RIMS Kôkyûroku Bessatsu
B23 (2010), 33–47

Structure of the Extreme Points of the Subtour Elimination Polytope of the STSP

By

Sylvia Boyd[*] and Paul Elliott-Magwood[**]

Abstract

The Symmetric Travelling Salesman Problem (STSP) is the problem of finding a minimum weight Hamiltonian cycle in a weighted complete graph on n vertices. This problem is well known to be NP-hard. One direction which seems promising for finding improved solutions for this and other NP-hard problems is the study of the structure of the extreme solutions associated with the problem's linear programming relaxation. This approach has led to new approximation algorithms and results for several NP-hard problems. Moreover it has not been possible thus far to obtain these results via other more traditional methods, i.e. knowledge of the structure of these extreme points was key.

In this paper we study the structure of the extreme solutions of the Subtour Elimination Problem (SEP), which is a linear programming relaxation of the STSP. We give some new results on both the underlying structure of these extreme solutions, as well as the structure of the defining cobasis for such solutions. We demonstrate the usefulness of these results by showing how this new theory facilitates the generation of all extreme solutions of the SEP for some values of n that were previously unattainable. This allows for the first time, for these values, the verification of the well-known conjecture that the integrality gap is 4/3 for the metric STSP. We believe that with further exploration these results may also facilitate the development of improved approximation algorithms for the STSP.

§ 1. Introduction

Given the complete graph $K_n = (V, E)$ on n vertices with non-negative edge costs $c \in \mathbf{R}^E$, $c \neq 0$, the *Symmetric Travelling Salesman Problem* (henceforth STSP) is the problem of finding a Hamiltonian cycle (or *tour*) in K_n of minimum cost. When the costs satisfy the triangle inequality, i.e. when $c_{uv} + c_{vw} \geq c_{uw}$ for all distinct triples

Received September 15, 2008. Revised May 28, 2009.
2000 Mathematics Subject Classification(s): 90C27, 90C57
Key Words: polyhedral combinatorics, approximation algorithms, graph and network algorithms
Supported by Natural Sciences and Engineering Research Council of Canada.

[*] University of Ottawa, Ottawa, Canada.
[**] University of Ottawa, Ottawa, Canada.

$u, v, w \in V$, we call the problem the *metric* STSP. The STSP is known to be NP-hard, even in the metric case [16].

One approach taken for finding reasonably good solutions for the STSP is to look for a γ-approximation algorithm for the problem, that is, try to find a heuristic which finds a tour which is guaranteed to be of cost at most γ times the optimal STSP value for some constant $\gamma \geq 1$. Currently the best γ-approximation algorithm known for the metric STSP is Christofides algorithm [9] for which $\gamma = 3/2$. Note that for general costs there does not exist a γ-approximation algorithm unless $P = NP$ [16].

Another approach taken is to study a linear programming relaxation for the problem. Such relaxations are usually much easier to solve than the original problems, and provide good starting points for the application of the branch and cut method, as well as good bounds on the value of optimal solutions. Good bounds on the value of the optimal solution are useful in evaluating the effectiveness of a heuristic.

An interesting problem that is closely related to the problem of finding a γ-approximation algorithm is that of finding the integrality gap of a linear programming relaxation for a problem, which is the value of the worst-case ratio between the optimal value for the problem and the optimal value for the relaxation. This integrality gap is important for finding improved solutions for NP-hard problems, as it gives a measure of the quality of the bound provided for the original problem by the linear programming relaxation. Moreover, a polynomial-time constructive proof of an integrality gap of value α provides an α-approximation algorithm for the problem.

If we relax the integer requirement in the integer linear programming formulation of the STSP we obtain the subtour elimination problem (SEP) relaxation for the STSP. For the metric STSP, the integrality gap αTSP between STSP and SEP is not known, although it is known that $4/3 \leq \alpha TSP \leq 3/2$ ([18]). In fact, a well-known conjecture in combinatorial optimization says that $\alpha TSP = 4/3$ for the metric STSP. Even though this conjecture has been around for over 20 years, very little progress has been made towards proving or disproving it. Also, surprisingly, no one has been able to improve upon the Christofides 3/2-approximation algorithm in the last 30 years.

For problems such as the STSP where standard methods for obtaining approximation algorithms and the integrality gap have failed, it seems it may be necessary to develop new techniques to have any hope of success. One direction which seems promising for finding improved solutions for this and other NP-hard problems is the study of the structure of the extreme solutions of the linear programming relaxation. This approach has led to new approximation algorithms and results for several NP-hard problems (see, for example, [15], [13], [12], [8], [11], [6] and [1]), moreover it has not been possible so far to obtain these results via other more traditional methods, i.e. knowledge of the structure of these extreme solutions was key in obtaining the results.

In this paper we examine the structure of the extreme solutions of the SEP. It is our hope that this will lead to improved approximation algorithms for the STSP and also information regarding the integrality gap . In Section 2 we give some new overall structural results on these extreme solutions, as well as some operations that allow us to generate extreme solutions from those of smaller versions of the problem, and vice versa. In Section 3 we examine the structure of the cobases for these extreme solutions, and give some results. Finally, in Section 4 we demonstrate the usefulness of our results by showing how they can be used to facilitate the the generation of all extreme solutions for $n = 11$ and $n = 12$, something that has not been previously possible. This allowed the verification of the 4/3 conjecture for the integrality gap of SEP for these values of n for the first time.

Note that in order to keep this paper brief we have omitted the proofs for many results, although we have included some sketches of proofs to give the flavour of the methods used. Complete proofs for all results can be found in [2], [3] and [4].

We conclude this section with a few definitions and explanations of the notation used.

Given a graph G and vertex subset U of G, we use the notation $G[U]$ to denote the subgraph of G induced by U, and $\gamma(U)$ to denote the edges with both endpoints in U. Given a graph G and disjoint vertex subsets X and Y of G, we let $E(X : Y)$ denote all the edges with one end in X and one end in Y.

For any edge set $F \subseteq E$ and $x \in \mathbf{R}^E$, let $x(F)$ denote the sum $\sum_{e \in F} x_e$. For any vertex set $W \subset V$, let $\delta(W)$ denote $\{uv \in E : u \in W, v \notin W\}$. Let $\mathcal{S} = \{S \subset V, 2 \leq |S| \leq n-2\}$. Then we define an integer linear programming (ILP) formulation for the STSP is as follows:

$$
\begin{array}{lrl}
(1.1) & \text{minimize } cx & \\
(1.2) & \text{subject to: } x(\delta(v)) = 2 & \text{for all } v \in V, \\
(1.3) & x(\delta(S)) \geq 2 & \text{for all } S \in \mathcal{S}, \\
(1.4) & x_e \geq 0 & \text{for all } e \in E, \\
(1.5) & x & \text{integer.}
\end{array}
$$

The constraints (1.2) are called the *vertex equalities*, the constraints (1.3) are called the *cut constraints*, and the constraints (1.4) are called the *non-negativity constraints*.

If we drop the integer requirement (1.5) from the above ILP, we obtain a linear programming (LP) relaxation of the STSP called the *Subtour Elimination Problem* (SEP). The extreme solutions of this relaxation are the extreme points of the associated *SEP polytope*, which we denote by S^n for the problem on n vertices. The SEP polytope is the set of all vectors x satisfying the constraints of the SEP, i.e.

$$S^n = \{x \in \mathbf{R}^E : x \text{ satisfies } (1.2), (1.3), (1.4)\}.$$

Note that despite the fact that there is an exponential number of constraints (3), the SEP can be solved in polynomial-time (using the ellipsoid method) since there is an exact polynomial-time separation algorithm for each of its constraints [14]. However, no practical polynomial-time algorithm is currently known.

Given any feasible $x \in S^n$, the *weighted support graph* $G_x = (V_x, E_x)$ of x is the subgraph of K_n induced by the edge set $E_x = \{e \in E : x_e > 0\}$, with edge weights x_e for $e \in \mathbf{R}^{E_x}$. We say that a constraint is *tight* with respect to x if x satisfies the constraint with equality, and we say a vertex subset S of V is a *tight set* if $x(\delta(S)) = 2$.

§ 2. Results on extreme point structure and generation

In this section we report some new results pertaining to the structure of the extreme points of the SEP polytope S^n. These results fall into two types. First we have results that look at the overall structure of a support graph G_x for points x in S^n. These results are useful in identifying conditions that are necessary for a graph G to represent a support graph for points in S^n. In Section 4 we will use these results to greatly reduce the number of possible graphs in generating all the extreme points for S^n. Second we present operations that allow us to generate extreme points for S^n from extreme points of S^k where $k < n$. Again these results will prove very useful in Section 4 for greatly reducing the work required in generating all the extreme points for S^n.

The following are several results already known to be necessary for support graphs G_x for $x \in S^n$.

Theorem 2.1 (Boyd et. al. [7]). *Let x be an extreme point of S^n, $n \geq 3$. Then $|E(G_x)| + \frac{1}{2}q \leq 2n-3$, where q represents the number of vertices in G_x which have degree 3 and for which none of the corresponding incident edges $e \in E$ have value $x_e = 1$.* □

Theorem 2.2 (Boyd et. al. [7]). *Let x be an extreme point of S^n, $n \geq 3$. Then there are at least three edges e of G_x for which $x_e = 1$.* □

Theorem 2.3 (Goemans [13]). *Let x be an extreme point of S^n. Then for any $U \subset V$, $|E(G_x[U])| \leq 2|U| - 3$.* □

Without much extra work, we can refine the results of Theorem 2.3.

Corollary 2.4. *Let x be an extreme point of S^n. Then for any $U \subset V$, $|E(G_x[U])| \leq 2|U| - b(U) - 3$ where $b(U)$ is the number of bipartite components of $G_x[U]$.* □

There is also the following result, which is easily proved.

Proposition 2.5. *If $x \in S^n$ then G_x is 2-vertex-connected.* □

We have found that more general necessary conditions can be proven as well. Let $\kappa(G)$ denote the number of connected components of a graph G. We say that a graph, $G = (V, E)$ is *t-tough* if for every $T \subset V$ we have that

$$|T| \geq t * \kappa(G - T).$$

The concept of t toughness was first introduced and explored by Chvátal [5]. In his paper, Chvátal was interested in using toughness to find necessary and sufficient conditions for a graph to be Hamiltonian. He noted that every Hamiltonian graph must be 1-tough.

Notice that every 1-tough graph is also 2-vertex-connected, i.e. removing any vertex in a 1-tough graph leaves exactly one component. For our purposes, 1-toughness also plays a role as a necessary condition for support graphs of points in S^n. We can show the following result holds.

Proposition 2.6. *If $x \in S^n$ then G_x is 1-tough.* □

We can impose similar, but stronger, conditions on a graph which are necessary for it to be a support graph for a point in S^n. Given a graph, H, let $b_0(H)$ denote the number of blocks of H which contain no cut vertices and let $b_1(H)$ denote the number of blocks of H which contain exactly one cut vertex (we call these blocks *endblocks*). We will say that a graph, $G = (V, E)$, is *t-block-tough* if for every $T \subset V$ we have that

$$|T| \geq t \left(b_0(G - T) + \frac{1}{2} b_1(G - T) \right).$$

Note that it can be shown that every t-block-tough graph is t-tough. Hence the following result is a generalization of Proposition 2.6:

Theorem 2.7. *Let $x \in S^n$. Then G_x is 1-block-tough.*

Sketch of Proof. We basically show this by using a result from linear programming duality that says that a primal linear program is infeasible if its corresponding dual linear program is unbounded.

Let $G = (V, E)$ be a graph and suppose there exists some $T \subset V$ such that

$$|T| < b_0(G - T) + \frac{1}{2} b_1(G - T).$$

It is easy to show that we may assume that no cut vertex of $G - T$ is contained in two different endblocks.

Now let $Q_1, \ldots, Q_k$ be the endblocks of $G - T$ which contain cut vertices $v_1, \ldots, v_k$ respectively. Let $R_1, \ldots, R_l$ be the 2-connected components of $G - T$. Define $y \in \mathbf{R}^V$

such that y_v is -2 if $v \in T$, -1 if $v = v_i$ for some $1 \leq i \leq k$, and 0 otherwise. Define $d \in \mathbf{R}^{\mathcal{S}}$ such that d_S is 2 if $S = R_i$ for some $1 \leq i \leq l$, 1 if $S = Q_i$ or $S = Q_i \backslash \{v_i\}$ for some $1 \leq i \leq k$, and 0 otherwise.

It is straightforward to check that y and d are values for the variables of the dual linear program of the SEP which can be used to indicate that the dual linear program is unbounded. Thus by the theory of linear programming duality, the SEP is infeasible on the graph G. □

We next move on to several new results that allow us to generate extreme points for S^n from extreme points of S^k, $k < n$. In fact, the next theorems show, in particular, how any extreme point x of S^n for which there is a tight set $S \subset V$ of size 3 can be obtained directly from an extreme point of S^{n-1} or S^{n-2}. As will be shown in Section 4, this result greatly reduces the work required in generating all the extreme points for S^n, as many of these can be obtained directly from the extreme points of S^{n-1} or S^{n-2} via this operation.

Let $x \in S^n$ and let S be a tight set of x. We let $x \downarrow S$ denote the set of edge values induced on G_x/S (where S is identified to a single vertex v) as follows.

$$(x \downarrow S)_e = \begin{cases} x(E(u:S)) & \text{if } e = uv \\ x_e & \text{otherwise.} \end{cases}$$

The next two theorems deal with an operation that shows how we can "split" or "unsplit" an edge e in G_x for which $x_e = 1$.

Theorem 2.8. *Let x be an extreme point of the SEP-polytope on S^n, $n \geq 4$, and let u and v be vertices of G_x such that $x_{uv} = 1$. If there exists a vertex, w, of G_x such that $x_{uw} > 0$ and $x_{vw} > 0$ then $x \downarrow \{u, v\}$ is an extreme point of S^{n-1}.* □

Let $x \in S^n$ and let G_x be the support graph of x. If zw is an edge of G_x and we can partition the edges, apart from zw, which are incident to z in G_x into two parts, E_1 and E_2, such that $0 \leq x(E_1), x(E_2) \leq 1$ then we define $x \uparrow_z (zw, E_1, E_2)$ (as shown in Figure 1) by deleting the vertex z and adding two new vertices u and v where

$$(x \uparrow_z (zw, E_1, E_2))_e = \begin{cases} 1 & \text{if } e = uv \\ 1 - x(E_1) & \text{if } e = uw \\ 1 - x(E_2) & \text{if } e = vw \\ x_{qz} & \text{if } e = qu \text{ and } qz \in E_1 \\ x_{qz} & \text{if } e = qv \text{ and } qz \in E_2 \\ x_e & \text{if } e \text{ is an edge of } G_x - z \\ 0 & \text{otherwise.} \end{cases}.$$

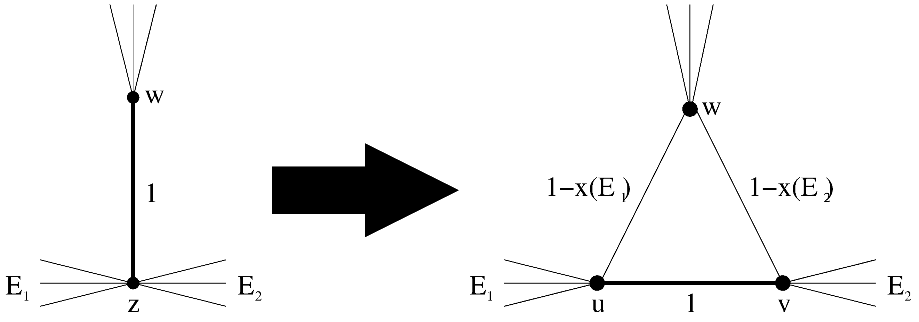

Figure 1. The edge-splitting operation

Theorem 2.9. *Let x be an extreme point of S^n. If zw is an edge of G_x and we can partition the edges, apart from zw, which are incident to z in G_x into two parts, E_1 and E_2, such that $0 \leq x(E_1), x(E_2) \leq 1$ then $x \uparrow_z (zw, E_1, E_2)$ is an extreme point of S^{n+1}.* □

Note that Theorem 2.9 is a generalization of a result by W. H. Cunningham reported in [7] that deals with the special case of the theorem for which the edge zw has value $x_{zw} = 1$ and z has degree 3 in G_x. Also Theorems 2.8 and 2.9 together imply the following result, as a special case:

Theorem 2.10 (Benoit et. al. [1]). *Consider $x \in \mathbf{R}^E$ such that for some vertex v we have $x_{uv} = x_{vw} = 1$. Let $\hat{x} = x \downarrow \{u, v\}$. Then $\hat{x}$ is an extreme point of S^{n-1} if and only if x is an extreme point of S^n.* □

We also have the following theorem which demonstrates the importance of the edge-splitting operation in its effect on the rank of the vertex equalities.

Theorem 2.11. *Let x be an extreme point of S^n, $n \geq 4$, and let F be the set of all 1-edges of x. If x cannot be obtained via the edge-splitting operation from an extreme point of S^{n-1} then the vertex equalities of $G_x - F$ have rank n or $n-1$.* □

The next two theorems tell us that we can shrink any fractional odd cycle that is tight in the support graph of an extreme point of S^n and get another extreme point, and we can also reverse this operation under certain conditions.

Theorem 2.12. *Let x be an extreme point of S^n and let S be a tight set of x such that $G_x[S]$ is an odd cycle. If $x_e < 1$ for every $e \in \gamma(S)$ in G_x then $x \downarrow S$ is an extreme point of $S^{n-|S|+1}$.*

Sketch of proof Here we will illustrate the ideas of the proof by sketching it only for the special case where $|S| = 3$.

We can show that each tight cut of x is either disjoint from S or contains S. We can then show that no two vertices of S are adjacent to a common vertex outside of S, since otherwise we have have a 4-cycle in G_x such that every tight cut of x contains 0 or 2 (adjacent) edges of this 4-cycle. Furthermore, all of the edges of this cycle obey $0 < x_e < 1$ so by choosing some small $\epsilon > 0$ we can alternately add and subtract ϵ to every edge in the cycle and obtain a new solution to the system of equations which are tight for x. This contradicts the fact that x is an extreme point. Hence every edge of G_x/S corresponds to a unique edge of G_x and thus contracting S in x results simply in removing the edges of $\gamma(S)$ and relabelling the edges of $\delta(S)$. Hence the tight sets of $x \downarrow S$ can be obtained from the tight sets of x simply by removing the 3 vertex equalities for the vertices in S. Since G_x/S has 3 fewer edges than G_x, these tight constraints form a set of equations for which $x \downarrow S$ is the unique solution. By the above, $x \downarrow S$ is also feasible for S^{n-2} thus $x \downarrow S$ is an extreme point of S^{n-2}. □

The next theorem is an analogous result that says that, under certain conditions, we can expand a vertex of an extreme point into a fractional odd cycle and obtain another extreme point.

Let $x \in S^n$ and let v be a vertex of G_x. Suppose we can partition the edges of G_x incident to v into k non-empty parts, $(E_1, \dots, E_k)$ where $k \geq 3$ is odd and for each $0 \leq i \leq k-1$ we have that

$$\sum_{j=0}^{\frac{1}{2}(k-3)} x(E_{i+2j+2}) < 1$$

(where all indices are taken modulo k). Then we define $x \uparrow_v (E_0, \dots, E_{k-1})$ as follows. Remove v from G_x and add k new vertices, $v_0, \dots, v_{k-1}$. Let $S = \{v_0, \dots, v_{k-1}\}$

$$(x \uparrow_v (E_0, \dots, E_{k-1}))_e = \begin{cases} x_e & \text{if } e \in \gamma(\overline{S}) \\ x_{uv} & \text{if } e = uv_i \in \delta(S) \\ \sum_{j=0}^{\frac{1}{2}(k-3)} x(E_{i+2j+2}) & \text{if } e = v_i v_{i+1} \\ 0 & \text{otherwise} \end{cases}.$$

An example for $k = 3$ is depicted in Figure 2.

Theorem 2.13. *Let let x be an extreme point of S^n, let $k \geq 3$ be an odd integer and let v be a vertex of G_x. If the edges incident to v in G_x can be partitioned into k non-empty parts, $(E_0, \dots, E_{k-1})$, such that for each $0 \leq i \leq k-1$ we have that*

$$\sum_{j=0}^{\frac{1}{2}(k-3)} x(E_{i+2j+2}) < 1$$

(where all indices are taken modulo k) then $x \uparrow_v (E_0, \dots, E_{k-1})$ is an extreme point of S^{n+k-1}. □

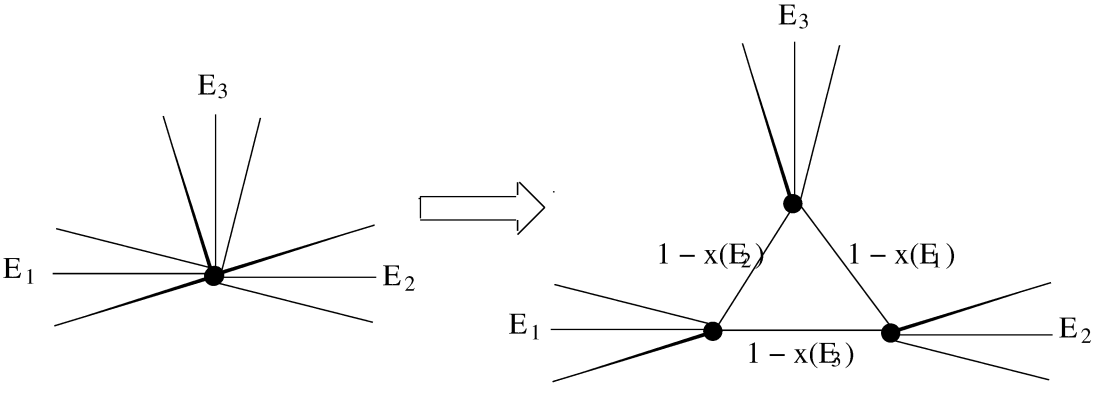

Figure 2. Inserting a 3-cycle

Note that Theorems 2.12 and 2.13 are a generalization of a result in [7], which deals with the special case of these theorems for which the odd cycle in the support graph has size 3, forms a tight set, and each vertex in the cycle has degree 3 and is not incident with an edge e for which $x_e = 1$.

The following corollary, which will be used extensively in the application in Section 4, follows directly from Theorems 2.8 and 2.12.

Corollary 2.14. *Let x be an extreme point of S^n and let $S \subset V$ be a tight set for x such that $|S| = 3$. Then we can obtain x from an extreme point of S^{n-1} or S^{n-2}.* □

§ 3. Reults on the cobasis structure of the extreme points of S^n

Given a system $Ax = b$ of equations, we will use the term *rank* for this system to denote the linear rank of matrix A.

Any extreme point $x \in S^n$ is uniquely determined by its tight constraints, i.e. it is the unique solution to the system $Ax = b$ composed of vertex equalities, and cut and non-negativity constraints that are tight with respect to x. Let the rank of $Ax = b$ be k. We say that a subsystem of k of these tight constraints form a *cobasis* for x if it has the same rank as A. The extreme points of S^n are highly degenerate in that for any extreme point there is a huge number of possible cobases. For example, for the extreme point x which is a tour for S^6 there are over 2000 possible cobases which determine x.

Many results from the previous section on extreme point structure arise from the knowledge that, for any extreme point $x \in S^n$, there always exists a cobasis that satisfies certain properties. In this section we extend what is currently known. These results will prove essential in generating all the extreme points for S^{11} and S^{12}, as described in Section 4, as they show it possible to consider a very small subset of the cobases for an extreme point rather than all of them.

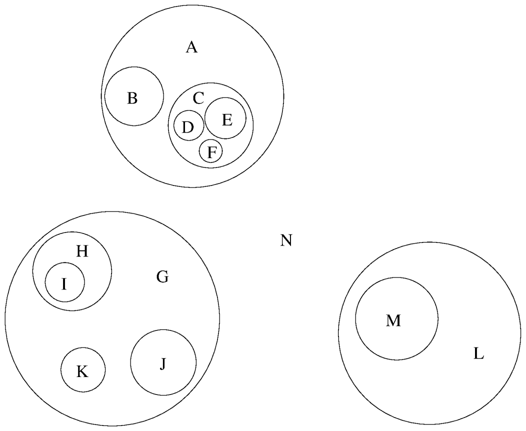

Figure 3. A Laminar Set

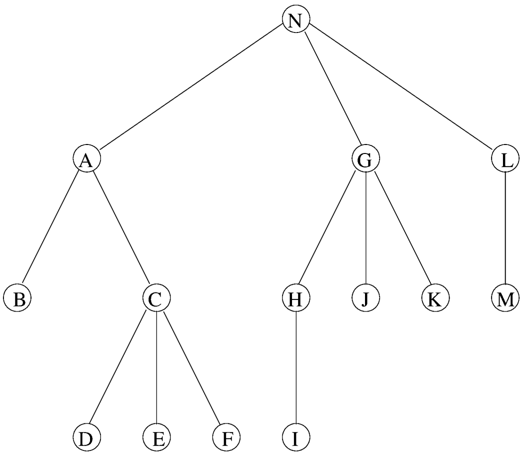

Figure 4. The LBT

A family of sets $\mathcal{L}$ is called *laminar* if no two sets in the family properly intersect each other, i.e. for any two sets $S, T \in \mathcal{L}$ we have that $S \subset T$ or $T \subset S$ or $S \cap T = \emptyset$. The following theorem is well-known:

Theorem 3.1 (Cornuéjols et al. [10]). *For any $x \in S^n$ there exists a cobasis for which the tight vertex subsets corresponding to the tight cut constraints form a laminar set.* □

We will show how the above theorem can be strengthened.

There is a way of compactly storing the laminar set of tight cuts in our cobasis by means of a labelled tree which we will call the *Laminar Basis Tree* or LBT for short. The LBT has one node (and one edge) for each tight cut in our laminar set plus one extra root node. The node corresponding to a set S in our laminar set will be labelled with the set of vertices of G_x which are in S but not in any proper subset of S in the laminar set. The root node will be labelled with the set of all vertices of G_x that are in no set in the laminar set. Two nodes will be adjacent if for the corresponding sets, one is the minimal set which contains the other. The root node will be adjacent to all the sets which are not contained in any other. As an example, Figure 3 shows a laminar set and Figure 4 shows the corresponding unlabelled LBT.

Notice now, that if we take a maximal subset in our laminar set and replace it with its complement, we get another laminar set which is also a cobasis (along with the tight non-negativity inequalities and the vertex equalities) for x. This occurs because our new laminar set induces the exact same cuts as the old one. In fact, the constraints in the cobasis are exactly the same. The LBT for this new laminar set will be identical to that of the first laminar set, the only difference being which node is the root. By repeating this process of replacing a maximal subset in the laminar set with its complement, we can make any node the root in the corresponding LBT. However, these changes to the laminar set do not actually change the constraints in the cobasis. Hence, for a given cobasis, we can construct an LBT which we will consider to be unrooted. We now proceed with some properties of LBTs.

Proposition 3.2. *Let T be a LBT for a support graph with n vertices and let n_1 and n_2 denote the numbers of nodes of T of degree 1 and 2 respectively. Then*

1. *T has at most $n-2$ nodes,*
2. *the total number of labels on the nodes of T is n,*
3. *every node of T of degree 1 must contain at least two labels,*
4. *every node of T of degree 2 must contain at least one label, and*
5. *$2n_1 + n_2 \leq n$.* □

Proposition 3.3. *Let x be an extreme point of S^n with support graph G_x and let S be the set of labels corresponding to some node, v, of a LBT, T, for some cobasis of x. Then each component of $G_x[S]$ is either a tree or a 1-tree with an odd cycle. Futhermore $G_x[S]$ has at most $deg_T(v)$ components.* □

Proposition 3.4. *Let x be an extreme point of S^n for which the minimum degree of G_x is at least 3. Then there is a cobasis, $\mathcal{B}$, for x such that the tight sets corresponding to the tight cut constraints in $\mathcal{B}$ form a laminar set and the leaf nodes of the associated LBT correspond exactly to the 1-edges of x.* □

We give a small example to illustrate the power of Proposition 3.4 in restricting the number of cobases we need to examine for an extreme point. Consider the extreme point, x of $\mathcal{S}^{10}$ with vertices $u_0, \ldots, u_4$ and $v_0, \ldots, v_4$ such that $u_i u_{i+1} = v_i v_{i+1} = 1/2$ for each $0 \leq i \leq 4$ and $u_i v_i = 1$ for eacb $0 \leq i \leq 4$. If we build up a cobasis for x from the vertex equalities and all tight nonnegativity constraints, then there are at least 1792 different families of tight sets which will give us a cobasis for x. There are 280 such families which are laminar, but only 1 meets the requirements of Proposition 3.4.

§ 4. An application: Finding all non-isomorphic extreme points for S^{11} and S^{12}

In [1], Benoit and Boyd were able to find the exact integrality gap αTSP for the STSP when the problems were restricted to have n vertices, $n \leq 10$. The method they used requires a list of all the non-isomorphic extreme points for S^n for each value of n considered. Although they were able to generate such a set for each value of $n \leq 10$, their methods and tools were completely impractical for $n = 11$.

In this section we explain how the results in the previous sections can be used to find a complete list of the non-isomorphic extreme points for S^n, and demonstrate this for $n = 11$ and $n = 12$. The basic idea is to take each graph G that could potentially represent a support graph G_x for an extreme point x of S^n, and then match it with all possible cobases that could go with that graph. For each such pairing we check the corresponding solution x to see if it is indeed an extreme point. Of course the number of possible cobases for a graph makes this method impractical, as can the number of potential support graphs themselves. However if we carefully apply our results from the previous two sections we can make both of these numbers quite manageable for $n = 11$ and $n = 12$.

We begin by outlining how to obtain a set of potential support graphs. Note that by Corollary 2.14 we can generate all extreme points x which have a tight set S of size 3 from the extreme points of S^{n-1} or S^{n-2}, so we will assume that the extreme points we seek do not have such a tight set (which implies, in particular, that we do not have two adjacent 1-edges in any support graphs). To obtain a reasonable-sized set of potential support graphs for all other extreme points x of S^n, we used the following:

Step 1 Find all the non-isomorphic 2-vertex connected graphs on n vertices that have at most $2n - 3$ edges and minimum degree of 3. This can be accomplished using the software tool NAUTY [17].

Step 2 For each graph G' from Step 1, check if it has a subset S where $|E(G'[S])| > 2|S| - b(S) - 3$. By Corollary 2.4 we can eliminate such a graph from our set.

Step 3 For each remaining graph G', check if it is 1-block tough. If not, we can eliminate it by Theorem 2.7.

We have outlined in Table 1 the effectiveness of these two ways of eliminating potential support graphs. Note that the process from Corollary 2.4 and for block-toughness were run independently, and thus there is some overlap in the numbers in the table (i.e. some graphs were eliminated by both processes).

n	Graphs produced by NAUTY	Graphs eliminated by Corollary 2.4	Graphs eliminated by Block-toughness	Remaining Graphs
11	54721	3649	11136	42087
12	956444	29654	211037	715753

Table 1. Elimination of Potential Support Graphs

Next we outline how to obtain a set of LBTs to be paired with each potential support graph. In this step we do not associate a set of vertices with each node of the

LBT, but rather one label that represents the number of vertices associated with that node of the LBT.

Step 1′ Find all the LBTs which obey the conditions of Proposition 3.2.

Step 2′ Since we are assuming that our support graphs have minimum degree 3, we know that each will have at least $\lceil \frac{3n}{2} \rceil$ edges. Thus for our cobasis to have full rank, we must have at least $\lceil \frac{3n}{2} \rceil - n$ tight cut constraints. So we remove any LBTs from Step 1′ that have fewer than $\lceil \frac{3n}{2} \rceil - n + 1$ nodes.

Step 3′ By Proposition 3.4 and Theorem 2.2 we can assume the LBTs have at least 3 leaf nodes, and all leaf nodes have label 2 (i.e. we can assume that the leaf nodes in the LBT correspond exactly to the 1-edges in the extreme point). Hence we can eliminate any remaining LBTs that do not satisfy these conditions.

Step 4′ By using Corollary 2.14 we can assume we have no tight sets of size three in our extreme points. Hence we can also eliminate all remaining LBTs that have a node with degree 2 with label 1 which has an adjacent leaf node with label 2.

In total, at the end of Step 4′, there were only 24 eligible LBTs for $n = 11$ and 92 for $n = 12$.

The final stage of the method involves pairing each potential support graph with each labelled LBT. Below we give a brief sketch of the steps for this which must be followed for each potential support graph G' in our list.

Step 1″ We begin by assigning some of the edges of G' to have value 1 in the corresponding extreme point. For each extreme point x we are generating, we know by our assumptions thus far that x_e will have value 1 for at least 3 edges, and that the edges with value 1 will form a matching in the support graph of x. So we find all matchings for G' of size between 3 and $\lfloor \frac{n}{2} \rfloor$.

Step 2″ For each matching in Step 1″, find the rank k of the system $\mathcal{K}$ of constraints consisting of the non-negativity constraints, vertex constraints, and constraints $x_e = 1$ for the edges in the matching. Take G' with this matching and pair it with each labelled LBT with the correct number of nodes and implied 1-edges from the leaf nodes such that this pairing could potentially result in a cobasis when we take the system $\mathcal{K}$ and add the cut constraints corresponding to the LBT. For each assignment of actual vertices for the node labels of the LBT, see if the resulting system of constraints does indeed result in an extreme point of S^n by checking its rank, and whether the corresponding solution is feasible for the SEP.

At the end of the pairing process, we directly generate all the extreme points with tight sets of size 3 from the extreme points of S^{n-1} and S^{n-2} and add these to our list of extreme points. As a final step, we remove all the isomorphic extreme points from the list, again using the software package NAUTY to do this.

Table 2 summarizes the results for $n = 11$ and $n = 12$. All of the numbers listed are the numbers of *non-isomorphic* extreme points, and include the tour. Also note that some identical extreme points were generated by some of the processes. Notice that over 85% of the extreme points in each case were generated by the edge-splitting operation (this is true for all $7 \leq n \leq 12$).

n	Points produced by pairing	Points produced by Theorem 2.13	Points produced by Theorem 2.9	Total number of points
11	673	15	4386	4972
12	9265	378	60390	68320

Table 2. Generating all extreme points

By successfully computing all the non-isomorphic extreme points of S^n, we were able to, for the first time, find the exact integrality gap αTSP for $n = 11$ and $n = 12$ using the method described in [1]. This value was 19/16 for $n = 11$ and 6/5 for $n = 12$, which verifies that the conjecture by Benoit and Boyd [1] about the exact value of the integrality gap for each value of n holds true for $n = 11$ and $n = 12$.

We conclude with some remarks on the time involved for solving these problems. Not surprisingly, the bottleneck in the time for finding the extreme points using our method was in completing Steps $1''$ and $2''$ above. For $n = 11$ these steps required just under 20 hours when running on a SUNW UltraSPARC-II. For $n = 12$ we partitioned the data into 8 pieces and ran them each on a different processor. These pieces took between 16 and 24 days each. Finally, in comparing to the time required by Benoit and Boyd [1], on the same machine their method required almost 4 days to find all the extreme points for $n = 10$, whereas our method required less than 7 minutes.

Acknowledgments

We would like to thank Maryam Haghighi for her help in developing the list of laminar basis trees for $n = 11$ and $n = 12$.

References

[1] Benoit, G., Boyd, S.: Finding the exact integrality gap for small Travelling Salesman Problems. Math. of Operations Research, accepted (2008).

[2] Boyd, S., Elliott-Magwood, P.: The Cobasis Structure of the Extreme Points of the SEP Polytope. Technical Report TR-2007-06, SITE, University of Ottawa, Ottawa, Canada (2007).

[3] Boyd, S., Elliott-Magwood, P.: Feasibility of the Held-Karp LP relaxation of the TSP. Technical Report TR-2007-07, SITE, University of Ottawa, Ottawa, Canada (2007).

[4] Boyd, S., Elliott-Magwood, P.: Operations for Generating SEP Extreme Points. Technical Report TR-2007-08, SITE, University of Ottawa, Ottawa, Canada (2007).

[5] Chvátal, V.: Tough Graphs and Hamiltonian Circuits. Discrete Mathematics **5** (1973) 215–228.

[6] Boyd, S., Carr. R.: A new bound for the ratio between the 2-matching problem and its linear programming relaxation. Mathematical Programming, Series A **86** (1999) 499–514.

[7] Boyd, S., Pulleyblank, W.R.: Optimizing over the subtour polytope of the travelling salesman problem. Mathematical Programming **49** (1991) 163–187.

[8] Carr, R., Ravi, R.: A new bound for the 2-edge connected subgraph problem. Proceedings of Integer Programming and combinatorial Optimization (IPCO), Lecture Notes in Computer Science, Springer (1998) 112–125.

[9] Christofides, N.: Worst case analysis of a new heuristic for the traveling salesman problem. Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh (1976).

[10] Cornuéjols, G., Fonlupt, J., and Naddef, D.: The Traveling Salesman Problem on a Graph and Some Related Integer Polyhedra. Mathematical Programming **33** (1985) 1–27.

[11] Fleischer, L.: A 2-approximaiton for minimum cost 0,1,2 vertex connectivity. Proceedings of Integer Programming and combinatorial Optimization (IPCO), Lecture Notes in Computer Science, Springer, (2001) 115–129.

[12] Fleischer, L., Jain, K., Williamson, D.: An iterative rounding 2-approximation algorithm for the element connectivity problem. 42nd Annual Symposium of the Foundations of Computer Science (2001).

[13] Goemans, M.: Minimum bounded degree spanning trees. Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (2006) 273–282.

[14] Grötschel, M., Lovasz, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization. Springer-Verlag, Berlin (1988).

[15] Jain, K.: A factor 2 approximation algorithm for the generalized Steiner network problem. Combinatorica **21** (2001) 39–60.

[16] Johnson, D.S., Papadimitriou, C.H.: Computational Complexity. In: The Traveling Salesman Problem (E.L. Lawler et al, eds.), John Wiley & Sons, Chichester (1985) 37–85.

[17] McKay, B.: NAUTY User's Guide (Version 1.5). Technical Report TR-CS-90-02, Department of Computer Science, Australia National University (1990).

[18] Shmoys, D.B., Williamson, D.P.: Analyzing the Held-Karp TSP bound: A monotonicity property with application. Inf. Process. Lett. **35** (1990) 281–285.