Let $U_1, \ldots, U_m \subset V$ be mutually disjoint subsets, which for some collection of customers $i_1, \ldots, i_m \in N$ satisfy*

$$+0,-0 \notin U_l, \qquad l = 1,\ldots,m, \tag{6}$$

$$+i_l, -i_{l+1} \in U_l, \qquad l = 1,\ldots,m. \tag{7}$$

Then

$$\sum_{l=1}^{m} x(U_l) \leq \sum_{l=1}^{m} |U_l| - m - 1 \tag{8}$$

is a valid inequality.

REMARK 6.3. An order constraint with fixed m is referred to as an m-order constraint.

PROOF. Let $R \subset E$ satisfy Conditions (i)–(iii) and fail (8) for $U_1, \dots, U_m$ and $i_l = l$. Because subtours are valid, for each U_l,

$$x(U_l) \le |U_l| - 1. \tag{9}$$

Summing these gives

$$\sum_{l=1}^{m} x(U_l) \le \sum_{l=0}^{m} |U_l| - m. \tag{10}$$

However by assumption,

$$\sum_{l=1}^{m} x(U_l) > \sum_{l=1}^{m} |U_l| - m - 1. \tag{11}$$

Hence, equality holds in (10), which in turn implies equality in each of (9).

So, the stops in each U_l are visited sequentially. The route R enters and exits each subset of stops U_l exactly once, as is shown in Figure 6. Define π a permutation of $(1, \dots, m)$ by the sequence of $U_1, \dots, U_m$ on R. R is a completion of the partial route

$$\langle +0, U_{\pi(1)}, U_{\pi(2)}, \dots, U_{\pi(m)}, -0 \rangle .$$

Regardless of the permutation, some precedence constraint will be violated. ∎

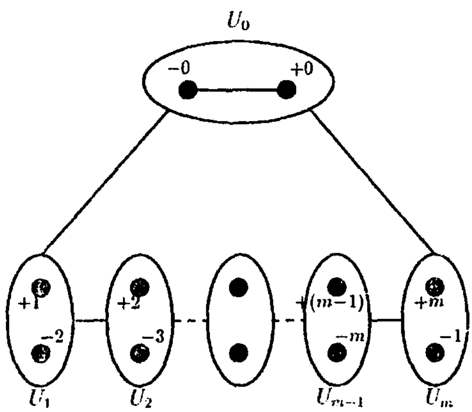


Figure 6. A violated generalized order constraint.

6.3. Order Matching Constraints

The trivial constraint

$$x(+i, -i) + x(+i, +j) + x(+j, -j) \le 2$$

shown in Figure 7, combines the precedence conditions of two customers. This constraint is generalized by replacing the edge $(+i, +j)$ by an arbitrary clique as in Figure 8.

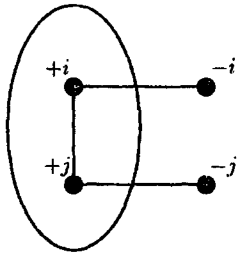


Figure 7. The simplest order matching constraint. Regardless of direction of travel, a precedence constraint is violated.

THEOREM 6.4. (ORDER MATCHING CONSTRAINT). *For any pair* $i, j \in N$, *and arbitrary collection of customer stops* $\{+i,+j\} \subset W \subset V \setminus \{-i,-j,\pm 0\}$

$$x(W) + x(+i,-i) + x(+j,-j) \leq |W| \tag{12}$$

is a valid inequality for PDP(G_n).

PROOF. The proof is by contradiction. Suppose $R \subset E$ satisfies Conditions (i)–(iii) but fails (12) for some i, j, and W. Suppose without loss of generality, $i = 1$ and $j = 2$. Then the subtour elimination constraint on W yields

$$|W| - 1 \geq x(W), \tag{13}$$

from which

$$\begin{aligned} |W| - 1 + x(+1,-1) + x(+2,-2) &\geq x(W) + x(+1,-1) + x(+2,-2) \\ &> |W|. \end{aligned}$$

This implies

$$x(+1,-1) + x(+2,-2) > 1,$$

or simply

$$\begin{aligned} x(+1,-1) &= 1, \\ x(+2,-2) &= 1. \end{aligned}$$

Again, because R fails (12),

$$\begin{aligned} x(W) + x(+1,-1) + x(+2,-2) &> |W|, \\ x(W) &> |W| - 2, \end{aligned}$$

and thus

$$|W| - 1 \geq x(W) > |W| - 2.$$

Therefore equality holds in the subtour elimination constraint (13) and R violates a precedence condition, either on 1 or 2. ∎

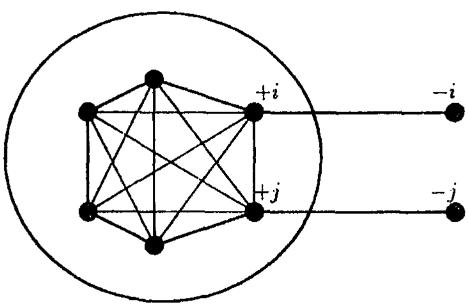


Figure 8. The order matching constraint with 2 teeth.

REMARK 6.5. Completely analogous results hold for the inequality when

$$\{-i,-j\} \subset W \subset V \setminus \{+i,+j,\pm 0\}.$$

Essentially, this constraint is the subtour constraint

$$x(W) \leq |W| - 1$$

strengthened by lifting the variables $x(+i,-i)$ and $x(+j,-j)$. This subtour is actually redundant because it is implied by the precedence constraint for $U = W \cup \{-0\}$.

This is by no means a complete description of PDP(G_n). We have failed to show any of the constraint classes actually define facets. Such proofs are usually constructions of affinely independent routes in the face defined by the constraint. However, empirical evidence show that for "small" problems (less than 10 customers) these constraints fairly well define the optimal solution.

7. BRANCH-AND-CUT ALGORITHM

We used the facial description above to construct a branch-and-cut algorithm to solve the integer program. This method is similar to the algorithm used to solve 2392-node TSP to optimality [1]. The research in the polyhedral structure of TSP(K_n) is very advanced with many known classes of facets and efficient identification routines. Our research shows the branch-and-cut algorithm, even with only a limited description of the polyhedron, can have good results. Further research into the PDP polyhedral structure will result in faster more efficient algorithms.

The branch-and-cut algorithm differs from traditional branch-and-bound integer program procedures by including cutting plane generation during the bounding phase. The branch-and-bound algorithm computes bounds on the optimal solution by solving a linear relaxation of the integer program. The branch-and-cut algorithm uses this fractional solution to compute cutting planes which improves the bounds, thereby reducing the size of the search tree.

This implementation of the branch-and-cut algorithm uses the mixed integer library MINTO [17,18] to manage the search tree, memory and active constraint set.

Simple routines were developed to identify some constraints from each of the four classes of inequalities described above. Violated subtour and precedence constraints were found by solving maximum flow subproblems. Additionally, some violated 2-order and order matching constraints were identified.

The initial upper bound in the procedure is generated using a greedy route construction heuristic. Each customer origin and destination were considered for insertion simultaneously. The customer whose insertion resulted in the least cost increase was added to the route. This heuristic is particularly good for small problems and has the advantage of always maintaining a feasible route of the inserted customers.

We tested the algorithm by solving problems randomly generated by selecting locations in the continental United States. These locations represent some 750 commercial, public, and military air fields. Even though the sample set is real data, no attempt was made to generate "realistic" problems. Similar results would probably be obtained by generating random problems in the unit square. Execution times on a Sun SparcServer 670 are shown in Table 1. Problems on 6 or fewer customers were solved nearly instantaneously primarily because of the quality of the solutions found by the heuristic.

Table 1. Median execution statistics for various sized problems.

Problem Size	Time (sec)	Depth	Nodes Searched	LP's Solved
7	3.580	1	3	9
9	11.780	5	13	32
11	58.565	11	59	104
13	189.65	13	123	212
15	1246.0	20	491	716

A significant time increase was experienced on larger problems—some particularly hard large problems required more than a half hour of processing time. The larger problems exhibit a tremendous growth in the search tree resulting from many iterations without generating additional constraints. The algorithm's ability to identify violated constraints could be improved by identifying more known facets of the TSP such as combs and clique trees [19], developing more robust schemes to identify the PDP specific inequalities described in this paper, and find new classes of faces for this problem.

REFERENCES

1. M. Padberg and G. Rinaldi, A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems, *SIAM Rev.* **33**, 60–100 (1991).
2. G. Dantzig, D. Fulkerson and S. Johnson, Solution of a large-scale traveling-salesman problem, *Oper. Res.* **2**, 393–410 (1954).

3. M. Grötschel and M. Padberg, Polyhedral theory, In *The Traveling Salesman Problem*, (Edited by E.L. Lawler *et al.*), pp. 307–360, John Wiley & Sons, (1985).
4. M. Grötschel and W.R. Pullyblank, Clique tree inequalities and the symmetric travelling salesman problem, *Math. Oper. Res.* **11**, 537–569 (1986).
5. D. Naddef and G. Rinaldi, The crown inequalities for the symmetric traveling salesman polytope, *Math. Oper. Res.* **17**, 308–326 (1992).
6. K.S. Ruland and E.Y. Rodin, Survey of facial results for the traveling salesman polytope, *Mathl. Comput. Modelling* (to appear).
7. L. Bianco, A. Mingozzi, S. Riccaiardelli and M. Spadoni, Exact and heuristic procedures for the traveling salesman problem with precedence constraints, based on dynamic programming, *INFOR* **32** (1), 19–31 (1994).
8. H.N. Psaraftis, An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows, *Transportation Sci.* **17** (3), 351–357 (1983).
9. T.R. Sexton and L.D. Bodin, Optimizing single vehicle many-to-many operations with desired delivery times: I. Scheduling, *Transportation Sci.* **19** (4), 378–410 (1985).
10. T.R. Sexton and L.D. Bodin, Optimizing single vehicle many-to-many operations with desired delivery times: II. Routing, *Transportation Sci.* **19** (4), 411–435 (1985).
11. Y. Dumas, J. Desrosiers and F. Soumis, The pickup and delivery problem with time windows, *European J. Oper. Res.* **54**, 7–22 (1991).
12. J.-J. Jaw, A.R. Odoni, H.N. Psaraftis and N.H. Wilson, A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows, *Transportation Res., Part B* **20B** (3), 243–257 (June 1986).
13. H.N. Psaraftis, k-Interchange procedures for local search in a precedence-constrained routing problem, *European J. Oper. Res.* **13**, 391–402 (1983).
14. H.N. Psaraftis, Analysis of an $O(n^2)$ heuristic for the single vehicle many-to-many Euclidean dial-a-ride problem, *Transportation Res.* **17B** (2), 133–145 (1983).
15. M.W.P. Savelsbergh and M. Sol, The general pickup and delivery problem, *Transportation Sci.* **29** (1), 17–29 (1995).
16. D. Naddef and G. Rinaldi, The graphical relaxation: A new framework for the symmetric travelling salesman polytope, *Math. Programming* **58**, 53–88 (1992).
17. G. Nemhauser, M.W.P. Savelsbergh and G. Sigismondi, MINTO, a mixed INTeger optimizer, *Oper. Res. Lett.* **15**, 47–58 (1994).
18. M.W.P. Savelsbergh and G.L. Nemhauser, *Functional Description of MINTO—Mixed INTeger Optimizer, Version 2.0*, Georgia Institute of Technology, (1994).
19. M. Padberg and G. Rinaldi, Facet identification for the symmetric traveling salesman polytope, *Math. Programming* **47**, 219–257 (1990).