An Approximation algorithm for a 2-Depot, Heterogeneous Vehicle Routing Problem

Sai Krishna Yadlapalli[1], Sivakumar Rathinam[2], Swaroop Darbha[3]

Abstract— **Routing problems involving heterogeneous vehicles naturally arise in several civil and military applications due to fuel and motion constraints of the vehicles. These vehicles can differ either in their motion constraints or sensing capabilities. Approximation algorithms are useful for solving these routing problems because they produce solutions that can be efficiently computed and are relatively less sensitive to the noise in the data. In this paper, we present the *first* approximation algorithm for a 2-Depot, Heterogeneous Vehicle Routing Problem when the cost of direct travel between any pair of locations is no costlier than the cost of travel between the same locations and going through any intermediate location.**

I. INTRODUCTION

Surveillance applications involving Unmanned Aerial Vehicles (UAVs) or ground robots require multiple vehicles with different capabilities to visit a set of locations. The cost of traveling between two locations can depend on the type of the vehicle being used. In these applications, it is reasonable to assume that the amount of fuel used by a vehicle is directly proportional to the distance traveled by the vehicle. This paper addresses an important routing problem involving two heterogeneous vehicles that arises due to fuel constraints. Specifically, the 2-Heterogeneous Vehicle Routing Problem (2-HVRP) is as follows: Given a set of targets and two heterogeneous vehicles, find a tour for each vehicle such that each target is visited exactly once and the sum of the costs traveled by both the vehicles is minimum. The cost of traveling between two targets depends on the type of vehicle used and the position of the targets.

There are several applications where routing problems such as the 2-HVRP could arise. In UAV applications, it is possible that the vehicles have different

1. Graduate Student, Department of Mechanical Engineering, Texas A & M University, College Station, Texas, U.S.A 77843.
2. Assistant Professor, Department of Mechanical Engineering, Texas A & M University, College Station, Texas, U.S.A 77843.
3. Associate Professor, Department of Mechanical Engineering, Texas A & M University, College Station, Texas, U.S.A 77843.

constraints on their maximum speeds depending on the vehicle type. Even if we ignore the minimum turning radius constraints of the vehicles when the targets are reasonably far apart, the distance travelled between any two targets is still dependent on the type of the vehicle. These routing problems could also arise in applications involving ground robots such as the reed-shepp vehicles.

The 2-HVRP is a generalization of the single Hamiltonian Path Problem (HPP) and is NP-Hard [1]. Therefore, we are interested in developing approximation algorithms for the 2-HVRP. An $\alpha-$approximation algorithm is an algorithm that

- has a polynomial-time running time, and
- returns a solution whose cost is within α times the optimal cost.

Aiming for approximation algorithms is reasonable in the context of path planning of unmanned aerial vehicles with motion constraints because the cost of traveling between any two targets for an unmanned aerial vehicle can depend on several factors including wind disturbances. Hence, it is appropriate to devise approximation algorithms for these planning problems that are relatively inexpensive than devise algorithms that opt for exact solutions.

Currently, there are no algorithms in the literature with a constant approximation factor for the 2-HVRP. We will assume that cost of traveling from an origin to a destination directly is no more expensive than the cost of traveling from the same origin to the destination through an intermediate location. We say that the costs associated with the problem satisfy the triangle inequality if they satisfy the above property. It is currently known that there cannot exist a constant factor approximation algorithm for a single Hamiltonian Path Problem or a Traveling Salesman Problem if the triangle inequality is not satisfied unless $P = NP$. In this paper, we present the *first* 8-approximation algorithm for the 2-HVRP when the costs satisfy the triangle inequality.

The 2-HVRP is related to a well known class of

problems that has received significant attention in the area of combinatorial optimization. These problems include the Traveling Salesman Problem (TSP) and the Hamiltonian Path Problem (HPP) and their generalizations [1], [2], [3], [4]. There are few approximation algorithms that are available for the generalizations of the TSP and the HPP. As this article deals with constant factor approximation algorithms, henceforth, in this article, we assume that the costs satisfy the triangle inequality[1]. The symmetric TSP has two well known approximation algorithms - the $2-$approximation algorithm obtained by doubling the minimum spanning tree (MST) and the $1.5-$approximation algorithm of Christofides obtained through the construction of MST and a weighted non-bipartite matching of nodes of MST with odd degree [5].

Rathinam et al. have provided $2-$approximation algorithms for variants of the homogenous, multiple TSP and HPP in [7],[8][9]. Currently, there are no approximation algorithms for any heterogeneous, multiple TSP or HPP known in the literature. In this paper, we present the first 8-approximation algorithm for the 2-HTSP when the cost of the edges joining any two targets satisfy the triangle inequality.

II. PROBLEM FORMULATION

Let T represent all the targets to be visited and $D = \{d_1, d_2\}$ denote the two depots (initial locations) corresponding to the first and the second vehicle respectively. Let E be the set of all the edges joining any two distinct vertices in $V = T \bigcup D$. Now, consider the undirected graph $G = (V, E)$. Let the cost of traversing from vertex i to vertex j for the first (second) vehicle be C_{ij}^1 (C_{ij}^2). The costs are assumed to satisfy the triangle inequality, *i.e.*, for every $i, j, k \in V, i \neq j \neq k$, $C_{ij}^1 + C_{jk}^1 \geq C_{ik}^1$ and $C_{ij}^2 + C_{jk}^2 \geq C_{ik}^2$.

Let the number of targets visited by the ith vehicle be k_i. In the trivial case where vehicle i does not visit any target, $k_i = 0$ and the corresponding travel cost for the ith vehicle is zero. In the general case where $k_i > 0$, let the sequence of vertices visited by vehicle i be denoted as $S_i = \{d_i, p_1^i, p_2^i, \cdots, p_{k_i}^i\}$ where $p_1^i, \cdot, p_{k_i}^i \in T$. The total travel cost corresponding to this sequence for the ith vehicle is defined as $Cost_i = C^i(d_i, p_1^i) + \sum_{j=2}^{k_i} C^i(p_{j-1}^i, p_j^i)$. The objective of 2-HVRP is to find the sequence of vertices to visited for each vehicle, *i.e.*, S_1, S_2 such that each target is visited at least once and

[1]If the costs do not satisfy the triangle inequality, we know that there cannot be constant factor approximation algorithms for the TSP or the HPP.

the total travel cost of both the vehicles, $Cost_1 + Cost_2$, is minimized.

III. APPROXIMATION ALGORITHM *via* A 2-COMPONENT HETEROGENEOUS MINIMUM SPANNING FOREST

We adapt the basic approach available for few variants of the TSPs to devise an approximation algorithm for the 2-HVRP. The main steps in this approach are as follows:

- Remove the degree constraints in the given variant of the TSP to find a suitable relaxation that can be solved in polynomial time. Solving this relaxation yields a network that spans all the vertices with minimum cost. For example, for a TSP, the network is a minimum spanning tree. For a multiple TSP, the network is a minimum spanning forest such that no two depots are connected.
- Double the edges in the minimum spanning network to obtain a connected, Eulerian subgraph for each vehicle.
- Find an Eulerian tour in each connected subgraph. The Eulerian tour traverses each edge in its connected subgraph exactly once.
- By short cutting each Eulerian tour, a path or a tour can be obtained for each vehicle such that each target is visited exactly once.

It has been shown that the above approach yields a 2-approximation algorithm for few variants of the TSP [10], [7], [8]. In this paper, we use a similar approach to develop an approximation algorithm for the 2-HVRP. The important step in the above approach is to find an appropriate relaxation of the given TSP or the routing problem that can can solved in polynomial time. By removing the degree constraints of the 2-HVRP, we obtain a relaxation that requires calculating a Heterogeneous Minimum Spanning Forest (HMSF).

A heterogeneous spanning forest consists of two *disjoint* trees rooted at d_1 and d_2, so that all the targets in T are spanned and the depots are not connected. The cost of the tree rooted at d_1 is computed with the edge costs associated with the first vehicle while the cost of tree rooted at d_2 is computed with the edge costs associated with the second vehicle. A HMSF is a heterogeneous spanning forest where the sum of the cost of the two trees is minimum. The problem of finding the HMSF is referred to as the HMSF problem in this paper. Though it is not clear whether there is a polynomial time algorithm that can find a HMSF, in

this paper, we prove that there is a 4-approximation algorithm for the HMSF problem.

The difficulty in developing an approximation algorithm for the HMSF lies in finding a suitable partition of the target vertices that must be visited for each of the vehicles. We do this by posing the HMSF problem as a multi-commodity flow problem as follows: Suppose there are n distinct commodities corresponding to each of the n targets, and at least one unit of each commodity is required to be delivered to its corresponding target by either of the vehicles. If the commodity is delivered by the ith vehicle to a target, then that commodity is routed through those edges that only carry commodities from the ith vehicle. The 2-HTSP may be posed as the construction of trees for the vehicles such that the combined cost is minimum and at least one unit of commodity specific to each target is delivered by either one of the vehicles.

Before we present the approximation algorithm for the HMSF problem, we formulate this multi-commodity flow problem as an integer program. Let p_{ij}^k denote the flow of kth commodity originating from the first depot and flowing from node i to node j. Let q_{ij}^k be the corresponding flow from the second depot through the directed edge (i, j) to the kth target. Though both the flows, p_{ij}^k, q_{ij}^k, can flow through (i, j), they are constrained in amount by the capacity of the arc (i, j). Let f_{ij} denote whether arc (i, j) is used by the first vehicle in its tour and similarly let g_{ij} denote whether arc (i, j) is used by the second vehicle. It should be noted that the directionality of arc is important here. The following capacity constraints naturally arise:

$$0 \leq p_{ij}^u \leq f_{ij} \quad \forall i, j \in T \cup d_1, \quad (1)$$
$$0 \leq q_{ij}^u \leq g_{ij} \quad \forall i, j \in T \cup d_2. \quad (2)$$

Consider an edge $e = (i, j) \in E$. Vertices i and j are essentially the endpoints of the edge e. Let x_e and y_e represent the variables which decide whether edge e is present in routes of first vehicle and second vehicle respectively. Edge e is present in the tour ($x_e = 1$) of the first vehicle if either there is a directed arc from i to j ($f_{ij} = 1$) or there is a directed arc from j to i ($f_{ji} = 1$). These conditions can be stated as follows:

$$f_{ij} + f_{ji} = x_e \quad \forall e \in E, \quad (3)$$
$$g_{ij} + g_{ji} = y_e \quad \forall e \in E. \quad (4)$$

A shipment of the uth commodity shipped from either of the depots can only be delivered to the uth target. Let ψ_u be the quantity of the uth commodity shipped

to the uth target from the first depot and let η_u be the corresponding quantity shipped from the second depot. The following are the flow balance equations for flows p and q respectively:

$$\sum_{j \in T} p_{ij}^k - p_{ji}^k = \begin{cases} \psi_k & \forall k \in T \text{ and } i = d_1, \\ 0 & \forall i, k \in T \text{ and } i \neq k, \\ -\psi_k & \forall i, k \in T \text{ and } i = k. \end{cases} \quad (5)$$

$$\sum_{j \in T} q_{ij}^k - q_{ji}^k = \begin{cases} \eta_k & \forall k \in T \text{ and } i = d_2, \\ 0 & \forall i, k \in T \text{ and } i \neq k, \\ -\eta_k & \forall i, k \in T \text{ and } i = k. \end{cases} \quad (6)$$

Since at least one unit of commodity is to be shipped for each $u \in T$, we have the following relation:

$$\psi_u + \eta_u \geq 1, \forall u \in T \quad (7)$$

The $2 - HMSF$ may thus be posed as the following integer program:

$$C_{HMSF^*} = \min \sum_{e \in E} C_e^1 x_e + C_e^2 y_e. \quad (8)$$

subject to capacity constraints [1, 2], flow balance constraints [5, 6], directed constraints [3, 4], coupling constraint [7] and the following restriction on the domain of the variables:

$$x_e, f_{ij} \in Z^+ \quad p_{ij}^k, \psi_k \in \Re^+ \quad (9)$$

$$y_e, g_{ij} \in Z^+ \quad q_{ij}^k, \eta_k \in \Re^+ \quad (10)$$

where Z^+ is the set of all positive integers.

The complexity of $2 - HMSF$ is not clear. However, in this article, we provide a 4-*approx* algorithm for the $2 - HMSF$ through the following algorithm.

HMSF Algorithm

1) Relax the integrality constraints in the above IP for the 2-HMSF and solve it. The relaxed program (call it LP^*) can be solved in polynomial time as the number of variables and constraints only scale polynomially with the size of V.

2) Find the optimal fractional quantities of each commodity shipped from both the depots. Partition the targets into two disjoint groups according to which depot ships the maximum amount of commodity to the targets. If both depots ship equal amount of commodity to a target, it does not matter to which group it belongs to. In essence, let $\mathcal{X} = \{k| \quad \psi_k \geq \frac{1}{2}\}$. $\mathcal{X}$ correspond to those targets who have received maximum

shipment of their commodity from the first depot, d_1. Let $\mathcal{Y}$ be the rest of targets.

3) Find a tree spanning the targets $\mathcal{X}$ and the depot d_1 of minimum cost. The minimum cost spanning tree (MST) is computed according to the cost of edges associated with the vehicle starting at depot d_1. Similarly find a minimum-cost tree spanning the targets $\mathcal{Y}$ and the depot d_2. Clearly, this is a feasible solution to the formulated integer program.

Let the optimal cost of the minimum spanning tree corresponding to the first vehicle be denoted by $C_{MST}^1(R)$ where $R = \mathcal{X} \bigcup \{d_1\}$. Similarly, let the optimal cost of the second minimum spanning tree be $C_{MST}^2(R')$ where $R' = \mathcal{Y} \bigcup \{d_2\}$. We now state the main result of this paper:

Theorem 3.1: The cost of the feasible solution produced by the HMSF algorithm is within *four times* the cost of the relaxed linear program and hence, is less than $4C_{HMSF^*}$. That is, the approximation factor of the HMSF Algorithm is 4.

We first outline the gist of the proof before presenting the details of the proof. Let the optimal cost of the relaxed linear program be C_{LP^*}. Corresponding to this cost, let ψ_k^*, η_k^* be the optimal quantities of kth commodity shipped from d_1 and d_2 respectively. We formulate a new linear program LP_1 by replacing the coupling constraint (7) with the following constraints.

$$\psi_k \geq 1 \quad \forall k \in \mathcal{X}, \tag{11}$$
$$\eta_k \geq 1 \quad \forall k \in \mathcal{Y}. \tag{12}$$

Let the optimal cost of this new LP be denoted by C_{LP_1}. The main steps in the proof of theorem 3.1 is as follows:

1) We first prove that the optimal cost of the LP relaxation of the HMSF problem is greater than half the optimal cost of the new linear program, LP_1. That is, $C_{LP^*} \geq \frac{1}{2}C_{LP_1}$.

2) In the HMSF problem, note that the only equations that couple the first set of variables, $\{x_e, f_{ij}, p_{ij}^k, \psi_k^2 \ \forall e = (i,j) \in E, k \in T\}$ with the second set of variables $\{y_e, g_{ij}, q_{ij}^k, \eta_k^2 \ \forall e = (i,j) \in E, k \in T\}$ is through equations (7). After replacing these coupling equations using (11,12), there are no constraints that relate both these sets of variables. Therefore, the new linear program,

LP_1, decomposes into two subproblems. In the first subproblem, $LP(\mathcal{X})$, the objective is to minimize $\sum_{e \in E} C_e^1 x_e$ subject to the constraints in (1,3,11) and $x_e, f_{ij}, p_{ij}^k, \psi_k \in \Re^+$. Similarly, in the second subproblem, $LP(\mathcal{Y})$, the objective is to minimize $\sum_{e \in E} C_e^2 y_e$ subject to the constraints in (2,4,12) and $y_e, g_{ij}, q_{ij}^k, \eta_k \in \Re^+$. Let the optimal cost of these two subproblems be defined as $C_{LP(\mathcal{X})}$ and $C_{LP(\mathcal{Y})}$ respectively. As the linear program, LP_1 decouples into two subproblems $LP(\mathcal{X})$ and $LP(\mathcal{Y})$, it is relatively easy to note that

$$C_{LP_1} = C_{LP(\mathcal{X})} + C_{LP(\mathcal{Y})}. \tag{13}$$

3) In the final step, we prove the following results:

$$\begin{aligned} C_{LP(\mathcal{X})} &\geq \frac{1}{2}C_{MST}^1(R), \\ C_{LP(\mathcal{Y})} &\geq \frac{1}{2}C_{MST}^2(R'). \end{aligned} \tag{14}$$

Recall that $C_{MST}^1(R)$ $(C_{MST}^2(R'))$ is the optimal cost of the minimum spanning tree for the set of vertices defined by $R = \mathcal{X} \bigcup \{d_1\}$ $(R' = \mathcal{Y} \bigcup \{d_2\})$.

Summarizing the results in each of the above steps gives

$$\begin{aligned} C_{LP^*} &\geq \frac{1}{2}C_{LP_1} \\ &\geq \frac{1}{2}C_{LP(\mathcal{X})} + \frac{1}{2}C_{LP(\mathcal{Y})} \\ &\geq \frac{1}{4}C_{MST}^1(R) + \frac{1}{4}C_{MST}^2(R'). \end{aligned} \tag{15}$$

In words, if the results in the outline of the proof are correct, then the sum of the cost of the two minimum spanning trees obtained from the HMSF algorithm is at most equal to four times the optimal LP relaxation cost of the HMSF problem. Since this optimal LP relaxation cost is a lower bound on the optimal cost of the HMSF problem, it follows that the approximation factor of the HMSF algorithm is 4. Therefore, to prove theorem 3.1, it is sufficient to prove the following two lemmas:

Lemma 3.1:

$$C_{LP^*} \geq \frac{1}{2}C_{LP_1}.$$

Lemma 3.2:

$$C_{LP(\mathcal{X})} \geq \frac{1}{2}C_{MST}(R),$$

$$C_{LP(\mathcal{Y})} \geq \frac{1}{2}C_{MST}(R').$$

A. Proof of Lemma 3.1

Let the optimal solution of the LP^* be denoted by $Sol^* = \{x_e^*, y_e^*, f_{ij}^*, g_{ij}^*, p_{ij}^{k*}, q_{ij}^{k*}, \psi_k^*, \eta_k^* \ \forall e = (i,j) \in E, k \in T\}$. We now prove that $2Sol^*$ is also a feasible solution for the linear program, LP_1. This would prove Lemma 3.1. Note that all the constraints in (1), (3) and (5) corresponding to the LP_1 are trivially satisfied as scaling the optimal solution, Sol^*, by a factor of 2 will still satisfy all these constraints. The only constraints that need to be checked for the LP_2 are the ones defined in (11) and (12). Now, note that for any $k \in \mathcal{X}$, $\psi_k^* \geq \frac{1}{2}$ or $2\psi_k^* \geq 1$. Similarly, any $k \in \mathcal{Y}$, $2\eta_k^* \geq 1$. We have shown that $2Sol^*$ is a feasible solution for LP_1. Therefore, $2C_{LP*} \geq C_{LP_1}$. Hence proved.

B. Proof of Lemma 3.2

To prove this Lemma, we first show that $C_{LP(\mathcal{X})} \geq \frac{1}{2}C_{MST}^1(R)$. The proof for $C_{LP(\mathcal{Y})} \geq \frac{1}{2}C_{MST}^2(R')$ follows exactly the same steps.

To prove $C_{LP(\mathcal{X})} \geq \frac{1}{2}C_{MST}^1(R)$, let us first summarize all the constraints and the objective of $LP(\mathcal{X})$ as follows:

$$C_{LP(\mathcal{X})} = \min \sum_{e \in E} C_e^1 x_e,$$

subject to

$$0 \leq p_{ij}^u \leq f_{ij} \quad \forall i,j \in V, u \in V\backslash\{d_1\} \quad (16)$$

$$f_{ij} + f_{ji} = x_e \quad \forall e \in E, \quad (17)$$

$$\sum_j p_{ij}^k - p_{ji}^k = \begin{cases} \psi_k & \forall k \in V\backslash\{d_1\}, i = d, j \in V \\ 0 & \forall i, k \in V\backslash\{d_1\}, i \neq k, j \in V \\ -\psi_k & \forall i, k \in V\backslash\{d_1\}, i = k, j \in V \end{cases} (18)$$

$$\psi_k \geq 1 \forall \quad k \in R\backslash\{d_1\} \quad (19)$$

$$\psi \geq 0, f \geq 0 \text{ and } x_e \in R^+ \text{ for all arcs } e \in E \quad (20)$$

Now, consider a target vertex $t \in R$ and a set $S \subset V$ such that the depot $d_1 \in S$ and $t \notin S$. The vertex t must receive at least one unit of commodity from the depot d_1. From the max-flow min-cut theorem [4],[12], there is a flow of at least one unit from the depot $(d \in S)$ to a terminal node, t, in $V \setminus S$ if and only if $\sum_{e \in \delta(S)} x_e \geq 1$. Since the cut set, $\sum_{e \in \delta(S)} x_e$, is

also equal to $\sum_{e \in \delta(V \setminus S)} x_e$, constraints in (16-19) is equivalent to the following set of constraints:

$$\sum_{e \in \delta(S)} x_e \geq 1 \quad \text{for} \quad S \subset V, S \cap R \neq \phi, R\backslash S \neq \emptyset$$

.

Therefore, the linear program $LP(\mathcal{X})$ can be written as follows:

$$C_{LP(\mathcal{X})} = \min \sum_{e \in E} C_e^1 x_e,$$

subject to

$$\sum_{e \in \delta(S)} x_e \geq 1 \quad \text{for} \quad S \subset V, S \cap R \neq \phi, R\backslash S \neq \phi$$

$$(21)$$

$$x_e \geq 0 \quad \text{for } e \in E \quad (22)$$

If we define $f(S)$ is equal to one whenever $S \subset V, S \cap R \neq \phi, R\backslash S \neq \phi$ and is equal to zero otherwise, the linear program $LP(\mathcal{X})$ can be rewritten as follows:

$$C_{LP(\mathcal{X})} = \min \sum_{e \in E} C_e^1 x_e,$$

subject to

$$\sum_{e \in \delta(S)} x_e \geq f(S) \quad \text{for} \quad S \subset V \quad (23)$$

$$x_e \geq 0 \quad \text{for } e \in E \quad (24)$$

The above formulation of $LP(\mathcal{X})$ is actually a linear programming relaxation of a well known problem in the literature called the steiner tree problem. Given an undirected graph $G = (V, E)$ with edge costs and subset of nodes, a set $R \subset V$, the objective of steiner tree problem is to find a minimum weight tree spanning all the nodes in R. The resulting tree may or may not have the optional nodes (i.e, nodes in $V \backslash R$). The optional nodes are often referred to as the Steiner nodes. Now, we use a result due to Goemans and Williamson [11] that is available for problems of this type to deduce that $C_{LP(\mathcal{X})} \geq \frac{1}{2}C_{MST}^1(R)$. If the edge costs satisfy the triangle inequality, Goemans and Williamson showed that the cost of the minimum spanning tree over the vertices in R is at most twice the optimal cost of $LP(\mathcal{X})$. This result is stated in the following theorem:

Theorem 3.2: (**Goemans and Williamson [11]**) If the costs satisfy the triangle inequality,

$$C_{MST}^1(R) \leq (2 - \frac{2}{|R|})C_{LP(\mathcal{X})}.$$

For more details on the above theorem, the readers are referred to [11],[4]. Using the above theorem it is clear that $C_{LP(\mathcal{X})} \geq \frac{1}{2}C^1_{MST}(R)$. Hence, Lemma 3.2 is proved.

IV. 8-Approximation Algorithm for the 2-HVRP

The approximation algorithm for the 2-HVRP is as follows:

1) Construct a feasible solution, $HMSF^f$, for the HMSF problem using the 4-approximation algorithm presented in the previous section. Let the two trees in this feasible solution corresponding to the two vehicles be denoted as $HMSF^f_1$ and $HMSF^f_2$ respectively.
2) For $i = 1, 2$, double the edges in $HMSF^f_i$ to get an Eulerian graph for vehicle i.
3) Find an Eulerian tour for each of the vehicles and short cut these tours to obtain a path for each of the vehicles such that each target is visited exactly once. Let the paths obtained for vehicle i be denoted as $PATH_i$ for $i = 1, 2$.

The above algorithm has an approximation factor of 8. To show this, let $C(PATH_i)$ be the total cost of all the edges present in $PATH_i$ for $i = 1, 2$. Now, since the costs satisfy the triangle inequality,

$$\sum_{i=1,2} C(PATH_i) \leq 2 \sum_{i=1,2} C_{HMSF^f_i}.$$

We also know that,

$$2 \sum_{i=1,2} C_{HMSF^f_i} = 2C_{HMSF^f}$$
$$\leq 8C_{HMSF^*}(\text{using theorem 3.1})$$
$$\leq 8C_{2-HVRP*}$$

where $C_{2-HVRP*}$ is the optimal cost of the 2-HVRP.

V. Conclusion

In this work, we have considered the heterogeneous version of a Vehicle Routing problem with two depots. Using the procedure adopted in this article, a $4p$ approximation algorithm can be obtained for the multiple depot version of the HVRP where p is the number of vehicles.

References

[1] The Travelling Salesman Problem and its variations, Kluwer Academic Publishers, G. Gutin, A.P. Punnen (Editors), 2002.
[2] Lawler, E., Combinatorial Optimization: Networks and Matroids, Dover Publications, 2000.
[3] Reinelt, G., "The Traveling Salesman - Computational Solutions for TSP Applications," *Lecture Notes in Computer Science*, Number 840, Springer-Verlag, 1994.
[4] Vazirani, V. V., *Approximation Algorithms* Springer, 2001.
[5] N. Christofides, Worst-case analysis of a new heuristic for the Traveling Salesman Problem, Technical Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA, 1976.
[6] M. Bellmore, S. Hong, A note on the symmetric Multiple Travelling Salesman Problem with fixed charges, Operations Research 25 (1977) 871-874.
[7] S. Rathinam, R. Sengupta, S. Darbha, A resource allocation algorithm for multiple vehicle systems with non-holonomic constraints, IEEE Transactions on Automation Science and Engineering 4(1) (2007) 98-104.
[8] Waqar Malik, Sivakumar Rathinam, Swaroop Darbha: An approximation algorithm for a symmetric Generalized Multiple Depot, Multiple Travelling Salesman Problem. Operations Research Letters, 35(6): 747-753 (2007).
[9] S. Rathinam and R. Sengupta, Lower and Upper Bounds for a Multiple Depot Routing Problem, IEEE Conference on Decision and Control, December 2006, San Diego.
[10] C. Papadimitrou and K. Steiglitz, Combinatorial optimization: algorithms and complexity, Prentice-Hall 1982, ; second edition by Dover, 1998.
[11] Goemans, M. X. and Williamson, D. P., A General Approximation Technique for Constrained Forest Problems, SIAM Journal of Computing, Vol. 24, p. 296-317, 1995.
[12] Magnanti Thomas, Wolsey Laurence, "Optimal trees", In: Network Models, Handbook in Operations Research and Management Science, North-Holland, 1995, pp.503-615.