Title

Lower and Upper Bounds for a Symmetric Multiple Depot, Multiple Travelling Salesman Problem

Authors

Rathinam, Sivakumar
Sengupta, Raja

Publication Date

2006-03-21

Institute of Transportation Studies
University of California at Berkeley

Lower and Upper Bounds for a Symmetric Multiple Depot, Multiple Travelling Salesman Problem

Sivakumar Rathinam, Raja Sengupta

RESEARCH REPORT
UCB-ITS-RR-2006-2

March 2006
ISSN 0192 4095

Lower and Upper Bounds for a Symmetric Multiple Depot, Multiple Travelling Salesman Problem

Sivakumar Rathinam[1], Raja Sengupta[2]

Abstract— **This paper extends the well known Held-Karp's lower bound available for a single Travelling Salesman Problem to the multiple depot case. The LP-relaxation of a symmetric multiple vehicle, multiple depot problem is shown to be lower bounded by an infinite family of bounds. Each lower bound can be computed in a tractable way using a matroid intersection algorithm. When the costs of travelling between any two locations satisfy triangle inequality, it is shown that there exists a 2-approximation algorithm for solving the multiple depot, multiple TSP. These results are useful in solving the following path planning problem of UAVs: Given a set of UAVs, their starting locations, a set of final UAV locations, a set of destinations to visit and the cost of travelling between any two locations, find a path for each UAV such that each destination is visited once by any one UAV and the total cost travelled by all the UAVs is minimum.**

I. Introduction

Resource allocation problems concerning Unmanned Aerial Vehicles (UAVs) have received significant attention in recent years. A collection of small autonomous UAVs with the necessary sensors can potentially replace a manned vehicle in dangerous environments and warfare. A common mission that can be carried out by a group of UAVs is a surveillance operation where a set of sites needs to be monitored. If the number of sites to be visited are higher than the number of UAVs available, then the following resource allocation questions naturally arises:
i. How to partition the set of sites into subsets such that each UAV gets a subset of sites to monitor?
ii. Given a subset for each UAV, how to determine the order in which the sites should be monitored?
iii. Can we answer questions 1 and 2 in an optimal way? That is, can we find a partition and a order for each vehicle such that total distance travelled by the UAVs is minimum or the total risk encountered is minimum?

These are the questions addressed in this paper. For example, if there are n sites and each site must be monitored exactly once with the help of one vehicle, then the number of ways in which the sites can be visited is $n!$. A naive algorithm of choosing a particular order that would minimize the total distance travelled would be to calculate the total distances travelled for all the possible $n!$ ways and pick the one that has minimum cost. Even for small numbers of n, such naive algorithms are extremely time consuming and inefficient in practice. This problem of finding the sequence of sites for a vehicle to visit that results in minimum total distance is referred to as the classic Travelling Salesman Problem or TSP in the operations research literature. TSP is well known to be NP-Hard [1],[2]. There are no algorithms in the literature that can solve the TSP problem exactly in polynomial time[1]. There are many efficient algorithms that exploits the structure of the problem and produces a solution that is close to the optimal solution. A approximation factor $\beta(P, A)$ of using an algorithm A to solve the problem P (objective is minimize some cost function) is defined as

$$\beta(P, A) = \sup_{I}\left(\frac{C(I, A)}{C_o(I)}\right), \tag{1}$$

where I is a problem instance, $C(I, A)$ is the cost of the solution by applying algorithm A to the instance I and $C_o(I)$ is the cost of the optimal solution of I. In simple terms, the algorithm A produces an approximate solution to every instance I of the problem P, whose cost is within $\beta(P, A)$ times the optimal solution of I. Constant factor approximation algorithms are useful in the sense that they give an upper bound to the cost of the resulting solution that is independent of the size of the problem. Also, in the context of UAVs where resource allocation algorithms are for planning purposes and where problems have simplifying assumptions on the dynamics of the vehicle, it is reasonable to aim for fast algorithms that can provide solutions with a guaranteed approximation factor.

Similarly, there are also algorithms that can provide tight lower bounds to the TSP. An advantage of deriving lower bounds is that they can be used in branch and bound solvers used to solve the TSP and get faster results. Also, if one can find lower bounds that are close to the optimal solution in a efficient way, then the quality of using an algorithm can be found out by comparing the solution produced by the algorithm directly with the lower bound than with the optimal solution that may require a large computation time.

Travelling salesman problem has received extensive treatment in the literature and one can refer to [1] for all the possible heuristics, algorithms that have been used to solve the problem. For a general cost function (i.e. the cost function determines the cost of travelling between two sites), it has been proved that there exists no constant factor approximation algorithm unless P=NP. If the cost function

1. Graduate Student, Department of Civil Engineering, University of California, Berkeley, CA 94703, **corresponding author** e-mail: rsiva@berkeley.edu

2. Assistant Professor, Department of Civil Engineering, University of California, Berkeley CA-94702

[1]An algorithm is said to run in polynomial time if the number of steps required to run the algorithm is a polynomial function in the input size of the problem

satisfies triangle inequality and is symmetric[2], then the following are the two approximation algorithms available for the single TSP:

- 2 approx algorithm [2].
- 1.5 approx algorithm by Christofides [3].

When the sites lie on a Euclidean plane, the cost function has additional properties that was exploited by Arora in [4]. Given any $\epsilon > 0$, Arora's algorithm finds a solution with an approximation factor of $1 + \epsilon$ in time $n^{O(\frac{1}{\epsilon})}$. For the multiple vehicle case, where each vehicle starts and ends at the same depot and where the cost function is symmetric and satisfies triangle inequality, Rathinam et al. has proposed a 2-approx algorithm in [5].

As far as the lower bounds are concerned, Held and Karp's result is the best known result for the single TSP. The experimental results in [1] show that even for large size problems, Held-Karp's lower bound gets within 1-2% of the optimal solution. An important feature of Held-Karp's algorithm is that the results are very close to the optimal solution for any general cost function (i.e. cost function doesn't have to satisfy triangle inequality). Hence, in the context of UAVs, this might be ideal as the cost function could be determined by the risk of travelling between any two sites and hence, may not satisfy triangle inequality constraints.

The contributions of this work is as follows:

- A 2-approx algorithm for a multiple depot, multiple TSP[3] when the cost function satisfies triangle inequality.
- Extension of the Held-Karp's lower bound available for the single TSP to the multiple vehicle, multiple depot case.

II. Formulation of the Multiple Depot, Multiple TSP

Let $V = \{1, 2, 3...n\}$ be the set of destinations to be visited. There are k ($k \leq n$) vehicles initially located at vertices $S = \{s_1, s_2...s_k\}$. Each vehicle is required to visit at least 1 destination and reach a final location. There are k final locations denoted by the set $F = \{f_1, f_2...f_k\}$. A feasible set of paths consists of k non-intersecting paths that start at S and reach F such that all destinations are visited exactly once and each vehicle visits at least one destination. A example for a 3 vehicle scenario is shown in figure 1. There exists no edges between any two vertices in S, F or between S and F. All the remaining edges are present and the edge joining vertex i to j has a cost C_{ij} associated with it. Costs are symmetric, i.e., $C_{ij} = C_{ji}$. To formulate the problem, two additional vertices r and r' are added (refer to figure 2) such that $\forall i \in S$, $C_{ri} = 0$ and $\forall i \in F$, $C_{r'i} = 0$. Let $\mathbf{x} := \{x_{ij} : \forall i, j \in \{r, r'\} \bigcup S \bigcup V \bigcup F, i < j$ if $i, j \in V\}$ denote the edge incidence matrix. $x_{ij} = 1$ implies that the edge between vertex i to j is chosen and $x_{ij} = 0$ implies otherwise. The **MVMDP** is formulated as follows:

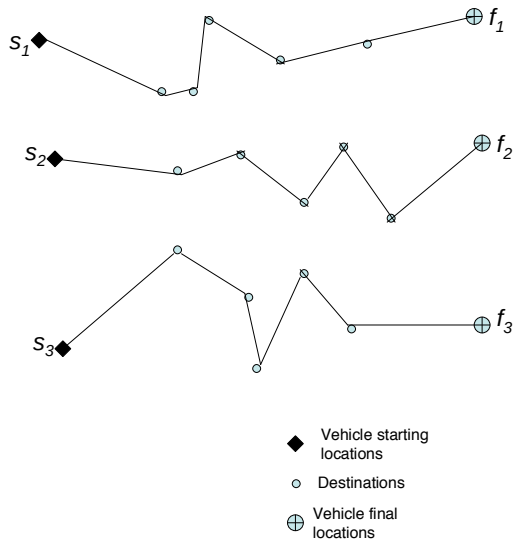

Fig. 1. An example of a 3 vehicle, 3 Depot TSP.

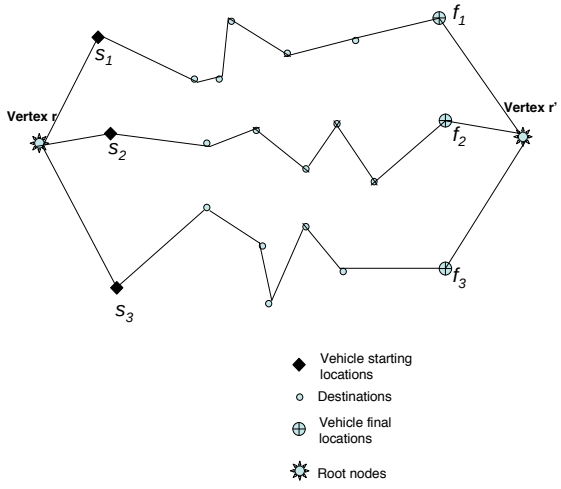

Fig. 2. Illustration of a Multiple Depot, Multiple TSP with root vertices.

Problem II.1: The objective is to find an incidence matrix $\mathbf{x}$ such that the following cost given by

$$C(\mathbf{x}) = \sum_{i \in S, j \in V} C_{ij} x_{ij} + \sum_{i \in V, j \in V, i < j} C_{ij} x_{ij} + \sum_{i \in F, j \in V} C_{ij} x_{ij} \quad (2)$$

is minimized subject to the following constraints.

i. $\sum_{j \in V} x_{ij} = 1, \forall i \in S$

ii. $\sum_{j \in S} x_{ij} + \sum_{j \in V, i < j} x_{ij} + \sum_{j \in V, j < i} x_{ji} + \sum_{j \in F} x_{ij} = 2, \forall i \in V$

[2]If i, j, k denote the sites to be visited and C_{ij}, the distance to travel from the i^{th} site to the j^{th} site, then satisfying triangle inequality means that $C_{ij} \leq C_{ik} + C_{kj}$. The cost function is symmetric if $C_{ij} = C_{ji}$.

[3]The problem discussed in this paper is different from the multi-vehicle, multi-depot TSP problem considered in [5]. In [5], each vehicle starts and ends at the same depot location, whereas in this paper, the starting and the final location of the vehicle are different. This paper also differs in the aspect that each vehicle is allowed to reach any one of the possible final locations.

iii. $\sum_{j\in V} x_{ij} = 1, \forall i \in F$

iv. $x_{ri} = 1, \forall i \in S$

v. $\sum_{\{i,j\}\in U_1} x_{ij} \leq |U_1| - 1, \ \forall U_1 \subseteq \{r\} \bigcup S \bigcup V \bigcup F$

vi. $x_{r'i} = 1, \forall i \in F$

vii. $\sum_{\{i,j\}\in U_2} x_{ij} \leq |U_2| - 1, \ \forall U_2 \in \{\{r'\} \bigcup T : T \subseteq S \bigcup V \bigcup F\}$

viii. $x_{ij} \in \{0,1\}, \forall i,j \in \{r,r'\} \bigcup S \bigcup V \bigcup F, i < j$ if $i,j \in V$.

Constraints i,ii,iii enforce the degree constraints on each vertex. Constraint v removes any possibility of a cycle in the graph induced by the vertices $\{r\} \bigcup S \bigcup V \bigcup F$. Since the edges connecting the root vertex r and each of the vertices in S must be selected by constraint iv, constraints iv and v state a requirement that there exists no path through vertices only in $V \bigcup F$ that connects any two, distinct vertices in S. Constraints vi and vii also play a similar role as iv and v. They state the requirement that there exists no path through vertices only in $S \bigcup V$ that connects any two, distinct vertices in F. To facilitate further analysis, an additional constraint is added to the above problem without changing the set of feasible solutions. This is stated in the following lemma.

Lemma II.1: The following additional constraint can be added to problem II.1 without changing its set of feasible solutions:

$$\sum_{i\in S, j\in V} x_{ij} + \sum_{i\in V, j\in V, i<j} x_{ij} + \sum_{i\in F, j\in V} x_{ij} = n + k \quad (3)$$

Proof: Summing the constraint i in problem II.1 for all vertices in S, we get $\sum_{i\in S, j\in V} x_{ij} = k$. Similarly, summing constraint iii, we get $\sum_{i\in F, j\in V} x_{ij} = k$. Summing constraint ii for all vertices in V and using the fact that $\sum_{i\in V, j\in V, i<j} x_{ij} = \sum_{i\in V, j\in V, j<i} x_{ji}$, we get,

$$\begin{aligned} \sum_{i\in S, j\in V} x_{ij} + 2 \sum_{i\in V, j\in V, i<j} x_{ij} + \sum_{i\in F, j\in V} x_{ij} &= 2n \\ \Rightarrow \sum_{i\in V, j\in V, i<j} x_{ij} &= n - k. \end{aligned}$$

Therefore,

$$\begin{aligned} \sum_{i\in S, j\in V} x_{ij} + \sum_{i\in V, j\in V, i<j} x_{ij} + \sum_{i\in F, j\in V} x_{ij} &= k + (n-k) + k \\ &= n + k. \end{aligned} \quad (4)$$

■

Therefore problem II.1 can be reformulated with the additional constraint as follows:

Problem II.2: The objective is to find the incidence matrix $\mathbf{x}$ such that the following cost given by

$$C(\mathbf{x}) = \sum_{i\in S, j\in V} C_{ij}x_{ij} + \sum_{i\in V, j\in V, i<j} C_{ij}x_{ij} + \sum_{i\in F, j\in V} C_{ij}x_{ij} \quad (5)$$

is minimized subject to the following constraints.

i. $\sum_{j\in V} x_{ij} = 1, \forall i \in S$

ii. $\sum_{j\in S} x_{ij} + \sum_{j\in V, i<j} x_{ij} + \sum_{j\in V, j<i} x_{ji} + \sum_{j\in F} x_{ij} = 2, \forall i \in V$

iii. $\sum_{j\in V} x_{ij} = 1, \forall i \in F$

iv. $\sum_{i\in S, j\in V} x_{ij} + \sum_{i\in V, j\in V, i<j} x_{ij} + \sum_{i\in F, j\in V} x_{ij} = n + k$

v. $x_{ri} = 1, \forall i \in S$

vi. $\sum_{\{i,j\}\in U_1} x_{ij} \leq |U_1| - 1, \ \forall U_1 \subseteq \{r\} \bigcup S \bigcup V \bigcup F$

vii. $x_{r'i} = 1, \forall i \in F$

viii. $\sum_{\{i,j\}\in U_2} x_{ij} \leq |U_2| - 1, \ \forall U_2 \in \{\{r'\} \bigcup T : T \subseteq S \bigcup V \bigcup F\}$

ix. $x_{ij} \in \{0,1\}, \forall i,j \in \{r,r'\} \bigcup S \bigcup V \bigcup F, i < j$ if $i,j \in V$.

Let the constraints i,ii,iii be denoted by $A_1\mathbf{x} = B_1$. The constraints defined by iv,v,vi,vii,viii can be written as $A_2\mathbf{x} \leq B_2$. Since the cycle elimination constraints ensure that each x_{ij} can never exceed 1, the above minimization problem can be restated as

$$C_{opt} = \min_{\mathbf{x}} \{C(\mathbf{x}) : A_1\mathbf{x} = B_1, A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0, \mathbf{x} \text{ is an integer}\}. \quad (6)$$

The LP relaxation of this problem is:

$$C_{lp} = \min_{\mathbf{x}} \{C(\mathbf{x}) : A_1\mathbf{x} = B_1, A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0\}. \quad (7)$$

Before, we present the algorithms for calculating the lower and the upper bounds, just as how the spanning tree played an important role in the single vehicle problem [7], a forest with k disjoint trees satisfying the following constraint plays an important role in **MVMDP**: Each tree in the forest spans exactly one vertex from S, exactly one vertex from F and a subset of vertices from V. An illustration of such a forest with 3 components is shown in figure 3. The problem of finding such a constrained forest of minimum cost can be formulated as follows:

Problem II.3: The objective is to find an incidence matrix $\mathbf{x}$ such that the cost given by

$$C(\mathbf{x}) = \sum_{i\in S, j\in V} C_{ij}x_{ij} + \sum_{i\in V, j\in V, i<j} C_{ij}x_{ij} + \sum_{i\in F, j\in V} C_{ij}x_{ij} \quad (8)$$

is minimized subject to the following constraints.

i. $\sum_{i\in S, j\in V} x_{ij} + \sum_{i\in V, j\in V, i<j} x_{ij} + \sum_{i\in F, j\in V} x_{ij} = n + k$

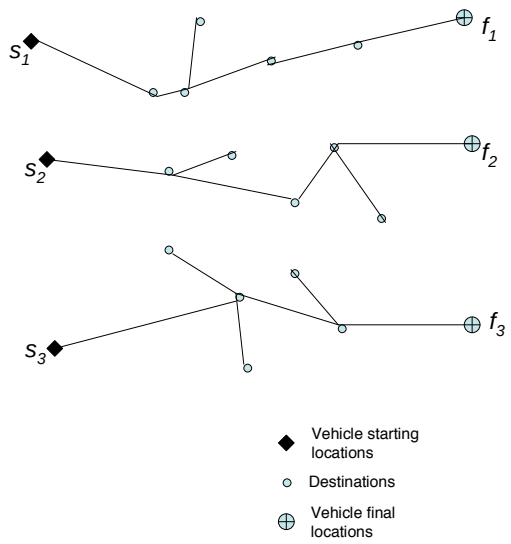

Fig. 3. An example of a constraint forest having 3 vehicles.

ii. $x_{ri} = 1, \forall i \in S$

iii. $\sum_{\{i,j\}\in U_1} x_{ij} \leq |U_1| - 1,\ \forall U_1 \subseteq \{r\} \bigcup S \bigcup V \bigcup F$

iv. $x_{r'i} = 1, \forall i \in F$

v. $\sum_{\{i,j\}\in U_2} x_{ij} \leq |U_2| - 1,\ \forall U_2 \in \{\{r'\} \bigcup T : T \subseteq S \bigcup V \bigcup F\}$

vi. $x_{ij} \in \{0,1\}, \forall i,j \in \{r, r'\} S \bigcup V \bigcup F, i < j$ if $i,j \in V$.

The difference between the problem formulations in II.2 and II.3 is the presence of the degree constraints on each of the vertices in problem II.2. Using the notation in equation 6, the constrained forest problem can be formulated as:

$$C_f = \min_{\mathbf{x}} \{C(\mathbf{x}) : A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0, \mathbf{x} \text{ is an integer}\}. \quad (9)$$

A useful property of the set of feasible solutions denoted by $\{A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0, \mathbf{x}$ is an integer$\}$ is stated in the following theorem. Due to space constraints, this is proved in the appendix.

Theorem II.1: The optimal solutions of $\min_{\mathbf{X}}\{C(\mathbf{x}) : A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0\}$ are integers. This implies that $\min_{\mathbf{X}}\{C(\mathbf{x}) : A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0\} = \min_{\mathbf{X}}\{C(\mathbf{x}) : A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0, \mathbf{x}$ is an integer$\}$.

III. Upper bounds

As discussed in the introduction of the paper, a constant factor approximation algorithm is not yet known for TSP problems for any general cost function. However, approximation algorithms are possible if the costs satisfy triangle inequality. For example, if the cost of travelling between any two vertices is just based on the Euclidean distances, then triangle inequality is easily satisfied. An approximation algorithm for the **MVMDP** is given below.

i. Find the optimal constrained forest (formulated in problem II.3) using the Edmond's matroid intersection algorithm (Refer to the appendix). The output of this algorithm for an example with five vehicles is shown in figure 4.

ii. For each tree corresponding to a vehicle, double its edges to construct its Eulerian graph (figure 5).

iii. Then construct a path for each vehicle based on its Eulerian graph. A path for each vehicle visits each destination in its corresponding Eulerian tour exactly once and reaches the final location (figure 6). This step is similar to the Tarjan's algorithm available for the single Travelling Salesman problem [2].

The following theorem shows this algorithm has a approximation factor of 2.

Theorem III.1: The algorithm solves the **MVMDP** with an approximation factor of 2 in $O((n+2k)^6)$ steps when the costs satisfy triangle inequality.

Proof: From lemma VI.3 in the appendix, computing the minimum constrained forest takes $O((n+2k)^6)$ steps. Steps ii and iii essentially finds an Eulerian tour for each vehicle and requires $O((n+2k)^2)$ steps. Therefore, the complexity of the algorithm is dominated by the first step and hence the algorithm runs in $O((n+2k)^6)$ steps. To prove the bound, note that a feasible solution for the **MVMDP** is also a feasible solution for the constrained forest problem. Hence, $C_f \leq C_{opt}$ (refer to equations 9 and 6). Also, since each edge in doubled in step ii of the algorithm to construct a Eulerian graph for each vehicle, the cost of the solution C_s obtained by short cutting some of the edges in step iii would be upper bounded by $2C_f$ (short cutting does not increase the cost because of the assumption on the triangle inequality). Hence, $C_f \leq C_{opt} \leq C_s \leq 2C_f$. Therefore, the approximation factor is 2. ■

IV. Lower Bounds

In this section, we derive lower bounds for **MVMDP** and show that its LP-relaxation is a tractable problem that can be solved using a matroid intersection algorithm. Let $\mathbf{P}$ denote the set of all feasible paths for the **MVMDP** as defined in equation 6. That is, $\mathbf{P} := \{A_1\mathbf{x} = B_1, A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0, \mathbf{x}$ is an integer$\}$. Similarly, let $\mathbf{F}$ denote the set of all feasible solutions for the constrained forest problem. That is, $\mathbf{F} := \{A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0, \mathbf{x}$ is an integer$\}$. Now, let us perturb the costs C_{ij} to $\overline{C}_{ij} := C_{ij} + \pi_i + \pi_j$ where π_i is a weight assigned to vertex i. The objective function of **MVMDP** in problem II.2 gets modified to $\min_{\mathbf{X}\in\mathbf{P}} \overline{C}(\mathbf{x})$, where,

$$\overline{C}(\mathbf{x}) = C(\mathbf{x}) + \sum_{i\in S} \pi_i + \sum_{i\in V} 2\pi_i + \sum_{i\in F} \pi_i. \quad (10)$$

Similarly, the objective function for the constrained forest problem gets modified to $\min_{\mathbf{X}\in\mathbf{P}} \widetilde{C}(\mathbf{x})$, where,

$$\widetilde{C}(\mathbf{x}) \quad = \quad C(\mathbf{x}) + \sum_{i\in S} \pi_i \sum_{j\in V} x_{ij} +$$

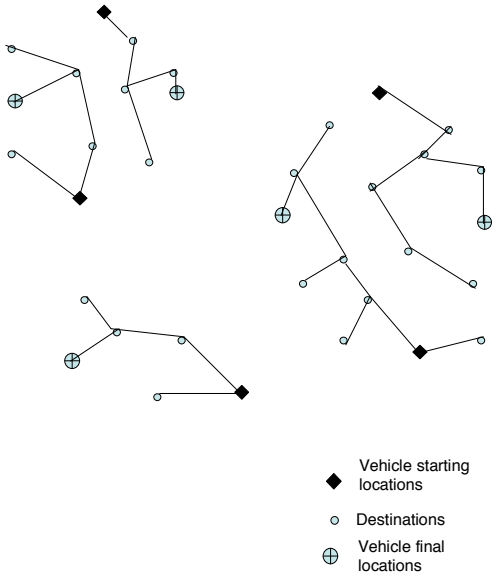

Fig. 4. Step 1 of 2-approx algorithm: Find the optimal constrained forest.

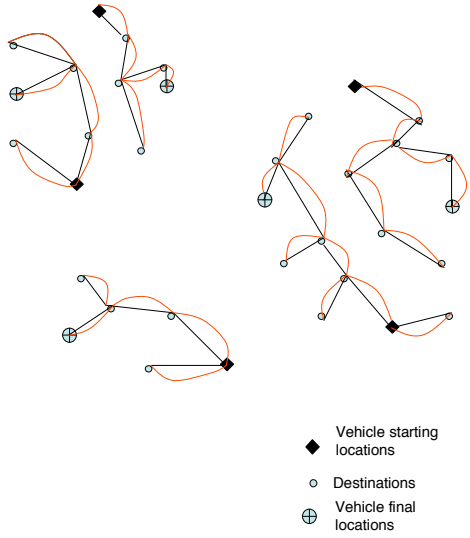

Fig. 5. Step 2 of 2-approx algorithm: Double the edges in each tree to get a Eulerian graph for each vehicle.

$$\sum_{i\in V}\pi_i(\sum_{j\in S}x_{ij}+\sum_{j\in V,i<j}x_{ij}+\sum_{j\in V,j<i}x_{ji}+\sum_{j\in F}x_{ij}) + \sum_{i\in F}\pi_i\sum_{j\in V}x_{ij} \tag{11}$$

Note that a feasible solution for the **MVMDP** is also a feasible solution for the constrained forest problem. Hence we must have, $\min_{\mathbf{X}\in\mathbf{F}}\widetilde{C}(\mathbf{x}) \leq \min_{\mathbf{X}\in\mathbf{P}}\overline{C}(\mathbf{x})$. Substituting for $\overline{C}(\mathbf{x})$ and $\widetilde{C}(\mathbf{x})$ using equations 10, 11 and rearranging the terms we get the following lower bound for **MVMDP**.

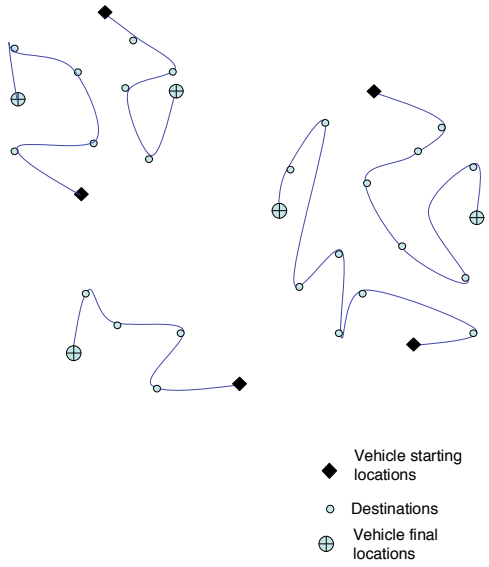

Fig. 6. Step 3 of 2-approx algorithm: Construct a path out of each Eulerian graph.

$$\min_{\mathbf{X}\in\mathbf{F}}\widetilde{C}(\mathbf{x})-\sum_{i\in S}\pi_i-\sum_{i\in V}2\pi_i-\sum_{i\in F}\pi_i\leq\min_{\mathbf{X}\in\mathbf{P}}C(\mathbf{x}) \tag{12}$$

Since the above equation is true for any π, we get the following lemma:

Lemma IV.1:

$$\max_{\pi}w(\pi)\leq\min_{\mathbf{X}}C(\mathbf{x}), \tag{13}$$

where $w(\pi)$ is defined as follows:

$$\begin{aligned} w(\pi) &= \min_{\mathbf{X}\in\mathbf{F}}[C(\mathbf{x})+\sum_{i=S}\pi_i(\sum_{j\in V}x_{ij}-1)+ \\ &\quad \sum_{i\in V}\pi_i(\sum_{j\in S}x_{ij}+\sum_{j\in V,i<j}x_{ij}+\sum_{j\in V,j<i}x_{ji}+\sum_{j\in F}x_{ij}-2) \\ &\quad +\sum_{i\in F}\pi_i(\sum_{j\in V}x_{ij}-1)], \end{aligned} \tag{14}$$

$$= \min_{\mathbf{X}\in\mathbf{F}}[C(\mathbf{x})+\pi^T(A_1\mathbf{x}-B_1)]. \tag{15}$$

The left hand side of equation 13 provides a lower bound to the **MVMDP**. Note that for any fixed π, the inner minimization problem in equation 14 is that of calculating an optimal constrained forest. This can be solved using a matroid intersection algorithm as shown in the appendix. For the single vehicle problem, Held and Karp showed that the LP-relaxation of the single vehicle TSP is actually *equal* to $\max_\pi w(\pi)$. This result was pivotel in realizing tight lower bounds for the single vehicle TSP. We present a similar result for the multiple vehicle case. We state this in the following theorem.

Theorem IV.1: Let $w(\pi)$ and C_{lp} be given by equations 14 and 7 respectively. Then,

$$\max_{\pi} w(\pi) = C_{lp}. \tag{16}$$

Remark: This result hinges on theorem *II.1.* Once the fact that $\min_{\mathbf{X}}\{C(\mathbf{x}) : A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0\} = \min_{\mathbf{X}}\{C(\mathbf{x}) : A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0, \mathbf{x} \text{ is an integer}\}$ is accepted, the remaining part of the proof can be found in Held-Karp's paper [7]. Due to space constraints, the authors refer to a similar proof in [7].

Now the fact that $w(\pi)$ is a concave function of π can be used to generate lower bounds using a ascent algorithm. Essentially we are interested in finding π^* such that $w(\pi^*) = \max_{\pi} w(\pi)$. We present here an ascent algorithm similar to the one discussed in [8]. Let $v := A_1\mathbf{x} - B_1$ (refer equation 15). Let π^k denote the iterate of π at the k^{th} step of the ascent algorithm. Let v^k denote the iterate of v at the k^{th} step of the ascent algorithm. Note that v^k is a function of π^k also. To start with let $\pi^1 = 0$. The ascent method chooses a sequences of vectors π^k such that $\pi^{k+1} = \pi^k + \bar{t}v^k$, where $\bar{t}$ is a appropriate constant. This choice of sequence for π^k would eventually yield an optimal solution π^* and the relevant convergence properties of this method is shown in [8]. Each iterate π^k gives a lower bound $w(\pi^k)$ for the **MVMDP**.

V. Conclusions

This paper presents a 2-approximation algorithm and a lower bounding algorithm for a symmetric multiple vehicle, multiple depot problem. The well known Held-Karp's lower bound available for a single Travelling Salesman Problem is extended to the multiple depot, multiple Travelling Salesman Problem. The algorithms essentially require calculating a constrained forest which can be computed using Edmond's matroid intersection algorithm. Currently, the authors are testing the derived algorithms in this paper with the standard instances available in the literature.

References

[1] The Travelling Salesman Problem and its Variations, Kluwer Academic Publishers, 2002.

[2] Papadimitriou, C.H., and Steiglitz, K., Combinatorial optimization: algorithms and complexity, Prentice-Hall 1982, Dover publications 1998.

[3] Christofides, N., Worst-case analysis of a new heuristic for the travelling salesman problem. In: J. F. Traub (Editor), *Algorithms and Complexity: New Directions and Recent Results*, Academic Press, pp.441, 1976.

[4] Arora, S., Polynomial-time Approximation Schemes for Euclidean TSP and other Geometric Problems, *Proceedings of the 37th Annual Symposium on the Foundations of Computer Science*, pp:2-11, 1996.

[5] Rathinam, S., Sengupta, R., and Swaroop, D., "A Resource Allocation Algorithm for Multi Vehicle Systems with Non-Holonomic Constraints", Accepted in IEEE Transactions on Automation Science and Engineering, 2005.

[6] Lawler, E. L, Combinatorial optimization: networks and matroids, 1976.

[7] Held, M., and Karp, R. M., "The Traveling-Salesman Problem and Minimum Spanning Trees," *Operations Research*, vol. 18, pp.1138-1162, 1970.

[8] Held, M., and Karp, R. M., "The Travelling Salesman Problem and Minimum Spanning Trees: Part II," *Mathematical Programming*, vol.18, pp.1138-1162, 1971.

VI. Appendix

A. Matroids and Matroid Polyhedra

This section presents some of the properties of matroids and their intersections which will be used to prove theorem *II.1.* The discussion in this paper primarily follows the notations and presentation given in Lawler [6]. A matroid $M = (E, \mathcal{I})$ is a structure in which E has a finite set of elements and $\mathcal{I}$ is a family of subsets of E such that the following holds:
i. $\Phi \in \mathcal{I}$.
ii. If $S_1 \in \mathcal{I}$ and $S_2 \subset S_1$, then $S_2 \in \mathcal{I}$.
iii. Let $I_1 \in \mathcal{I}$ and $I_2 \in \mathcal{I}$. If $\mid I_2 \mid > \mid I_1 \mid$, then $\exists e \in I_2 - I_1$ such that $I_1 \bigcup \{e\} \in \mathcal{I}$.
Any subset $I \in \mathcal{I}$ is said to be an independent set of the matroid $M = (E, \mathcal{I})$. The rank of any subset $S \subseteq E$, denoted by $r(S)$, is defined as the cardinality of the maximal independent subset of S. The span of a set $S \subset E$, $span(S)$, is defined as the maximal superset of S having the same rank as S. A set S is called a closed set if $S = span(S)$.

An example of a matroid is $M = (E, \mathcal{I})$ defined on a graph G, where E contains the edges of G and $I = \{S : S \subseteq E$ and S contains no cycles$\}$. Note that any spanning tree of G is an element in $\mathcal{I}$. Infact, a spanning tree corresponds to a maximal element in $\mathcal{I}$. A maximal independent set of a matroid is called the base. Hence, computing a maximum weighted spanning tree over a graph G is equivalent to finding the base in the matroid that has maximum weight. Assuming that the each edge in E has a non-negative weight, finding the maximum weighted spanning tree problem can be posed as a problem of finding an element in $\mathcal{I}$ that has maximum weight. Let the set of edges in E be denoted as $\{e_1, e_2 ... e_m\}$. Let the matrix A denote the incidence matrix of the closed sets. That is, $A_{ij} = 1$ if edge e_i is present in the closet set j and 0 otherwise. Let the rank vector be $r = \{r_1, r_2 ... r_p\}$, where r_j denotes the rank of the closed set j and p the number of closed sets present in E. Let x be the incidence vector which determines whether an edge is picked or not(i.e. $x_i = 1$ if e_i is chosen and $x_i = 0$ otherwise). Let c_i denote the cost of an edge $e_i \in E$. Then the maximum spanning tree problem can be formulated as

$$\max\{cx : Ax \leq r, x \geq 0, x_i \text{ an integer}\}. \tag{17}$$

A theorem by Edmonds shows that the extremal points of $\{x : Ax \leq r, x \geq 0\}$ are integers. In fact, the vertices of the convex polyhedron defined by $\{x : Ax \leq r, x \geq 0\}$ are in one-to-one correspondence to the elements in $\mathcal{I}$. This enables one to formulate the maximal weight spanning tree problem as a linear programming problem. This step was crucial in the results of the Held-Karp's algorithm [7] for the single TSP. In this paper we show that the constrained forest as discussed in section II.3 also shares this property. It turns out that this constrained forest is an element that lies in the intersection of two matroids. Edmonds showed

that that the intersection of two matroid polyhedra has integer solutions. This is stated in the following theorem. Let M_1 and M_2 be any two matroids defined over the same set of elements E. Let A and B be the closed set incidence matrices of M_1 and M_2 respectively. Let r and s be the rank vectors associated with these two matrices.

Theorem VI.1: (Edmonds) For any two matroids M_1 and M_2, all vertices of the convex polyhedron defined by the system of linear equations $\{x : Ax \leq r, Bx \leq s, x \geq 0\}$ have integer solutions. Moreover, the vertices and the intersections of the two matroids are in one-to-one correspondence.

B. Proof of Theorem II.1

It is first shown that solving the minimum cost constraint forest problem formulated in section II is equivalent to finding a related subgraph of maximum weight. The equivalent subgraph problem is formulated as follows:

Problem VI.1: The objective is to find an incidence matrix $\mathbf{x}$ such that the cost given by

$$\begin{aligned} W(\mathbf{x}) &= \sum_{j\in S} W_{rj}x_{rj} + \sum_{i\in S, j\in V} W_{ij}x_{ij} + \sum_{i\in V, j\in V, i<j} W_{ij}x_{ij} \\ &\quad + \sum_{i\in F, j\in V} W_{ij}x_{ij} + \sum_{j\in F} W_{r'j}x_{r'j} \end{aligned}$$

is maximized subject to the following constraints.

i. $\sum_{\{i,j\}\in U_1} x_{ij} \leq |U_1| - 1$, for any $U_1 \subseteq \{r\}\bigcup S\bigcup V\bigcup F$

ii. $\sum_{\{i,j\}\in U_2} x_{ij} \leq |U_2| - 1$, for any $U_2 \in \{\{r'\}\bigcup T : T \subseteq S\bigcup V\bigcup F\}$

iii. $x_{ij} \in \{0,1\}, \forall i,j \in \{r,r'\}S\bigcup V\bigcup F, i<j$ if $i,j \in V$.

In this problem the weights $W_{ij} > 0$ (defined later) for all $i,j \in \{r,r'\}\bigcup S\bigcup V\bigcup F$. Also let the constraints in the above subgraph problem be written as $\{A_s\mathbf{x} \leq B_s, \mathbf{x} \geq 0, \mathbf{x}$ is an integer$\}$. Therefore problem VI.1 can be concisely formulated as

$$\max_{\mathbf{x}}\{W(\mathbf{x}) : A_s\mathbf{x} \leq B_s, \mathbf{x} \geq 0, \mathbf{x} \text{ is an integer}\} \qquad (18)$$

The equivalency is a result stated in the following theorem.

Theorem VI.2: There exists a cost function W($\mathbf{x}$) such that:

$$\begin{aligned} &\text{i. } \arg\min_{\mathbf{x}}\{C(\mathbf{x}) : A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0, \mathbf{x} \text{ is an integer}\} = \\ &\quad \arg\max_{\mathbf{x}}\{W(\mathbf{x}) : A_s\mathbf{x} \leq B_s, \mathbf{x} \geq 0, \mathbf{x} \text{ is an integer}\} \\ &\text{ii. Also, } \arg\min_{\mathbf{x}}\{C(\mathbf{x}) : A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0\} = \\ &\quad \arg\max_{\mathbf{x}}\{W(\mathbf{x}) : A_s\mathbf{x} \leq B_s, \mathbf{x} \geq 0\}. \end{aligned}$$

Proof: The optimal solution for problem II.3 is given by

$$\begin{aligned} \mathbf{x}_{opt} &= \arg\min_{\mathbf{x}}\{C(\mathbf{x}) : A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0, \mathbf{x} \text{ is an integer}\} \\ &= \arg\max_{\mathbf{x}}\{-C(\mathbf{x}) : A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0, \mathbf{x} \text{ is an integer}\} \\ &= \arg\max_{\mathbf{x}}\{-C(\mathbf{x}) + \alpha(n+k) \\ &\quad : \alpha \text{ is some constant}, A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0, \mathbf{x} \text{ is an integer}\}. \\ &= \arg\max_{\mathbf{x}}\{-C(\mathbf{x}) + \alpha(\sum_{i\in S, j\in V} x_{ij} + \sum_{i\in V, j\in V, i<j} x_{ij} \\ &\quad + \sum_{i\in F, j\in V} x_{ij}) : A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0, \mathbf{x} \text{ is an integer}\}. \\ &= \arg\max_{\mathbf{x}}\{W'(\mathbf{x}) : A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0, \mathbf{x} \text{ is an integer}\}. \end{aligned} \qquad (19)$$

In the above equation, $W'(\mathbf{x})$ is defined as:

$$W'(\mathbf{x}) = \sum_{i\in S, j\in V} W_{ij}x_{ij} + \sum_{i\in V, j\in V, i<j} W_{ij}x_{ij} + \sum_{i\in F, j\in V} W_{ij}x_{ij}, \qquad (20)$$

where $W'_{ij} = -C_{ij} + \alpha$ with α being a constant $= 1 + \max_{i,j\in S\bigcup V\bigcup F} C_{ij}$. This ensures that the weights W'_{ij} are ¿ 0. Continuing with the above argument, the optimal solution for problem II.3 is given by:

$$\begin{aligned} \mathbf{x}_{opt} &= \arg\max_{\mathbf{x}}\{W'(\mathbf{x}) : A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0, \mathbf{x} \text{ is an integer}\} \\ &= \arg\max_{\mathbf{x}}\{W'(\mathbf{x}) + 2kM : M \text{ a constant}, \\ &\quad A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0, \mathbf{x} \text{ is an integer}\} \\ &= \arg\max_{\mathbf{x}}\{W'(\mathbf{x}) + M(\sum_{j\in S} x_{rj} + \sum_{j\in S} x_{r'j}) : \\ &\quad M \text{ a constant}, A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0, \mathbf{x} \text{ is an integer}\} \\ &= \arg\max_{\mathbf{x}}\{W(\mathbf{x}) : A_2\mathbf{x} \leq B_2, \mathbf{x} \geq 0, \mathbf{x} \text{ is an integer}\}, \end{aligned} \qquad (21)$$

where $W(\mathbf{x})$ is defined such that $W_{ij} = W'_{ij}$ for $i,j \in S\bigcup V\bigcup F$, $W_{rj} = M$ for $j \in S$ and $W_{r'j} = M$ for $j \in F$. M is chosen such that it is a large value compared to the other weights in the graph, i.e. $M = 1 + \max_{i,j\in S\bigcup V\bigcup F} W'_{ij}$.

Since, the problem is a maximization problem with positive weights, it is evident that the optimal solution in equation 21 will have $x_{rj} = 1\ \forall j \in S$ and $x_{r'j} = 1\ \forall j \in F$. Therefore the constraints $x_{rj} = 1\ \forall j \in S$ and $x_{r'j} = 1\ \forall j \in F$ in $A_2\mathbf{x} \leq B_2$ are redundant and can be removed. Due to the same reason $\sum_{i\in S, j\in V} x_{ij} + \sum_{i\in V, j\in V, i<j} x_{ij} + \sum_{i\in F, j\in V} x_{ij} = n + k$ can be replaced by $\sum_{j\in S} x_{rj} + \sum_{i\in S, j\in V} x_{ij} + \sum_{i\in V, j\in V, i<j} x_{ij} + \sum_{i\in F, j\in V} x_{ij} = n + 2k$ and $\sum_{i\in S, j\in V} x_{ij} + \sum_{i\in V, j\in V, i<j} x_{ij} + \sum_{i\in F, j\in V} x_{ij} + \sum_{j\in F} x_{r'j} = n + 2k$. These two constraints are just a restatement of the following two:

$$\sum_{\{i,j\}\in U_1} x_{ij} = |U_1| - 1, U_1 = \{r\}\bigcup S\bigcup V\bigcup F,$$

$$\sum_{\{i,j\}\in U_2} x_{ij} \quad = \quad |U_2|-1, U_2 = \{r'\}\bigcup S\bigcup V\bigcup F. \tag{22}$$

Again, since the problem is a maximization problem with positive weights, equations 22 can be replaced with

$$\begin{aligned}\sum_{\{i,j\}\in U_1} x_{ij} &\leq |U_1|-1, U_1 = \{r\}\bigcup S\bigcup V\bigcup F,\\ \sum_{\{i,j\}\in U_2} x_{ij} &\leq |U_2|-1, U_2 = \{r'\}\bigcup S\bigcup V\bigcup F.\end{aligned} \tag{23}$$

But these two constraints are already part of the cycle elimination constraints present in iii and v of the constrained forest problem II.3. They are redundant and can be removed. Essentially we have removed constraints i,ii and iii from the constrained forest problem (i.e. removed from $A_2\mathbf{x} \leq B_2$). We are left with only two cycle elimination constraints which are essentially what is present in the maximum subgraph problem. Therefore, part i of the theorem has been proved. The same proof can be used to prove part ii as removing the constraint $\{\mathbf{x}$ is an integer$\}$ doesn't alter any step of the proof. ■

The claim now is that the maximum subgraph problem can be formulated as a maximum weighted matroid intersection problem. To see this, we write the feasible set of the maximum subgraph problem as the intersection of two sets in the following lemma:

Lemma VI.1: The feasible set in the maximum subgraph problem can be written as the intersection of sets S_1 and S_2, where S_1 and S_2 are defined in the following way:

$$\begin{aligned}S_1 \quad = \quad \{\mathbf{x}: &\sum_{\{i,j\}\in U_1} x_{ij} \leq |U_1|-1 \ \forall U_1 \subseteq \{r\}\bigcup S\bigcup V\bigcup F,\\ &x_{ij}\in\{0,1\}\ \forall i,j\in\{r,r'\}S\bigcup V\bigcup F, i<j \text{ if } i,j\in V\}\end{aligned}$$

$$\begin{aligned}S_2 \quad = \quad \{\mathbf{x}: &\sum_{\{i,j\}\in U_2} x_{ij} \leq |U_2|-1 \ \forall U_2 \subseteq \{r'\}\bigcup S\bigcup V\bigcup F,\\ &x_{ij}\in\{0,1\}\ \forall i,j\in\{r,r'\}S\bigcup V\bigcup F, i<j \text{ if } i,j\in V\}\end{aligned}$$

As discussed in the previous section on matroids, note that the structure defined by $(E_1,\mathcal{G}_1)$ with $E_1 = \{e_{rj} : j\in S\}\bigcup\{e_{ij} : i\in S, j\in V\}\bigcup\{e_{ij} : i\in S, j\in S, i<j\}\bigcup\{e_{ij} : i\in V, j\in F\}$ and $\mathcal{G}_1 = \{Y : Y\subseteq E_1$ and Y contains no cycles$\}$ is a matroid. So is $(E_2,\mathcal{G}_2)$ with $E_2 = \{e_{ij} : i\in S, j\in V\}\bigcup\{e_{ij} : i\in S, j\in S, i<j\}\bigcup\{e_{ij} : i\in V, j\in F\}\bigcup\{e_{r'j} : j\in F\}$ and $\mathcal{G}_2 = \{Y : Y\subseteq E_2$ and Y contains no cycles$\}$. Let $R = \{e_{rj} : j\in S\}$ and $R' = \{e_{r'j} : j\in F\}$. Now define a structure $M_1 = (E,\mathcal{I}_1)$ such that $E = E_1\bigcup R'$ and $\mathcal{I}_1 = \{Y\bigcup Z : Y\subseteq E_1$ and Y contains no cycles, $Z\subseteq R'\}$. It is easy to see that each element in $\mathcal{I}_1$ contains a set of edges that corresponds to a feasible solution in S_1 and vice versa. Similarly define $M_2 = (E,\mathcal{I}_2)$ on the same set of edges E with $\mathcal{I}_2 = \{Y\bigcup Z : Y\subseteq E_2$ and Y contains no cycles, $Z\subseteq R\}$.

Lemma VI.2: The structures M_1 and M_2 are matroids.

Proof: We prove that M_1 is a matroid. A similar proof carries over to M_2 also. According to the definition of a matroid in the previous section, the condition to check is iii as conditions i and ii are trivially satisfied. Condition iii says that if $I_1\in\mathcal{I}_1$ and $I_2\in\mathcal{I}_1$ and if $|\ I_2\ |>|\ I_1\ |$, then there should exist an edge $a\in I_2-I_1$ such that $I_1\bigcup\{a\}\in\mathcal{I}_1$. If $|\ I_2\ |>|\ I_1\ |$, then there are two possibilities: either $|\{e : e\in R', e\in I_2\}| > |\{e : e\in R', e\in I_1\}|$ or $|e : e\in E_1, e\in I_2| > |e : e\in E_1, e\in I_1|$. If the first case is true, then there exists an edge $a\in\{e : e\in R', e\in I_2-I_1\}$ such that $\{a\}\bigcup I_1\in\mathcal{I}_1$. If the second case is true, then since both $|e : e\in E_1, e\in I_2|$ and $|e : e\in E_1, e\in I_1|$ are independent sets of $\mathcal{G}_1$, there must exist an edge a in $\{e : e\in E_1, e\in I_2-I_1\}$ again such that $\{a\}\bigcup I_1\in\mathcal{I}_1$. ■

From the discussion on the intersection of matroid polyhedra, we get the following theorem:

Theorem VI.3:

$$\text{The extreme points of } \{\mathbf{x} : A_s\mathbf{x}\leq B_s, \mathbf{x}\geq 0\} \text{ are integers.} \tag{24}$$

Proof: From the definition of the matroids M_1 and M_2 preceding lemma *VI.2*, any element in S_1 and S_2 is in one to one correspondence with an independent set in M_1 and M_2 respectively. Now, using Edmonds theorem *VI.1*, we know that the extremal points in the intersection of two matroid polyhedra are integers. Hence the theorem follows. ■

Theorem II.1 follows by combining theorems *VI.2* and *VI.3*.

Lemma VI.3: The constrained forest can be found in $O(|E|^3)$ steps where $|E| = (n+2k)^2$.

Proof: The primal algorithm for the matroid intersection problem as explained in [6] has a complexity of $O(|E|R^3 + |E|R^2C(|E|))$ where $R = \min\{R_1, R_2\}$ and $C(|E|) = \max\{C_1(|E|), C_2(|E|)\}$. R_1 and R_2 are the ranks of matroids M_1 and M_2 respectively. Similarly $C_1(|E|)$ and $C_2(|E|)$ is the running time of the subroutine available for independence testing of matroids M_1 and M_2. For the constrained forest problem with M_1 and M_2 as defined in lemma VI.1, $R = O(\sqrt{(}|E|))$ and $C(|E|) = |E|$ where $|E| = (n+2k)^2$. Therefore the number of computations required is $O(|E|^3)$. ■