Alternate Solution Approaches for Competitive Hub Location Problems

Richa Tiwari
Sachin Jayaswal
Ankur Sinha

W.P. No. WP 2019-12-01
December 2019

INDIAN INSTITUTE OF MANAGEMENT
AHMEDABAD – 380015
INDIA

Alternate Solution Approaches for Competitive Hub Location Problems

Richa Tiwari
Sachin Jayaswal
Ankur Sinha

Abstract

In this paper, we study the hub location problem of an entrant airline that tries to maximize its share in a market with already existing competing players. The problem is modelled as a non-linear integer program, which is intractable for off-the-shelf commercial solvers, like CPLEX and Gurobi, etc. Hence, we propose four alternate approaches to solve the problem. The first among them uses the Kelly's cutting plane method, the second is based on a mixed integer second order conic program reformulation, the third uses the Kelly's cutting plane method within Lagrangian relaxation, while the fourth uses second order conic program within Lagrangian relaxation. The main contribution of this paper lies in the fourth approach, which along with refinements is the most efficient. Many of the problem instances that were not solvable using standard techniques, like the Kelly's cutting plane method, have been solved in less than 2 hours of CPU time within 1% optimality gap.

`Competitive Hub and Spoke Network, Non-Linear Integer Program, Kelly's Cutting Plane, Second Order Conic Program, Lagrangian Relaxation`

1 Introduction

Hub and spoke network was pioneered in the airline industry by Delta Airlines in 1955 to compete with the low cost Eastern Airlines (Delta History, 1955). In a hub and spoke network, every origin-destination (O-D) pair is connected through an intermediate node, called hub. Hubs are consolidation points, where traffic from various non-hub nodes, called spokes, is aggregated, thereby generating economies of scale. This leads to a lower operational cost, compared to alternative network configurations with direct O-D connections (O'kelly, 1986; Hamacher et al., 2004; Chen, 2007). A hub and spoke network also results in a lower setup cost since it requires fewer links to connect various origins and destinations in the network. Several studies have documented the benefits, leading to competitive advantage, from the use of hub and spoke network (McShan and Windle, 1989; Oum et al., 1995; Bania et al., 1998; Martins de Sá et al., 2015). Since the deregulation of the US airline industry in 1978, hub and spoke network has become almost a default choice for airline networks. Besides the airline industry, hub and spoke networks are also used in other industries, for instance, telecommunications (Klincewicz, 1998), energy (Lumsden et al., 1999), road transportation (Üster and Agrahari, 2011).

Designing a hub and spoke network calls for solving a hub location problem (HLP), which determines the optimal location of hubs and the path through some of those hubs between every O-D pair (Campbell and O'Kelly, 2012). In a multi-player setting, HLPs can be broadly classified either as cooperative games or non-cooperative games. Cooperative games focus on coalitions and joint actions among firms, thereby determining their collective payoffs, as opposed to non-cooperative games, where individual players focus on their own payoffs under competition. Lin and Lee (2010) studies a cooperative game in freight services in an oligopolistic market. In a non-cooperative setting, an HLP is solved either by an entrant airline that intends to set up its network by locating the hubs, or by an incumbent airline that intends to revamp its existing network. In this paper, we focus on this network design problem faced by an entrant. Most of the extant literature on HLP has studied the above problem in a monopolistic setting, without accounting for the presence of competing firms in the market. The literature on the entrant's problem in a competitive setting can be broadly classified in the following two categories: (i) the best response of the entrant explicitly accounting for the competitors' reaction (Sasaki and Fukushima, 2001; Sasaki, 2005; Sasaki et al., 2014; Mahmutogullari and Kara, 2016) (ii) the best response of the entrant without accounting for the reaction from the competitors (Marianov et al., 1999; Wagner, 2008; Eiselt and Marianov, 2009; Lüer-Villagra and Marianov, 2013). In this paper, we specifically focus on the problem of maximizing the market share of an entrant airline in the second category of competitive HLP. The problem results in an non-linear integer program (NLIP), which is challenging to solve using off-the-shelf solvers like CPLEX, Gurobi, etc. To the best of our knowledge, ours is the first study to solve the above problem exactly.

Through this paper, we make the following contributions to the related literature. We propose four alternate approaches to solve the problem. The first approach (CPA) is based on the Kelly's cutting plane algorithm to address the non-linearity in the problem. The second approach (MISOCP) relies on a mixed integer second order conic program based reformulation of the problem. The third approach (LR-CPA) employs Kelly's cutting plane algorithm within Lagrangian relaxation, while the fourth approach (LR-SOCP) uses a Lagrangian relaxation of the mixed integer second order conic program. Further, we compare the above four solution approaches based on extensive computational experiments. Our analysis highlights the superiority of LR-SOCP for most of the test instances. We also propose a refinement for the LR-SOCP approach, using which we are able to solve all the test instances within an optimality gap of 1% in less than 2 hours of CPU Time.

The rest of the paper is organized as follows. In Section 2, we present a review of the literature on HLP and its variants in a competitive setting. The problem description, followed by its mathematical formulation, is presented in Section 3. In Section 4, we present our alternate solution approaches, followed by extensive computational results in Section 5. Finally, the conclusions and directions for future research are presented in Section 6.

2 Literature Review

HLPs can be classified as p-median, p-center, covering or fixed-charge on the basis of their objective functions. The objective in the p-median HLP is to minimize the total transportation cost (O'kelly, 1986; Skorin-Kapov et al., 1996; Campbell, 1996; Ernst and Krishnamoorthy, 1996), while that for a p-center HLP is to minimize the maximum transportation cost between any pair of nodes (Campbell, 1994; Kara and Tansel, 2000). A covering HLP can either be hub set-covering if the objective is to minimize the number of hubs to cover all the nodes (Campbell, 1994; Kara and Tansel, 2000), or maximal hub-covering if the objective is to maximize the demand covered with a given number of hubs to locate (Campbell, 1994). Further, an HLP can be single allocation (O'kelly, 1986; Campbell, 1994) or multiple allocation (Campbell, 1992; Skorin-Kapov et al., 1996; Ernst and Krishnamoorthy, 1996) depending on whether the immediate hub visited by any flow originating from (or destined to) a given node is the same irrespective of their destinations (or origins), or they can be different. HLPs can also be classified as complete or incomplete on the basis of the hub topology. Most of the papers in the literature assume a complete hub network topology, wherein there exists a direct connection between every pair of hubs. Incomplete HLPs, on the other hand, assume a specific hub topology like star (Labbé and Yaman, 2008), tree (Contreras et al., 2010), etc. For an extensive review on HLPs, the readers may refer to Campbell and O'Kelly (2012). Next, we review literature on HLPs in a competitive setting (non-cooperative games), which we further categorize into two classes.

The first class of non-cooperative games in HLPs consists of papers that do not take into account the response of the competitors while deciding the hub locations for a player. Marianov et al. (1999) belongs to this category, and to the best of our knowledge, is also the first to introduce competition in the area of hub location. The problem models the decision of an entrant, to locate a set of hubs so as to maximize the demand flow captured from its competitors (without taking into account the response of the competitors). This results in a mixed integer linear program (MILP), which is solved using a tabu search heuristic. Wagner (2008) highlights certain shortcomings of the model proposed by Marianov et al. (1999), and proposes ways to correct them. Lüer-Villagra and Marianov (2013) studies the problem of an entrant that wants to determine the prices so as to maximize its profit, given the pricing scheme of the competitor, and solves the resulting model using genetic algorithm. Eiselt and Marianov (2009) also studies the problem of an entrant airline that wants to maximize its market share, wherein the customers' choice of an airline depends on their utility (instead of just cost). The problem is formulated as an NLIP, which is again solved using genetic algorithm.

In contrast to the above cited papers, the second class of papers take into account the competitor's response while solving the HLP. Sasaki and Fukushima (2001) studies the perspective of a leader who competes with several existing firms to maximize his/her profit. The problem is modelled as a bilevel Stackelberg game, and solved using sequential quadratic programming. Sasaki (2005) extends the problem from a continuous network to a discrete one, which is solved using complete enumeration and greedy heuristics. Sasaki et al. (2014) studies the problem of a leader who tries to locate hub arcs, as opposed to locating hub nodes, to maximize revenue. The resulting bilevel program is solved using implicit enumeration. Mahmutogullari and Kara (2016) studies a duopoly model in a Stackelberg framework, where two competitors sequentially choose their respective hub locations with the aim to maximize their captured flow. The problem is formulated as a bilevel HLP, and solved using implicit enumeration of the leader's problem.

The focus of this paper is on the first category. As evident from the review above, the literature in this area is scarce. Further, the extant studies resort to heuristic approaches. Specifically, we study the problem of market share maximization by the entrant firm in a competitive airline industry. The market share of the entrant is modeled as a probabilistic function of its hub location decisions, which introduces non-linearity in the problem. The resulting mathematical program is a non-linear IP, for which we propose alternate exact solution approaches. Next, we provide a detailed problem description, followed by its mathematical formulation.

3 Problem Description and Model Formulation

Consider a network of cities, represented by a complete graph $G = (N, A)$, in which a set $C = \{1, 2, \ldots, |C|\}$ of airlines are competing for their respective market shares. There exists a given demand f_{ij} for each pair $(i \in N, j \in N, i \neq j)$ of cities. Each airline $c \in C$ operates in a hub and spoke network with a set $H^c \subset N$ of hubs, such that the traffic between any city pair (i, j) is routed via a maximum of two intermediate hubs $(k, l \in H^c)$, so as to exploit the benefits arising from the economies of scale in inter-hub transportation. Let B^c_{ij} be the transportation cost per unit traffic for a direct connection between a city pair (i, j), which is the same for every airline. The cost benefit due to economies of scale is captured using a constant $\alpha < 1$, such that the cost per unit inter-hub traffic between hubs k and l is given by αB^c_{kl}. We further assume that the costs B^c_{ij} follow the triangular inequality, as a result of which a route involving more than two hubs does not generate any additional benefit, and hence the above restriction of a maximum of two hubs on any path between an origin-destination pair. Therefore, a path between an origin i and destination j can always be denoted using four ordered indices (i, j, k, l), where k and l are the hubs along the path $(i \rightarrow k \rightarrow l \rightarrow j)$.

A new airline, hereafter referred to as an entrant, aims to capture a portion of this market by opening a set $(H^e \in N)$ of p hubs. As a newcomer to the market, the entrant focuses on maximizing its market share. It also enjoys the same discount factor α, as does its competitors, for any inter-hub transportation. Hence, the cost per unit inter-hub traffic between hubs k and l is given by αB^e_{kl}. The proportion of the market share captured by any airline is commonly calculated in the extant literature using the proximity rule (Drezner, 1994). The rule, as used in the competitive hub location literature, suggests that for a given origin-destination pair (i, j), customers select the path $(i \rightarrow k \rightarrow l \rightarrow j)$ that minimizes either their total travel distance or time. As a result of this, all the passengers traveling between any origin-destination pair will always traverse through the same sequence of hubs. However, Alamdari and Black (1992) argue that "simple all or nothing models, which assume the cheapest airline is chosen by all the passengers, are not suitable for determining airlines' market share. Passenger demand is influenced by a combination of fare and many other attributes that make up the quality of service provided". The drawback of the proximity rule is addressed by the probabilistic choice model, which proposes that market share captured by a player is proportional to the utility that customers derive from using its product/service vis-a-vis that derived from using the competitors' product/service (Leeflang et al., 2013; Bell et al., 1975). Mathematically, the proportion of market share (ρ^e_{ijkl}) between any city pair (i, j) via hubs k and l captured by the entrant is given by:

$$\rho^e_{ijkl} = \frac{u^e_{ijkl}}{\sum_{k \in H^e} \sum_{l \in H^e} u^e_{ijkl} + \sum_{k \in H^c} \sum_{l \in H^c} u^c_{ijkl}} \tag{1}$$

For customers choosing between airlines, the utility derived often depends on various factors like cost of travel, travel time and attractiveness of the airline (safety record, mileage points provided, in-flight entertainment, quality of food & service, the number of hubs on the route, their location, the convenience they offer to passengers, among other factors) (Eiselt and Marianov, 2009). Using the gravity model (Huff, 1964, 1966), the utility (u^e_{ijkl}) that the customers derive from the entrant airline e is modelled as follows :

$$u^e_{ijkl} = \frac{A^e_{kl}}{\gamma\left(T^e_{ijkl}\right)^\beta + (1-\gamma)\left(B^e_{ijkl}\right)^\delta} \tag{2}$$

where A^e_{kl} is the basic attractiveness index of a pair of hubs (k,l) used for the trip, B^e_{ijkl} and T^e_{ijkl} denotes the the cost and the total time required by the flight respectively for traveling along the route $(i \rightarrow k \rightarrow l \rightarrow j)$, parameters β and δ denote the attraction decay of travel time and cost, respectively. The customers' utility from choosing a competing airline c, u^c_{ijkl}, can be similarly computed.

The total capture of passengers using the route $(i \rightarrow k \rightarrow l \rightarrow j)$ offered by the entrant is denoted by $f_{ij}\rho^e_{ijkl}$. The objective of the airline is to capture as large a market share as possible, by locating a fixed p number of hubs. There is a set of binary location variable y_k, which is 1 if a hub is located at node k, and zero otherwise. Also there is a set of binary hub pair location variable w_{kl}, which is 1 if hubs are located at both the nodes k and l. We assume that each competitor can have multiple routes between nodes i and j, going through different hubs or pairs of hubs. All competitors in the market have same information of the demand structure. Airfares are proportional to the costs incurred by the airlines.

To mathematically describe the entrant airline's problem, let us define a decision variable $y_k = 1$ if it locates a hub at node k, 0 otherwise. Further, let $w_{kl} = 1$ if the entrant airline locates its hubs at both the nodes k and l. To state the entrant airline's problem, we first summarize the notation used.

3.1 Notation

i	:	Index for source nodes, $i \in N$;
j	:	Index for destination nodes, $j \in N$;
k	:	Index for first hub, $k \in N$;
l	:	Index for second hub, $l \in N$;
α	;	Discount factor for trans-shipment (hub to hub), $(k \to l)$
H	:	set of all hubs, $H \subseteq N$;
H^c	:	set of hubs for competing airlines;
H^e	:	set of potential hubs of entrant airline;
f_{ij}	:	Flow from origin i to destination j;
T^c_{ijkl}	:	Time taken by any passenger for travelling from $i \to k \to l \to j$ using any competing airline c;
T^e_{ijkl}	:	Time taken by any passenger for travelling from $i \to k \to l \to j$ using the entrant airline c;
B^c_{ijkl}	:	Transportation cost incurred by any competing airline c for flying any passenger from $i \to k \to l \to j$;
B^e_{ijkl}	:	Transportation cost incurred by the entrant airline e for flying any passenger from $i \to k \to l \to j$;
u^c_{ijkl}	:	The utility of any passenger for travelling from $i \to k \to l \to j$ using any competing airline c;
u^e_{ijkl}	:	The utility of any passenger for travelling from $i \to k \to l \to j$ using the entrant airline e;
A^c_{kl}	:	Basic attractiveness index of a pair of hubs (k, l) for any competing airline c ;
A^e_{kl}	:	Basic attractiveness index of a pair of hubs (k, l) for the entrant airline e ;
ρ^e_{ijkl}	:	The proportion of market share captured by the entrant airline e between any city pair (i, j) via hubs k and l ;
p	:	No. of hubs to be located;
y_k	:	1, if hub is located, 0 otherwise.
w_{kl}	:	1, if hubs are located at k and l, 0 otherwise.

3.2 Model

Using the above notation, the entrant airline's problem can be mathematically described as:

$[COHLP]$:

$$\theta(y, w) = \max \quad \sum_i \sum_j f_{ij} \left\{ \frac{\sum_{k \in H^e} \sum_{l \in H^e} u^e_{ijkl} w_{kl}}{\sum_{k \in H^e} \sum_{l \in H^e} u^e_{ijkl} w_{kl} + \sum_{k \in H^c} \sum_{l \in H^c} u^c_{ijkl}} \right\} \tag{3}$$

$$\text{s.t. } w_{kl} \leqslant y_k \qquad \forall\, k, l \in H^e \tag{4}$$

$$w_{kl} \leqslant y_l \qquad \forall\, k,l \in H^e \tag{5}$$
$$\sum_k y_k = p \tag{6}$$
$$y_k \in \{0,1\} \qquad \forall\, k \in H^e \tag{7}$$
$$w_{kl} \in \{0,1\} \qquad \forall\, k,l \in H^e \tag{8}$$

The objective function (3) maximizes the total demand captured by the entering airline, given the competitors' hub locations. Constraint sets (4) and (5) are the linking constraints between w and y variables. Constraint (6) enforces p hubs to be open. Constraint sets (7) and (8) are the binary constraints on y and w variables, respectively. $COHLP$ is an NLIP, which off-the-shelf solvers like CPLEX and Gurobi cannot handle (since they cannot handle non-linear problems that are non-quadratic).

Please note that $COHLP$ can be transformed into a non-linear mixed integer program by relaxing the binary variables w_{kl} to take continuous values in $[0,1]$ using the following argument: whenever both y_k and y_l, which are binary, take a value 1, w_{kl} will always assume a value of 1 (due to (3)). On the other hand, when one of y_k or y_l is 0, w_{kl} takes a value 0. Therefore, (8) can be relaxed as:

$$w_{kl} \in [0,1] \tag{9}$$

However, the resulting non-linear mixed integer program is still difficult for off-the-shelf solvers. Next, we discuss alternate solution methods to solve the problem efficiently, exploiting (9) wherever possible.

4 Solution Methods

In this section, we propose four alternate approaches to solve $COHLP$. The first approach is based on the linearization of the non-linear term in (3) using Kelly's cutting plane approach (CPA) (Kelley, 1960). In the second approach, we reformulate $COHLP$ into a mixed integer second order conic program (MISOCP), which can be solved efficiently using an off-the-shelf solver (Alizadeh and Goldfarb, 2003). The third and fourth approaches are based on Lagrangian relaxation of $COHLP$, which separates the resulting problem into a linear integer program and a non-linear program (NLP) with continuous variables. The third approach solves the resulting NLP using CPA, whereas the fourth approach reformulates it as a second order conic program (SOCP).

4.1 Cutting Plane Algorithm

This method exploits the special structure of the non-linear term in the objective function through a transformation, followed by a piece-wise linear approximation, which results in a mixed integer linear program (MILP). The constraints required for piece-wise linear approximation are generated using Kelly's Cutting Plane approach. To describe the approach, let us define the following non-negative variables:

$$R_{ij} = \sum_{k \in H^e} \sum_{l \in H^e} w_{kl} u^e_{ijkl} \tag{10}$$
$$Z_{ij}(R_{ij}) = \frac{R_{ij}}{\sum_{k,l \in H^c} u^c_{ijkl} + R_{ij}} \qquad \forall\, i,j$$

Clearly, Z_{ij} is concave in R_{ij} (since $\frac{\partial^2 Z_{ij}}{\partial R_{ij}^2} \leqslant 0$). The concavity of Z_{ij} implies that for a given set of points, indexed by $g \in G$, Z_{ij} can be approximated arbitrarily closely by a set of piece-wise linear functions that are tangents to Z_{ij} at a set of points $\mathbb{G} = \{R^g_{ij}\}_{g \in G}$, such that

$$Z'_{ij}(R_{ij}) = \min_{g \in G} \left\{ Z(R^g_{ij}) + \frac{R_{ij} - R^g_{ij}}{(\sum_{k \in H^c} \sum_{l \in H^c} u^c_{ijkl} + R^g_{ij})^2} \right\} \quad \forall\, i, j,$$

which is equivalent to the following constraint set:

$$Z'_{ij}(R_{ij}) \leqslant Z(R^g_{ij}) + \frac{R_{ij} - R^g_{ij}}{(\sum_{k \in H^c} \sum_{l \in H^c} u^c_{ijkl} + R^g_{ij})^2} \quad \forall\, i, j, g \tag{11}$$

Using the above transformations, $COHLP$ can be approximated using the following MILP:

$$[COHLP_{MILP(\mathbb{G})}]: \tag{12}$$

$$\max \quad \sum_i \sum_j f_{ij} Z'_{ij}$$

$$\text{s.t.}(4) - (7), (9), (11)$$

$$Z'_{ij} \in [0, 1] \qquad \forall\, i, j \tag{13}$$

$$R_{ij} \geqslant 0 \qquad \forall\, i, j \tag{14}$$

In $COHLP_{MILP}$, we have used w_{kl} as continuous variables (9) instead of binary (8), which obviates any branching on the w_{kl} variables in the branch-and-bound tree. $COHLP_{MILP}$ is a linear mixed integer program with two additional sets of continuous variables R_{ij} and Z'_{ij}, and a large set of linear constraints (11). Adding a large set of constraints (11) a priori can provide an arbitrarily close approximation of $COHLP$; however, many of these constraints may be redundant, which may adversely affect the computational performance. Hence, a more prudent way to solve the problem is to add constraints (11) on the fly. Such an approach of adding the constraints on the fly is called a cutting plane algorithm (CPA), the steps of which, as applied to $COHLP_{MILP}$, are described in Algorithm 1.

Algorithm 1 Cutting Plane Algorithm for $COHLP_{MILP}$

```
1: q ← 1; UB^q ← ∞; LB^q ← −∞; G^q ← Φ; 𝔾^q ← Φ
2: do
3:     Solve COHLP_{MILP(𝔾^q)}, and obtain its solution as (y_k^q, w_kl^q, Z'_ij^q, R_ij^q)
4:     Update the UB^{q+1} ← Σ_i Σ_j f_ij Z'_ij^q
5:     Update the LB^{q+1} ← max{LB^q, θ(y^q, w^q)}.
6:     G^{q+1} ← G^q ∪ {q + 1}; 𝔾^{q+1} = {R_ij^g}_{g∈G}
7:     q ← q + 1
8: while (UB^q − LB^q)/UB^q > ε
```

CPA is known to converge, within a given tolerance ϵ, in a finite number of iterations (Elhedhli, 2005; Vidyarthi and Jayaswal, 2014; Vidyarthi et al., 2016; Jayaswal et al., 2017). Although in Algorithm 1, the set $\mathbb{G}^0$ is empty, it is suggested in the literature that the algorithm can be speeded up by initializing $\mathbb{G}^0$ with a carefully chosen set of points. The constraints of the form (11) that are generated from the points in $\mathbb{G}^0$ are called *a priori* cuts (Vidyarthi and Jayaswal, 2014). In our numerical experiments in Section 5, we set $\mathbb{G}^0$ so as to approximate $Z_{ij}(R_{ij})$ within a gap of 0.1%.

4.2 Mixed Integer Second Order Conic Program (MISOCP)

Second order conic programs (SOCPs) are of particular interest as they can be solved efficiently with widely available commercial solvers, like CPLEX and Gurobi. Hence, SOCPs have recently been employed to a variety of problems, like portfolio optimization, value-at-risk minimization, machine scheduling, supply chain network design and airline rescheduling with speed control (Vielma et al. (2008); Aktürk et al. (2014); Antoniou and Lu (2007)). Mixed Integer second order comic programs (MISOCPs) have also been studied in a variety of problems like hub location problems with congestion and assortment problems (Sen et al., 2017) among many others. $COHLP$, as discussed in Section 3.2, is an NLIP, which cannot be solved using off-the-self solvers. Therefore, in this section, we reformulate $COHLP$ as a mixed integer second order conic program (MISOCP). To convert the COHLP into an MISOCP, we introduce the following sets of variables:

$$Q_{ij} = 1/\Big(\sum_{k\in H^e}\sum_{l\in H^e} u^e_{ijkl}w_{kl} + \sum_{k\in H^c}\sum_{l\in H^c} u^c_{ijkl}\Big) \tag{15}$$

$$r_{ij} = \sum_{k,l\in H^e}\sum_{l\in H^e} u^e_{ijkl}w_{kl} + \sum_{k\in H^c}\sum_{l\in H^c} u^c_{ijkl} \tag{16}$$

$$V_{ij} = Q_{ij}\sum_{k\in H^e}\sum_{l\in H^e} u^e_{ijkl}w_{kl} \tag{17}$$

The objective function (3) in $COHLP$ can be rewritten as :

$$\theta(y,w) = \quad \sum_i\sum_j \hat{f}_i - \sum_i\sum_j \hat{f}_i + \sum_i\sum_j f_{ij}\left\{\frac{\sum_{k\in H^e}\sum_{l\in H^e} u^e_{ijkl}w_{kl}}{\sum_{k\in H^e}\sum_{l\in H^e} u^e_{ijkl}w_{kl} + \sum_{k\in H^c}\sum_{l\in H^c} u^c_{ijkl}}\right\}, \tag{18}$$

where $\hat{f}_i = \max_j f_{ij}$. Further, using the newly defined sets of variables, (18) can be restated as:

$$\theta(V) = \quad \sum_{i,j}\hat{f}_i - \sum_{i,j}\hat{f}_i + \sum_i\sum_j f_{ij}V_{ij} \tag{19}$$

From (15) & (17), we can conclude that:

$$V_{ij} + \sum_{k\in H^c}\sum_{l\in H^c} u^c_{ijkl}Q_{ij} = 1 \tag{20}$$

Introducing the relation (20) in (19), we get:

$$\theta(Q,V) = \quad \sum_i\sum_j \hat{f}_i - \sum_i\sum_j \hat{f}_i\Big(V_{ij} + \sum_{k\in H^c}\sum_{l\in H^c} u^e_{ijkl}Q_{ij}\Big) + \sum_i\sum_j f_{ij}V_{ij} \tag{21}$$

On rearranging the terms, (21), can be restated as:

$$\theta(Q,V) = \quad \sum_i\sum_j \hat{f}_i - \sum_i\left\{\sum_j\sum_{k\in H^c}\sum_{l\in H^c} u^c_{ijkl}\hat{f}_iQ_{ij} + \sum_j(\hat{f}_i - f_{ij})V_{ij}\right\}$$

Using the above form of the objective function, $COHLP$ can be restated as:

$$\max \quad \sum_i\sum_j\hat{f}_i - \sum_i\left\{\sum_j\sum_{k\in H^c}\sum_{l\in H^c} u^c_{ijkl}\hat{f}_iQ_{ij} + \sum_j(\hat{f}_i - f_{ij})V_{ij}\right\} \tag{22}$$

$$\text{s.t.}(4)-(8),(16)$$

$$Q_{ij}r_{ij} \geqslant 1 \qquad \forall\, i,j\in N \tag{23}$$

$$V_{ij}r_{ij} \geqslant \sum_{k\in H^e}\sum_{l\in H^e} u^e_{ijkl}w_{kl} \qquad \forall\, i,j\in N \tag{24}$$

$$Q_{ij}, V_{ij}, r_{ij} \geqslant 0 \qquad \forall\, i,j\in N \tag{25}$$

In (22)-(25), the constraint set (23) is derived using (15) and (16). Constraint set (24) is derived using (15) and (17). Also, since $w_{kl} \in \{0,1\}$, (24) can be rewritten as $V_{ij} r_{ij} \geqslant \sum_{k \in H^e} \sum_{l \in H^e} u^e_{ijkl} w^2_{kl} \ \forall\ i,j \in N$, which is a rotated second order conic constraint. The rotated second order cone (SOC) can be transformed to a standard SOC as follows (Alamdari and Black, 1992):

$$(V_{ij} + r_{ij})^2 \geqslant 2 \sum_{k \in H^e} \sum_{l \in H^e} u^e_{ijkl} w^2_{kl} + V^2_{ij} + r^2_{ij} \quad \forall\ i,j \in N \tag{26}$$

Similarly, (23), which is also a rotated SOC constraint, can be transformed to a standard SOC as follows:

$$(Q_{ij} + r_{ij})^2 \geqslant 2 + Q^2_{ij} + r^2_{ij} \quad \forall\ i,j \in N \tag{27}$$

The above transformations result in the following MISOCP based reformulation of $COHLP$:

$$[COHLP_{MISOCP}]: \tag{28}$$

$$\begin{aligned} \max \quad & \sum_i \sum_j \hat{f}_i - \sum_i \left\{ \sum_j \sum_{k \in H^c} \sum_{l \in H^c} u^c_{ijkl} \hat{f}_i Q_{ij} + \sum_j (\hat{f}_i - f_{ij}) V_{ij} \right\} \\ \text{s.t.} \quad & (4) - (7), (9), (16), (26) - (27) \end{aligned}$$

In $COHLP_{MISOCP}$, similar to $COHLP_{MILP(\mathbb{G})}$, we have used w_{kl} as continuous variables (9) instead of binary (8), which obviates any branching on the w_{kl} variables in the branch-and-bound tree. $COHLP_{MISOCP}$ has $3N^2$ additional variables and $5N^2$ additional constraints, out of which $2N^2$ are SOC constraints. The program can be solved directly by CPLEX using either of the two parameter settings; *miqcpstrat 1* and *miqcpstrat 2*. In *miqcpstrat 1*, it uses an SOCP based branch-and-bound algorithm, wherein at each node, the continuous relaxation is solved using an interior point algorithm specifically designed for SOCPs. In *miqcpstrat 2*, CPLEX uses outer approximation of the MISOCP, which produces an LP at each node of the branch-and-bound tree. In our numerical experiments, reported in Section 5, we use the default setting of CPLEX, which is *miqcpstrat 0* to allow CPLEX to choose the best strategy, depending on the problem structure.

4.3 Lagrangian Relaxation with CPA

Lagrangian relaxation (LR) is a popular technique that has been used to solve a wide variety of integer/ mixed integer linear and non-linear programs (Narula et al., 1977; Mirchandani et al., 1985; Aykin, 1994; Pirkul and Schilling, 1998, 1991). In this section, we apply LR to $COHLP$. The challenge with LR method is to correctly identify the constraints to be relaxed. For this particular work, we relax (4) and (5) using α_{kl} and β_{kl} as their respective Lagrange multipliers, which produces the following Lagrangian sub-problem:

$$[COHLP_{SUB}]:$$

$$\begin{aligned} \theta_{SUB}(\alpha,\beta) = \max \quad & \sum_i \sum_j f_{ij} \frac{\sum_{k \in H^e} \sum_{l \in H^e} w_{kl} u^e_{ijkl}}{\sum_{k \in H^e} \sum_{l \in H^e} w_{kl} u^e_{ijkl} + \sum_{k \in H^c} \sum_{l \in H^c} u^c_{ijkl}} + \sum_{k \in H^e} \sum_{l \in H^e} \alpha_{kl}(y_k - w_{kl}) \\ & + \sum_{k \in H^e} \sum_{l \in H^e} \beta_{kl}(y_l - w_{kl}) \qquad (29) \\ \text{s.t.} \quad & (6) - (8) \\ & \alpha_{kl}, \beta_{kl} \in [0, \infty] \qquad (30) \end{aligned}$$

For a given set of (α, β), the Lagrangian sub-problem provides an upper bound (UB) to $COHLP$. The tightest (smallest) UB is obtained by solving the following Lagrangian dual problem:

$$\min_{\alpha \geqslant 0, \beta \geqslant 0} \theta_{SUB}(\alpha, \beta) \tag{31}$$

(31) is non-linear optimization problem, which is popularly solved using the sub-gradient algorithm (Held et al., 1974; Fisher, 1981) as elaborated in Algorithm (4). A feasible solution to $COHLP$ can be obtained by turning the (infeasible) solution obtained from Lagrangian sub-problem into a feasible solution, which provides a lower bound (LB) to $COHLP$. The best feasible solution (maximum of the known LBs) is reported, and the relative optimality gap is calculated at every iteration as 1-(LB/UB), which is used as a termination criterion for the sub-gradient algorithm.

For a given set of Lagrange multipliers (α, β), the sub-problem (31) decomposes into the following two independent sub-problems, with one involving only y_k variables, while the other involving only w_{kl} variables.

$$\begin{aligned}
&[SUB_1]: \\
&\theta_{SUB1} = \max \quad \sum_k \sum_l (\alpha_{kl} y_k + \beta_{kl} y_l) && (32) \\
&\qquad \text{s.t.} \quad \sum_{k \in H^e} y_k = p \\
&\qquad\qquad y_k \in \{0,1\} && \forall\, k \in H^e \\
&[SUB_2]: \\
&\theta_{SUB2} = \max \quad \sum_i \sum_j f_{ij} \frac{\sum_{k \in H^e} \sum_{l \in H^e} w_{kl} u^e_{ijkl}}{\sum_{k \in H^e} \sum_{l \in H^e} w_{kl} u^e_{ijkl} + \sum_{k \in H^c} \sum_{l \in H^c} u^c_{ijkl}} - \sum_k \sum_l \Big\{ \alpha_{kl} + \beta_{kl} \Big\} w_{kl} && (33) \\
&\qquad \text{s.t.} \quad (8)
\end{aligned}$$

SUB_1 can be solved optimally for a given set of (α, β) by locating p hubs with the maximum contribution to objective function (32). The steps of the method to solve SUB_1 are summarized in Algorithm (2)

Algorithm 2 Optimal Solution for SUB_1

1: Define an ordered set S=$\{i : \zeta_i \geqslant \zeta_{i+1}, \zeta_i \in \{\alpha_{kl}, \beta_{kl} : k, l \in H^e\}\}$. Let S_j denote an element in S, $j \in 1, \ldots, |S|$.
2: Set $y_j = 1\ \forall\ j = S_1, \ldots, S_p$; $y_j = 0\ \forall\ j = S_{p+1}, \ldots, S_{|S|}$

SUB_2 is an unconstrained non-linear binary program, which can be solved using Kelly's cutting plane approach, as explained in Section (4.1). The linear approximation of SUB_2, which is solved at every iteration in Kelly's cutting plane method, is provided below as SUB_2'. However, we introduce the following modifications in SUB_2' to solve it more efficiently. First, (8) can be replaced by (9), although this may not guarantee binary values for w_{kl} variables, and therefore, may result in a relatively weaker Lagrangian UB. Nonetheless, we still prefer to use (9) since it makes SUB_2' a linear program (LP). Second, we add $\sum_{k \in H^e} \sum_{l \in H^e} w_{kl} = p^2$ to SUB_2', which is redundant to $COHLP$, but helps to strengthen the Lagrangian UB.

$$
\begin{aligned}
&[SUB_2'] : \\
&\theta_{SUB_2'} = \max \quad \sum_i \sum_j f_{ij} Z_{ij}' - \sum_k \sum_l \Big\{\alpha_{kl} + \beta_{kl}\Big\} w_{kl} && (34) \\
&\qquad\qquad \text{s.t.} \sum_{k \in H^e} \sum_{l \in H^e} w_{kl} = p^2 && (35) \\
&\qquad\qquad (9), (11), (13) - (14)
\end{aligned}
$$

We now state the following two propositions, which are used in the development of the complete sub-gradient algorithm.

Proposition 1. *For any given set of (α^q, β^q), (36) provides an UB on the optimal objective function value of COHLP, where (y_k^q), $(w_{kl}^q, Z_{ij}'^q)$ are the optimal solutions to SUB_1 and SUB_2' with the objective function values $\theta_{SUB_1}^q$ and $\theta_{SUB_2'}^q$, respectively.*

$$
UB^q = \theta_{SUB_1}^q + \theta_{SUB_2'}^q = \sum_k \sum_l (\alpha_{kl}^q y_k^q + \beta_{kl}^q y_l^q) + \sum_i \sum_j f_{ij} Z_{ij}'^q - \sum_k \sum_l \left\{\alpha_{kl}^q + \beta_{kl}^q\right\} w_{kl}^q \tag{36}
$$

Proof. Since $COHLP_{SUB}$ is a Lagrangian relaxation of the full problem COHLP, the objective function value of $COHLP_{SUB}$, given by θ_{SUB}, provides an UB on the optimal objective function value of $COHLP$. □

Proposition 2. *For any given set of (α^q, β^q), (37) provides a LB to the optimal objective function value of $COHLP$, where (y_k^q) is an optimal solution to SUB_1.*

$$
LB^q = \theta(y^q, y_k^q y_l^q) = \sum_{i,j \in N} f_{ij} \frac{\sum_{k,l \in H^e} y_l^q y_l^q u_{ijkl}^c}{\sum_{k,l \in H^e} y_l^q y_l^q u_{ijkl} + \sum_{k,l \in H^c} u_{ijkl}^c} \tag{37}
$$

Proof. If (y_k^q) is an optimal solution to SUB_1, then (y_k^q, w_{kl}^q), where $w_{kl}^q = y_l^q y_l^q$, is a feasible solution to $COHLP$ (since it satisfies constraints (4)-(5), which were relaxed in the Langrangian relaxation) as also illustrated in Algorithm 3. Hence, the objective function of $COHLP$ evaluated at $(y_k^q, y_k^q y_l^q)$, given by (37), provides a LB on the optimal objective of $COHLP$. □

A good, but not necessarily optimal, set of Lagrange multipliers to $COHLP$ is obtained using the standard sub-gradient optimization, as summarized in Algorithm 4. At each iteration, the algorithm also produces a feasible solution, the best among which is reported as the final solution once the algorithm terminates, which happens either on reaching the specified maximum time limit (M) or upon reaching an optimality gap of ϵ.

Algorithm 3 Feasible

1: Set y_k using optimal y from Algorithm (2) $\quad \forall\ k$.
2: Set $w_{kl} = y_k * y_l \quad \forall\ k, l$.

Algorithm 4 Sub-Gradient Optimisation Algorithm

$q \leftarrow 1$, $\alpha^q \leftarrow 0, \beta^q \leftarrow 0$, $UB^q \leftarrow \infty$, $LB^q \leftarrow -\infty$, $UB \leftarrow UB^q$ and $LB \leftarrow LB^q$, $t^q \leftarrow 0$.
Initialise Δ (the step-size multiplier), N_I (maximum iterations with no improvement in UB), ϵ (optimality gap), M (maximum CPU time limit).
do
    Solve SUB_1, SUB_2, and obtain UB^q, LB^q from (36), (37), respectively.
    $UB \leftarrow UB^q$, $LB \leftarrow max\{LB, LB^q\}$.
    $(y^*, w^*) \leftarrow \arg\max_{y,w}\{LB\}$ and (α^*, β^*) be the corresponding lagrange multipliers.
    Find a feasible solution using Algorithm (3).
    Adjust the multipliers as :

$$
\begin{aligned}
&\alpha_{kl}^{q+1} \leftarrow \alpha_{kl}^{q} + t^q(y_k^q - w_{kl}^q) \quad \forall\, k,l \\
&\beta_{kl}^{q+1} \leftarrow \beta_{kl}^{q} + t^q(y_l^q - w_{kl}^q) \quad \forall\, k,l \\
&\text{where, } t^{q+1} \leftarrow \Delta\left\{\frac{UB - LB}{\sum_k \sum_l \left\{(y_k^q - w_{kl}^q)^2 + (y_l^q - w_{kl}^q)^2\right\}}\right\}.
\end{aligned}
$$

    If no improvement in UB in N_I consecutive iterations, $\Delta \leftarrow \Delta/2$ and $(\alpha, \beta) \leftarrow (\alpha^*, \beta^*)$
    $q \leftarrow q + 1$.
while $(1 - LB^q/UB^q \leqslant \epsilon \vee \text{CPU Time} \leqslant M)$

4.4 Lagrangian relaxation with SOCP

In this section, like in Section 4.3, we use the sub-gradient algorithm (refer to Algorithm 4) to solve $COHLP$. However, the UB to $COHLP$ is computed by solving SUB_2 in a different way, as opposed to using CPA. For this, we exploit the fact that SUB_2, given by (33), can be reformulated as the following MISCOP, using the transformations (15), (16) and (17) from Section 4.2. We also include (35), which we used in SUB_2' in Section 4.3, to obtain a tighter Lagrangian UB.

$[SUB_2'']$:

$$
\begin{aligned}
\theta''_{SUB_2} = \max & \sum_i \sum_j \hat{f}_i - \sum_i \left\{ \sum_j \sum_k \sum_l u_{ijkl}^c \hat{f}_i Q_{ij} + \sum_j (\hat{f}_i - f_{ij}) V_{ij} \right\} - \sum_k \sum_l \left\{ \alpha_{kl} + \beta_{kl} \right\} w_{kl} && (38) \\
\text{s.t. } & r_{ij} = \sum_{k \in H^e} \sum_{l \in H^e} u_{ijkl}^e w_{kl} + \sum_{k \in H^c} \sum_{l \in H^c} u_{ijkl}^c && \forall i,j \in N \\
& (V_{ij} + r_{ij})^2 \geqslant 2 \sum_{k \in H^e} \sum_{l \in H^e} u_{ijkl}^e w_{kl}^2 + V_{ij}^2 + r_{ij}^2 && \forall i,j \in N \\
& (Q_{ij} + r_{ij})^2 \geqslant 2 + Q_{ij}^2 + r_{ij}^2 && \forall i,j \in N \\
& \sum_{k \in H^e} \sum_{l \in H^e} w_{kl} = p^2 \\
& w_{kl} \in [0,1] && \forall k,l \in H^e \\
& Q_{ij}, V_{ij}, r_{ij} \geqslant 0 && \forall\, i,j \in N
\end{aligned}
$$

The above model is an SOCP, which has a polynomial time complexity, and hence can be solved efficiently using off-the-shelf solvers (Andersen et al., 2003; Monteiro and Tsuchiya, 2000) . It is worth noting that in $COHLP$ (3)-(8), w_{kl} is binary. One could have chosen to retain w_{kl} as binary variables in $SUB"_2$, which would have produced a tighter Lagrangian UB. However, this would have resulted in an MISOCP model for $SUB"_2$, which is not polynomial time solvable.

With the above transformation, $COHLP$ can be solved using a sub-gradient algorithm, similar to Algorithm 4, with the exception that SUB_2 is solved as an SOCP (SUB_2''). The lower and upper bounds (LB^q and UB^q) are updated accordingly.

5 Numerical Experiments

We first describe in Section 5.1 the relevant data used in our experiments, followed by a discussion of our results in Section 5.2.

5.1 Data Generation

All our computational experiments are based on the following two popular data sets from the HLP literature: Civil Aeronautics Board (CAB) and Australian Post (AP) data-set. The CAB data-set introduced by O'kelly (1987) consists of data on passenger flow volumes and distances among 25 US cities. The AP data-set, introduced by Ernst and Krishnamoorthy (1996), consists of nodes representing district postcodes, along with their coordinates and mail flow volumes. In the context of competitive HLP, the AP data-set has been used by Eiselt and Marianov (2009); Marianov et al. (1999), which are the closest to our work. However, there is no information available on computation of customers' utility in either of the data-sets, and hence we provide below the scheme used in this paper to generate them.

In all our experiments, we assume there is only one incumbent player, denoted by c, operating with a set of hubs in the network, and the entrant, denoted by e, intends to capture a part of its market by locating its own set of p hubs. As discussed in Section 3, the share of the market captured by the entrant is given by (1), wherein the customer utility appearing in (1) are given by (2). The parameters used in (2) are set as follows: $\beta = 1, \delta = 1, \gamma = 0.75$, and $A^a_{kl} = 1.25 \; \forall \; k, l : l = k, a \in \{c, e\}; 1$ otherwise. The attractiveness index $A^a_{kl} \; \forall \; a \in \{c, e\}$ is set 25% higher when $l = k$ to capture the fact that single hub routes are more attractive to customers than multiple hub routes. We assume that the the travel time for the same route ($i \to k \to l \to j$) is the same for both the airlines, which is computed as $T^a_{ijkl} = T^a_{ik} + T^a_{kl} + T^a_{lj} \; \forall \; a \in \{c, e\}$. The travel time (in minutes) between any two cities i and k is given by $30 + 0.12d_{ik}$, where d_{ik} is the distance (in miles) between the cities i and k, and 30 is an approximation for the layover time (in minutes) at the two cities (Grove and O'Kelly, 1986). The transportation cost per unit volume is also assumed to be the same for a given route ($i \to k \to l \to j$), which is computed as $B^a_{ijkl} = \chi T^a_{ik} + \alpha T^a_{kl} + \eta T^a_{lj} \; \forall \; a \in \{c, e\}$, where χ and η are the transportation costs per unit volume on the collection and distribution legs, respectively. All the parameter values used in our experiments are summarized in Table 2. All the experiments are run on a personal computer

Table 2: Parameters

Common Parameters			Entrant's Parameters			
N	p	α	γ	β	δ	A^a
$\{10, 15, 20, 25\}$	$\{2, 3, 4\}$	$\{.1, .2, .3, .4, .5, .6, .7, .8, .9, 1\}$	.75	1	1	1

with 2.20 GHz Intel(R) Xenon(R) E5-2630 CPU and 64 GB RAM. The four solution methods, as

described in Section 4, are coded in C++, and ILOG CPLEX 12.7.1 is used as the default solver. The optimality gap $(UB - LB)/LB$ for all the iterative algorithms is set at $\epsilon = .01$, and the maximum CPU time limit (M) of 4 hours is used for each instance. For MISOCP in Section 4.2, we use the default value of 0 for the CPLEX parameter *miqcpstrat*, as discussed earlier in Section 4.2. For the Lagrangian based methods (LR-CPA and LR-SOCP), the parameters in the sub-gradient algorithm are initialized as: $\Delta = 6$ and $N_I = 50$.

5.2 Results and Analysis

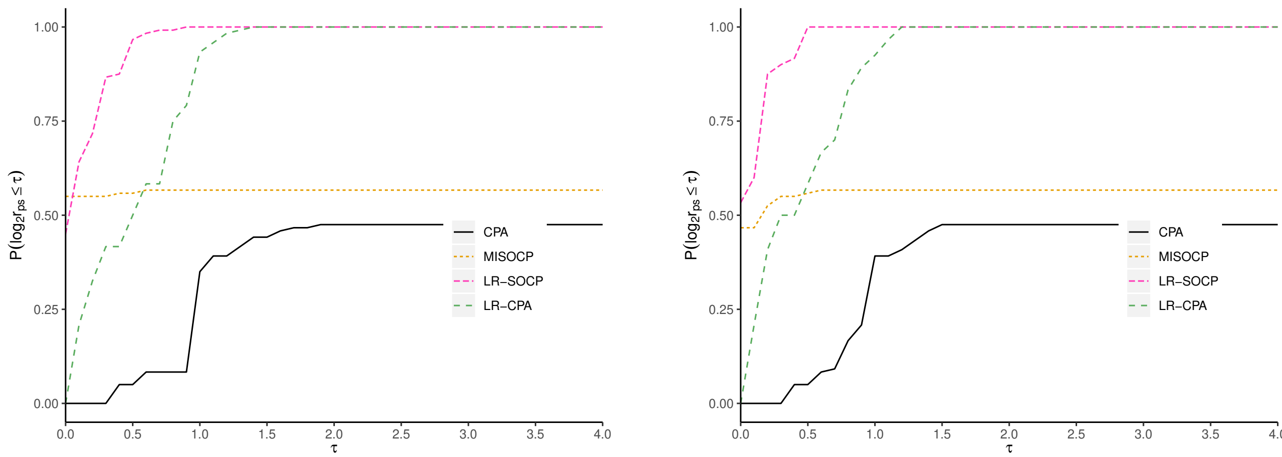
Tables 3 and 4 present the computational performances (in terms of CPU time) of the four alternate solution approaches corresponding to different combinations of $|N|, p, \alpha$ for the CAB and AP datasets, respectively. In each of these tables, the column "CPU Time" reports the time required to reach the final solution, while the column "Gap%" reports the optimality gap (in %) at algorithm termination. For the iterative methods (CPA, LR-CPA, LR-SOCP), the termination criterion used is: 1% optimality gap or 4 hours of CPU time. For MISOCP, which is directly solved using the CPLEX solver, we use the default termination criteria of CPLEX or 4 hours of CPU time.

As obvious from Table 3, for all the instances that CPA could solve, it was consistently the worst in terms of CPU time. There are other instances, particularly for $|N| \geqslant 15$, for which CPA runs out of memory, which are indicated using **. For smaller instances, especially with $|N| = 10, 15$, MISOCP outperforms all the other methods. For $|N| \geqslant 20$, it performs well for few instances, but consistently goes out of memory for most of the other instances. On the other hand, the Lagrangian relaxation based extensions, i.e., LR-CPA and LR-SOCP are able to solve all the instances within 1% optimality gap. Between LR-CPA and LR-SOCP, the latter consistently outperforms the former in terms of CPU time. Therefore, for larger instances, LR-SOCP turns out to be the best, and for smaller instances, the CPU time required by it is comparable to the best performing method (MISOCP). The above observations are also true for the AP dataset, the results for which are provided in Table 4. Interestingly, while LR-SOCP outperforms all the other method on larger instances, it also performs the best on certain smaller instances. Also, the computational performances (in terms of CPU time) of the three iterative solution approaches; namely CPA, LR-CPA, LR-SOCP to reach optimality gaps of $5\%, 2\%, 1\%$ are presented in the Appendix (refer Table 6, 7).

To further tease out the difference in the performances of the four different methods, we show their performance profiles (Dolan and Moré, 2002) through Figures 1a-1b. For this, let $t_{p,s}$ be the CPU time taken to solve the instance $p \in P$ using method $s \in S$. Then, performance ratio $r_{p,s}$ is calculated as:

$$r_{p.s} = \frac{t_{p,s}}{\min_{s \in S} t_{p,s}}$$

Treating $r_{p,s}$ as a random variable, a performance profile $\rho_s(\tau) = P(r_{p,s} \leqslant 2^\tau)$ gives its cumulative probability distribution. In other words, it gives the probability with which the CPU time taken by a given approach s does not exceed 2^τ times the CPU time required by the best among all the approaches under study. Specifically, the intercept on the y-axis for a given approach s gives the proportion of instances for which it is the best. As clear from Figure 1a, LR-SOCP is the best for 45% of the 30×4 instances in CAB dataset and from Figure 1b that LR-SOCP is the best for 54% of the 30×4 instances in AP dataset. Further, LR-SOCP is able to solve 100% of the instances at $\tau = .85$ for CAB dataset and at $\tau = .5$ for AP dataset, whereas CPA is able to solve only 47.5% of the 30×4 instances in both CAB and AP datasets even at $\tau = 4$.



(a) CAB dataset

(b) AP dataset

Figure 1: Performance Profile of proposed exact solution methods for CAB and AP dataset

5.3 Further Refinement

As witnessed in the previous section, LR-SOCP is the best performing method for most of the test instances. In this section, we explore the possibility of further improving upon this method. For this, recall from Algorithm 4 that LR-SOCP requires solving SUB_1 and SUB_2'' to compute an UB, while it uses the solution only from SUB_1 to compute an LB, at each iteration of the sub-gradient algorithm. Further, an UB is useful only to the extent of identifying when to terminate the algorithm. Hence, computing an UB, which requires solving an expensive SUB_2'', at every iteration of the sub-gradient algorithm can be avoided. We exploit this idea to further improve upon the computational efficiency of LR-SOCP by solving SUB_2'' only after a fixed number, say ψ, of iterations of the sub-gradient algorithm. The drawback of this proposal is that it misses the UBs from the intermediate iterations, one of which could have terminated the algorithm. As a result of this, the modified algorithm may take more iterations to terminate, which may offset the computational savings from avoiding solving SUB_2''. Hence, the value of ψ needs to be carefully chosen. Our limited experiments suggested $\psi = 50$ to be a good choice.

The results obtained with the above proposed refinement are reported in Table 5, where we show a comparison between the computational performances of MISOCP, LR-SOCP (original), and LR-SOCP* (with refinement) for CAB data-set. Clearly, with this refinement, the computational time further improves, and LR-SCOP* consistently outperforms all the other methods across all the test instances, including those for which LR-SOCP was outperformed by MISOCP.

6 Conclusions and Directions for Future Research

In this paper, we studied a competitive hub location problem ($COHLP$), wherein an entrant is making a strategic decision of locating its hubs in a market with already existing competing players. The entrant tries to maximize its market share, which is a function of the utility that customers get from using its services. The resulting problem is an NLIP, which is computationally intractable for off-the-shelf solvers. Hence, papers that have studied similar problems (Marianov et al., 1999; Eiselt and Marianov, 2009) have resorted to metaheuristics. In this paper, we proposed four alternate solution

Table 3: Comparison of computational performance for all solution approaches (CAB dataset)

	CPU Time				Gap %				CPU Time				Gap %			
p	CPA	MISOCP	LR-CPA	LR-SOCP	CPA	MISOCP	LR-CPA	LR-SOCP	CPA	MISOCP	LR-CPA	LR-SOCP	CPA	MISOCP	LR-CPA	LR-SOCP
								N=10								
				α = .1								α = .2				
2	254.40	**98.40**	218.40	123.60	0.00	0.00	0.00	0.00	266.01	**108.82**	220.05	128.47	0.00	0.00	0.00	0.00
3	454.80	**241.20**	440.40	256.80	0.00	0.00	0.00	0.00	466.16	**241.60**	446.05	266.94	0.00	0.00	0.00	0.00
4	700.80	**358.80**	685.20	380.40	0.00	0.00	0.00	0.00	711.58	**367.60**	695.11	381.83	0.00	0.00	0.00	0.00
				α = .3								α = .4				
2	273.29	**109.81**	223.78	129.54	0.00	0.00	0.00	0.00	275.44	**116.91**	226.14	138.82	0.00	0.00	0.00	0.00
3	477.12	**242.86**	453.38	271.29	0.00	0.00	0.00	0.00	486.05	**245.64**	461.17	281.53	0.00	0.00	0.00	0.00
4	723.19	**375.59**	705.56	387.61	0.00	0.00	0.00	0.00	732.00	**380.82**	713.16	387.71	0.00	0.00	0.00	0.00
				α = .5								α = .6				
2	283.12	**119.80**	231.32	144.67	0.00	0.00	0.00	0.00	279.09	**108.70**	219.86	143.70	0.00	0.00	0.00	0.00
3	493.74	**250.56**	461.26	283.26	0.00	0.00	0.00	0.00	490.13	**244.94**	458.63	281.66	0.00	0.00	0.00	0.00
4	738.77	**391.37**	718.08	392.38	0.00	0.00	0.00	0.00	732.44	**381.77**	713.93	386.89	0.00	0.00	0.00	0.00
				α = .7								α = .8				
2	278.57	**96.82**	212.40	142.70	0.00	0.00	0.00	0.00	278.22	**93.88**	201.74	142.05	0.00	0.00	0.00	0.00
3	483.83	**237.69**	450.81	277.65	0.00	0.00	0.00	0.00	482.01	**237.36**	450.18	274.03	0.00	0.00	0.00	0.00
4	723.78	**375.43**	704.59	383.81	0.00	0.00	0.00	0.00	711.91	**374.70**	699.10	380.89	0.00	0.00	0.00	0.00
				α = .9								α = 1				
2	268.14	**83.50**	192.66	133.47	0.00	0.00	0.00	0.00	263.16	**73.25**	181.91	132.02	0.00	0.00	0.00	0.00
3	474.81	**234.34**	449.14	265.81	0.00	0.00	0.00	0.00	473.82	**222.93**	442.83	261.99	0.00	0.00	0.00	0.00
4	704.49	**364.13**	695.72	373.71	0.00	0.00	0.00	0.00	694.66	**360.53**	685.76	372.73	0.00	0.00	0.00	0.00
								N=15								
				α = .1								α = .2				
2	2410.80	**1057.74**	2090.40	1124.04	0.00	0.00	0.00	0.00	2422.77	**1060.81**	2101.86	1131.62	0.00	0.00	0.00	0.00
3	2876.40	**1266.84**	2552.40	1455.54	0.00	0.00	0.00	0.00	2879.81	**1273.75**	2554.30	1463.24	0.00	0.00	0.00	0.00
4	3711.60	**2138.94**	3375.60	2869.26	0.00	0.00	0.00	0.00	3721.19	**2148.03**	3385.93	2875.33	0.00	0.00	0.00	0.00
				α = .3								α = .4				
2	2424.86	**1060.99**	2109.82	1133.79	0.00	0.00	0.00	0.00	2432.79	**1065.20**	2113.56	1134.25	0.00	0.00	0.00	0.00
3	2881.25	**1277.88**	2558.78	1469.22	0.00	0.00	0.00	0.00	2887.77	**1282.45**	2565.74	1476.95	0.00	0.00	0.00	0.00
4	3727.29	**2156.37**	3387.74	2875.91	0.00	0.00	0.00	0.00	3732.15	**2162.27**	3391.26	2879.45	0.00	0.00	0.00	0.00
				α = .5								α = .6				
2	2436.60	**1070.10**	2124.44	1139.36	0.00	0.00	0.00	0.00	2431.10	**1064.96**	2124.09	1131.84	0.00	0.00	0.00	0.00
3	2899.46	**1292.32**	2570.98	1486.02	0.00	0.00	0.00	0.00	2890.56	**1282.73**	2560.33	1482.60	0.00	0.00	0.00	0.00
4	**	**2172.13**	3400.30	2881.75	0.00	0.00	0.00	0.00	**	**2168.41**	3399.01	2872.54	0.00	0.00	0.00	0.00
				α = .7								α = .8				
2	2422.60	**1062.55**	2113.51	1122.18	0.00	0.00	0.00	0.00	2422.07	**1060.01**	2108.33	1117.67	0.00	0.00	0.00	0.00
3	2880.59	**1280.49**	2555.50	1481.00	0.03	0.00	0.00	0.00	**	**1272.10**	2554.79	1472.99	0.00	0.00	0.00	0.00
4	**	**2162.41**	3390.70	2869.19	0.00	0.00	0.00	0.00	**	**2160.24**	3388.04	2865.93	0.00	0.00	0.13	0.13
				α = .9								α = 1				
2	2421.99	**1059.42**	2097.56	1111.44	0.00	0.00	0.00	0.00	2419.63	**1052.24**	2096.78	1105.63	0.00	0.00	0.00	0.00
3	**	**1270.28**	2548.33	1472.80	0.00	0.00	0.00	0.00	**	**1264.82**	2547.27	1471.32	0.00	0.00	0.10	0.10
4	**	**2152.16**	3380.02	2860.33	0.00	0.00	0.56	0.54	**	**2148.04**	3377.80	2860.07	0.00	0.00	0.38	0.38
								N=20								
				α = .1								α = .2				
2	7462.80	**4962.30**	6627.60	5099.52	0.00	0.00	0.00	0.00	7464.46	**4964.86**	6635.39	5102.58	0.00	0.00	0.00	0.00
3	7840.80	**5265.24**	6826.80	5277.12	0.00	0.00	0.19	0.22	7842.43	**5270.84**	6828.94	5282.76	0.00	0.00	0.52	0.46
4	8204.40	**5305.02**	7438.80	5496.96	0.00	0.00	0.89	0.88	**	**	7442.65	**5505.88**	0.00	0.00	0.91	0.90
				α = .3								α = .4				
2	7469.40	**4968.30**	6643.19	5110.69	0.00	0.00	0.00	0.00	**	**	6649.73	**5112.93**	0.00	0.00	0.67	0.67
3	**	**	6835.76	**5286.14**	0.00	0.00	0.81	0.74	**	**	6847.27	**5289.02**	0.00	0.00	0.43	0.36
4	**	**	7446.61	**5506.74**	0.00	0.00	0.52	0.45	**	**	7448.31	**5516.05**	0.00	0.00	0.46	0.44
				α = .5								α = .6				
2	**	**	6650.83	**5117.93**	0.00	0.00	0.53	0.54	**	**	6642.90	**5116.91**	0.00	0.00	0.28	0.98
3	**	**	6853.44	**5291.36**	0.00	0.00	0.76	0.74	**	**	6848.21	**5284.49**	0.00	0.00	0.59	0.54
4	**	**	7453.64	**5523.93**	0.00	0.00	0.72	0.93	**	**	7442.10	**5515.32**	0.00	0.00	0.39	0.20
				α = .7								α = .8				
2	**	**	6637.00	**5115.88**	0.00	0.00	0.75	0.75	**	**	6634.03	**5113.13**	0.00	0.00	0.54	0.54
3	**	**	6845.88	**5276.97**	0.00	0.00	0.58	0.42	**	**	6842.48	**5272.00**	0.00	0.00	0.82	0.79
4	**	**	7439.47	**5514.52**	0.00	0.00	0.36	0.06	**	**	7438.56	**5510.01**	0.00	0.00	0.43	0.37
				α = .9								α = 1				
2	**	**	6623.62	**5110.39**	0.00	0.00	0.29	0.13	**	**	6620.73	**5106.52**	0.00	0.00	0.46	0.44
3	**	**	6839.70	**5266.14**	0.00	0.00	0.75	0.73	**	**	6837.70	**5264.47**	0.00	0.00	0.42	0.43
4	**	**	7438.26	**5501.67**	0.00	0.00	0.59	0.48	**	**	7435.44	**5498.18**	0.00	0.00	0.29	0.14
								N=25								
				α = .1								α = .2				
2	**	9181.80	9858.00	**7433.28**	0.00	0.00	0.55	0.53	**	**	9863.50	**7441.90**	0.00	0.00	0.13	0.11
3	**	10547.10	13537.20	**7515.84**	0.00	0.00	0.62	0.37	**	**	13546.48	**7524.78**	0.00	0.00	0.67	0.65
4	**	**	14384.40	**9814.08**	0.00	0.00	0.49	0.48	**	**	14394.77	**9817.81**	0.00	0.00	0.96	0.97
p				α = .3								α = .4				
2	**	**	9866.01	**7442.82**	0.00	0.00	0.43	0.38	**	**	9868.86	**7452.03**	0.00	0.00	0.92	0.87
3	**	**	13547.74	**7527.01**	0.00	0.00	0.32	0.30	**	**	13556.54	**7534.32**	0.00	0.00	0.88	0.89
4	**	**	14044.10	**9823.65**	0.00	0.00	0.19	0.12	**	**	14052.05	**9832.41**	0.00	0.00	0.62	0.60
p				α = .5								α = .6				
2	**	**	9873.72	**7453.63**	0.00	0.00	0.17	0.12	**	**	9863.95	**7447.83**	0.00	0.00	0.61	0.56
3	**	**	13558.51	**7542.03**	0.00	0.00	0.16	0.19	**	**	13546.79	**7536.04**	0.00	0.00	0.32	0.34
4	**	**	14063.25	**9841.30**	0.00	0.00	0.71	0.75	**	**	14058.07	**9836.10**	0.00	0.00	0.35	0.37
p				α = .7								α = .8				
2	**	**	9858.28	**7444.49**	0.00	0.00	0.41	0.37	**	**	9855.48	**7443.25**	0.00	0.00	0.34	0.01
3	**	**	13538.87	**7531.64**	0.00	0.00	0.19	0.16	**	**	13537.30	**7527.37**	0.00	0.00	0.72	0.65
4	**	**	14054.51	**9834.78**	0.00	0.00	0.38	0.35	**	**	14045.84	**9826.11**	0.00	0.00	0.63	0.67
p				α = .9								α = 1				
2	**	**	9853.12	**7434.66**	0.00	0.00	0.52	0.51	**	**	9848.51	**7425.66**	0.00	0.00	0.19	0.25
3	**	**	13528.94	**7521.18**	0.00	0.00	0.68	0.66	**	**	13526.34	**7519.01**	0.00	0.00	0.29	0.17
4	**	**	14039.29	**9816.75**	0.00	0.00	0.35	0.34	**	**	14031.71	**9807.41**	0.00	0.00	0.57	0.56

** out of memory

Table 4: Comparison of computational performance for all solution approaches (AP dataset)

p	CPU Time				Gap %				CPU Time				Gap %			
	CPA	MISOCP	LR-CPA	LR-SOCP	CPA	MISOCP	LR-CPA	LR-SOCP	CPA	MISOCP	LR-CPA	LR-SOCP	CPA	MISOCP	LR-CPA	LR-SOCP
									N=10							
				α =.1								α = .2				
2	256.80	**100.80**	218.40	128.40	0.00	0.00	0.00	0.00	260.79	**103.05**	229.74	131.41	0.00	0.00	0.00	0.00
3	457.20	300.00	440.40	**262.80**	0.00	0.00	0.00	0.00	460.68	303.44	447.80	**273.62**	0.00	0.00	0.00	0.00
4	703.20	**361.20**	685.20	393.60	0.00	0.00	0.00	0.00	714.63	**368.45**	687.12	404.25	0.00	0.00	0.00	0.00
				α = .3								α = .4				
2	267.34	**112.15**	231.67	133.52	0.00	0.00	0.00	0.00	273.01	**116.28**	242.92	134.77	0.00	0.00	0.00	0.00
3	467.30	311.88	455.38	**284.63**	0.00	0.00	0.00	0.00	470.67	316.66	463.73	**285.75**	0.00	0.00	0.00	0.00
4	725.61	**368.72**	693.93	415.30	0.00	0.00	0.00	0.00	734.27	**373.32**	699.24	426.35	0.00	0.00	0.00	0.00
				α = .5								α = .6				
2	283.93	**126.88**	251.27	140.86	0.00	0.00	0.00	0.00	281.58	**124.00**	250.56	137.57	0.00	0.00	0.00	0.00
3	474.69	328.10	469.92	**294.81**	0.00	0.00	0.00	0.00	471.75	323.44	463.08	**284.67**	0.00	0.00	0.00	0.00
4	735.65	**382.07**	703.26	437.40	0.00	0.00	0.00	0.00	733.32	**379.67**	696.32	432.22	0.00	0.00	0.00	0.00
				α = .7								α = .8				
2	281.52	**119.77**	242.17	131.33	0.00	0.00	0.00	0.00	279.45	**112.54**	230.67	120.68	0.00	0.00	0.00	0.00
3	469.22	320.35	452.59	**280.76**	0.00	0.00	0.00	0.00	462.25	319.85	450.42	**271.34**	0.00	0.00	0.00	0.00
4	727.71	**374.89**	685.92	425.38	0.00	0.00	0.00	0.00	725.11	**372.51**	683.72	423.18	0.00	0.00	0.00	0.00
				α = .9								α = 1				
2	271.56	**100.66**	224.46	111.10	0.00	0.00	0.00	0.00	264.17	**97.46**	216.35	102.55	0.00	0.00	0.00	0.00
3	451.01	314.80	440.83	**259.44**	0.00	0.00	0.00	0.00	441.76	306.45	437.32	**256.15**	0.00	0.00	0.00	0.00
4	723.82	**370.44**	674.12	419.69	0.00	0.00	0.00	0.00	715.89	**361.34**	664.54	417.06	0.00	0.00	0.00	0.00
									N=15							
				α = .1								α = .2				
2	2413.20	**1062.84**	2090.40	1131.18	0.00	0.00	0.00	0.00	2420.69	**1064.94**	2101.80	1132.24	0.00	0.00	0.00	0.00
3	2882.40	**1301.52**	1712.40	1459.62	0.00	0.00	0.00	0.00	2882.65	**1309.69**	1715.60	1464.81	0.00	0.00	0.00	0.00
4	3716.40	**2140.98**	3375.60	2878.44	0.00	0.00	0.00	0.00	3726.99	**2141.94**	3380.85	2882.25	0.00	0.00	0.00	0.00
				α = .3								α = .4				
2	2428.32	**1066.15**	2108.64	1142.19	0.00	0.00	0.00	0.00	2431.29	**1072.30**	2120.64	1149.88	0.00	0.00	0.00	0.00
3	2885.34	**1316.38**	1725.36	1471.40	0.00	0.00	0.00	0.00	2891.17	**1317.60**	1736.68	1480.43	0.00	0.00	0.00	0.00
4	3734.91	**2151.67**	3391.80	2886.56	0.00	0.00	0.00	0.00	3740.35	**2152.05**	3398.68	2893.28	0.00	0.00	0.00	0.00
				α = .5								α = .6				
2	2442.35	**1077.48**	2126.43	1153.84	0.00	0.00	0.00	0.00	2438.24	**1076.36**	2115.09	1144.98	0.00	0.00	0.00	0.00
3	2895.93	**1325.21**	1739.86	1482.54	0.00	0.00	0.00	0.00	2884.95	**1319.48**	1734.61	1475.46	0.00	0.00	0.00	0.00
4	**	**2154.76**	3404.09	2895.82	0.00	0.00	0.00	0.00	**	**2149.70**	3397.24	2889.19	0.00	0.00	0.00	0.00
				α = .7								α = .8				
2	2430.70	**1067.07**	2107.37	1144.18	0.00	0.00	0.00	0.00	2424.40	**1058.71**	2102.39	1143.20	0.00	0.00	0.00	0.00
3	2873.91	**1314.65**	1726.42	1475.23	0.03	0.00	0.00	0.00	**	**1306.82**	1716.93	1474.96	0.00	0.00	0.00	0.00
4	**	**2148.00**	3392.79	2882.19	0.00	0.00	0.00	0.00	**	**2142.94**	3390.24	2874.15	0.00	0.00	0.13	0.13
				α = .9								α = 1				
2	2414.92	**1055.44**	2099.57	1136.75	0.00	0.00	0.00	0.00	2408.49	**1050.96**	2098.97	1129.51	0.00	0.00	0.00	0.00
3	**	**1304.99**	1708.89	1472.50	0.00	0.00	0.00	0.00	**	**1297.00**	1706.02	1465.48	0.00	0.00	0.10	0.10
4	**	**2139.80**	3379.60	2872.03	0.00	0.00	0.56	0.54	**	**2131.36**	3371.93	2864.35	0.00	0.00	0.38	0.38
									N=20							
				α = .1								α = .2				
2	7465.20	**4978.62**	6627.60	5104.32	0.00	0.00	0.00	0.00	7468.53	**4981.23**	6635.85	5111.62	0.00	0.00	0.00	0.00
3	7855.20	**5272.38**	6826.80	5291.52	0.00	0.00	0.19	0.22	7866.94	**5281.81**	6828.37	5296.17	0.00	0.00	0.52	0.46
4	8269.20	**5359.08**	7438.80	5501.76	0.00	0.00	0.89	0.88	**	**	7448.19	**5506.48**	0.00	0.00	0.91	0.90
				α = .3								α = .4				
2	7480.43	**4986.02**	6643.10	5112.35	0.00	0.00	0.00	0.00	**	**	6649.51	**5119.64**	0.00	0.00	0.67	0.67
3	**	**	6838.54	**5299.75**	0.00	0.00	0.81	0.74	**	**	6843.04	**5307.08**	0.00	0.00	0.43	0.36
4	**	**	7450.76	**5510.76**	0.00	0.00	0.52	0.45	**	**	7453.28	**5517.43**	0.00	0.00	0.46	0.44
				α = .5								α = .6				
2	**	**	6649.83	**5125.42**	0.00	0.00	0.53	0.54	**	**	6638.98	**5117.04**	0.00	0.00	0.28	0.98
3	**	**	6851.71	**5308.20**	0.00	0.00	0.76	0.74	**	**	6848.75	**5304.85**	0.00	0.00	0.59	0.54
4	**	**	7454.81	**5518.91**	0.00	0.00	0.72	0.93	**	**	7452.25	**5518.44**	0.00	0.00	0.39	0.20
				α = .7								α = .8				
2	**	**	6627.45	**5110.62**	0.00	0.00	0.75	0.75	**	**	6625.53	**5106.12**	0.00	0.00	0.54	0.54
3	**	**	6848.73	**5303.42**	0.00	0.00	0.58	0.42	**	**	6845.16	**5297.81**	0.00	0.00	0.82	0.79
4	**	**	7441.91	**5509.63**	0.00	0.00	0.36	0.06	**	**	7439.64	**5500.72**	0.00	0.00	0.43	0.37
				α = .9								α = 1				
2	**	**	6623.44	**5104.45**	0.00	0.00	0.29	0.13	**	**	6621.47	**5096.46**	0.00	0.00	0.46	0.44
3	**	**	6840.24	**5292.39**	0.00	0.00	0.75	0.73	**	**	6840.18	**5286.16**	0.00	0.00	0.42	0.43
4	**	**	7433.56	**5497.01**	0.00	0.00	0.59	0.48	**	**	7425.33	**5494.90**	0.00	0.00	0.29	0.14
									N=25							
				α = .1								α = .2				
2	**	9212.40	9858.00	**7437.12**	0.00	0.00	0.55	0.53	**	**	9867.18	**7439.19**	0.00	0.00	0.13	0.11
3	**	10611.90	13537.20	**7518.72**	0.00	0.00	0.62	0.37	**	**	13546.20	**7519.15**	0.00	0.00	0.67	0.65
4	**	**	14384.40	**9860.16**	0.00	0.00	0.49	0.48	**	**	14394.47	**9865.79**	0.00	0.00	0.96	0.97
p				α = .3								α = .4				
2	**	**	9878.69	**7447.15**	0.00	0.00	0.43	0.38	**	**	9878.93	**7456.55**	0.00	0.00	0.92	0.87
3	**	**	13553.44	**7528.02**	0.00	0.00	0.32	0.30	**	**	13557.71	**7534.88**	0.00	0.00	0.88	0.89
4	**	**	14045.58	**9868.68**	0.00	0.00	0.19	0.12	**	**	14053.29	**9871.96**	0.00	0.00	0.62	0.60
p				α = .5								α = .6				
2	**	**	9880.78	**7464.29**	0.00	0.00	0.17	0.12	**	**	9880.08	**7460.62**	0.00	0.00	0.61	0.56
3	**	**	13569.18	**7541.57**	0.00	0.00	0.16	0.19	**	**	13561.40	**7535.74**	0.00	0.00	0.32	0.34
4	**	**	14053.49	**9879.33**	0.00	0.00	0.71	0.75	**	**	14051.54	**9872.01**	0.00	0.00	0.35	0.37
p				α = .7								α = .8				
2	**	**	9875.10	**7459.17**	0.00	0.00	0.41	0.37	**	**	9868.34	**7453.98**	0.00	0.00	0.34	0.01
3	**	**	13560.47	**7529.59**	0.00	0.00	0.19	0.16	**	**	13554.34	**7522.31**	0.00	0.00	0.72	0.65
4	**	**	14041.42	**9867.35**	0.00	0.00	0.38	0.35	**	**	14040.53	**9861.10**	0.00	0.00	0.63	0.67
p				α = .9								α = 1				
2	**	**	9860.44	**7444.39**	0.00	0.00	0.52	0.51	**	**	9856.14	**7437.85**	0.00	0.00	0.19	0.25
3	**	**	13551.50	**7514.02**	0.00	0.00	0.68	0.66	**	**	13548.45	**7508.89**	0.00	0.00	0.29	0.17
4	**	**	14038.73	**9857.57**	0.00	0.00	0.35	0.34	**	**	14038.32	**9849.29**	0.00	0.00	0.57	0.56

** out of memory

Table 5: Comparison of computational performance with modified LR-SOCP method (CAB dataset)

	CPU Time			Gap %			CPU Time			Gap %		
p	MISOCP	LR-SOCP	LR-SOCP*	MISOCP	LR-SOCP	LR-SOCP*	MISOCP	LR-SOCP	LR-SOCP*	MISOCP	LR-SOCP	LR-SOCP*
						N=10						
			α = .1						α = .2			
2	98.40	123.60	**82.01**	0.00	0.00	0.00	108.82	128.47	**86.03**	0.00	0.00	0.00
3	241.20	256.80	**185.41**	0.00	0.00	0.00	241.60	266.94	**204.75**	0.00	0.00	0.00
4	358.80	380.40	**334.25**	0.00	0.00	0.00	367.60	381.83	**335.33**	0.00	0.00	0.00
			α = .3						α = .4			
2	109.81	129.54	**96.30**	0.00	0.00	0.00	116.91	138.82	**103.40**	0.00	0.00	0.00
3	242.86	271.29	**238.28**	0.00	0.00	0.00	245.64	281.53	**237.11**	0.00	0.00	0.00
4	375.59	387.61	**340.89**	0.00	0.00	0.00	380.82	387.71	**340.58**	0.00	0.00	0.00
			α = .5						α = .6			
2	119.80	144.67	**108.08**	0.00	0.00	0.00	108.70	143.70	**107.22**	0.00	0.00	0.00
3	250.56	283.26	**248.71**	0.00	0.00	0.00	244.94	281.66	**237.20**	0.00	0.00	0.00
4	391.37	392.38	**344.87**	0.00	0.00	0.00	381.77	386.89	**340.18**	0.00	0.00	0.00
			α = .7						α = .8			
2	96.82	142.70	**91.82**	0.00	0.00	0.00	93.88	142.05	**77.33**	0.00	0.00	0.00
3	237.69	277.65	**214.02**	0.00	0.00	0.00	237.36	274.03	**230.32**	0.00	0.00	0.00
4	375.43	383.81	**337.39**	0.00	0.00	0.00	374.70	380.89	**334.50**	0.00	0.00	0.00
			α = .9						α = 1			
2	83.50	133.47	**81.57**	0.00	0.00	0.00	73.25	132.02	**72.24**	0.00	0.00	0.00
3	234.34	265.81	**233.45**	0.00	0.00	0.00	222.93	261.99	**220.50**	0.00	0.00	0.00
4	364.13	373.71	**328.08**	0.00	0.00	0.00	360.53	372.73	**327.06**	0.00	0.00	0.00
						N=15						
			α = .1						α = .2			
2	1057.74	1124.04	**989.11**	0.00	0.00	0.00	1060.81	1131.62	**995.75**	0.00	0.00	0.00
3	1266.84	1455.54	**1180.21**	0.00	0.00	0.00	**1273.75**	1463.24	1287.50	0.00	0.00	0.00
4	2138.94	2869.26	**2024.21**	0.00	0.00	0.00	**2148.03**	2875.33	2530.23	0.00	0.00	0.00
			α = .3						α = .4			
2	1060.99	1133.79	**997.64**	0.00	0.00	0.00	1065.20	1134.25	**997.53**	0.00	0.00	0.00
3	1277.88	1469.22	**1262.38**	0.00	0.00	0.00	1282.45	1476.95	**1278.84**	0.00	0.00	0.00
4	2156.37	2875.91	**2030.29**	0.00	0.00	0.00	2162.27	2879.45	**2133.20**	0.00	0.00	0.00
			α = .5						α = .6			
2	1070.10	1139.36	**1002.38**	0.00	0.00	0.00	1064.96	1131.84	**995.57**	0.00	0.00	0.00
3	1292.32	1486.02	**1286.74**	0.00	0.00	0.00	1282.73	1482.60	**1263.80**	0.00	0.00	0.00
4	2172.13	2881.75	**2035.82**	0.00	0.00	0.00	2168.41	2872.54	**2127.27**	0.00	0.00	0.00
			α = .7						α = .8			
2	1062.55	1122.18	**987.16**	0.00	0.00	0.00	1060.01	1117.67	**982.96**	0.00	0.00	0.00
3	1280.49	1481.00	**1278.29**	0.00	0.00	0.00	1272.10	1472.99	**1255.44**	0.00	0.00	0.00
4	2162.41	2869.19	**2124.07**	0.00	0.00	0.00	2160.24	2865.93	**2102.01**	0.00	0.13	0.13
			α = .9						α = 1			
2	1059.42	1111.44	**977.73**	0.00	0.00	0.00	1052.24	1105.63	**972.51**	0.00	0.00	0.00
3	1270.28	1472.80	**1265.77**	0.00	0.00	0.00	1264.82	1471.32	**1234.69**	0.00	0.10	0.10
4	2152.16	2860.33	**2106.22**	0.00	0.54	0.56	2148.04	2860.07	**2126.04**	0.00	0.38	0.38
						N=20						
			α = .1						α = .2			
2	4962.30	5099.52	**4486.71**	0.00	0.00	0.00	4964.86	5102.58	**4489.29**	0.00	0.00	0.00
3	5265.24	5277.12	**4643.58**	0.00	0.22	0.11	5270.84	5282.76	**4647.93**	0.00	0.46	0.37
4	5305.02	5496.96	**4836.98**	0.00	0.88	0.71	**	5505.88	**4844.65**	0.00	0.90	0.72
			α = .3						α = .4			
2	4968.30	5110.69	**4496.98**	0.00	0.00	0.00	**	5112.93	**4498.79**	0.00	0.67	0.51
3	**	5286.14	**4650.84**	0.00	0.74	0.41	**	5289.02	**4653.97**	0.00	0.36	0.40
4	**	5506.74	**4845.77**	0.00	0.45	0.31	**	5516.05	**4853.58**	0.00	0.44	0.28
			α = .5						α = .6			
2	**	5117.93	**4503.21**	0.00	0.54	0.47	**	5116.91	**4502.63**	0.00	0.98	0.20
3	**	5291.36	**4656.12**	0.00	0.74	0.71	**	5284.49	**4649.88**	0.00	0.54	0.27
4	**	5523.93	**4861.00**	0.00	0.93	0.65	**	5515.32	**4852.90**	0.00	0.20	0.98
			α = .7						α = .8			
2	**	5115.88	**4501.00**	0.00	0.75	0.61	**	5113.13	**4498.59**	0.00	0.54	0.09
3	**	5276.97	**4643.19**	0.00	0.42	0.39	**	5272.00	**4639.28**	0.00	0.79	0.92
4	**	5514.52	**4852.49**	0.00	0.06	0.27	**	5510.01	**4848.66**	0.00	0.37	0.76
			α = .9						α = 1			
2	**	5110.39	**4496.23**	0.00	0.13	0.22	**	5106.52	**4493.01**	0.00	0.44	0.44
3	**	5266.14	**4633.29**	0.00	0.73	0.43	**	5264.47	**4632.13**	0.00	0.43	0.28
4	**	5501.67	**4841.07**	0.00	0.48	0.59	**	5498.18	**4837.95**	0.00	0.14	0.99
						N=25						
			α = .1						α = .2			
2	9181.80	7433.28	**6540.75**	0.00	0.53	0.14	**	7441.90	**6548.05**	0.00	0.11	0.23
3	10547.10	7515.84	**6613.91**	0.00	0.37	0.57	**	7524.78	**6620.97**	0.00	0.65	0.69
4	**	9814.08	**7135.84**	0.00	0.48	0.37	**	9817.81	**7038.91**	0.00	0.97	0.27
			α = .3						α = .4			
2	**	7442.82	**6549.45**	0.00	0.38	0.77	**	7452.03	**6556.98**	0.00	0.87	0.81
3	**	7527.01	**6622.93**	0.00	0.30	0.50	**	7534.32	**6629.84**	0.00	0.89	0.79
4	**	9823.65	**7136.09**	0.00	0.12	0.74	**	9832.41	**7039.68**	0.00	0.60	0.41
			α = .5						α = .6			
2	**	7453.63	**6558.22**	0.00	0.12	0.22	**	7447.83	**6553.78**	0.00	0.56	0.57
3	**	7542.03	**6636.54**	0.00	0.19	0.87	**	7536.04	**6630.72**	0.00	0.34	0.91
4	**	9841.30	**7136.74**	0.00	0.75	0.64	**	9836.10	**7040.30**	0.00	0.37	0.29
			α = .7						α = .8			
2	**	7444.49	**6550.84**	0.00	0.37	0.98	**	7443.25	**6549.33**	0.00	0.01	0.47
3	**	7531.64	**6627.18**	0.00	0.16	0.54	**	7527.37	**6623.13**	0.00	0.65	0.61
4	**	9834.78	**7136.92**	0.00	0.35	0.13	**	9826.11	**7040.60**	0.00	0.67	0.85
			α = .9						α = 1			
2	**	7434.66	**6541.74**	0.00	0.51	0.87	**	7425.66	**6534.48**	0.00	0.25	0.24
3	**	7521.18	**6618.32**	0.00	0.66	0.95	**	7519.01	**6616.04**	0.00	0.17	0.17
4	**	9816.75	**7137.91**	0.00	0.34	0.43	**	9807.41	**7040.92**	0.00	0.56	0.79

** out of memory

approaches, namely, CPA, MISOCP, LR-CPA, LR-SOCP. Besides proposing various reformulations for the problem, our primary contribution lied in using second order cone programming within Lagrangian relaxation to solve the problem efficiently. The LR-SOCP method along with its refinement, which relied on this idea, was able to solve all the problem instances within 1% optimality gap in less than 2 hours of CPU time.

There are several directions in which the current work can be extended. First, prompted by our success in the current study, we foresee the application of SOCP within Lagrangian relaxation in similar classes of problems, where non-linearity may arise due to competition, congestion, economies of scale, uncertainties, etc. Second, the problem studied in this paper does not take into account the response of the competing firms to the entrant's actions. One possible extension of $COHLP$ could be to model it as a leader-follower game, wherein the entrant takes into account the follower's response while solving its problem.

Acknowledgement

This research was supported by the Research fellowship provided to the first author by Indian Institute of Management, Ahmedabad.

References

Aktürk, M. S., Atamtürk, A., and Gürel, S. (2014). Aircraft rescheduling with cruise speed control. *Operations Research*, 62(4):829–845.

Alamdari, F. E. and Black, I. G. (1992). Passengers' choice of airline under competition: the use of the logit model. *Transport Reviews*, 12(2):153–170.

Alizadeh, F. and Goldfarb, D. (2003). Second-order cone programming. *Mathematical Programming*, 95(1):3–51.

Andersen, E. D., Roos, C., and Terlaky, T. (2003). On implementing a primal-dual interior-point method for conic quadratic optimization. *Mathematical Programming*, 95(2):249–277.

Antoniou, A. and Lu, W.-S. (2007). Semidefinite and second-order cone programming. *Practical Optimization: Algorithms and Engineering Applications*, pages 449–499.

Aykin, T. (1994). Lagrangian relaxation based approaches to capacitated hub-and-spoke network design problem. *European Journal of Operational Research*, 79(3):501–523.

Bania, N., Bauer, P. W., and Zlatoper, T. J. (1998). Us air passenger service: a taxonomy of route networks, hub locations, and competition. *Transportation Research Part E: Logistics and Transportation Review*, 34(1):53–74.

Bell, D. E., Keeney, R. L., and Little, J. D. (1975). A market share theorem. *Journal of Marketing Research*, 12(2):136–141.

Campbell, J. F. (1992). Location and allocation for distribution systems with transshipments and transportion economies of scale. *Annals of Operations Research*, 40(1):77–99.

Campbell, J. F. (1994). Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, 72(2):387–405.

Campbell, J. F. (1996). Hub location and the p-hub median problem. *Operations Research*, 44(6):923–935.

Campbell, J. F. and O'Kelly, M. E. (2012). Twenty-five years of hub location research. *Transportation Science*, 46(2):153–169.

Chen, J.-F. (2007). A hybrid heuristic for the uncapacitated single allocation hub location problem. *Omega*, 35(2):211–220.

Contreras, I., Fernández, E., and Marín, A. (2010). The tree of hubs location problem. *European Journal of Operational Research*, 202(2):390–400.

Delta History, D. F. (1955). History.

Dolan, E. D. and Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213.

Drezner, T. (1994). Locating a single new facility among existing, unequally attractive facilities. *Journal of Regional Science*, 34(2):237–252.

Eiselt, H. A. and Marianov, V. (2009). A conditional p-hub location problem with attraction functions. *Computers & Operations Research*, 36(12):3128–3135.

Elhedhli, S. (2005). Exact solution of a class of nonlinear knapsack problems. *Operations Research Letters*, 33(6):615–624.

Ernst, A. T. and Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science*, 4(3):139–154.

Fisher, M. L. (1981). The lagrangian relaxation method for solving integer programming problems. *Management Science*, 27(1):1–18.

Grove, P. G. and O'Kelly, M. E. (1986). Hub networks and simulated schedule delay. *Papers in Regional Science*, 59(1):103–119.

Hamacher, H. W., Labbé, M., Nickel, S., and Sonneborn, T. (2004). Adapting polyhedral properties from facility to hub location problems. *Discrete Applied Mathematics*, 145(1):104–116.

Held, M., Wolfe, P., and Crowder, H. P. (1974). Validation of subgradient optimization. *Mathematical Programming*, 6(1):62–88.

Huff, D. L. (1964). Defining and estimating a trading area. *The Journal of Marketing*, pages 34–38.

Huff, D. L. (1966). A programmed solution for approximating an optimum retail location. *Land Economics*, 42(3):293–303.

Jayaswal, S., Vidyarthi, N., and Das, S. (2017). A cutting plane approach to combinatorial bandwidth packing problem with queuing delays. *Optimization Letters*, 11(1):225–239.

Kara, B. Y. and Tansel, B. C. (2000). On the single-assignment p-hub center problem. *European Journal of Operational Research*, 125(3):648–655.

Kelley, Jr, J. E. (1960). The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712.

Klincewicz, J. G. (1998). Hub location in backbone/tributary network design: a review. *Location Science*, 6(1-4):307–335.

Labbé, M. and Yaman, H. (2008). Solving the hub location problem in a star–star network. *Networks*, 51(1):19–33.

Leeflang, P. S., Wittink, D. R., Wedel, M., and Naert, P. A. (2013). *Building models for marketing decisions*, volume 9. Springer Science & Business Media.

Lin, C.-C. and Lee, S.-C. (2010). The competition game on hub network design. *Transportation Research Part B: Methodological*, 44(4):618–629.

Lüer-Villagra, A. and Marianov, V. (2013). A competitive hub location and pricing problem. *European Journal of Operational Research*, 231(3):734–744.

Lumsden, K., Dallari, F., and Ruggeri, R. (1999). Improving the efficiency of the hub and spoke system for the skf european distribution network. *International Journal of Physical Distribution & Logistics Management*, 29(1):50–66.

Mahmutogullari, A. I. and Kara, B. Y. (2016). Hub location under competition. *European Journal of Operational Research*, 250(1):214–225.

Marianov, V., Serra, D., and ReVelle, C. (1999). Location of hubs in a competitive environment. *European Journal of Operational Research*, 114(2):363–371.

Martins de Sá, E., Contreras, I., Cordeau, J.-F., Saraiva de Camargo, R., and de Miranda, G. (2015). The hub line location problem. *Transportation Science*, 49(3):500–518.

McShan, S. and Windle, R. (1989). The implications of hub-and-spoke routing for airline costs. *Logistics and Transportation Review*, 25(3):209.

Mirchandani, P. B., Oudjit, A., and Wong, R. T. (1985). 'multidimensional'extensions and a nested dual approach for the m-median problem. *European Journal of Operational Research*, 21(1):121–137.

Monteiro, R. D. and Tsuchiya, T. (2000). Polynomial convergence of primal-dual algorithms for the second-order cone program based on the mz-family of directions. *Mathematical programming*, 88(1):61–83.

Narula, S. C., Ogbu, U. I., and Samuelsson, H. M. (1977). An algorithm for the p-median problem. *Operations Research*, 25(4):709–713.

O'kelly, M. E. (1986). The location of interacting hub facilities. *Transportation Science*, 20(2):92–106.

O'kelly, M. E. (1987). A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, 32(3):393–404.

Oum, T. H., Zhang, A., and Zhang, Y. (1995). Airline network rivalry. *Canadian Journal of Economics*, pages 836–857.

Pirkul, H. and Schilling, D. A. (1991). The maximal covering location problem with capacities on total workload. *Management Science*, 37(2):233–248.

Pirkul, H. and Schilling, D. A. (1998). An efficient procedure for designing single allocation hub and spoke systems. *Management Science*, 44(12-part-2):S235–S242.

Sasaki, M. (2005). Hub network design model in a competitive environment with flow threshold. *Journal of the Operations Research Society of Japan*, 48(2):158–171.

Sasaki, M., Campbell, J. F., Krishnamoorthy, M., and Ernst, A. T. (2014). A stackelberg hub arc location model for a competitive environment. *Computers & Operations Research*, 47:27–41.

Sasaki, M. and Fukushima, M. (2001). Stackelberg hub location problem. *Journal of the Operations Research Society of Japan*, 44(4):390–402.

Sen, A., Atamturk, A., and Kaminsky, P. (2017). A conic integer programming approach to constrained assortment optimization under the mixed multinomial logit model. *arXiv preprint arXiv:1705.09040*.

Skorin-Kapov, D., Skorin-Kapov, J., and O'Kelly, M. (1996). Tight linear programming relaxations of uncapacitated p-hub median problems. *European Journal of Operational Research*, 94(3):582–593.

Üster, H. and Agrahari, H. (2011). A benders decomposition approach for a distribution network design problem with consolidation and capacity considerations. *Operations Research Letters*, 39(2):138–143.

Vidyarthi, N. and Jayaswal, S. (2014). Efficient solution of a class of location–allocation problems with stochastic demand and congestion. *Computers & Operations Research*, 48:20–30.

Vidyarthi, N., Jayaswal, S., and Chetty, V. B. T. (2016). Bandwidth packing problem with queueing delays: modelling and exact solution approach. *Journal of Global Optimization*, 65(4):745–776.

Vielma, J. P., Ahmed, S., and Nemhauser, G. L. (2008). A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs. *INFORMS Journal on Computing*, 20(3):438–450.

Wagner, B. (2008). A note on "location of hubs in a competitive environment". *European Journal of Operational Research*, 184(1):57–62.

Appendices

Table 6: Comparison of computation time for various optimality tolerance across approaches (CAB dataset)

p	% Gap	CPU Time														
		CPA	LR-CPA	LR-SOCP	CPA	LR-CPA	LR-SOCP	CPA	LR-CPA	LR-SOCP	CPA	LR-CPA	LR-SOCP	CPA	LR-CPA	LR-SOCP
									N=10							
			$\alpha = .1$			$\alpha = .2$			$\alpha = .3$			$\alpha = .4$			$\alpha = .5$	
	5	93.83	72.53	**54.64**	94.19	78.00	**56.92**	87.92	81.46	**53.04**	95.07	81.84	**54.77**	94.01	76.00	**56.99**
2	2	173.31	147.59	**59.74**	178.02	151.19	**60.03**	185.84	148.68	**66.42**	185.25	156.51	**62.99**	196.80	158.39	**80.24**
	1	254.40	218.40	**123.60**	266.01	220.05	**128.47**	273.29	223.78	**129.54**	275.44	226.14	**138.82**	283.12	231.32	**144.67**
	5	154.46	146.67	**100.83**	157.65	151.94	**103.07**	149.63	148.07	**118.10**	157.34	157.17	**113.22**	158.89	163.17	**112.15**
3	2	305.12	288.66	**134.43**	309.48	305.98	**134.09**	304.27	306.32	**135.56**	323.90	309.47	**141.19**	323.35	320.85	**143.13**
	1	454.80	440.40	**256.80**	466.16	446.05	**266.94**	477.12	453.38	**271.29**	486.05	461.17	**281.53**	493.74	461.26	**283.26**
	5	228.88	225.76	**153.39**	241.13	226.84	**155.68**	237.25	229.49	**164.68**	243.01	231.14	**159.71**	239.17	242.14	**159.61**
4	2	471.90	459.07	**194.40**	483.91	462.18	**193.82**	489.83	460.47	**199.05**	481.53	463.00	**197.88**	486.50	476.14	**205.13**
	1	700.80	685.20	**380.40**	711.58	695.11	**381.83**	723.19	705.56	**387.61**	732.00	713.16	**387.71**	738.77	718.08	**392.38**
			$\alpha = .6$			$\alpha = .7$			$\alpha = .8$			$\alpha = .9$			$\alpha = 1$	
	5	97.77	80.50	**63.83**	97.10	1404.09	**64.60**	90.70	70.45	**55.56**	88.27	75.44	**55.56**	83.50	72.93	**57.39**
2	2	188.47	147.87	**76.48**	181.36	2095.62	**74.68**	175.64	144.23	**70.13**	179.35	166.41	**70.13**	178.28	134.74	**65.27**
	1	279.09	219.86	**143.70**	278.57	212.40	**142.70**	278.22	201.74	**142.05**	268.14	192.66	**133.47**	263.16	181.91	**132.02**
	5	163.56	150.63	**113.72**	156.54	114.94	**109.82**	155.82	155.88	**99.46**	159.95	115.12	**99.46**	156.41	149.62	**105.38**
3	2	318.86	304.31	**145.87**	313.47	173.59	**133.28**	314.02	302.58	**127.08**	307.84	170.03	**127.08**	304.45	290.97	**128.02**
	1	490.13	458.63	**281.66**	483.83	450.81	**277.65**	482.01	450.18	**274.03**	474.81	449.14	**265.81**	473.82	442.83	**261.99**
	5	241.09	239.56	**157.82**	245.36	2269.27	**161.43**	235.95	230.45	**148.50**	247.34	2268.75	**148.50**	237.76	221.40	**142.99**
4	2	486.47	474.83	**200.80**	486.98	3382.36	**195.11**	480.06	463.45	**187.93**	485.51	3372.51	**187.93**	478.34	451.50	**180.56**
	1	732.44	713.93	**386.89**	723.78	704.59	**383.81**	711.91	699.10	**380.89**	704.49	695.72	**373.71**	694.66	685.76	**372.73**
									N=15							
			$\alpha = .1$			$\alpha = .2$			$\alpha = .3$			$\alpha = .4$			$\alpha = .5$	
	5	801.86	689.13	**528.22**	804.58	692.09	**536.06**	806.08	691.28	**544.43**	814.88	700.19	**542.19**	811.29	707.38	**546.68**
2	2	1618.58	1402.40	**664.98**	1619.55	1407.64	**668.33**	1627.14	1403.32	**676.39**	1642.46	1406.14	**678.87**	1645.38	1422.33	**675.85**
	1	2410.80	2090.40	**1124.04**	2422.77	2101.86	**1131.62**	2424.86	2109.82	**1133.79**	2432.79	2113.56	**1134.25**	2436.60	2124.44	**1139.36**
	5	955.58	**573.95**	686.94	949.74	**574.82**	684.83	950.72	**573.31**	696.70	956.81	**576.88**	690.13	961.24	**582.77**	692.47
3	2	1929.29	1143.09	**855.76**	1929.07	1150.67	**858.81**	1932.82	1152.99	**866.08**	1929.68	1168.36	**867.81**	1946.97	1162.39	**871.25**
	1	2876.40	2552.40	**1455.54**	2879.81	2554.30	**1463.24**	2881.25	2558.78	**1469.22**	2887.77	2565.74	**1476.95**	2899.46	2570.98	**1486.02**
	5	1232.99	**1118.30**	1351.51	1222.98	**1116.99**	1353.38	1235.78	**1115.17**	1353.51	1237.51	**1129.34**	1359.37	1328.00	**1127.46**	1359.70
4	2	2488.06	2262.19	**1696.03**	2483.11	2261.35	**1693.75**	2501.83	2266.78	**1692.54**	2502.54	2282.71	**1699.34**	2527.70	2280.60	**1701.32**
	1	3711.60	3375.60	**2869.26**	3721.19	3385.93	**2875.33**	3727.29	3387.74	**2875.91**	3732.15	3391.26	**2879.45**	**	3400.30	**2881.75**
			$\alpha = .6$			$\alpha = .7$			$\alpha = .8$			$\alpha = .9$			$\alpha = 1$	
	5	814.90	693.25	**546.25**	813.65	699.82	**546.25**	809.12	697.39	**540.37**	806.85	697.22	**534.80**	795.27	694.13	**534.94**
2	2	1642.94	1413.91	**676.86**	1633.41	1404.09	**672.30**	1626.98	1402.32	**675.69**	1629.76	1407.44	**666.75**	1620.48	1403.70	**666.09**
	1	2431.10	2124.09	**1131.84**	2422.60	2113.51	**1122.18**	2422.07	2108.33	**1117.67**	2421.99	2097.56	**1111.44**	2419.63	2096.78	**1105.63**
	5	958.68	**572.43**	699.23	951.30	**568.30**	696.14	881.74	**571.61**	694.61	860.53	**563.89**	685.52	851.73	**564.68**	691.29
3	2	1925.49	1159.98	**865.44**	1937.01	1158.94	**870.37**	1919.14	1157.58	**866.74**	1850.97	1145.12	**855.89**	1758.10	1151.18	**857.28**
	1	2890.56	2560.33	**1482.60**	2880.59	2555.50	**1481.00**	**	2554.79	**1472.99**	**	2548.33	**1472.80**	**	2547.27	**1471.32**
	5	1334.88	**1125.79**	1357.43	1400.08	**1117.63**	1362.30	1468.80	**1115.76**	1350.63	1485.81	**1120.86**	1357.55	1504.70	**1113.48**	1354.39
4	2	2576.35	2270.12	**1699.11**	**	2269.27	**1698.80**	**	2269.90	**1697.38**	**	2268.75	**1699.00**	**	2266.69	**1692.04**
	1	**	3399.01	**2872.54**	**	3390.70	**2869.19**	**	3388.04	**2865.93**	**	3380.02	**2860.33**	**	3377.80	**2860.07**
									N=20							
			$\alpha = .1$			$\alpha = .2$			$\alpha = .3$			$\alpha = .4$			$\alpha = .5$	
	5	2462.61	**2192.99**	2560.20	2474.30	**2188.18**	2556.31	3366.79	**2188.65**	2561.07	3390.42	**2197.69**	2560.29	3420.02	**2194.62**	2554.01
2	2	5001.67	4436.82	**3197.86**	5013.06	4441.28	**3186.75**	5021.27	4443.29	**3197.96**	5052.18	4448.79	**3197.18**	5098.68	4444.78	**3190.47**
	1	7462.80	6627.60	**5099.52**	7464.46	6635.39	**5102.58**	7469.40	6643.19	**5110.69**	**	6649.73	**5112.93**	**	6650.83	**5117.93**
	5	2590.43	**2256.29**	2637.88	2595.45	**2260.06**	2642.16	3520.78	**2257.65**	2644.52	3468.83	**2261.56**	2650.73	3431.78	**2258.10**	2649.95
3	2	5253.68	4570.42	**3301.60**	5266.89	4568.35	**3305.84**	5246.21	4581.17	**3305.55**	5169.59	4582.34	**3311.21**	5111.45	4590.46	**3313.15**
	1	7840.80	6826.80	**5277.12**	7842.43	6828.94	**5282.76**	**	6835.76	**5286.14**	**	6847.27	**5289.02**	**	6853.44	**5291.36**
	5	2714.77	**2456.25**	2747.65	3690.91	**2459.39**	2751.15	3663.41	**2462.12**	2750.25	3641.69	**2469.07**	2758.12	3612.33	**2467.44**	2750.88
4	2	5506.05	4990.55	**3441.11**	5496.66	4991.21	**3432.61**	5467.76	5005.02	**3436.36**	5433.35	5001.48	**3439.11**	**	5006.78	**3445.15**
	1	8204.40	7438.80	**5496.96**	**	7442.65	**5505.88**	**	7446.61	**5506.74**	**	7448.31	**5516.05**	**	7453.64	**5523.93**
			$\alpha = .6$			$\alpha = .7$			$\alpha = .8$			$\alpha = .9$			$\alpha = 1$	
	5	3381.43	**2196.56**	2559.70	3369.91	**2192.08**	2551.26	3334.73	**2183.62**	2551.01	3312.95	**2181.46**	2547.86	3298.19	**2187.25**	2536.78
2	2	5036.81	4445.66	**3203.22**	5015.28	4445.34	**3190.68**	4970.61	4432.81	**3185.43**	4937.24	4427.31	**3189.31**	4918.96	4422.86	**3175.99**
	1	**	6642.90	**5116.91**	**	6637.00	**5115.88**	**	6634.03	**5113.13**	**	6623.62	**5110.39**	**	6620.73	**5106.52**
	5	3418.95	**2261.84**	2655.91	3364.15	**2252.00**	2655.69	3358.66	**2257.61**	2651.40	3299.10	**2255.23**	2650.84	3283.99	**2253.38**	2642.89
3	2	5093.73	4583.49	**3318.28**	5017.58	4572.45	**3313.46**	4999.96	4581.21	**3311.04**	4922.42	4576.70	**3314.90**	4883.78	4578.20	**3305.85**
	1	**	6848.21	**5284.49**	**	6845.88	**5276.97**	**	6842.48	**5272.00**	**	6839.70	**5266.14**	**	6837.70	**5264.47**
	5	3575.41	**2463.34**	2757.65	3556.21	**2470.62**	2745.60	3542.07	**2466.27**	2740.48	3529.40	**2460.72**	2747.58	3496.04	**2451.66**	2738.44
4	2	5330.41	5001.42	**3434.32**	5297.87	4993.01	**3432.78**	5277.77	4991.86	**3423.92**	5255.13	4991.31	**3424.54**	5208.15	4980.50	**3425.38**
	1	**	7442.10	**5515.32**	**	7439.47	**5514.52**	**	7438.56	**5510.01**	**	7438.26	**5501.67**	**	7435.44	**5498.18**
									N=25							
			$\alpha = .1$			$\alpha = .2$			$\alpha = .3$			$\alpha = .4$			$\alpha = .5$	
	5	2714.77	**2456.25**	2747.65	3690.91	**2459.39**	2751.15	3663.41	**2462.12**	2750.25	3641.69	**2469.07**	2758.12	3612.33	**2467.44**	2750.88
2	2	5506.05	4990.55	**3441.11**	5496.66	4991.21	**3432.61**	5467.76	5005.02	**3436.36**	5433.35	5001.48	**3439.11**	**	5006.78	**3445.15**
	1	**	9858.00	**7433.28**	**	9863.50	**7441.90**	**	9866.01	**7442.82**	**	9868.86	**7452.03**	**	9873.72	**7453.63**
	5	8291.04	**3257.23**	3715.00	8326.63	**3263.12**	3719.91	8406.48	**3260.43**	3722.02	8479.43	**3257.99**	3719.35	8514.11	**3271.50**	3733.07
3	2	10671.60	6612.57	**4651.08**	**	6610.64	**4649.01**	**	6613.37	**4652.39**	**	6612.21	**4649.05**	**	6625.89	**4654.93**
	1	**	13537.20	**7515.84**	**	13546.48	**7524.78**	**	13547.74	**7527.01**	**	13556.54	**7534.32**	**	13558.51	**7542.03**
	5	8459.00	4468.41	**3764.54**	8580.14	4471.56	**3768.16**	8457.82	4473.10	**3765.11**	8783.84	4475.39	**3763.38**	8579.26	4474.82	**3769.95**
4	2	**	9073.15	**4702.79**	**	9080.39	**4699.12**	**	9081.00	**4700.63**	**	9086.17	**4702.40**	**	9085.90	**4709.75**
	1	**	14384.40	**9814.08**	**	14394.77	**9817.81**	**	14044.10	**9823.65**	**	14052.05	**9832.41**	**	14063.25	**9841.30**
			$\alpha = .6$			$\alpha = .7$			$\alpha = .8$			$\alpha = .9$			$\alpha = 1$	
	5	3575.41	**2463.34**	2757.65	3556.21	**2470.62**	2745.60	3542.07	**2466.27**	2740.48	3529.40	**2460.72**	2747.58	3496.04	**2451.66**	2738.44
2	2	5330.41	5001.42	**3434.32**	5297.87	4993.01	**3432.78**	5277.77	4991.86	**3423.92**	5255.13	4991.31	**3424.54**	5208.15	4980.50	**3425.38**
	1	**	9863.95	**7447.83**	**	9858.28	**7444.49**	**	9855.48	**7443.25**	**	9853.12	**7434.66**	**	9848.51	**7425.66**
	5	8628.60	**3268.02**	3717.13	8488.47	**3266.43**	3723.02	8490.45	**3262.60**	3717.76	8598.02	**3266.59**	3707.63	8493.18	**3262.01**	3703.55
3	2	**	6621.43	**4653.14**	**	6616.07	**4647.88**	**	6619.39	**4649.40**	**	6616.93	**4641.67**	**	6617.78	**4637.53**
	1	**	13546.79	**7536.04**	**	13538.87	**7531.64**	**	13537.30	**7527.37**	**	13528.94	**7521.18**	**	13526.34	**7519.01**
	5	8715.25	4472.97	**3761.69**	8646.46	4473.90	**3764.66**	8749.77	4471.37	**3761.37**	8948.32	4466.46	**3761.38**	8704.08	4465.20	**3752.83**
4	2	**	9079.27	**4705.26**	**	9078.93	**4702.12**	**	9078.23	**4700.58**	**	9063.17	**4696.48**	**	9068.92	**4691.59**
	1	**	14058.07	**9836.10**	**	14054.51	**9834.78**	**	14045.84	**9826.11**	**	14039.29	**9816.75**	**	14031.71	**9807.41**

** out of memory

Table 7: Comparison of computation time for various optimality tolerance (AP dataset)

p	% Gap	CPU Time														
		CPA	LR-CPA	LR-SOCP	CPA	LR-CPA	LR-SOCP	CPA	LR-CPA	LR-SOCP	CPA	LR-CPA	LR-SOCP	CPA	LR-CPA	LR-SOCP
									N=10							
			$\alpha = .1$			$\alpha = .2$			$\alpha = .3$			$\alpha = .4$			$\alpha = .5$	
	5	89.34	71.18	**54.39**	92.37	86.33	**52.21**	89.29	73.86	**58.09**	90.69	88.32	**57.31**	95.82	85.11	**66.65**
2	2	176.71	144.54	**65.37**	177.40	160.86	**64.07**	179.27	157.67	**69.99**	182.97	164.65	**76.17**	192.28	170.84	**79.29**
	1	256.80	218.40	**128.40**	260.79	229.74	**131.41**	267.34	231.67	**133.52**	273.01	242.92	**134.77**	283.93	251.27	**140.86**
	5	150.55	150.51	**104.25**	156.90	155.53	**112.77**	152.15	150.23	**118.40**	156.83	159.07	**117.63**	162.28	156.61	**118.28**
3	2	313.25	297.88	**137.58**	311.37	304.45	**144.00**	313.29	301.55	**140.70**	323.93	307.28	**145.52**	323.43	313.29	**156.44**
	1	457.20	440.40	**262.80**	460.68	447.80	**273.62**	467.30	455.38	**284.63**	470.67	463.73	**285.75**	474.69	469.92	**294.81**
	5	240.51	230.07	**155.04**	233.52	230.74	**166.02**	239.63	233.64	**165.52**	242.16	235.85	**172.34**	241.99	233.05	**182.12**
4	2	468.38	463.18	**201.73**	474.20	455.78	**211.39**	488.25	465.23	**207.14**	500.49	467.11	**212.46**	492.76	471.71	**213.19**
	1	703.20	685.20	**393.60**	714.63	687.12	**404.25**	725.61	693.93	**415.30**	734.27	699.24	**426.35**	735.65	703.26	**437.40**
			$\alpha = .6$			$\alpha = .7$			$\alpha = .8$			$\alpha = .9$			$\alpha = 1$	
	5	98.24	90.08	**59.83**	101.72	93.02	**52.48**	98.58	76.22	**53.47**	98.87	79.60	**53.47**	95.95	71.27	**44.81**
2	2	198.08	172.53	**68.30**	183.87	183.52	**62.33**	193.63	151.69	**63.32**	181.82	179.95	**63.32**	177.00	149.02	**47.64**
	1	281.58	250.56	**137.57**	281.52	242.17	**131.33**	279.45	230.67	**120.68**	271.56	224.46	**111.10**	264.17	216.35	**102.55**
	5	165.69	155.45	**121.41**	161.34	151.73	**118.80**	152.58	150.00	**107.57**	152.96	138.24	**107.57**	156.60	150.13	**102.85**
3	2	321.60	306.80	**143.56**	317.01	283.89	**137.91**	313.38	307.59	**134.61**	299.49	1705.82	**134.61**	303.90	290.85	**129.64**
	1	471.75	463.08	**284.67**	469.22	452.59	**280.76**	462.25	450.42	**271.34**	451.01	440.83	**259.44**	441.76	437.32	**256.15**
	5	244.99	238.15	**176.91**	246.93	229.52	**179.98**	246.47	230.31	**166.16**	243.38	227.57	**166.16**	234.04	223.36	**168.26**
4	2	491.75	469.30	**223.99**	487.45	338.72	**222.09**	490.21	454.06	**211.17**	489.80	375.05	**211.17**	483.33	440.53	**205.20**
	1	733.32	696.32	**432.22**	727.71	685.92	**425.38**	725.11	683.72	**423.18**	723.82	674.12	**419.69**	715.89	664.54	**417.06**
									N=15							
			$\alpha = .1$			$\alpha = .2$			$\alpha = .3$			$\alpha = .4$			$\alpha = .5$	
	5	802.80	690.56	**531.26**	806.37	695.09	**535.00**	805.49	695.74	**533.70**	807.01	705.06	**546.00**	806.08	705.15	**545.51**
2	2	1613.45	1398.53	**669.57**	1626.38	1412.18	**667.32**	1635.31	1410.64	**671.09**	1636.66	1416.48	**675.80**	1645.89	1432.85	**675.01**
	1	2413.20	2090.40	**1131.18**	2420.69	2101.80	**1132.24**	2428.32	2108.64	**1142.19**	2431.29	2120.64	**1149.88**	2442.35	2126.43	**1153.84**
	5	954.45	**570.41**	692.30	952.15	**568.57**	685.68	952.96	**571.25**	688.70	964.05	**580.67**	702.66	963.49	**581.43**	701.49
3	2	1929.25	1148.48	**864.94**	1934.53	1154.68	**865.10**	1937.11	1153.14	**862.54**	1943.87	1164.13	**874.82**	1939.06	1164.41	**875.08**
	1	2882.40	1712.40	**1459.62**	2882.65	1715.60	**1464.81**	2885.34	1725.36	**1471.40**	2891.17	1736.68	**1480.43**	2895.93	1739.86	**1482.54**
	5	1232.66	**1117.30**	1360.91	1239.19	**1122.81**	1353.03	1233.25	**1123.70**	1367.17	1234.78	**1122.67**	1365.59	1328.00	**1134.57**	1366.79
4	2	2498.45	2258.48	**1691.35**	2507.70	2260.81	**1693.07**	2506.77	2273.50	**1699.70**	2500.79	2282.78	**1706.22**	2527.70	2281.49	**1712.39**
	1	3716.40	3375.60	**2878.44**	3726.99	3380.85	**2882.25**	3734.91	3391.80	**2886.56**	3740.35	3398.68	**2893.28**	**	3404.09	**2895.82**
			$\alpha = .6$			$\alpha = .7$			$\alpha = .8$			$\alpha = .9$			$\alpha = 1$	
	5	815.88	702.16	**543.81**	800.32	705.90	**546.06**	804.61	701.06	**545.67**	798.40	699.20	**535.87**	801.89	701.61	**537.72**
2	2	1639.80	1421.44	**680.81**	1628.88	1413.02	**675.81**	1627.86	1417.00	**673.27**	1620.26	1409.60	**666.14**	1614.50	1416.93	**672.90**
	1	2438.24	2115.09	**1144.98**	2430.70	2107.37	**1144.18**	2424.40	2102.39	**1143.20**	2414.92	2099.57	**1136.75**	2408.49	2098.97	**1129.51**
	5	951.22	**577.31**	695.92	948.80	**571.39**	701.70	881.74	**569.28**	703.86	860.53	**573.10**	690.94	851.73	**560.32**	692.14
3	2	1939.17	1160.59	**875.14**	1926.24	1163.73	**871.23**	1919.14	1152.07	**873.21**	1850.97	1146.24	**872.24**	1758.10	1140.30	**866.30**
	1	2884.95	1734.61	**1475.46**	2873.91	1726.42	**1475.23**	**	1716.93	**1474.96**	**	1708.89	**1472.50**	**	1706.02	**1465.48**
	5	1334.88	**1126.35**	1359.62	1400.08	**1126.34**	1365.38	1468.80	**1118.48**	1355.86	1485.81	**1123.23**	1358.79	1504.70	**1115.74**	1351.22
4	2	2576.35	2279.35	**1701.11**	**	2281.52	**1699.84**	**	2272.96	**1694.16**	**	2271.57	**1697.19**	**	2264.75	**1691.86**
	1	**	3397.24	**2889.19**	**	3392.79	**2882.19**	**	3390.24	**2874.15**	**	3379.60	**2872.03**	**	3371.93	**2864.35**
									N=20							
			$\alpha = .1$			$\alpha = .2$			$\alpha = .3$			$\alpha = .4$			$\alpha = .5$	
	5	2469.13	**2188.12**	2554.46	2464.24	**2197.22**	2558.99	3363.92	**2189.61**	2555.62	3390.42	**2198.33**	2565.43	3420.02	**2200.51**	2565.48
2	2	5002.07	4441.97	**3192.76**	5006.18	4448.37	**3192.31**	5005.23	4453.69	**3194.19**	5052.18	4453.49	**3196.34**	5098.68	4456.70	**3206.97**
	1	7465.20	6627.60	**5104.32**	7468.53	6635.85	**5111.62**	7480.43	6643.10	**5112.35**	**	6649.51	**5119.64**	**	6649.83	**5125.42**
	5	2592.06	**2254.91**	2648.87	2600.14	**2253.47**	2654.25	3520.78	**2255.73**	2655.12	3468.83	**2257.33**	2656.57	3431.78	**2262.62**	2657.78
3	2	5268.57	4576.57	**3310.07**	5277.03	4568.08	**3308.21**	5246.21	4578.94	**3320.39**	5169.59	4587.31	**3319.72**	5111.45	4596.29	**3323.18**
	1	7855.20	6826.80	**5291.52**	7866.94	6828.37	**5296.17**	**	6838.54	**5299.75**	**	6843.04	**5307.08**	**	6851.71	**5308.20**
	5	2729.50	**2461.29**	2755.46	3690.91	**2455.92**	2760.36	3663.41	**2461.17**	2766.54	3641.69	**2464.00**	2761.20	3612.33	**2467.13**	2761.61
4	2	5542.37	4988.35	**3439.40**	5496.66	4994.56	**3448.75**	5467.76	4992.72	**3450.33**	5433.35	5002.74	**3453.20**	**	4993.50	**3450.08**
	1	8269.20	7438.80	**5501.76**	**	7448.19	**5506.48**	**	7450.76	**5510.76**	**	7453.28	**5517.43**	**	7454.81	**5518.91**
			$\alpha = .6$			$\alpha = .7$			$\alpha = .8$			$\alpha = .9$			$\alpha = 1$	
	5	3381.43	**2196.81**	2561.39	3369.91	**2187.99**	2563.57	3334.73	**2191.17**	2555.47	3312.95	**2186.85**	2556.74	3298.19	**2185.78**	2558.69
2	2	5036.81	4451.52	**3205.66**	5015.28	4443.13	**3198.97**	4970.61	4440.39	**3196.64**	4937.24	4435.10	**3190.22**	4918.96	4436.20	**3193.58**
	1	**	6638.98	**5117.04**	**	6627.45	**5110.62**	**	6625.53	**5106.12**	**	6623.44	**5104.45**	**	6621.47	**5096.46**
	5	3418.95	**2262.59**	2657.95	3364.15	**2261.88**	2655.26	3358.66	**2264.79**	2650.94	3299.10	**2257.18**	2647.27	3283.99	**2261.19**	2652.81
3	2	5093.73	4587.72	**3321.27**	5017.58	4591.31	**3313.50**	4999.96	4584.38	**3306.42**	4922.42	4581.78	**3309.19**	4883.78	4585.34	**3303.66**
	1	**	6848.75	**5304.85**	**	6848.73	**5303.42**	**	6845.16	**5297.81**	**	6840.24	**5292.39**	**	6840.18	**5286.16**
	5	3575.41	**2467.53**	2764.13	3556.21	**2461.77**	2755.10	3542.07	**2452.70**	2756.09	3529.40	**2455.13**	2746.81	3496.04	**2453.12**	2753.02
4	2	5330.41	4989.46	**3447.28**	**5297.87**	4995.66	**3448.01**	**5277.77**	4982.35	**3446.74**	**5255.13**	4981.52	**3437.14**	5208.15	4979.08	**3445.96**
	1	**	7452.25	**5518.44**	**	7441.91	**5509.63**	**	7439.64	**5500.72**	**	7433.56	**5497.01**	**	7425.33	**5494.90**
									N=25							
			$\alpha = .1$			$\alpha = .2$			$\alpha = .3$			$\alpha = .4$			$\alpha = .5$	
	5	2729.50	**2461.29**	2755.46	3690.91	**2455.92**	2760.36	3663.41	**2461.17**	2766.54	3641.69	**2464.00**	2761.20	3612.33	**2467.13**	2761.61
2	2	5542.37	4988.35	**3439.40**	5496.66	4994.56	**3448.75**	5467.76	4992.72	**3450.33**	5433.35	5002.74	**3453.20**	**	4993.50	**3450.08**
	1	**	9858.00	**7437.12**	**	9867.18	**7439.19**	**	9878.69	**7447.15**	**	9878.93	**7456.55**	**	**7445.56**	7464.29
	5	8338.81	**3260.35**	3722.32	8390.04	**3263.86**	3719.23	8469.72	**3260.64**	3726.75	8489.18	**3262.74**	3730.09	8550.28	**3264.26**	3740.08
3	2	10671.60	6609.86	**4646.16**	**	6608.91	**4647.21**	**	6624.16	**4658.06**	**	6615.77	**4657.80**	**	6624.06	**4668.20**
	1	**	13537.20	**7518.72**	**	13546.20	**7519.15**	**	13553.44	**7528.02**	**	13557.71	**7534.88**	**	13569.18	**7541.57**
	5	8555.15	4473.29	**3764.59**	8458.07	4477.10	**3764.53**	8480.50	4474.89	**3770.22**	8732.17	4483.44	**3769.55**	8881.35	4482.03	**3777.81**
4	2	**	9075.67	**4697.97**	**	9079.79	**4701.20**	**	9077.72	**4709.94**	**	9088.90	**4711.89**	**	9087.55	**4716.03**
	1	**	14384.40	**9860.16**	**	14394.47	**9865.79**	**	14045.58	**9868.68**	**	14053.29	**9871.96**	**	14053.49	**9879.33**
			$\alpha = .6$			$\alpha = .7$			$\alpha = .8$			$\alpha = .9$			$\alpha = 1$	
	5	3575.41	**2467.53**	2764.13	3556.21	**2461.77**	2755.10	3542.07	**2452.70**	2756.09	3529.40	**2455.13**	2746.81	3496.04	**2453.12**	2753.02
2	2	5330.41	4989.46	**3447.28**	5297.87	4995.66	**3448.01**	5277.77	4982.35	**3446.74**	5255.13	4981.52	**3437.14**	5208.15	4979.08	**3445.96**
	1	**	9880.08	**7460.62**	**	9875.10	**7459.17**	**	9868.34	**7453.98**	**	9860.44	**7444.39**	**	9856.14	**7437.85**
	5	8625.57	**3264.11**	3737.09	8496.36	**3260.34**	3735.51	8557.70	**3264.40**	3728.75	8636.78	**3265.08**	3727.94	8505.16	**3256.56**	3723.55
3	2	**	6624.07	**4670.63**	**	6616.75	**4665.41**	**	6605.43	**4654.53**	**	6611.60	**4661.52**	**	6608.81	**4649.86**
	1	**	13561.40	**7535.74**	**	13560.47	**7529.59**	**	13554.34	**7522.31**	**	13551.50	**7514.02**	**	13548.45	**7508.89**
	5	8787.31	4484.15	**3777.04**	8854.99	4478.47	**3765.44**	8624.09	4473.32	**3769.06**	8944.90	4478.49	**3760.90**	8632.15	4473.17	**3754.23**
4	2	**	9092.38	**4709.60**	**	9079.95	**4713.14**	**	9079.26	**4704.31**	**	9073.29	**4701.13**	**	9081.80	**4698.01**
	1	**	14051.54	**9872.01**	**	14041.42	**9867.35**	**	14040.53	**9861.10**	**	14038.73	**9857.57**	**	14038.32	**9849.29**

** out of memory