



A Benders decomposition method for locating stations in a one-way electric car sharing system under demand uncertainty

Hatice Calik, Bernard Fortz

► To cite this version:

Hatice Calik, Bernard Fortz. A Benders decomposition method for locating stations in a one-way electric car sharing system under demand uncertainty. 2018. hal-01962059

HAL Id: hal-01962059

<https://hal.inria.fr/hal-01962059>

Submitted on 20 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Benders decomposition method for locating stations in a one-way electric car sharing system under demand uncertainty

Hatice Calik^{*1,2} and Bernard Fortz^{†1,3}

¹Department of Computer Science, Université Libre de Bruxelles, Brussels, Belgium

²LORIA - University of Lorraine, 54506 Vandoeuvre-les-Nancy, France

³INOCs, INRIA Lille Nord-Europe, France

February 25, 2018

Abstract

We focus on a problem of locating recharging stations in one-way station based electric car sharing systems which operate under demand uncertainty. We model this problem as a mixed integer stochastic program and develop a Benders decomposition algorithm based on this formulation. We integrate a stabilization procedure to our algorithm and conduct a large-scale experimental study on our methods. To conduct the computational experiments, we developed a demand forecasting method allowing to generate many demand scenarios. The method was applied to real data from Manhattan taxi trips. We are able to solve problems with 100 to 500 scenarios, each scenario including 1000 to 5000 individual customer requests, under high and low cost values and 5 to 15 mins of accessibility restrictions, which is measured as the maximum walking time to the operating stations.

Keywords: Location, Urban Mobility, Electric Car Sharing, Benders Decomposition, Mixed Integer Stochastic Programming, Stochastic Demand

1 Introduction

The increasing number of privately owned cars causes significant amount of pollution, traffic congestion, and parking problems in urban cities. The private cars are usually parked for very long hours and their utilization rates are very low. The car sharing systems are based on shared usage of vehicles by multiple people and the utilization rate of these vehicles, usually owned by companies or organizations, are much higher compared to the private ones. They usually serve their users based on a subscription system and charge each usage by travelling distance or time. Due to this kind of pricing, the users tend to drive less and therefore, create less traffic. Together with the integration of environmental friendly vehicles, car sharing systems have a high potential of reducing these crucial urban problems.

The car sharing systems can be grouped into two categories as station based and free floating depending on whether the customers are required to visit certain stations or not. In station based

*hatice.calik@ulb.ac.be (corresponding author)

†bernard.fortz@ulb.ac.be

systems, where the customers need to visit stations to pick-up and drop the cars, two types of trips can be observed: i) round trips which force to leave the cars to the stations that they were taken from and ii) one-way trips where the cars can be dropped at stations different than the originating one. One-way systems provide more flexibility to the users but requires a more sophisticated management as otherwise there might be significant load imbalances at stations. In free-floating systems, the cars can be parked anywhere within a pre-defined zone and customers can pick the cars from wherever they are parked. Although they are more attractive from a flexibility perspective, free floating systems do not guarantee an available parking space to the users at the end of their trips and the cars need to be parked with at least a certain level of fuel or battery. Moreover, these systems in general require relocation of cars in order to balance the availability at different parts of the city and to perform refuelling/recharging operations.

In this work, we focus on design of a one way electric car sharing system that needs to be operated under demand uncertainty. We assume that the users will be served with a fleet of identical electric cars. Our main goal is to decide on the location and number of stations with limited capacities and the initial level of cars at each station to operate the system in order to maximize the expected profit of the operating company. The expected profit is equal to the expected revenue obtained from served customers minus the fixed cost of opening stations and possession cost of cars. The electric cars have a shorter range than the conventional cars and the recharging operations take much longer. Therefore, the location of stations plays a crucial role in service quality. Knowing this fact, we develop exact methods to design a system that takes into account accessibility of users to located stations, availability of a car at the departure station, and availability of an empty spot at the arrival station for forecasted customer requests. We measure accessibility by the walking distance from origin and destination points, in other words, a station is accessible by a customer if their origin and destination points are within a certain walking distance to some operating stations.

We consider general capacities (non-identical) for stations and assume that the number of charging units is equal to the number of parking spots at the stations. The cars are plugged as soon as they are parked to a station and they are recharged until full battery capacity before being available for the next customer.

We aim to solve large scale problems of locating recharging stations without aggregation of demand. In order to represent uncertainty in demand, we consider multiple scenarios with occurrence probability. Each scenario consists of a certain number of customer requests and each request is associated with an origin node, a destination node, and a starting time. The problem under consideration requires decisions on the number and location of stations, the number of cars available at each station, the customers to be served, the stations they need to visit, and the time of each visit.

We formulate this problem as a two stage stochastic programming model and develop a Benders decomposition algorithm based on its deterministic equivalent. In order to test our methods, we develop a demand forecasting method that uses historical data to generate new requests. We evaluate the performance of our algorithm under different parametric and structural settings by using problem instances obtained from a real taxi trip data of Manhattan. From the experimental studies we conducted on the formulation, we observe that the average linear programming relaxation (LP) gap of the formulation is less than 0.5% on the instances tested. Moreover, a partial relaxation of the formulation provides the same objective value with the original model for most problems with an average gap less than 0.01% in our test bed. We further strengthen our Benders method with a stabilization procedure, which improve the average performance of our algorithm, especially,

on instances with larger number of scenarios.

The rest of this paper is organized as follows: In Section 2, we present a brief summary of the related works in the literature. In Section 3, we provide a mathematical description of our problem with some notation and provide the details of our mixed integer programming formulation. Section 4 presents the results of the computational study we conducted on our formulation and its relaxations. We dedicate Section 5 to a brief summary of the classical Benders decomposition method, its adaptation to pure integer programming (IP) problems in the literature, and a detailed description of the Benders algorithm that we propose. Section 6 follows with the computational study on our Benders algorithm and a stabilized version of it. In this section, we also explain our demand forecasting method. We conclude in Section 7 with a summary of our results and perspectives on future research.

2 Related works

The current studies in the car sharing literature mostly focus on tactical and operational level decision problems. Tactical and operational level problems concerning electric vehicles already attracted attention in logistics systems and several studies in the literature focused on different variants of electric vehicle routing problems [25, 27, 42, 41]. Within the context of car sharing systems, the most commonly considered problem is the relocation of cars to increase availability in high demand areas and avoid surplus of cars at the low demand points [4, 15, 18, 21, 34, 37, 38, 43, 32, 11]. On the other hand, the car sharing literature is relatively sparse in terms of the studies that focus on strategic level decision problems involving location of stations [12].

Optimal location of stations in one-way car sharing systems is first addressed by Correia and Antunes [23]. The authors propose a mixed integer programming formulation that maximizes the profit of the operator by considering several cost and revenue factors. They compare the model, which enforces service via closest stations to origins and destinations, under three service strategies: (i) The operator has the possibility to decide on which customer requests to serve without any restriction (ii) all requests have to be served (iii) the operator can reject a request only if there is no car available at the starting station. The experiments on a real data from Lisbon, Portugal, show that serving all customers might decrease the profit significantly. Later, Correia et al. [24] extend the formulations of Correia and Antunes [23] to a more flexible system where the assumption of service through closest stations is relaxed. The experiments in the latter study reveal that this kind of flexibility increased the profit of the operator together with the introduction of vehicle stock information.

In addition to the optimization models mentioned above, a discrete event simulation based model that analyzes the impact of strategy changes in car sharing systems is introduced by Fassi et al. [26]. The strategies considered include opening new stations, increasing station capacities, merging or splitting stations. The model is evaluated on a car sharing data from Montreal, Canada.

Location of recharging stations in electric car sharing systems is first studied by Boyacı et al. [10]. The authors focus on a multi-objective problem that combines strategic and tactical decisions in a one-way electric car sharing system. The first objective is to maximize the net profit of the operating company and the second objective is to maximize the net benefit of the clients. The authors propose a mixed integer linear programming formulation for solving this problem but due to the intractability of this formulation caused by the large number of relocation variables, they employ an aggregated demand structure and assumed that the charging periods are provided as

input parameters. The proposed method is tested on a real data from Nice, France.

In a recent paper, Çalik and Fortz [16] focus on location of stations in a one-way electric car sharing where the demand is known in advance. The problem includes decisions on location and number of the stations, initial level of cars at operated stations, the subset of requests to serve, and their service routes. The authors provide a mathematical formulation that maximizes the profit of the operator by taking into account the revenue obtained from the served demand and the fixed costs of stations and car purchases. They conduct experiments on large scale problem instances obtained through a real data of Manhattan (New York, USA) taxi trips. In this paper, we extend the mathematical formulation of Çalik and Fortz [16] to the case where there is uncertainty in demand and based on this formulation, we develop a decomposition algorithm that can solve large scale problems in reasonable amount of time. Another recent paper by Biesinger et al. [7] provides a bi-level heuristic algorithm that aims to find the location of recharging stations and the initial number of cars in an electric car-sharing system. The algorithm is tested on real-world instances from Vienna, Austria.

We are also aware of a few recent works [14, 13] that focus on heuristic methods for location of stations in electric car sharing systems. Among them, Brandstätter et al. [14] consider a system with perfect demand information. The authors provide integer programming formulations and construction heuristics which maximize the profit under budget constraints. They conduct experiments on randomly generated grid-graph instances and instances from Vienna (relatively larger in size). Brandstätter et al. [13] extend the methods of Brandstätter et al. [14] to the case with demand uncertainty where demand is represented by multiple scenarios. They consider a maximum walking time of five minutes and seven scenarios in their experiments on the Vienna instances. The model is tested on small-size grid-graph instances (maximum 100 requests, 50 stations, 5 scenarios) and a reduced version of the Vienna instances (1060 requests, 201 stations, 7 scenarios). With this model, most instances are solved to optimality within the one-week time limit enforced by the authors.

Other than the car sharing systems, location of charging stations were also considered for private electric vehicles and urban taxi providers. The literature is relatively denser for these type of problems and it is possible to find both exact solution methodologies [31, 45, 20, 3, 19, 1, 2, 39] and heuristic approaches [33, 35, 44] for location of public charging units.

3 Problem definition and formulation

In this section, we present a formal definition and a mathematical formulation of our problem. Given a city network $G = (V, A)$ with arc set A , node set V , and set of potential stations $J \subset V$, we are required to select a subset \bar{J} of J and locate an initial number of cars to the selected stations in order to satisfy a part of the customer demand represented by multiple scenarios consisting of individual requests. Each request must be served by a car available at a station accessible from the origin node at the time of departure and the car should be left at an arrival station accessible from the destination of the request with a free parking/charging spot at the time of arrival. Additionally, the battery consumption between the departure and the arrival stations assigned to a request should not exceed the battery level of the car and the total length (time) of the trip from the origin node to the destination should not be larger than a given threshold Δ_k for request k . The travelling time and the energy consumption for each request are computed based on the shortest path between the departure and arrival stations allocated. One can also consider the case where these values are known explicitly for each request. A car that completes its service is plugged into a charger at

its arrival station and it does not become available until the beginning of the first period after it has full battery level. For any scenario, the total number of cars (available or being charged) at a station should not exceed the capacity of that station at any time. The objective of the problem is to maximize the total expected profit that is equivalent to the total expected revenue obtained from the requests served minus the total cost of stations selected and cars owned.

Before going into the details of our mathematical formulation, we provide an initial set of parameter definitions used throughout the paper here.

- $V = \{1, \dots, n\}$ is the set of nodes.
- $J = \{1, \dots, m\}$ is the set of potential stations where $J \subset V$.
- f_j is the fixed cost of locating a station on vertex $j \in J$.
- g is the cost of having a car.
- C_j is the capacity of station $j \in J$.
- $T = \{0, \dots, \tau\}$ is the set of time slots (identical length)
- $S = \{1, 2, \dots, |S|\}$ is an index set corresponding to the scenarios.
- q_s is the probability that scenario $s \in S$ will occur.
- K_s is the set of requests in scenario $s \in S$, with origin $O_k \in V$, destination $D_k \in V$, starting time $T_k \in T$, and revenue p_k for $k \in K_s$. $K = \bigcup_{s \in S} K_s$.
- δ_{ij} is the battery usage on the way from station $i \in J$ to station $j \in J$.
- β is the restriction on the battery usage for each car.
- d_{ij} is the travel time from station $i \in J$ to station $j \in J$.
- d_{ij}^w is the walking distance from node $i \in V$ to node $j \in S$.
- β^w is the maximum walking distance between the origin (destination) points of requests and the departure (arrival) stations they are assigned to.
- Δ_k is the maximum length that a route assigned to customer k can have.

We formulate our problem as a two stage stochastic program, in which location of stations and initial number of cars at each station are the first stage decisions whereas the requests served within each scenario and allocation of stations to the requests are second stage decisions. In this paper, we focus on the deterministic equivalent of our model. We propose a path based formulation with a set of decision variables associated with the paths (ordered list of stations to visit) allocated to the requests.

We initially apply a pre-processing procedure to generate a set H_k of all feasible paths (trips) for each request $k \in K$. A path is feasible if it satisfies the accessibility, total length, and battery restrictions. Define $S_s = \bigcup_{k \in K_s} H_k$ for scenario $s \in S$ and $H = \bigcup_{k \in K} H_k = \bigcup_{s \in S} S_s$, then for each $h \in H$, we know the stations to visit on the trip (P_h) together with the time of visit, the amount of time required to recharge the car at the arrival station, thus, the time slot that the car will be

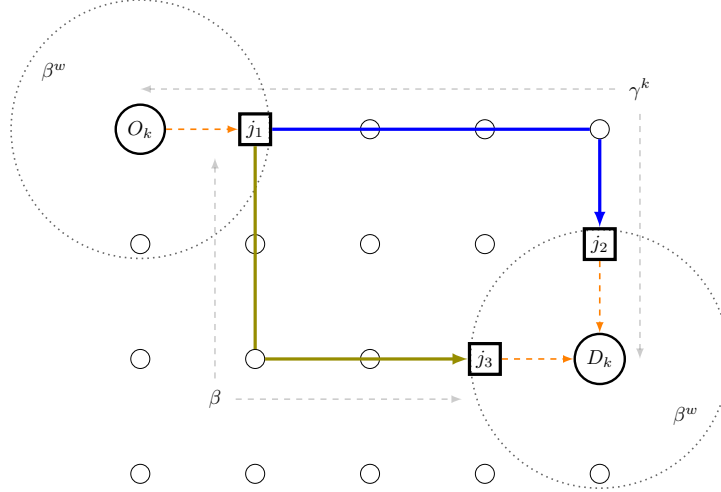


Figure 1: Feasible trips generated for a single request

ready for another customer. We illustrate this procedure for a single request on a simple example in Figure 1. In this example, two feasible trips are illustrated for request k . Both trips start at station j_1 and one ends at station j_2 whereas the other one ends at station j_3 .

Our methods can be easily adapted to more challenging variants of our problem with slight modifications on this pre-processing procedure. One of these variants allows customers to visit intermediate stations when it is not possible to serve a request without recharging the battery or changing the car due to the large distance between the origin and destination nodes. We present a pseudo-code of our pre-processing procedure for this generalized case in Algorithm 1, but we do not consider visits to intermediate stations in our computational results as we focus on the application inside cities where it is unlikely to have these type of requests. Another possible extension would be to consider time dependent travelling times and energy consumptions in the problem.

After the pre-processing, we retrieve the following parameters:

- $b_{h,j}^t = 1$ if the car used for trip $h \in S_s$ of scenario $s \in S$ exits station $j \in J$ at time $t \in T$, 0 otherwise.
- $\mu_{h,j}^t = 1$ if the car used for trip $h \in S_s$ of scenario $s \in S$ is being recharged at station $j \in J$ at time $t \in T$, 0 otherwise.
- $\lambda_{h,j}^t = 1$ if the charging of the car used for trip $h \in S_s$ of scenario $s \in S$ is completed at station $j \in J$ at time $t \in T$, 0 otherwise.

Then, we define the following decision variables for our formulation:

- $u_h = 1$ if trip $h \in S_s$ of scenario $s \in S$ is chosen, 0 otherwise.
- L_j^{ts} is the number of available cars at station $j \in J$ at the beginning of time $t \in T$ for scenario $s \in S$.
- L_j^0 is the number of available cars at station $j \in J$ at time 0.

Algorithm 1 Pre-processing for path (trip) generation

```
1:  $H \leftarrow \emptyset$ ;  
2: for  $s = 1$  to  $|S|$  do  
3:   for  $k = 1$  to  $|K_s|$  do  
4:      $H_k \leftarrow \emptyset$ ;  
5:     for  $i = 1$  to  $m$  do  
6:       if  $d_{O_k i}^w \leq \beta^w$   
7:         for  $j = 1$  to  $m : j \neq i$  do  
8:           if  $d_{j D_k}^w \leq \beta^w$   
9:             if  $d_{O_k i}^w + d_{j D_k}^w + d_{ij} \leq \Delta^k$  and  $\delta_{ij} < \beta$   
10:               $P_1 \leftarrow \emptyset$ ;  
11:              Insert  $i$  to  $P_1$  as the departure station;  
12:              Insert  $j$  to  $P_1$  as the arrival station;  
13:               $P \leftarrow P \cup P_1$ ;  
14:               $H_k \leftarrow H_k \cup \{1\}$ ;  
15:              for  $h = 1$  to  $|H_k|$  do  
16:                for  $u = 1$  to  $m : u \notin P_h$  do  
17:                   $P_{h+u} \leftarrow P_h$ ;  
18:                   $l$  is follower of  $i$  in  $P_{h+u}$ ;  
19:                  if  $\delta_{iu} < \beta$  and  $\delta_{ul} < \beta$   
20:                    if inserting  $u$  after  $i$  does not violate  $\Delta_k$   
21:                      Insert  $u$  after  $i$  in  $P_{h+u}$ ;  
22:                       $P \leftarrow P \cup P_{h+u}$ ;  
23:                       $H_k \leftarrow H_k \cup \{h + u\}$ ;  
24:    $S_s \leftarrow S_s \cup H_k$ ;  
25:    $H \leftarrow H \cup H_k$ ;  
26: return  $S_1, \dots, S_{|S|}, H, P$ ;
```

- $y_j = 1$ if a station is located at vertex $j \in I$, 0 otherwise.

Now, we can write our path based formulation PF as follows:

$$(PF) \quad \max \sum_{s \in S} q_s \sum_{h \in S_s} p_h u_h - \sum_{j \in J} f_j y_j - g \sum_{j \in J} L_j^0 \quad (1)$$

$$\text{s.t.} \quad \sum_{h \in H_k} u_h \leq 1, \quad \forall k \in K \quad (2)$$

$$u_h \leq y_j, \quad \forall j \in J, h \in H : j \in P_h \quad (3)$$

$$\sum_{h \in S_s} b_{hj}^t u_h \leq L_j^{ts}, \quad \forall j \in J, t \in T, s \in S \quad (4)$$

$$L_j^{ts} + \sum_{h \in S_s} (\mu_{hj}^t - b_{hj}^t) u_h \leq C_j y_j, \quad \forall j \in J, t \in T, s \in S \quad (5)$$

$$L_j^{ts} = L_j^{(t-1)s} + \sum_{h \in S_s} (\lambda_{hj}^t - b_{hj}^{(t-1)}) u_h, \quad \forall j \in J, t \in T : t \geq 1, s \in S \quad (6)$$

$$L_j^{0s} = L_j^0, \quad \forall j \in J, s \in S \quad (7)$$

$$0 \leq L_j^{ts} \leq C_j y_j, \quad \forall j \in J, t \in T, s \in S \quad (8)$$

$$L_j^0 \in \mathbb{Z}^+, \quad \forall j \in J \quad (9)$$

$$u_h \in \{0, 1\}, \quad \forall h \in H \quad (10)$$

$$y_j \in \{0, 1\}, \quad \forall j \in J \quad (11)$$

The objective function (1) maximizes the expected profit. The first term in this function gives the expected revenue obtained by serving the customers, the second term is the total fixed cost of opened stations, and the third term is total cost of car possession.

By Constraints (2), a customer is served with at most one trip and by Constraints (3) every station on a selected trip is forced to be opened.

For each scenario, Constraints (4)-(8) ensure the following restrictions: Constraints (4) restrict the number of cars leaving a station with the number of available cars at that station for each time slot. Constraints (5) ensure that the capacity of each station is respected, so that parking a car is not allowed if there is no free space at the station. Constraints (6) balance the number of cars at each station at each time slot. Constraints (7) initialize the number of cars available for each station and Constraints (8) restrict this number with the capacity of that station for each time zone.

Finally, Constraints (9)-(11) give the non-negative integrality and binary restrictions.

In the following section, we present the test results on our formulation PF, its LP relaxation, and the partial relaxation RPF where we keep y variables as binary and relax the integrality of u variables as follows:

$$(RPF) \quad \begin{aligned} & \max (1) \\ & \text{s.t.} (2) - (9), (11) \\ & \quad 0 \leq u_h \leq 1, \quad \forall h \in H \end{aligned} \quad (12)$$

4 Computational study on PF and its relaxations

In order to test our methods, we generate instances with multiple scenarios based on a real data obtained from Manhattan (New York, USA) taxi trips. The data file is based on a city network in Manhattan with 10556 nodes, 85 potential station locations, and 25592 edges. It contains 27549 requests with origin, destination, starting time, revenue, and duration; the cost values for the stations of non-identical capacities, installation of slow or fast charging units, and purchase of three types of electric cars; charging speed for both types of charging units; and the distance and time dependent maximum travelling speed for each edge. Based on the information given in the Manhattan data file, we are able to obtain the values of all the parameters of an instance.

In our experiments, we consider a single car type, a period of 5 minutes, a capacity of a station identical to its number of charging units, and only fast charging units. Through our demand forecasting method (detailed in Section 6.1), we generate instances that include requests with different origin-destination pairs than the ones in the original data file. We compute the revenue of these newly generated requests by using the formula of the original data based on the shortest path distances between the origin and the destination nodes. We obtain the shortest paths for walking and traveling times via Dijkstra’s algorithm.

We conduct our experiments by using IBM ILOG CPLEX 12.7 in the Java environment on an Intel(R) Xeon(R) E5-2630 v3 CPU at 2.40GHz with 16 cores and 32 GB of RAM.

For the Dijkstra’s algorithm, we use the implementation of the JGraphT 1.0.1 package. We impose a one hour time limit and 16 GB memory limit to each experiment.

As the profit values are very small compared to the cost values of the data, in order to have solutions that open stations and serves customers, we introduce a parameter *cost factor* that divides the cost values. This parameter can also be considered as expected repetitions of similar requests in the long run. In our experiments, we use two different values, namely, 10^5 and 10^6 as *cost factor*. Moreover, we use $\Delta_k = 1.1 \times d_{O_k D_k}$ for every $k \in K$.

In the tables of this section, $|K_s|$ denote the number of requests in each scenario generated and this number is the same for every scenario of a single problem instance. The β^w column gives the maximum walking time restriction in minutes, all other time values are given in seconds. The columns labelled ‘Obj’ give the solution value obtained from the corresponding method at termination due to optimality or resource limitations (time or memory). We report two types of gap percentages in our tables: $g1(\%)$ gives the gap relative to the best known PF solution value, say Obj^* , then, $g1(\%) = 100 \times [(Obj - Obj^*)/Obj^*]$; and $g2(\%)$ denotes the gap provided by CPLEX at the termination. We here note that for some instances, Obj^* value might be obtained via the methods that we propose in Section 5. We further indicate the instances that cannot be solved to optimality within the time limit with ‘TL’ under the ‘time’ column. We do not hit the memory limit in any of the instances.

In Table 1, we present the numbers related to pre-processing on problems with 100 scenarios and with and 200 scenarios. For each value of β^w , column $|K_a|$ reports the total number of requests that satisfy accessibility and maximum length restrictions over all scenarios, $|K_{ad}|$ is the number of such distinct requests, and $|H|$ is the total number of trips generated i.e. the cardinality of the \mathbf{u} variables in the model. Finally, ‘PP time’ gives the total amount of time spent during the preprocessing of the corresponding instance.

We provide results regarding PF, its LP relaxation, and RPF for *cost factor* values 10^5 and 10^6 in Tables 2 and 3, respectively, on instances with $|S| = 100$. In these tables, we observe that the LP relaxations provide very tight bounds with an average gap of 0.64% and 0.03% for 10^5 and 10^6 ,

Table 1: Pre-processing results for $|S| = 100, 200$

$ K_s $	β^w	$ S = 100$				$ S = 200$			
		$ K_a $	$ K_{ad} $	$ H $	PP time	$ K_a $	$ K_{ad} $	$ H $	PP time
1000	5	5715	1520	10197	29.87	11409	1592	20218	44.67
1000	6	10009	2695	21490	60.05	20078	2818	42756	64.32
1000	7	15566	4218	39246	80.80	31176	4397	78283	119.25
1000	8	22856	6153	65916	131.45	45784	6429	131995	138.59
1000	9	30233	8099	104752	207.60	60499	8460	210193	204.72
1000	10	37658	10081	155145	271.23	75292	10528	310244	334.02
1000	15	76527	19981	716437	1800.31	152828	20860	1431576	1566.76
2000	5	11409	1592	20218	44.10	22624	1594	40064	76.34
2000	6	20078	2818	42756	67.67	39789	2823	84643	97.71
2000	7	31176	4397	78283	111.69	62207	4406	155859	116.85
2000	8	45784	6429	131995	190.21	91535	6439	263305	194.93
2000	9	60499	8460	210193	223.84	120535	8478	418134	273.68
2000	10	75292	10528	310244	289.60	149971	10549	616824	373.44
2000	15	152828	20860	1431576	1698.36	305226	20907	2851841	1400.74
3000	5	16979	1594	30128	58.30	33907	1594	60464	80.21
3000	6	29905	2823	63812	84.49	59593	2823	127077	112.58
3000	7	46683	4406	117372	121.09	93193	4406	234083	155.48
3000	8	68646	6439	198109	134.50	137283	6439	394892	204.53
3000	9	90536	8476	314551	245.62	180778	8478	627646	265.60
3000	10	112625	10546	463870	328.44	224855	10549	925580	361.29
3000	15	228923	20904	2144200	1784.09	458144	20907	4280165	1359.83
4000	5	22457	1594	40017	66.25	44906	1594	80122	120.19
4000	6	39764	2823	84486	93.28	79192	2823	168424	141.16
4000	7	61802	4406	154983	131.33	123282	4406	308972	183.73
4000	8	91513	6439	262714	175.64	182787	6439	524433	210.10
4000	9	120614	8478	417605	253.41	240840	8478	834089	317.06
4000	10	149707	10549	615214	362.25	299345	10549	1229742	388.21
4000	15	305270	20907	2838284	1762.09	610535	20907	5682785	1567.85
5000	5	27931	1594	49851	91.74	56107	1594	100058	142.61
5000	6	49719	2823	105194	103.88	99451	2823	210730	170.24
5000	7	77459	4406	192986	127.08	155197	4406	387333	176.42
5000	8	114467	6439	327936	194.83	229417	6439	656427	234.15
5000	9	150649	8478	521632	251.97	301944	8478	1042957	256.36
5000	10	187213	10549	769974	342.48	375029	10549	1540125	421.67
5000	15	381970	20907	3572493	1304.47	763499	20907	7131274	1346.57
Avg.:					377.83				377.77
Max:					1800.31				1567.85

respectively. The largest LP gap observed is 3.88% in Table 2 and 0.14% in Table 3.

As expected, PF spends a larger amount of time to solve problems with larger β^w values. PF is not able to find any solution with a positive profit value in instances with $\beta^w = 15$ mins and it spends all the time given for the presolve of CPLEX when $|K_s| \geq 2000$ for $\beta^w = 15$.

We present similar results for $|S| = 200$ in Tables 4 and 5. As the instances with $\beta^w = 15$, $|K_s| \geq 3000$ are intractable, we omit corresponding problems for $|S| = 200$ tables.

Our experiments reveal that the computational difficulty increases parallel to the number of scenarios as the number of constraints and variables increases significantly. In order to overcome this difficulty, we describe a Benders decomposition algorithm in the following section.

Another interesting observation on these results is that the objective value of RPF is the same as the objective value of PF for almost every instance and the difference is very small for the nonidentical ones. As it will be explained in the next section, this observation encourages us to focus on two phase solution methodologies that will first focus on solving RPF which has less number variables with integrality restrictions compared to PF and then reach to the optimality of PF by using the information obtained from RPF.

5 Decomposition algorithm (BAC)

Before presenting the details of our algorithm, we give a general description of the Benders decomposition method [6]. For a mixed integer program with a group of integer variables and a group of continuous variables, the algorithm consists of solving the formulation by temporarily removing all continuous variables. This problem is called the master problem. Then, the Benders algorithm focuses on the dual of a subproblem, which is simply a restriction of the original problem where all the integer variables are fixed to the values of the master problem. From the dual problem, a so called feasibility cut for each extreme ray and an optimality cut for each extreme point is obtained. These cuts are added to the master problem to obtain another solution and the procedure is repeated. Since enumeration of all extreme rays and extreme points is not very practical, cutting plane algorithms are commonly used in the literature.

The convergence rate of the classical Benders decomposition method might be very slow especially if the subproblem is difficult to solve. On the other hand, it might be very efficient if the subproblem can be decomposed further into smaller and easy-to-solve problems as in multi-commodity, multi-period, or multi-scenario problems [8]. Motivated by this fact, we aim further decomposing the second-stage problem of our two-stage recourse problem into smaller problems, each one corresponding to a single demand scenario. For this purpose, we decide to choose y, L^0 as first stage variables that are kept in the master problem and u, L as the variables of the subproblem. However, as u variables are integral, we cannot apply the classical Benders decomposition to PF without any further adaptations. On the other hand, with this type of categorization of variables, RPF is a good candidate for applying the Benders procedure as u, L variables are continuous in this model. In this case, we need additional steps to make sure that the optimal solution of our algorithm provides an integral vector u . Several successful adaptations of the classical Benders decomposition algorithm to different IP problems are available due to [9, 29, 36]. In our implementation, we use the ‘‘Combinatorial Benders cuts’’, which are first named by Codato and Fischetti [22].

We implemented our Benders algorithm as a two phase method. In the first phase of our algorithm, we solve RPF with a classical Benders framework. Let $v(RPF)$ be the optimal value

Table 2: Model results for $|S| = 100$ and $costfactor = 10^5$ - gaps & solution values

$ K_s $	b^w	LP		RPF		PF		time
		Obj	g1(%)	Obj	g2(%)	Obj	g2(%)	
1000	5	7342.31	1.71	7218.65	0.00	7218.65	0.00	4.12
1000	6	15605.92	3.88	15022.45	0.00	15022.45	0.00	15.76
1000	7	27573.03	1.80	27085.46	0.00	27085.46	0.00	46.71
1000	8	37518.81	1.45	36980.87	0.00	36980.87	0.00	247.10
1000	9	44984.52	0.58	44723.11	0.00	44723.11	0.00	656.09
1000	10	52304.87	0.28	52158.64	0.00	52158.64	0.00	723.22
1000	15	89960.52	0.13	0.00	NA	0.00	NA	TL
2000	5	21016.22	0.85	20839.93	0.00	20839.93	0.00	11.72
2000	6	32538.52	1.90	31931.30	0.00	31931.30	0.00	33.56
2000	7	45264.42	0.80	44907.15	0.00	44907.15	0.00	97.92
2000	8	53763.37	0.32	53593.97	0.00	53593.97	0.00	294.15
2000	9	61039.66	0.06	61004.03	0.00	61004.03	0.00	850.56
2000	10	68256.41	0.12	68173.08	0.00	68173.08	0.00	2987.40
2000	15	120772.52	0.05	0.00	NA	NA	NA	TL
3000	5	25213.00	1.74	24781.00	0.00	24781.00	0.00	18.78
3000	6	42570.33	0.87	42204.00	0.00	42204.00	0.00	86.56
3000	7	53092.93	0.24	52964.00	0.00	52964.00	0.00	794.61
3000	8	63073.87	0.22	62933.00	0.00	62933.00	0.00	784.47
3000	9	72583.38	0.15	72480.00	0.00	72476.00	0.04	TL
3000	10	78374.00	0.02	78358.00	0.00	78358.00	0.00	TL
3000	15	141114.40	NA	0.00	NA	NA	NA	TL
4000	5	32185.67	0.64	31980.00	0.00	31980.00	0.00	27.40
4000	6	50607.83	0.32	50444.00	0.00	50444.00	0.00	94.88
4000	7	63640.95	0.25	63485.00	0.00	63485.00	0.00	1720.41
4000	8	70676.08	0.14	70577.00	0.00	70577.00	0.00	2042.30
4000	9	76461.20	0.04	76346.00	0.12	76431.00	0.01	1655.19
4000	10	79728.28	0.03	79666.00	0.00	79708.00	0.00	2329.47
4000	15	149452.50	NA	0.00	NA	NA	NA	TL
5000	5	38549.29	0.99	38169.82	0.00	38169.82	0.00	68.72
5000	6	53705.84	0.41	53488.83	0.00	53488.83	0.00	312.48
5000	7	65663.05	0.17	65554.10	0.00	65554.10	0.00	1877.70
5000	8	72589.00	0.05	72554.02	0.00	72554.02	0.00	2357.71
5000	9	79130.76	0.16	79005.40	0.00	79005.40	0.04	TL
5000	10	85073.95	0.09	84986.33	0.00	84998.90	0.00	TL
5000	15	161011.19	NA	NA	NA	NA	NA	TL
Avg.:			0.64		0.00		0.00	1518.65
Max:			3.88		0.12		0.04	3777.61

Table 3: Model results with $|S| = 100$ and $costfactor = 10^6$ - gaps & solution values

$ K_s $	b^w	LP		RPF		PF		time
		Obj	g1(%)	Obj	g2(%)	Obj	g2(%)	
1000	5	16980.75	0.03	16975.83	0.00	16975.83	0.00	5.59
1000	6	27698.48	0.08	27675.73	0.00	27675.73	0.00	14.35
1000	7	41089.05	0.14	41030.45	0.00	41030.45	0.00	42.35
1000	8	51404.74	0.05	51380.12	0.00	51380.12	0.00	116.04
1000	9	58869.93	0.05	58838.80	0.00	58838.80	0.00	543.88
1000	10	66417.96	0.02	66402.34	0.00	66402.34	0.00	521.08
1000	15	106736.62	0.00	0.00	NA	0.00	NA	TL
2000	5	33364.02	0.04	33349.13	0.00	33349.13	0.00	15.77
2000	6	45939.10	0.07	45904.90	0.00	45904.90	0.00	41.48
2000	7	59443.25	0.07	59399.62	0.00	59399.62	0.00	139.59
2000	8	67929.84	0.04	67903.56	0.00	67903.56	0.00	1096.80
2000	9	75852.91	0.01	75843.23	0.00	75843.23	0.00	TL
2000	10	83831.89	0.01	83820.27	0.00	83820.27	0.00	TL
2000	15	139678.53	0.00	0.00	NA	NA	NA	TL
3000	5	36036.00	0.09	36003.00	0.00	36003.00	0.00	23.38
3000	6	55492.75	0.05	55466.00	0.00	55466.00	0.00	96.77
3000	7	67553.75	0.04	67527.00	0.00	67527.00	0.00	248.83
3000	8	78213.28	0.01	78209.00	0.00	78209.00	0.00	701.07
3000	9	88457.67	0.01	88446.00	0.00	88446.00	0.00	2693.32
3000	10	94913.75	0.01	94909.00	0.00	94909.00	0.00	2726.03
3000	15	160490.00	NA	0.00	NA	NA	NA	TL
4000	5	45154.33	0.04	45137.00	0.00	45137.00	0.00	32.39
4000	6	65262.25	0.02	65252.00	0.00	65252.00	0.00	115.26
4000	7	79767.07	0.02	79753.00	0.00	79753.00	0.00	1746.54
4000	8	86871.00	0.00	86868.00	0.00	86868.00	0.00	1035.95
4000	9	93277.00	0.00	93277.00	0.00	93277.00	0.00	1655.19
4000	10	96947.00	0.00	96947.00	0.00	96947.00	0.00	1824.23
4000	15	168970.00	NA	0.00	NA	NA	NA	TL
5000	5	51641.00	0.05	51614.58	0.00	51614.58	0.00	53.21
5000	6	68400.66	0.06	68362.75	0.00	68362.75	0.00	423.10
5000	7	81719.65	0.03	81695.37	0.00	81695.37	0.00	1223.83
5000	8	89801.18	0.02	89786.21	0.00	89786.21	0.00	2706.08
5000	9	96651.38	0.01	96645.94	0.00	96645.70	0.00	3337.07
5000	10	103253.62	0.00	103252.48	0.00	103252.48	0.00	3024.76
5000	15	180913.30	NA	0.00	NA	NA	NA	TL
Avg.:			0.03		0.00		0.00	1491.68
Max:			0.14		0.00		0.00	3888.52

Table 4: Model results for $|S| = 200$ and $costfactor = 10^5$ - gaps & solution values

$ K_s $	b^w	LP		RPF		PF		time
		Obj	g1(%)	Obj	g2(%)	Obj	g2(%)	
1000	5	6969.82	2.00	6832.88	0.00	6832.88	0.00	14.46
1000	6	15444.12	3.74	14887.28	0.00	14887.28	0.00	62.97
1000	7	27442.90	1.66	26994.52	0.00	26994.52	0.00	265.89
1000	8	37319.10	1.50	36740.66	0.18	36766.38	0.00	2550.04
1000	9	44926.41	0.64	0.00	NA	44496.02	0.38	TL
1000	10	52354.37	0.30	0.00	NA	0.00	NA	TL
1000	15	0.02	NA	0.00	NA	NA	NA	TL
2000	5	17369.70	0.69	17251.00	0.00	17251.00	0.00	31.43
2000	6	32614.25	1.22	32220.00	0.00	32220.00	0.00	147.28
2000	7	44027.19	1.16	43229.00	0.74	43524.00	0.09	TL
2000	8	52368.69	0.14	0.00	NA	52294.00	0.00	2088.65
2000	9	62119.02	1.55	0.00	NA	0.00	NA	TL
2000	10	68569.30	1.74	0.00	NA	0.00	NA	TL
2000	15	0.00	NA	0.00	NA	NA	NA	TL
3000	5	25213.00	1.74	24781.00	0.00	24781.00	0.00	63.67
3000	6	42570.33	0.87	42204.00	0.00	42204.00	0.00	485.12
3000	7	53092.93	0.24	52964.00	0.00	52964.00	0.00	3286.40
3000	8	63073.87	2.76	0.00	NA	0.00	NA	TL
3000	9	72583.38	1.42	0.00	NA	0.00	NA	TL
3000	10	78374.00	0.75	0.00	NA	0.00	NA	TL
4000	5	32046.28	0.64	31842.00	0.00	31842.00	0.00	160.84
4000	6	50500.08	0.32	50337.72	0.00	50337.72	0.00	575.42
4000	7	63537.28	0.25	63005.95	0.62	0.00	NA	TL
4000	8	70579.57	0.14	0.00	NA	0.00	NA	TL
4000	9	76396.01	0.03	0.00	NA	0.00	NA	TL
4000	10	79660.24	0.05	0.00	NA	NA	NA	TL
5000	5	38280.05	0.91	37934.12	0.00	37934.12	0.00	673.09
5000	6	53494.06	0.36	53216.43	0.16	53300.98	0.00	1903.53
5000	7	65459.66	0.15	0.00	NA	0.00	NA	TL
5000	8	72521.60	0.05	0.00	NA	0.00	NA	TL
5000	9	79074.33	0.16	0.00	NA	0.00	NA	TL
5000	10	85072.10	0.09	0.00	NA	NA	NA	TL
Avg.:			0.91		0.11		0.03	2438.34
Max:			3.74		0.74		0.38	3739.77

Table 5: Model results with $|S| = 200$ and $costfactor = 10^6$ - gaps & solution values

$ K_s $	b^w	LP		RPF		PF		time
		Obj	g1(%)	Obj	g2(%)	Obj	g2(%)	
1000	5	16766.10	0.00	16766.10	0.00	16766.10	0.00	12.64
1000	6	27615.35	0.07	27597.19	0.00	27597.19	0.00	30.69
1000	7	41037.19	0.11	40992.58	0.00	40992.58	0.00	79.90
1000	8	51395.72	0.05	51371.33	0.00	51371.33	0.00	354.34
1000	9	58943.52	0.07	58855.20	0.11	58899.58	0.01	TL
1000	10	66533.64	0.01	66524.12	0.01	66528.32	0.00	1782.33
1000	15	0.21	NA	0.00	NA	NA	NA	TL
2000	5	27767.50	0.03	27758.00	0.00	27758.00	0.00	24.26
2000	6	44523.50	0.09	44485.00	0.00	44485.00	0.00	89.46
2000	7	56945.73	0.04	56921.00	0.00	56921.00	0.00	1051.88
2000	8	65833.38	0.01	65824.00	0.00	65824.00	0.00	TL
2000	9	76688.18	0.80	0.00	NA	0.00	NA	TL
2000	10	83826.47	0.21	0.00	NA	0.00	NA	TL
2000	15	0.01	NA	0.00	NA	NA	NA	TL
3000	5	36036.00	0.09	36003.00	0.00	36003.00	0.00	134.59
3000	6	55492.75	0.05	55466.00	0.00	55466.00	0.00	388.46
3000	7	67553.75	0.04	67527.00	0.00	67527.00	0.00	1150.57
3000	8	78213.28	0.01	78206.00	0.00	78209.00	0.00	1877.87
3000	9	88457.67	0.16	0.00	NA	0.00	NA	TL
3000	10	94913.75	0.22	0.00	NA	0.00	NA	TL
4000	5	45063.29	0.04	45047.24	0.00	45047.24	0.00	177.75
4000	6	65174.09	0.02	65164.01	0.00	65164.01	0.00	747.65
4000	7	79657.46	0.02	79643.34	0.00	79643.34	0.01	TL
4000	8	86775.66	0.01	86768.81	0.00	0.00	NA	TL
4000	9	93220.05	0.00	93220.05	0.00	0.00	NA	TL
4000	10	96908.38	0.01	0.00	NA	0.00	NA	TL
5000	5	51525.43	0.03	51508.09	0.00	51508.09	0.00	429.57
5000	6	68232.66	0.03	68215.50	0.00	68215.50	0.00	1099.12
5000	7	81548.30	0.03	81521.92	0.00	81521.92	0.00	2349.23
5000	8	89735.76	0.02	89720.81	0.00	89714.20	0.02	TL
5000	9	96601.81	0.00	0.00	NA	NA	NA	TL
5000	10	103249.59	0.00	0.00	NA	0.00	NA	TL
Avg.:			0.07		0.01		0.00	2081.68
Max:			0.80		0.11		0.02	3768.34

of RPF and $(\mathbf{y}^*, \mathbf{L}^*)$ be the optimal values of the first stage variables $(\mathbf{y}, \mathbf{L}^0)$. Then, we solve PF by fixing values of y, L^0 variables to $(\mathbf{y}^*, \mathbf{L}^*)$ to see if there exists an integer feasible solution for RP with objective value $v(RPF)$. If yes, we stop as we have an optimal solution of PF on hand. Otherwise, we move to the second phase of the algorithm. The motivation behind this first phase is the fact that RPF gives the optimal value of PF in most instances; so, the values of the first stage decision variables obtained from RPF might possibly provide a complete integral solution for PF and avoid excessive number of integrality checks for addition of relatively weak combinatorial cuts.

In Phase II, we restart a branch-and-cut framework for RPF and restrict the objective value with $v(RPF)$ from above. Then, for integer solutions of RPF tree, we solve individual subproblems with integrality restrictions on u variables for each scenario and add combinatorial Benders cuts if these subproblems are integer infeasible. Our experimental study reveals that the two-phase procedure converges faster than introducing the combinatorial cuts at the first phase in most instances and it is much more efficient in overall as most problems do not require the second phase.

Now we present the details of Phase I and Phase II of our algorithm.

Phase I:

For practical reasons, we make a modification on our formulation so that we obtain only feasibility cuts from our dual subproblem. We define a nonnegative decision variable z_s for each scenario $s \in S$ and ensure that its value will be equal to the revenue obtained from that scenario as follows:

$$(PF2) \quad \max \sum_{s \in S} q_s z_s - \sum_{j \in J} f_j y_j - g \sum_{j \in J} L_j^0 \quad (13)$$

$$\text{s.t.} \quad \sum_{h \in S_s} p_h u_h \geq z_s, \quad \forall s \in S \quad (14)$$

$$(2) - (11)$$

$$0 \leq z_s \leq \sum_{k \in K_s} p_k, \quad \forall s \in S \quad (15)$$

To solve $(PF2)$ in a Benders fashion, we keep $\mathbf{z}, \mathbf{y}, \mathbf{L}^0$ variables in the master problem and deal only with \mathbf{u} variables in the subproblem.

In order to make our master problem stronger, we introduce two sets of valid inequalities (18) and (19). Thus, we solve (MP) given below as the master problem of our Benders algorithm.

$$(MP) \quad \max \sum_{s \in S} q_s z_s - \sum_{j \in J} f_j y_j - g \sum_{j \in J} L_j^0 \quad (16)$$

$$\text{s.t.} \quad \sum_{m=1}^{C_j} x_{jm} \leq y_j, \quad \forall j \in J \quad (17)$$

$$z_s \leq \sum_{k \in K_s} p_k \sum_{j \in N_{O_k}} y_j, \quad \forall s \in S \quad (18)$$

$$z_s \leq \sum_{k \in K_s} p_k \sum_{j \in N_{D_k}} y_j, \quad \forall s \in S \quad (19)$$

$$z_s \geq 0, \quad \forall s \in S \quad (20)$$

(9), (11)

In this formulation, N_{O_k} and N_{D_k} denote the set of stations accessible by, respectively, origin and destination, of request $k \in K$. Once we introduce fixed values $(\bar{\mathbf{z}}, \bar{\mathbf{y}}, \bar{\mathbf{L}}^0)$ to our subproblem, we observe that it can be decomposed into $|S|$ problems, one for each scenario, that can be solved independently.

At this point, we make some simplifications and modifications to have smaller number of constraints and variables in our primal and dual subproblems. The first observation is as follows:

Property 1. *Constraints (8) are redundant.*

Proof. (i) $L_j^{ts} \geq 0, \forall j \in J, t \in T, s \in S$:

By Constraints (4), $L_j^{ts} - \sum_{h \in S_s} b_{hj}^t u_h \geq 0, \forall j \in J, t \in T, s \in S$. Then, by Constraints (6), $L_j^{ts} = L_j^{(t-1)s} - \sum_{h \in S_s} b_{hj}^{(t-1)} u_h + \sum_{h \in S_s} \lambda_{hj}^t u_h \geq 0, \forall j \in J, t \in T, s \in S$ as $\lambda_{hj}^t, u_h \in \{0, 1\}, \forall j \in J, t \in T, h \in H$.

(ii) $L_j^{ts} \leq C_j y_j, \forall j \in J, t \in T, s \in S$:

By Constraints (5), $\sum_{h \in S_s} \mu_{hj}^t u_h \leq C_j y_j + \sum_{h \in S_s} b_{hj}^t u_h - L_j^{ts}, \forall j \in J, t \in T, s \in S$. Moreover, $\sum_{h \in S_s} \lambda_{hj}^{(t+1)} u_h \leq \sum_{h \in S_s} \mu_{hj}^t u_h, \forall j \in J, t < \tau$ as the number of cars who finished recharging at $t+1$ cannot be larger than the number of cars being charged at t . Then, $\sum_{h \in S_s} \lambda_{hj}^{(t+1)} u_h + L_j^{ts} - \sum_{h \in S_s} b_{hj}^t u_h \leq C_j y_j$ implying that $L_j^{(t+1)s} \leq C_j y_j, \forall j \in J, t \in T : t < \tau, s \in S$ via Constraints (6). □

Moreover, the following replacement of L_j^{ts} is possible due to Constraints (6) and (7):

$$L_j^{ts} = L_j^0 + \sum_{h \in S_s} \left(\sum_{r=1}^t \lambda_{hj}^r - \sum_{r=0}^{t-1} b_{hj}^r \right) u_h, \quad \forall j \in J, t \in T : t \geq 1, s \in S \quad (21)$$

If we replace L_j^{ts} with the right hand side of (21) throughout PF2 and omit equality Constraints (6) and (7), we obtain a formulation with $\mathbf{z}, \mathbf{y}, \mathbf{u}, \mathbf{L}^0$ variables only. Then, for each scenario $s \in S$, we can express the corresponding subproblem (SP_s) as follows:

$$(SP_s) \quad \max 0 \quad (22)$$

$$\text{s.t.} \quad \sum_{h \in S_s} p_h u_h \geq \bar{z}_s, \quad (23)$$

$$\sum_{h \in H_k} u_h \leq 1, \quad \forall k \in K_s \quad (24)$$

$$u_h \leq \bar{y}_j, \quad \forall h \in S_s, j \in P_h \quad (25)$$

$$\sum_{h \in S_s} \left(\sum_{r=0}^t b_{hj}^r - \sum_{r=1}^t \lambda_{hj}^r \right) u_h \leq \bar{L}_j^0, \quad \forall j \in J, t \in T \quad (26)$$

$$\sum_{h \in S_s} \left(\sum_{r=1}^t \lambda_{hj}^r - \sum_{r=0}^t b_{hj}^r + \mu_{hj}^t \right) u_h \leq C_j \bar{y}_j - \bar{L}_j^0, \quad \forall j \in J, t \in T \quad (27)$$

$$u_h \in \{0, 1\}, \quad \forall h \in S_s \quad (28)$$

Let LP_s denote the linear programming (LP) relaxation of SP_s , where the integrality restriction on \mathbf{u} variables is removed, therefore, $0 \leq u_h \leq 1, \forall h \in S_s$; DLP_s denote the dual of LP_s ; and $\alpha, \epsilon_k, \gamma_{jh}, \theta_{jt}$, and ω_{jt} be the dual variables associated with Constraints (23)-(27), respectively. Here, we note that $u_h \leq 1$ restriction is already satisfied due to (24) and (25) $\forall h \in S_s$, so, we do not include an additional set of constraints for this restriction. Then, we can express DLP_s as follows:

$$(DLP_s) \quad \min -\bar{z}_s \alpha + \sum_{k \in K_s} \epsilon_k + \sum_{h \in S_s, j \in P_h} \bar{y}_j \gamma_{hj} + \sum_{j,t} \bar{L}_j^0 \theta_{jt} \\ + \sum_{j,t} (C_j \bar{y}_j - \bar{L}_j^0) \omega_{jt} \quad (29)$$

$$\text{s.t.} \quad -p_h \alpha + \sum_{k: h \in H_k} \epsilon_k + \sum_{j \in P_h} \gamma_{hj} + \sum_{j,t} \left(\sum_{r=0}^t b_{hj}^r - \sum_{r=1}^t \lambda_{hj}^r \right) \theta_{jt} \\ + \sum_{j,t} \left(\sum_{r=1}^t \lambda_{hj}^r - \sum_{r=0}^t b_{hj}^r + \mu_{hj}^t \right) \omega_{jt} \geq 0, \quad \forall h \in S_s \quad (30)$$

$$\alpha \geq 0, \quad (31)$$

$$\epsilon_k \geq 0, \quad \forall k \in K_s \quad (32)$$

$$\gamma_{hj} \geq 0, \quad \forall h \in S_s, j \in P_h \quad (33)$$

$$\theta_{jt}, \omega_{jt} \geq 0, \quad \forall j \in J, t \in T \quad (34)$$

In our implementation, we introduce $\alpha \leq 1$ to (DLP_s) and associate unboundedness of the problem with a negative objective value. Therefore, as soon as the optimal value obtained from (DLP_s) is negative, we add feasibility cuts (35) to our master problem. An important property of DLP_s is that the feasible region of the model does not change by the solution of the master problem. We make use of this property by generating the individual dual problems once in our implementation

and changing only the objective function at each iteration. This modification in the implementation improves the efficiency of our algorithm significantly.

$$-\bar{\alpha}z_s + \sum_j \left(\sum_{h:j \in P_h} \bar{\gamma}_{hj} + C_j \sum_t \bar{\omega}_{jt} \right) y_j + \sum_j \left(\sum_t \bar{\theta}_{jt} - \sum_t \bar{\omega}_{jt} \right) L_j^0 \geq - \sum_{k \in S_s} \bar{\epsilon}_k \quad (35)$$

In addition to every integral solution of the branch-and-bound tree, we add feasibility cuts (35) for violated fractional solutions at the root node as well. We use the *LazyConstraintCallback* and *UserCutCallback* of CPLEX for separation of, respectively, integral and fractional solutions.

After this branch-and-cut first phase, the feasibility of the optimal solution is checked as follows. Let $v(RPF)$ be the optimal value of RPF and $(\mathbf{y}^*, \mathbf{L}^*)$ be the optimal values of the first stage variables $(\mathbf{y}, \mathbf{L}^0)$. Then, we solve PF by fixing values of $(\mathbf{y}, \mathbf{L}^0)$ variables to $(\mathbf{y}^*, \mathbf{L}^*)$ to see if there exists an integer feasible solution for RP with objective value $v(RPF)$. If yes, we stop as we have an optimal solution of PF on hand. Otherwise, we move to the second phase of the algorithm.

Phase II:

In order to start Phase II, we make use of $v(RPF)$ and $(\mathbf{y}^*, \mathbf{L}^*)$ obtained from Phase I. At this stage, we introduce a new set of binary variables \mathbf{x} that will represent integral \mathbf{L}^0 variables with the variable transformation $L_j^0 = \sum_{m=1}^{C_j} mx_{jm}, j \in J$ in MP. Doing so, we can introduce combinatorial Benders cuts associated with these variables when needed. We further add the following constraints to MP:

$$\sum_{s \in S} q_s z_s - \sum_{j \in J} f_j y_j - g \sum_{j \in J} \sum_{m=1}^{C_j} mx_{jm} \leq v(RPF) \quad (36)$$

$$\sum_{m=1}^{C_j} x_{jm} \leq y_j, \quad \forall j \in J \quad (37)$$

Similar to Phase I, we solve DLP_s for separation not only at integer solutions of the branch-and-bound tree but also at fractional solutions at the root node. But this time, the separation cut would be (38).

$$-\bar{\alpha}z_s + \sum_j \left(\sum_{h:j \in P_h} \bar{\gamma}_{hj} + C_j \sum_t \bar{\omega}_{jt} \right) y_j + \sum_j \left(\sum_t \bar{\theta}_{jt} - \sum_t \bar{\omega}_{jt} \right) \sum_{m=1}^{C_j} mx_{jm} \geq - \sum_{k \in S_s} \bar{\epsilon}_k \quad (38)$$

Additionally, once DLP_s is bounded (i.e. LP_s feasible) for every $s \in S$, we check if we have an integral solution for $SPMax_s$ for the current $(\bar{\mathbf{y}}, \bar{\mathbf{x}})$ values of the master problem iteratively for $s = 1, \dots, |S|$.

$$\begin{aligned} (SP_s Max) \quad & \max \sum_{h \in S_s} p_h u_h \\ & \text{s.t. (24) - (28).} \end{aligned} \quad (39)$$

If $SPMax_s$ is infeasible for some $s \in S$, we add the combinatorial feasibility cut (40) and do not check integer feasibility for the remaining subproblems. If $SPMax_s$ is feasible but the optimal value is greater than \bar{z}_s for some $s \in S$, then we add the optimality cut (41). This cut requires either the value of z_s variable to be tightened by $\sum_{h \in S_s} p_h u_h^*$ for the same (\bar{y}, \bar{x}) solution or to change the value of y_j for some $j \in J$ or x_{jm} for some (j, m) where $j \in J, m = 1, \dots, C_j$. At the termination of this branch-and-cut procedure, we obtain an optimal solution for PF.

$$F(\bar{y}, \bar{x}) \geq 1 \tag{40}$$

$$z_s \leq \sum_{h \in S_s} p_h u_h^* + \sum_{k \in K_s} p_k F(\bar{y}, \bar{x}) \tag{41}$$

where \mathbf{u}^* is the solution vector obtained from $SPMax_s$ of $s \in S$ and

$$F(\bar{y}, \bar{x}) = \sum_{j: \bar{y}_j=0} y_j + \sum_{j: \bar{y}_j=1} (1 - y_j) + \sum_{j, m: \bar{x}_{jm}=0} x_{jm} + \sum_{j, m: \bar{x}_{mj}=1} (1 - x_{jm}).$$

6 Computational Study on BAC

In this section, we provide a detailed analysis of the experimental study on our algorithm. Before going into this analysis, we briefly describe our demand generation method in Section 6.1.

6.1 Demand Forecasting Models

In order to generate multiple demand scenarios which follow the pattern of the available data to certain extent, we fit the data into several probability distribution (mass) functions (PDF) from the literature. Among these PDFs, we choose the one that gives the minimum of least squared errors and generated scenarios, each with a certain number of requests. Finally, we compute the probability of scenarios based on the probability of individual requests in each scenario.

Below we provide the details of our method to obtain probability distributions associated with the occurrence of customer requests based on the historical data on hand and generate scenarios by using these probability distributions. We initially retrieve the following parametric values from the Manhattan data to use in our demand generation method:

- $O_{u,t}$: The number of requests with origin $u \in V$ and starting time $t \in T$.
- $D_{v,t}$: The number of requests with destination $v \in V$ and starting time $t \in T$.
- $\bar{P}_{u,v,t}$: The number of requests with origin $u \in V$, destination $v \in V$, and starting time $t \in T$.
- d_{uv} : The traveling time from $u \in V$ to $v \in V$.

Then, we propose to obtain a probability distribution P such that $P_{u,v,t}$ is the probability that a request from node $u \in V$ to node $v \in V$ with starting time zone $t \in T$. In our experiments, we try fitting the probabilities in one of the following forms:

- (1) $P_{u,v,t} = \psi O_{u,t} D_{v,t} \exp(-\phi d_{uv}),$
- (2) $P_{u,v,t} = \psi O_{u,t} D_{v,t} \exp(\sigma \bar{P}_{u,v,t} - \phi d_{uv}),$

$$(3) P_{u,v,t} = \psi O_{u,t} D_{v,t} \exp(-(d_{uv} - \sigma)^2 / (2\xi^2)).$$

$$(4) P_{u,v,t} = \psi O_{u,t} D_{v,t} \exp(\phi \bar{P}_{u,v,t} - (d_{uv} - \sigma)^2 / (2\xi^2)).$$

Functions (1) and (2) are gravity models used in estimation of demand in communication and transportation systems [30] and Functions (3) and (4) are derived from the well-known Gaussian function.

We obtain the values of ψ, ϕ, σ, ξ parameters in the above functions by solving the following nonlinear least squares problem:

$$\min_{\psi, \phi, \sigma, \xi > 0}, \sum_{u,v,t} (\bar{P}_{u,v,t} - P_{u,v,t})^2 \quad (42)$$

$$\text{s.t.} \sum_{u,v,t} P_{u,v,t} = 1, \quad (43)$$

$$0 \leq P_{u,v,t} \leq 1, \quad \forall u, v \in V, t \in T. \quad (44)$$

To solve this problem, we initially focus on the minimization function (42) without restrictions (43) and (44). We solve this minimization problem by using the nonlinear least squares method of MATLAB. Once we obtain the optimal ψ, ϕ, σ, ξ values, we normalize the $P_{u,v,t}$ values to ensure constraints (43) and (44).

The computational experiments we conduct to compare the least squares values obtained from the three probability distributions reveal that the Function (4) provides the smallest one (slightly better than (3)). Therefore, we decide to use this function for our scenario generation steps.

After obtaining the probabilities for each $\{u, v, t\}$ pair in the data, where u is the origin node for some $k_1 \in K$, v is the destination node for some $k_2 \in K$, and t is the starting time for k_1 and k_2 , we generate scenarios by using the *datasample* function of MATLAB based on these probabilities. This function generates a random sample of elements with given probabilities and sample size. Let us denote $S = \{1, 2, \dots, |S|\}$ as the index set corresponding to the scenarios that we generate and K_s as the set of requests in scenario $s \in S$, that is, $|S|$ is the number of scenarios and $|K_s|$ is the number of requests in scenario $s \in S$. Then, one can form the scenarios in two ways: 1) Call *datasample* function to generate $|S| \times |K_s|$ requests and form $|S|$ scenarios of $|K_s|$ requests. 2) Call *datasample* for $|S|$ times and generate $|K_s|$ requests at step $s = 1, 2, \dots, |S|$. In our implementation, we choose the latter one and obtain an independent output file for each scenario. Here, we note that as the number of $\{u, v, t\}$ combinations which do not exist as requests in the original data might be excessive, we include only the $\{u, v\}$ pairs that appear at least twice in the data.

After we generate the scenarios, we also need to obtain the probability q_s associated with each scenario $s \in S$.

Let $q_k = P_{O_k, D_k, T_k}$ be the probability that request k will occur, then, $q_s^0 = \prod_{k \in K_s} q_k$ is the probability that requests in scenario K_s will occur together since the requests are independent from each other. In order to ensure that the probabilities of the scenarios sum up to 1, we normalize these values by setting $q_s = q_s^0 / (\sum_{s \in S} q_s^0)$.

6.2 Results

In addition to the metrics reported in earlier tables, we also report here the number of nodes explored in the branch-and-bound tree (nodes) and the number of stations opened in the best

solution ($|\bar{J}|$). Moreover, regarding our algorithms, we give the number of integral and fractional solutions separated during branch-and-cut procedures via columns ‘lazy’ and ‘user’, respectively.

In Tables 6 and 7, we compare model PF with algorithm BAC for *costfactor* values 10^5 and 10^6 , respectively, on instances with $|S| = 100$. In these tables, we see that BAC performs better in tackling instances with large β^w values and instances with $|K_s| \geq 4000$. However, the problems with $\beta^w = 15$ become intractable also by BAC when $|K_s| \geq 3000$. Among the problems that we can obtain feasible solutions, we need Phase II only in four instances for 10^5 and in two instances for 10^6 . The average gap from the best solution is 0.16% for 10^5 and 0.00% for 10^6 . The maximum gap is 1.96% for 10^5 and 0.03% for 10^6 . Considering the solution times and values, BAC performs better in 21 of 35 instances for 10^5 and in 24 of 35 instances for 10^6 .

In Tables 8 and 9, we compare model PF with algorithm BAC on instances with $|S| = 200$. Through these tables, we see that both PF and BAC needed more time on average to reach optimality. They hit the time limit for larger number of instances but the average and maximum gaps from the best solution value were much larger for PF compared to BAC. The better performance of BAC compared to PF becomes more visible in these tables. BAC is able to find very high quality solutions within one hour for most instances whereas PF struggles with finding a non-zero feasible solution in many of them. The difference is more obvious in Table 9 where the cost values are lower. Model PF hits the time limit in fifteen problems of Table 9 whereas BAC does so in only eight instances.

On average, BAC needs to visit more branch-and-bound nodes than PF and usually, it needs to separate a larger number of fractional solutions than integer solutions even though the fractional solutions are separated only at the root node.

When we look at the number of stations opened, we see that this number is larger when cost values are larger. An important observation here is that the increase in the number of stations opened when β^w is increased from 5 minutes to 6 minutes is more significant compared to other increments, for example, from 6 minutes to 7 minutes. This is worth to express as most recent works in the literature restrict the walking time by 5 minutes but we see through our experiments that a one-minute increase of this parameter might cover a much larger portion of the demand and yield a significant increase in the expected profit.

6.3 Stabilization

When we compare the performance of BAC on problems with $|S| = 200$ with those of $|S| = 100$, we observe that more iterations of separation are needed in Phase I of the algorithm and Phase II iterations are needed in a larger portion of problems, seven problems with the high cost values and eight problems with the low cost values. To overcome these difficulties, we introduce a stabilization procedure that enables addition of potentially stronger cuts for separation of infeasible solutions at individual iterations of our Benders algorithm. This method was originally proposed by Ben-Ameur and Neto [5] for cutting-plane and column generation algorithms and it was successfully adapted to several Benders decomposition implementations [28, 17, 40].

In the original form of our algorithm, for each optimal solution, say $\pi^m = (z^m, y^m, L^m)$, we solve the dual of subproblem *SP1* for π^m and add a cut associated with π^m to our master problem if there is a violation and move to subproblem *SP2* to solve its dual for π^m and repeat for each subproblem. Now, instead of moving to next subproblem immediately after finding a violation with the current one, we do the following: Let π^f be a global feasible solution to the original problem PF and $\varphi \in (0, 1)$. We find another point $\pi^1 = \varphi\pi^m + (1 - \varphi)\pi^f$ that is located on the line segment

Table 6: Comparison of PF with BAC for $|S| = 100$ and $costfactor = 10^5$

$ K_s $	b^w	PF			BAC		BAC Phase I			BAC Phase II			$ \bar{J} $
		g1(%)	time	nodes	g1(%)	time	lazy	user	nodes	lazy	user	nodes	
1000	5	0.00	4.12	0	0.00	36.44	24	44	17				49
1000	6	0.00	15.76	0	0.00	95.51	42	82	104				62
1000	7	0.00	46.71	4	0.00	80.21	15	63	3				67
1000	8	0.00	247.10	500	0.00	1518.94	148	87	27853				67
1000	9	0.00	656.09	40	0.00	2826.21	21	60	241	6	53	197	66
1000	10	0.00	723.22	83	0.00	498.66	19	53	328				66
1000	15	100.00	3629.30	0	0.00	2942.05	8	19	15	6	0	0	76
2000	5	0.00	11.72	0	0.00	6.96	4	1	0				60
2000	6	0.00	33.56	0	0.00	18.87	4	1	0				67
2000	7	0.00	97.92	0	0.00	25.42	2	0	0				72
2000	8	0.00	294.15	0	0.00	52.37	4	1	0				68
2000	9	0.00	850.56	0	0.00	120.18	4	1	0				70
2000	10	0.00	2987.40	1068	0.00	604.06	9	7	13	3	0	0	72
2000	15	NA	TL	NA	0.00	1763.26	1	6	0				81
3000	5	0.00	18.78	0	0.81	TL	130	353	58812	165	1040	8146	58
3000	6	0.00	86.56	0	0.40	TL	134	634	7781				66
3000	7	0.00	794.61	230	0.19	TL	51	493	0				67
3000	8	0.00	784.47	20	1.96	TL	16	84	0				69
3000	9	0.00	TL	511	1.90	TL	11	81	0				74
3000	10	0.00	TL	969	0.00	3481.80	21	84	985				76
3000	15	NA	TL	NA	NA	TL	1	4	0				NA
4000	5	0.00	27.40	0	0.00	11.66	4	1	0				64
4000	6	0.00	94.88	0	0.00	22.17	4	1	0				68
4000	7	0.00	1720.41	1111	0.00	184.13	21	6	64				74
4000	8	0.00	2042.30	525	0.00	262.33	10	6	47				74
4000	9	0.00	1655.19	967	0.00	385.09	4	7	5				75
4000	10	0.00	2329.47	99	0.00	636.52	6	7	3				76
4000	15	NA	TL	NA	NA	TL	0	6	0				NA
5000	5	0.00	68.72	36	0.00	62.78	23	38	75				66
5000	6	0.00	312.48	66	0.00	106.57	14	22	80				69
5000	7	0.00	1877.70	1177	0.00	359.93	27	20	354				75
5000	8	0.00	2357.71	2052	0.00	481.01	10	13	35				77
5000	9	0.00	TL	942	0.00	637.82	6	11	22				78
5000	10	0.00	TL	16	0.00	1087.72	13	8	60				79
5000	15	NA	TL	NA	NA	TL	NA	NA	NA				NA
Avg.:		3.23	1518.65		0.16	1460.36							
Max:		100.00	3777.61		1.96	7208.06							

Table 7: Comparison of PF with BAC for $|S| = 100$ and $costfactor = 10^6$

$ K_s $	b^w	PF			BAC		BAC Phase I			BAC Phase II			$ \bar{J} $
		g1(%)	time	nodes	g1(%)	time	lazy	user	nodes	lazy	user	nodes	
1000	5	0.00	5.59	0	0.00	9.70	19	22	0				68
1000	6	0.00	14.35	0	0.00	14.74	19	27	0				74
1000	7	0.00	42.35	2	0.00	35.82	34	44	44				76
1000	8	0.00	116.04	11	0.00	49.91	9	23	10				77
1000	9	0.00	543.88	14	0.00	90.99	10	41	24				73
1000	10	0.00	521.08	0	0.00	184.31	15	24	10				73
1000	15	100.00	TL	0	0.00	1691.83	6	17	0	6	0	0	80
2000	5	0.00	15.77	0	0.00	7.06	3	0	0				71
2000	6	0.00	41.48	0	0.00	14.47	4	1	0				75
2000	7	0.00	139.59	0	0.00	24.15	6	2	0				78
2000	8	0.00	1096.80	1557	0.00	63.43	10	5	10				76
2000	9	0.00	TL	75	0.00	134.13	6	7	9				76
2000	10	0.00	TL	858	0.00	284.24	8	6	56				77
2000	15	NA	TL	NA	0.00	2633.47	3	10	0				NA
3000	5	0.00	23.38	0	0.00	962.87	60	1004	179				64
3000	6	0.00	96.77	0	0.00	1371.60	50	765	331				69
3000	7	0.00	248.83	0	0.00	1876.95	50	695	1084				75
3000	8	0.00	701.07	0	0.03	TL	42	368	0				76
3000	9	0.00	2693.32	0	0.01	TL	17	219	0				77
3000	10	0.00	2726.03	24	0.00	TL	21	145	62				80
3000	15	NA	TL	NA	NA	TL	1	4	0				NA
4000	5	0.00	32.39	0	0.00	9.56	3	0	0				71
4000	6	0.00	115.26	0	0.00	21.85	3	1	0				76
4000	7	0.00	1746.54	1017	0.00	70.61	5	7	18				77
4000	8	0.00	1035.95	0	0.00	177.57	8	8	0				77
4000	9	0.00	1655.19	0	0.00	218.38	3	2	0				79
4000	10	0.00	1824.23	0	0.00	300.02	1	2	0				79
4000	15	NA	TL	NA	NA	TL	0	5	0				NA
5000	5	0.00	53.21	0	0.00	53.51	22	82	50				75
5000	6	0.00	423.10	90	0.00	82.80	21	28	24				76
5000	7	0.00	1223.83	17	0.00	141.18	14	16	36				78
5000	8	0.00	2706.08	546	0.00	201.82	2	8	55				80
5000	9	0.00	3337.07	28	0.00	492.26	11	7	0	5	0	0	80
5000	10	0.00	3024.76	0	0.00	437.84	2	1	0				81
5000	15	NA	TL	NA	NA	TL	0	0	0				NA
Avg.:		3.23	1485.64		0.00	959.53							
Max:		100.00	3888.52		0.03	3699.71							

Table 8: Comparison of PF with BAC for $|S| = 200$ and $costfactor = 10^5$

$ K_s $	b^w	PF			BAC		BAC Phase I			BAC Phase II			$ \bar{J} $
		g1(%)	time	nodes	g1(%)	time	lazy	user	nodes	lazy	user	nodes	
1000	5	0.00	14.46	0	0.00	52.38	31	35	195				48
1000	6	0.00	62.97	0	0.00	268.96	31	70	206	9	0	0	62
1000	7	0.00	265.89	3	0.00	155.95	25	46	11	10	0	0	68
1000	8	0.00	2550.04	1079	0.00	2606.79	98	76	21161	11	0	0	66
1000	9	0.32	TL	3	0.00	969.98	46	56	1194				66
1000	10	100.00	TL	0	0.00	866.58	18	45	581				66
1000	15	NA	TL	0	NA	TL	0	12	0				NA
2000	5	0.00	31.43	0	0.00	1280.23	51	417	489				53
2000	6	0.00	147.28	0	1.52	TL	58	569	0				62
2000	7	0.00	TL	173	0.44	TL	133	974	1103				66
2000	8	0.00	2088.65	0	0.21	TL	43	589	10				64
2000	9	100.00	TL	0	0.00	TL	21	203	0				71
2000	10	100.00	TL	0	0.00	TL	13	53	0				70
2000	15	NA	TL	0	NA	TL	1	3	0				NA
3000	5	0.00	63.67	0	5.26	TL	170	368	23932	61	1186	0	58
3000	6	0.00	485.12	0	0.81	TL	42	439	0				66
3000	7	0.00	TL	144	2.75	TL	30	190	0				67
3000	8	100.00	TL	0	0.00	TL	14	63	0				71
3000	9	100.00	TL	0	0.00	TL	12	51	0				72
3000	10	100.00	TL	0	0.00	TL	7	36	0				76
4000	5	0.00	160.84	0	0.00	27.79	5	2	0				64
4000	6	0.00	575.42	0	0.00	45.32	3	3	0				68
4000	7	100.00	TL	0	0.00	264.77	17	8	47				74
4000	8	100.00	TL	0	0.00	335.08	5	6	86				74
4000	9	100.00	TL	0	0.00	3494.47	6	5	7	4	7	51	75
4000	10	NA	TL	0	0.00	867.07	3	6	71				76
5000	5	0.00	673.09	25	0.00	69.24	14	30	92				66
5000	6	0.00	1903.53	0	0.00	264.90	14	25	50				69
5000	7	100.00	TL	0	0.00	463.73	17	21	151				75
5000	8	100.00	TL	0	0.00	TL	0	11	37	10	12	9	75
5000	9	100.00	TL	0	0.00	813.33	6	9	35				77
5000	10	NA	TL	0	0.00	TL	0	6	11	8	9	0	78
Avg.:		42.87	2438.34		0.37	2236.12							
Max:		100.00	3739.77		5.26	7207.33							

Table 9: Comparison of PF with BAC for $|S| = 200$ and $costfactor = 10^6$

$ K_s $	b^w	PF			BAC		BAC Phase I			BAC Phase II			$ \bar{J} $
		g1(%)	time	nodes	g1(%)	time	lazy	user	nodes	lazy	user	nodes	
1000	5	0.00	12.64	0	0.00	9.75	18	14	0				68
1000	6	0.00	30.69	0	0.00	61.61	36	42	87	8	0	0	75
1000	7	0.00	79.90	0	0.00	66.85	28	28	13	9	0	0	77
1000	8	0.00	354.34	0	0.00	209.72	17	24	56	10	0	0	77
1000	9	0.01	TL	0	0.00	TL	26	51	91200	18	29	7044	76
1000	10	0.00	1782.33	0	0.00	484.39	7	25	45				74
1000	15	NA	TL	0	0.00	TL	9	13	158	15	0	0	80
2000	5	0.00	24.26	0	0.00	804.65	56	641	83				56
2000	6	0.00	89.46	0	0.00	2546.39	57	1520	416				64
2000	7	0.00	1051.88	0	0.00	2201.05	47	861	243				68
2000	8	0.00	TL	147	0.00	3081.48	45	626	333				68
2000	9	100.00	TL	0	0.00	TL	18	101	0				73
2000	10	100.00	TL	0	0.00	TL	25	109	0				76
2000	15	NA	TL	0	NA	TL	1	4	0				NA
3000	5	0.00	134.59	0	0.00	1719.39	50	890	84				64
3000	6	0.00	388.46	0	0.00	2744.97	43	856	314				69
3000	7	0.00	1150.57	7	0.00	3051.89	45	553	2983				74
3000	8	0.00	1877.87	0	0.05	TL	40	302	0				76
3000	9	100.00	TL	0	0.00	TL	22	121	0				79
3000	10	100.00	TL	0	0.00	TL	13	44	0				83
4000	5	0.00	177.75	0	0.00	74.96	32	0	184				71
4000	6	0.00	747.65	0	0.00	66.64	5	9	0				76
4000	7	0.00	TL	361	0.00	138.63	7	8	25				77
4000	8	100.00	TL	0	0.00	306.16	5	9	38				78
4000	9	100.00	TL	0	0.00	343.61	3	3	0				80
4000	10	100.00	TL	0	0.00	500.06	2	2	0				79
5000	5	0.00	429.57	153	0.00	154.87	32	64	22				77
5000	6	0.00	1099.12	0	0.00	730.80	14	29	40	13	34	123	76
5000	7	0.00	2349.23	180	0.00	488.06	19	13	117	10	0	0	79
5000	8	0.01	TL	0	0.00	756.91	8	8	36				80
5000	9	NA	TL	0	0.00	922.29	3	3	0	10	0	0	80
5000	10	NA	TL	0	0.00	1404.04	3	3	0				81
Avg.:		25.00	2081.68		0.00	1758.44							
Max:		100.00	3768.34		0.05	7226.77							

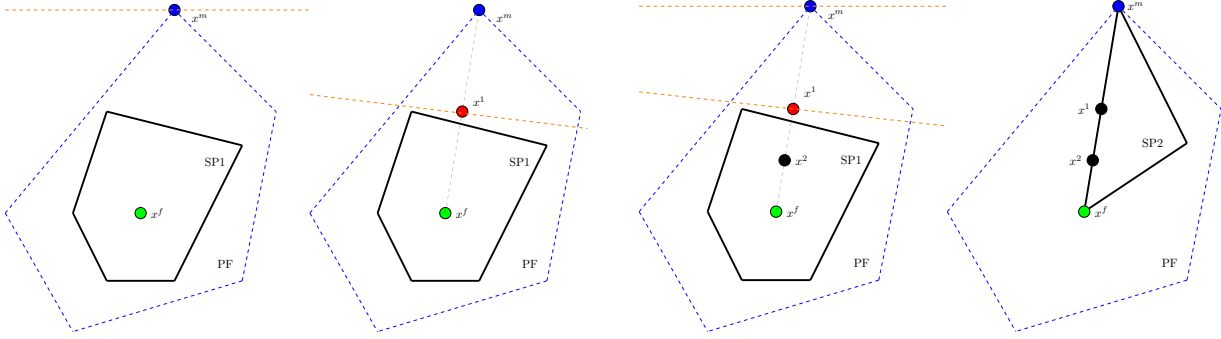


Figure 2: A sample stabilization iteration

Table 10: Summary of stabilization results results for $|S| = 100$

		BAC					BAC + Stabilization				
		g2(%)	time	lazy	user	nodes	g2(%)	time	lazy	user	nodes
High	Avg.:	0.16	1460.36	23.85	67.76	2849.91	0.10	1304.90	16.09	39.94	2652.23
f,g	Max:	1.96	7208.06	148	634	58812	1.33	4172.63	102	307	59000
Low	Avg.:	0.00	959.53	13.94	102.74	57.20	0.00	894.29	10.51	44.66	89.54
f,g	Max:	0.03	3699.71	60	1004	1084	0.04	3983.73	35	369	2180

between π^m and π^f , and check if the dual of the current subproblem is violated by this point π^1 . If yes, then we obtain a potentially stronger cut that separates both π^m and π^1 , so, we add the new cut to our master problem. Then, we move to another point $\pi^2 = \varphi\pi^1 + (1 - \varphi)\pi^f$ and check if there is a violation. If there is no violation, we move to another point $\pi^3 = \varphi\pi^1 + (1 - \varphi)\pi^2$ and we keep moving on the line segment between the last feasible point and the last separation point found for certain number of iterations. In our implementation, we move for only two iterations between π^m and π^f for each subproblem and then, we move to the next subproblem to repeat the same stabilization steps. We further choose $\varphi = 0.5$ in our experiments. We provide an illustration of the stabilization procedure on a sample polyhedra in Figure 2).

A summary of the results comparing the performance BAC with and without stabilization is presented in Tables 10-14. Detailed results can be found in Tables 16-25 in Appendix A. Even though there is not an absolute winner among the two methods, the stabilized version performs slightly better on average solving time. It is also better on the average gap for instances with $|S| = 100, 300, 400$ whereas the non-stabilized version is better for those with $|S| = 200, 500$. Stabilization decreases the average number of *LazyConstraintCallback* iterations in all tables for low cost values and it decreases the average number of *UserCutCallback* iterations by around 50% in each table for both high and low cost cases. The impact of stabilisation on the number of nodes explored does not follow a clear pattern and it is difficult to draw any conclusion on this aspect.

We provide a summary of our observations regarding the effects of changes in parameters *costfactor*, β^w , $|S|$, $|K_s|$ on problem outputs in Table 15.

When we look at the detailed comparison of the stabilized and non-stabilized BAC in Tables 16-25, we see that the stabilization is helpful in solving more challenging problems that the non-stabilized version could not handle easily but it does provide much improvement on the instances

Table 11: Summary of stabilization results results for $|S| = 200$

		BAC					BAC + Stabilization				
		g2(%)	time	lazy	user	nodes	g2(%)	time	lazy	user	nodes
High	Avg.:	0.37	2236.12	29.19	138.34	1545.91	0.73	1841.16	30.03	60.22	1880.88
f,g	Max:	5.26	7207.33	170	974	23932	10.78	3901.83	327	237	28470
Low	Avg.:	0.00	1758.44	23.00	218.00	3014.91	0.09	1454.17	15.13	86.09	2595.78
f,g	Max:	0.05	7226.77	57	1520	91200	1.27	7221.45	35	455	79000

Table 12: Summary of stabilization results results for $|S| = 300$

		BAC					BAC + Stabilization				
		g2(%)	time	lazy	user	nodes	g2(%)	time	lazy	user	nodes
High	Avg.:	2.30	3260.01	28.87	164.63	4021.37	2.05	2930.75	25.87	76.37	1199.23
f,g	Max:	11.81	5698.23	144	978	105803	8.63	3962.23	144	208	20959
Low	Avg.:	0.50	2758.24	29.87	218.70	1752.00	0.26	2274.90	21.93	88.10	4568.00
f,g	Max:	3.09	5325.02	88	1121	35490	2.13	4864.87	53	458	88798

Table 13: Summary of stabilization results results for $|S| = 400$

		BAC					BAC + Stabilization				
		g2(%)	time	lazy	user	nodes	g2(%)	time	lazy	user	nodes
High	Avg.:	3.59	3238.77	25.17	146.03	702.21	2.56	3185.44	31.87	80.13	1479.27
f,g	Max:	13.11	6977.37	67	829	12759	12.22	7244.98	188	297	17961
Low	Avg.:	0.94	3116.34	23.53	227.79	1402.80	0.47	2794.51	17.37	95.83	998.47
f,g	Max:	5.97	7238.90	62	1309	35813	1.86	5997.19	39	448	14444

Table 14: Summary of stabilization results results for $|S| = 500$

		BAC					BAC + Stabilization				
		g2(%)	time	lazy	user	nodes	g2(%)	time	lazy	user	nodes
High	Avg.:	3.18	3287.89	24.77	114.73	4416.93	3.33	3289.77	28.17	67.23	4432.60
f,g	Max:	16.20	7140.04	107	587	105545	15.04	7056.08	187	237	80182
Low	Avg.:	0.71	3161.01	23.43	212.63	1111.83	0.98	2861.56	17.67	100.37	209.50
f,g	Max:	4.27	7271.27	72	1341	19500	11.04	5667.87	55	481	2175

Table 15: Correlation between the changes on problem parameters and outputs

	$costfactor \uparrow$	$\beta^w \uparrow$	$ S \uparrow$	$ K_s \uparrow$
Obj	\uparrow	\uparrow	\downarrow	\uparrow
$ \mathcal{J} $	\uparrow	mostly \uparrow		no pattern
PP time	NA	\uparrow	no pattern	mostly \uparrow
PF time	decreases in average	mostly \uparrow	\uparrow	\uparrow
BAC time	decreases in average	mostly \uparrow	\uparrow	no pattern
BAC-S time	decreases in average	mostly \uparrow	\uparrow	no pattern
BAC Phase II	no pattern	no pattern	no pattern	no pattern
BAC-S Phase II	no pattern	no pattern	no pattern	no pattern
Avg. gap %	\downarrow	no pattern	mostly \uparrow	no pattern

that can already be tackled by the non-stabilized version. More specifically, the stabilized BAC provides a better solution value for 97 instances among 314 whereas non-stabilized BAC finds a better solution in 57 instances (for two of them, PF provides a better solution). In 12 of the remaining problems, none of our methods can find a feasible solution within one hour ($\beta^w = 15$ for 8 of these problems). For the remaining problems, both algorithms give solutions of the optimal objective value and they are proven to be optimal in both versions for all except one, for which the optimality can be proven only by the stabilized version.

7 Conclusion

In this paper, we introduce a strategic decision-making problem arising in the design of a one-way station based electric car sharing system that operates under demand uncertainty. The strategic decisions include the number and location of the recharging stations which need to be selected meticulously as opening of these kind of facilities requires large amount of investment in money and time, and it is undesirable to replace them frequently. Our model takes into account the potential customer demand based on a relevant historical data and the expected revenue of the demand portion that could be served in a feasible way with the opened stations. As we aim to solve this problem optimally, we develop a mathematical formulation and a Benders decomposition algorithm that we further enrich with a stabilization procedure. We observe that our algorithm has the potential to solve real size problems to optimality, if not we can obtain very high quality solutions within very short amount of time.

We observe through our experimental study that the problem is harder to solve when the costs (station opening and car possession) are higher and as expected, less number of stations are operated and less profit is made. Moreover, even a small portion of increase in the maximum walking distance for customers increases the profit significantly. Therefore, the accessibility measures should be selected very carefully when operating the system. The accessibility measures can also be used as a parameter in formation of pricing strategies. Another parameter that needs to be taken into account in design or operation of electric car sharing systems is the number of different demand scenarios. As we observe in this study, considering too few scenarios in taking decisions might cause an overestimation of the actual profit or misplacement of stations. The length of the planning horizon is also very effective in the profit of the system constructed. We observe that the profit increases as the number of requests per scenario increases, which can be interpreted as a longer planning period. A future study could focus on deciding the optimal length of the planning horizon for strategic decisions such as location of stations or recharging units.

The main focus of this work is on exact methods but the methods we propose can be easily modified or combined with practical procedures to obtain near optimal solutions.

As mentioned in Section 3, we can easily adapt our methods to more complicated variations of the problem we proposed here. These variations include considering time-dependent travelling time and energy consumption as well as visiting intermediate stations. An interesting and more challenging extension would be to relax full battery restriction for availability and develop methods that can ensure sufficient battery at the beginning of each trip. In this case, an additional index for each operating car would be needed and the mathematical models might have problems in tackling large scale problems. On the other hand, some meta-heuristic methods might be developed to solve these problems heuristically.

Acknowledgements

This research is conducted under e4-share (Models for Ecological, Economical, Efficient, Electric Car-Sharing) project funded by FFG, INNOVIRIS and MIUR via JPI Urban Europe. This research is supported also by the Interuniversity Attraction Poles P7/36 COMEX initiated by the Belgian Science Policy Office.

References

References

- [1] Arslan, O., Karaşan, O. E., 2016. A Benders decomposition approach for the charging station location problem with plug-in hybrid electric vehicles. *Transportation Research Part B: Methodological* 93, 670–695.
- [2] Asamer, J., Reinthaler, M., Ruthmair, M., Straub, M., Puchinger, J., 2016. Optimizing charging station locations for urban taxi providers. *Transportation Research Part A: Policy and Practice* 85, 233–246.
- [3] Baouche, F., Billot, R., Trigui, R., El Faouzi, N.-E., 2014. Efficient allocation of electric vehicles charging stations: Optimization model and application to a dense urban network. *IEEE Intelligent Transportation Systems Magazine* 6 (3), 33–43.
- [4] Barth, M., Todd, M., Xue, L., 2004. User-based vehicle relocation techniques for multiple-station shared-use vehicle systems. In: *Transportation Research Board, 80th Annual Meeting*. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.360.8614>
- [5] Ben-Ameur, W., Neto, J., 2007. Acceleration of cutting-plane and column generation algorithms: Applications to network design. *Networks* 49 (1), 3–17.
- [6] Benders, J. F., 1962. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik* 4 (1), 238–252.
- [7] Biesinger, B., Hu, B., Stubenschrott, M., Ritzinger, U., Prandtstetter, M., 2017. Optimizing charging station locations for electric car-sharing systems. *Evolutionary Computation in Combinatorial Optimization*, 157–172.
- [8] Birge, J. R., Louveaux, F., 2011. Two-stage recourse problems. In: *Introduction to Stochastic Programming*. Springer, pp. 181–263.
- [9] Botton, Q., Fortz, B., Gouveia, L., Poss, M., 2013. Benders decomposition for the hop-constrained survivable network design problem. *INFORMS Journal on Computing* 25 (1), 13–26.
- [10] Boyacı, B., Zografos, K. G., Geroliminis, N., Feb. 2015. An optimization framework for the development of efficient one-way car-sharing systems. *European Journal of Operational Research* 240 (3), 718–733.

- [11] Boyacı, B., Zografos, K. G., Geroliminis, N., 2017. An integrated optimization-simulation framework for vehicle and personnel relocations of electric carsharing systems with reservations. *Transportation Research Part B: Methodological* 95, 214–237.
- [12] Brandstätter, G., Gambella, C., Leitner, M., Malaguti, E., Masini, F., Puchinger, J., Ruthmair, M., Vigo, D., 2016. Overview of optimization problems in electric car-sharing system design and management. In: *Dynamic Perspectives on Managerial Decision Making*. Springer, pp. 441–471.
- [13] Brandstätter, G., Kahr, M., Leitner, M., 2017. Determining optimal locations for charging stations of electric car-sharing systems under stochastic demand. Tech. rep., Department of Statistics and Operations Research, University of Vienna, Vienna, Austria.
- [14] Brandstätter, G., Leitner, M., Ljubić, I., 2016. Location of charging stations in electric car sharing systems. Tech. rep., Department of Statistics and Operations Research, University of Vienna, Vienna, Austria.
- [15] Bruglieri, M., Colorni, A., Luè, A., Feb. 2014. The vehicle relocation problem for the one-way electric vehicle sharing: An application to the milan case. *Procedia - Social and Behavioral Sciences* 111, 18–27.
- [16] Çalık, H., Fortz, B., 2017. Location of stations in a one-way electric car sharing system. In: *Computers and Communications (ISCC), 2017 IEEE Symposium on*. IEEE, pp. 134–139.
- [17] Calik, H., Leitner, M., Luipersbeck, M., 2017. A Benders decomposition based framework for solving cable trench problems. *Computers & Operations Research* 81, 128–140.
- [18] Carrier, A., Munier-Kordon, A., Klaudel, W., 2015. Mathematical model for the study of relocation strategies in one-way carsharing systems. *Transportation Research Procedia* 10, 374–383.
- [19] Cavadas, J., Correia, G. H., Gouveia, J., Mar. 2015. A mip model for locating slow-charging stations for electric vehicles in urban areas accounting for driver tours. *Transportation Research Part E: Logistics and Transportation Review* 75, 188–201.
- [20] Chen, T. D., Kockelman, K. M., Khan, M., et al., 2013. The electric vehicle charging station location problem: a parking-based assignment method for seattle. In: *92nd Annual Meeting of the Transportation Research Board*. Washington DC, USA.
URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.300.2883>
- [21] Clemente, M., Fanti, M. P., Mangini, A. M., Ukovich, W., 2013. The vehicle relocation problem in car sharing systems: Modeling and simulation in a petri net framework. *Lecture Notes in Computer Science* 7927, 250–269.
- [22] Codato, G., Fischetti, M., 2006. Combinatorial Benders’ cuts for mixed-integer linear programming. *Operations Research* 54 (4), 756–766.
- [23] Correia, G. H., Antunes, A. P., Jan. 2012. Optimization approach to depot location and trip selection in one-way carsharing systems. *Transportation Research Part E: Logistics and Transportation Review* 48 (1), 233–247.

- [24] Correia, G. H., Jorge, D. R., Antunes, D. M., Jun. 2014. The added value of accounting for users' flexibility and information on the potential of a station-based one-way car-sharing system: An application in Lisbon, Portugal. *Journal of Intelligent Transportation Systems* 18 (3), 299–308.
- [25] Erdoğan, S., Miller-Hooks, E., Jan. 2012. A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review* 48 (1), 100–114.
- [26] Fassi, A. E., Awasthi, A., Viviani, M., Jun. 2012. Evaluation of carsharing network's growth strategies through discrete event simulation. *Expert Systems with Applications* 39 (8), 6692–6705.
- [27] Felipe, Á., Ortuño, M. T., Righini, G., Tirado, G., 2014. A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Transportation Research Part E: Logistics and Transportation Review* 71 (0), 111–128.
- [28] Fischetti, M., Ljubić, I., Sinnl, M., 2016. Redesigning Benders decomposition for large-scale facility location. *Management Science* 63 (7), 2146–2162.
- [29] Fortz, B., Poss, M., 2009. An improved Benders decomposition applied to a multi-layer network design problem. *Operations Research Letters* 37 (5), 359–364.
- [30] Fortz, B., Thorup, M., 2004. Increasing internet capacity using local search. *Computational Optimization and Applications* 29 (1), 13–48.
- [31] Frade, I., Ribeiro, A., Gonçalves, G., Antunes, A. P., Dec. 2011. Optimal location of charging stations for electric vehicles in a neighborhood in lisbon, Portugal. *Transportation Research Record: Journal of the Transportation Research Board* 2252, 91–98.
- [32] Gambella, C., Malaguti, E., Masini, F., Vigo, D., 2017. Optimizing relocation operations in electric car-sharing. Tech. rep., OR-17-3, DEI-University of Bologna.
- [33] Ge, S., Feng, L., Liu, H., Sep. 2011. The planning of electric vehicle charging station based on grid partition method. In: 2011 International Conference on Electrical and Control Engineering (ICECE). IEEE.
- [34] Herrmann, S., Schulte, F., Voß, S., 2014. Increasing acceptance of free-floating car sharing systems using smart relocation strategies: a survey based study of car2go hamburg. In: *Computational Logistics*. Springer, pp. 151–162.
- [35] Hess, A., Malandrino, F., Reinhardt, M. B., Casetti, C., Hummel, K. A., Barceló-Ordinas, J. M., 2012. Optimal deployment of charging stations for electric vehicular networks. In: *Proceedings of the First Workshop on Urban Networking. UrbaNe '12*. ACM, New York, NY, USA, pp. 1–6.
- [36] Hooker, J. N., Ottosson, G., 2003. Logic-based Benders decomposition. *Mathematical Programming* 96 (1), 33–60.
- [37] Jorge, D., Correia, G. H., Barnhart, C., Aug. 2014. Comparing optimal relocation operations with simulated relocation policies in one-way carsharing systems. *IEEE Transactions on Intelligent Transportation Systems* 15 (4), 1667–1675.

- [38] Kek, A. G., Cheu, R. L., Meng, Q., Fung, C. H., Jan. 2009. A decision support system for vehicle relocation operations in carsharing systems. *Transportation Research Part E: Logistics and Transportation Review* 45 (1), 149–158.
- [39] Li, X., Ma, J., Cui, J., Ghiasi, A., Zhou, F., 2016. Design framework of large-scale one-way electric vehicle sharing systems: A continuum approximation model. *Transportation Research Part B: Methodological* 88, 21–45.
- [40] Álvarez Miranda, E., Ljubić, I., Luipersbeck, M., Sinnl, M., 2017. Solving minimum-cost shared arborescence problems. *European Journal of Operational Research* 258 (3), 887 – 901.
- [41] Pelletier, S., Jabali, O., Laporte, G., 2016. 50th anniversary invited article—goods distribution with electric vehicles: Review and research perspectives. *Transportation Science* 50 (1), 3–22.
- [42] Schneider, M., Stenger, A., Goeke, D., Nov. 2014. The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science* 48 (4), 500–520.
- [43] Schulte, F., Voß, S., 2015. Decision support for environmental-friendly vehicle relocations in free-floating car sharing systems: The case of car2go. *Procedia CIRP* 30, 275–280.
- [44] Wang, H., Huang, Q., Zhang, C., Xia, A., Dec. 2010. A novel approach for the layout of electric vehicle charging station. In: *The 2010 International Conference on Apperceiving Computing and Intelligence Analysis Proceeding*. IEEE.
- [45] Wang, Y.-W., Lin, C.-C., Nov. 2013. Locating multiple types of recharging stations for battery-powered electric vehicle transport. *Transportation Research Part E: Logistics and Transportation Review* 58, 76–87.

A Detailed computational results

Table 16: Stabilization results for $|S| = 100$ and $cost\ factor = 10^5$

$ K_s $	b^w	PF			BAC			BAC + Stabilization		
		Obj	g2(%)	time	Obj	g2(%)	time	Obj	g2(%)	time
1000	5	7218.65	0.00	4.12	7218.65	0.00	36.44	7218.65	0.00	26.98
1000	6	15022.45	0.00	15.76	15022.45	0.00	95.51	15022.45	0.00	71.62
1000	7	27085.46	0.00	46.71	27085.46	0.00	80.21	27085.46	0.00	75.22
1000	8	36980.87	0.00	247.10	36980.87	0.00	1518.94	36980.87	0.00	815.25
1000	9	44723.11	0.00	656.09	44723.11	0.00	2826.21	44723.11	0.00	442.56
1000	10	52158.64	0.00	723.22	52158.64	0.00	498.66	52158.64	0.00	512.92
1000	15	0.00	NA	3629.30	89847.52	0.00	2942.05	89847.52	0.00	2988.01
2000	5	20839.93	0.00	11.72	20839.93	0.00	6.96	20839.93	0.00	9.41
2000	6	31931.30	0.00	33.56	31931.30	0.00	18.87	31931.30	0.00	16.34
2000	7	44907.15	0.00	97.92	44907.15	0.00	25.42	44907.15	0.00	25.81
2000	8	53593.97	0.00	294.15	53593.97	0.00	52.37	53593.97	0.00	49.67
2000	9	61004.03	0.00	850.56	61004.03	0.00	120.18	61004.03	0.00	127.87
2000	10	68173.08	0.00	2987.40	68173.08	0.00	604.06	68173.08	0.00	529.17
2000	15	NA	NA	3668.57	120714.21	0.00	1763.26	120714.21	0.00	1766.10
3000	5	24781.00	0.00	18.78	24581.00	0.81	7208.06	24781.00	0.57	3606.36
3000	6	42204.00	0.00	86.56	42036.00	0.40	3609.21	41845.00	1.33	3608.93
3000	7	52964.00	0.00	794.61	52862.00	0.19	3614.63	52964.00	0.00	3611.91
3000	8	62933.00	0.00	784.47	61701.00	1.96	3625.94	62613.00	0.73	3617.60
3000	9	72476.00	0.04	3615.67	71100.00	1.90	3634.37	72397.00	0.26	3636.24
3000	10	78358.00	0.00	3619.10	78358.00	0.00	3481.80	78013.00	0.46	3652.60
3000	15	NA	NA	3702.94	NA	NA	3658.41	NA	NA	3728.56
4000	5	31980.00	0.00	27.40	31980.00	0.00	11.66	31980.00	0.00	15.32
4000	6	50444.00	0.00	94.88	50444.00	0.00	22.17	50444.00	0.00	27.39
4000	7	63485.00	0.00	1720.41	63485.00	0.00	184.13	63485.00	0.00	202.86
4000	8	70577.00	0.00	2042.30	70577.00	0.00	262.33	70577.00	0.00	287.03
4000	9	76431.00	0.01	1655.19	76431.00	0.00	385.09	76431.00	0.00	331.57
4000	10	79708.00	0.00	2329.47	79708.00	0.00	636.52	79708.00	0.00	704.05
4000	15	NA	NA	3747.21	NA	NA	3675.97	NA	NA	3782.78
5000	5	38169.82	0.00	68.72	38169.82	0.00	62.78	38169.82	0.00	43.27
5000	6	53488.83	0.00	312.48	53488.83	0.00	106.57	53488.83	0.00	82.66
5000	7	65554.10	0.00	1877.70	65554.10	0.00	359.93	65554.10	0.00	253.17
5000	8	72554.02	0.00	2357.71	72554.02	0.00	481.01	72554.02	0.00	272.96
5000	9	79005.40	0.04	3625.34	79005.40	0.00	637.82	79005.40	0.00	576.94
5000	10	84998.90	0.00	3628.12	84999.05	0.00	1087.72	84998.95	0.00	4319.78
5000	15	NA	NA	3777.61	NA	NA	3777.26	NA	NA	3683.00
Avg.:			0.00	1518.65		0.16	1460.36		0.10	1357.20
Max:			0.04	3777.61		1.96	7208.06		1.33	4319.78

Table 17: Stabilization results for $|S| = 100$ and $cost\ factor = 10^6$

$ K_s $	b^w	PF			BAC			BAC + Stabilization		
		Obj	g2(%)	time	Obj	g2(%)	time	Obj	g2(%)	time
1000	5	16975.83	0.00	5.59	16975.83	0.00	9.70	16975.83	0.00	10.49
1000	6	27675.73	0.00	14.35	27675.73	0.00	14.74	27675.73	0.00	25.01
1000	7	41030.45	0.00	42.35	41030.45	0.00	35.82	41030.45	0.00	50.99
1000	8	51380.12	0.00	116.04	51380.12	0.00	49.91	51380.12	0.00	57.08
1000	9	58838.8	0.00	543.88	58838.80	0.00	90.99	58838.80	0.00	119.79
1000	10	66402.34	0.00	521.08	66402.34	0.00	184.31	66402.34	0.00	194.35
1000	15	0.00	100.00	3633.69	106736.52	0.00	1691.83	106736.52	0.00	1485.02
2000	5	33349.13	0.00	15.77	33349.13	0.00	7.06	33349.13	0.00	7.10
2000	6	45904.90	0.00	41.48	45904.90	0.00	14.47	45904.90	0.00	16.02
2000	7	59399.62	0.00	139.59	59399.62	0.00	24.15	59399.62	0.00	29.10
2000	8	67903.56	0.00	1096.80	67903.56	0.00	63.43	67903.56	0.00	94.76
2000	9	75843.23	0.00	3613.47	75843.23	0.00	134.13	75843.23	0.00	160.78
2000	10	83820.27	0.00	3624.72	83820.27	0.00	284.24	83820.27	0.00	294.32
2000	15	NA	NA	3724.18	139677.45	0.00	2633.47	139677.45	0.00	2850.63
3000	5	36003.00	0.00	23.38	36003.00	0.00	962.87	36003.00	0.00	300.51
3000	6	55466.00	0.00	96.77	55466.00	0.00	1371.60	55466.00	0.00	781.81
3000	7	67527.00	0.00	248.83	67527.00	0.00	1876.95	67527.00	0.00	959.29
3000	8	78209.00	0.00	701.07	78209.00	0.03	3623.15	78209.00	0.00	2765.50
3000	9	88446.00	0.00	2693.32	88446.00	0.01	3636.20	88421.00	0.04	3633.88
3000	10	94909.00	0.00	2726.03	94909.00	0.00	3653.13	94904.00	0.01	3650.17
3000	15	NA	NA	3708.71	NA	NA	3635.12	NA	NA	3983.73
4000	5	45137.00	0.00	32.39	45137.00	0.00	9.56	45137.00	0.00	10.14
4000	6	65252.00	0.00	115.26	65252.00	0.00	21.85	65252.00	0.00	27.41
4000	7	79753.00	0.00	1746.54	79753.00	0.00	70.61	79753.00	0.00	65.25
4000	8	86868.00	0.00	1035.95	86868.00	0.00	177.57	86868.00	0.00	164.96
4000	9	93277.00	0.00	1655.19	93277.00	0.00	218.38	93277.00	0.00	229.87
4000	10	96947.00	0.00	1824.23	96947.00	0.00	300.02	96947.00	0.00	292.70
4000	15	NA	NA	3888.52	NA	NA	3679.14	NA	NA	3679.43
5000	5	51614.58	0.00	53.21	51614.58	0.00	53.51	51614.58	0.00	31.39
5000	6	68362.75	0.00	423.10	68362.75	0.00	82.80	68362.75	0.00	83.52
5000	7	81695.37	0.00	1223.83	81695.37	0.00	141.18	81695.37	0.00	198.08
5000	8	89786.21	0.00	2706.08	89786.21	0.00	201.82	89786.21	0.00	236.76
5000	9	96645.70	0.00	3337.07	96645.94	0.00	492.26	96645.94	0.00	544.23
5000	10	103252.48	0.00	3024.76	103252.48	0.00	437.84	103252.48	0.00	569.67
5000	15	NA	NA	3600.00	NA	NA	3699.71	NA	NA	3696.44
Avg.:			3.23	1485.64		0.00	959.53		0.00	894.29
Max:			100.00	3888.52		0.03	3699.71		0.04	3983.73

Table 18: Stabilization results for $|S| = 200$ and $costfactor = 10^5$

$ K_s $	b^w	PF			BAC			BAC + Stabilization		
		Obj	g2(%)	time	Obj	g2(%)	time	Obj	g2(%)	time
1000	5	6832.88	0.00	14.46	6832.88	0.00	52.38	6832.88	0.00	51.49
1000	6	14887.28	0.00	62.97	14887.28	0.00	268.96	14887.28	0.00	249.29
1000	7	26994.52	0.00	265.89	26994.52	0.00	155.95	26994.52	0.00	138.42
1000	8	36766.38	0.00	2550.04	36766.38	0.00	2606.79	36766.38	0.00	1158.99
1000	9	44496.02	0.32	3612.79	44640.44	0.00	969.98	44640.44	0.00	697.03
1000	10	0.00	100.00	3614.75	52195.96	0.00	866.58	52195.96	0.00	772.91
1000	15	NA	NA	3739.77	NA	NA	3494.73	89799.23	0.30	3901.83
2000	5	17251.00	0.00	31.43	17251.00	0.00	1280.23	17251.00	0.00	591.14
2000	6	32220.00	0.00	147.28	31730.00	1.52	3567.62	29464.00	10.78	3612.83
2000	7	43524.00	0.00	3608.65	43331.00	0.44	3618.73	43464.00	0.64	3615.43
2000	8	52294.00	0.00	2088.65	52183.00	0.21	3631.55	52294.00	0.01	3626.57
2000	9	0.00	100.00	3622.32	61171.00	0.00	3664.94	61901.00	0.16	3637.06
2000	10	0.00	100.00	3632.51	67395.00	0.00	3650.93	66677.00	3.22	3658.38
2000	15	NA	NA	3717.24	NA	NA	3442.37	NA	NA	3695.01
3000	5	24781.00	0.00	63.67	23477.00	5.26	7207.33	24629.00	1.08	3607.99
3000	6	42204.00	0.00	485.12	41862.00	0.81	3614.14	41726.00	2.01	3614.58
3000	7	52964.00	0.00	3286.40	51506.00	2.75	3632.46	52961.00	0.06	3626.75
3000	8	0.00	100.00	3621.42	61380.00	0.00	3641.34	61131.00	3.25	3639.04
3000	9	0.00	100.00	3626.71	71567.00	0.00	3670.11	72159.00	0.59	3672.42
3000	10	0.00	100.00	3653.59	77793.00	0.00	3708.01	78039.00	0.45	3715.01
4000	5	31842.00	0.00	160.84	31842.00	0.00	27.79	31842.00	0.00	24.48
4000	6	50337.72	0.00	575.42	50337.72	0.00	45.32	50337.72	0.00	53.98
4000	7	0.00	100.00	3623.55	63381.46	0.00	264.77	63381.46	0.00	207.64
4000	8	0.00	100.00	3627.30	70478.65	0.00	335.08	70478.65	0.00	580.71
4000	9	0.00	100.00	3635.25	76369.84	0.00	3494.47	76369.84	0.00	727.98
4000	10	NA	NA	3661.10	79618.45	0.00	867.07	79618.45	0.00	1597.52
5000	5	37934.12	0.00	673.09	37934.12	0.00	69.24	37934.12	0.00	75.48
5000	6	53300.98	0.00	1903.53	53300.98	0.00	264.90	53300.98	0.00	176.83
5000	7	0.00	100.00	3658.05	65361.57	0.00	463.73	65361.57	0.00	378.09
5000	8	0.00	100.00	3675.62	72486.98	0.00	3866.91	72486.98	0.00	839.90
5000	9	0.00	100.00	3697.10	78947.79	0.00	813.33	78947.79	0.00	4554.48
5000	10	NA	NA	3690.50	84995.61	0.00	4298.04	84995.61	0.00	1131.15
Avg.:			42.87	2438.34		0.37	2236.12		0.73	1925.95
Max:			100.00	3739.77		5.26	7207.33		10.78	4554.48

Table 19: Stabilization results for $|S| = 200$ and $costfactor = 10^6$

$ K_s $	b^w	PF			BAC			BAC + Stabilization		
		Obj	g2(%)	time	Obj	g2(%)	time	Obj	g2(%)	time
1000	5	16766.10	0.00	12.64	16766.10	0.00	9.75	16766.10	0.00	10.59
1000	6	27597.19	0.00	30.69	27597.19	0.00	61.61	27597.19	0.00	40.56
1000	7	40992.58	0.00	79.90	40992.58	0.00	66.85	40992.58	0.00	60.94
1000	8	51371.33	0.00	354.34	51371.33	0.00	209.72	51371.33	0.00	190.54
1000	9	58899.58	0.01	3623.01	58904.84	0.00	7226.77	58904.84	0.00	7221.45
1000	10	66528.32	0.00	1782.33	66528.32	0.00	484.39	66528.32	0.00	477.20
1000	15	NA	NA	3710.07	106940.45	0.00	4249.64	106940.45	0.00	4234.98
2000	5	27758.00	0.00	24.26	27758.00	0.00	804.65	27758.00	0.00	339.91
2000	6	44485.00	0.00	89.46	44485.00	0.00	2546.39	44485.00	0.00	494.45
2000	7	56921.00	0.00	1051.88	56921.00	0.00	2201.05	56921.00	0.00	712.54
2000	8	65824.00	0.00	3614.03	65824.00	0.00	3081.48	65824.00	0.00	1142.61
2000	9	0.00	100.00	3624.47	76076.00	0.00	3643.59	76091.00	0.00	3661.13
2000	10	0.00	100.00	3627.28	83654.00	0.00	3670.44	83045.00	1.05	3659.98
2000	15	NA	NA	3768.34	NA	NA	3735.65	NA	NA	4020.67
3000	5	36003.00	0.00	134.59	36003.00	0.00	1719.39	36003.00	0.00	371.48
3000	6	55466.00	0.00	388.46	55466.00	0.00	2744.97	55457.00	0.05	3613.28
3000	7	67527.00	0.00	1150.57	67527.00	0.00	3051.89	67527.00	0.00	1274.86
3000	8	78209.00	0.00	1877.87	78171.00	0.05	3600.70	78209.00	0.00	2802.01
3000	9	0.00	100.00	3633.52	88319.00	0.00	3673.39	87973.00	0.57	3676.39
3000	10	0.00	100.00	3639.12	94704.00	0.00	3600.57	93819.00	1.27	3712.83
4000	5	45047.24	0.00	177.75	45047.24	0.00	74.96	45047.24	0.00	82.18
4000	6	65164.01	0.00	747.65	65164.01	0.00	66.64	65164.01	0.00	77.37
4000	7	79643.34	0.00	3628.21	79643.34	0.00	138.63	79643.34	0.00	140.96
4000	8	0.00	100.00	3640.57	86768.81	0.00	306.16	86768.81	0.00	595.58
4000	9	0.00	100.00	3637.45	93220.05	0.00	343.61	93220.05	0.00	363.17
4000	10	0.00	100.00	3666.32	96903.06	0.00	500.06	96903.06	0.00	591.56
5000	5	51508.09	0.00	429.57	51508.09	0.00	154.87	51508.09	0.00	49.45
5000	6	68215.50	0.00	1099.12	68215.50	0.00	730.80	68215.50	0.00	181.74
5000	7	81521.92	0.00	2349.23	81521.92	0.00	488.06	81521.92	0.00	486.68
5000	8	89714.20	0.01	3634.53	89720.81	0.00	756.91	89720.81	0.00	558.53
5000	9	NA	NA	3703.99	96597.08	0.00	922.29	96597.08	0.00	895.08
5000	10	0.00	NA	3682.46	103248.51	0.00	1404.04	103248.51	0.00	792.70
Avg.:			25.00	2081.68		0.00	1758.44		0.09	1454.17
Max:			100.00	3768.34		0.05	7226.77		1.27	7221.45

Table 20: Stabilization results for $|S| = 300$ and $costfactor = 10^5$

$ K_s $	b^w	BAC			BAC + Stabilization		
		Obj	g2(%)	time	Obj	g2(%)	time
1000	5	5118.43	0.00	163.96	5118.43	0.00	131.37
1000	6	13261.47	0.00	2223.01	13261.47	0.00	859.48
1000	7	25525.38	0.00	458.94	25525.38	0.00	352.45
1000	8	35498.16	0.09	3636.83	35499.73	0.00	1395.10
1000	9	43627.25	0.00	2067.64	43627.25	0.00	948.39
1000	10	51048.52	0.01	5698.23	51051.27	0.00	1366.78
2000	5	17251.00	0.00	1464.73	17241.00	0.34	3609.55
2000	6	31617.00	3.17	3616.84	32220.00	0.00	2159.81
2000	7	43118.00	1.98	3627.27	42516.00	3.74	3625.05
2000	8	51555.00	1.89	3644.55	52264.00	0.00	3650.36
2000	9	60107.00	4.17	3686.92	60782.00	2.50	3672.42
2000	10	66322.00	4.33	3716.28	64160.00	8.22	3700.96
3000	5	24661.00	0.85	3612.85	24583.00	1.83	3610.22
3000	6	41964.00	1.46	3617.83	41403.00	2.77	3618.58
3000	7	48940.00	11.81	3668.16	50604.00	5.55	3633.27
3000	8	61325.00	5.24	3709.73	58827.00	7.49	3658.36
3000	9	69652.00	5.28	3800.05	71800.00	1.35	3710.49
3000	10	75161.00	4.98	3560.05	77841.00	0.92	3835.80
4000	5	35217.29	0.00	1109.58	35217.29	0.00	2444.20
4000	6	47880.91	2.51	3654.15	48280.77	1.33	3626.52
4000	7	61517.89	0.70	3674.61	61713.99	0.36	3644.20
4000	8	69826.73	0.53	3731.77	70068.15	0.03	3697.09
4000	9	74685.19	0.61	3909.67	71755.78	4.64	3754.13
4000	10	NA	NA	3728.61	78651.45	2.99	3962.23
5000	5	38237.46	0.65	3627.51	38335.90	0.00	860.88
5000	6	54125.53	1.85	3449.75	54412.86	0.91	3634.33
5000	7	64841.54	0.98	3710.97	64844.42	0.77	3654.81
5000	8	NA	NA	3459.35	73428.87	0.36	3544.12
5000	9	76216.19	3.98	3830.86	74148.72	6.66	3710.56
5000	10	78474.17	7.35	3939.60	76889.51	8.63	3850.98
Avg.:			2.30	3260.01		2.05	2930.75
Max:			11.81	5698.23		8.63	3962.23

Table 21: Stabilization results for $|S| = 300$ and $costfactor = 10^6$

$ K_s $	b^w	BAC			BAC + Stabilization		
		Obj	g2(%)	time	Obj	g2(%)	time
1000	5	15470.79	0.00	47.80	15470.79	0.00	39.03
1000	6	26183.97	0.00	104.91	26183.97	0.00	122.17
1000	7	39651.20	0.00	509.51	39651.20	0.00	928.72
1000	8	49877.19	0.00	907.81	49877.19	0.00	602.97
1000	9	57884.69	0.00	3646.46	57884.69	0.00	3632.67
1000	10	65819.10	0.00	1065.71	65819.10	0.00	908.24
2000	5	27758.00	0.00	2251.45	27758.00	0.00	965.29
2000	6	44485.00	0.00	2158.99	44485.00	0.00	799.38
2000	7	56756.00	0.92	3595.93	56921.00	0.00	3492.40
2000	8	65811.00	0.03	3640.92	65824.00	0.00	2039.04
2000	9	76614.00	0.16	3704.58	76643.00	0.03	3625.76
2000	10	82579.00	2.17	3728.30	83094.00	1.24	3759.31
3000	5	36003.00	0.00	791.41	36003.00	0.00	1365.40
3000	6	55460.00	0.06	3554.89	55466.00	0.00	1472.34
3000	7	66320.00	2.48	3464.68	67527.00	0.00	1408.05
3000	8	76925.00	2.68	3511.50	77391.00	1.24	3659.89
3000	9	87873.00	0.81	3564.95	86754.00	2.13	3681.23
3000	10	94581.00	0.45	3652.42	93493.00	1.70	3442.02
4000	5	48848.67	0.00	1274.81	48848.67	0.00	211.57
4000	6	63595.10	0.00	1334.39	63595.10	0.00	600.72
4000	7	77638.98	0.00	2514.29	77638.98	0.00	1326.52
4000	8	85652.04	0.35	3469.63	85899.42	0.02	3676.44
4000	9	91562.13	0.26	3585.75	91227.92	0.69	3332.54
4000	10	97216.45	0.91	3892.74	97910.97	0.17	3909.33
5000	5	52313.64	0.00	5325.02	52313.64	0.00	5236.76
5000	6	70004.18	0.00	1404.21	70004.18	0.00	741.27
5000	7	81781.01	0.01	3665.16	81783.00	0.01	3658.96
5000	8	90436.85	0.15	3802.49	90498.28	0.01	3303.29
5000	9	96049.71	0.49	4113.81	NA	NA	3031.39
5000	10	98058.73	3.09	4462.55	NA	NA	3410.32
Avg.:			0.50	2758.24		0.26	2279.43
Max:			3.09	5325.02		2.13	5236.76

Table 22: Stabilization results for $|S| = 400$ and $costfactor = 10^5$

$ K_s $	b^w	BAC			BAC + Stabilization		
		Obj	g2(%)	time	Obj	g2(%)	time
1000	5	5111.28	0.00	412.54	5111.28	0.00	243.97
1000	6	13251.40	0.00	1726.66	13251.40	0.00	1024.79
1000	7	25518.68	0.00	441.17	25518.68	0.00	380.21
1000	8	35491.26	0.00	6977.37	35491.51	0.04	3626.14
1000	9	43608.82	0.02	6296.94	43616.91	0.00	2161.73
1000	10	51039.05	0.00	3701.35	51039.05	0.00	2429.07
2000	5	15454.76	13.11	3528.43	16941.72	2.46	3610.73
2000	6	31768.71	2.58	3609.52	32088.70	0.00	3645.95
2000	7	42844.82	2.76	3653.06	43302.80	1.23	3632.50
2000	8	52118.83	0.49	3682.00	52240.83	0.15	3650.02
2000	9	58095.89	8.10	3579.03	55509.84	12.22	3513.72
2000	10	64267.92	10.17	2485.98	64065.88	8.35	3187.59
3000	5	24701.00	1.73	3606.19	24086.00	4.28	3612.72
3000	6	40838.00	6.85	3636.70	40936.00	4.44	3630.57
3000	7	51668.00	3.92	3681.41	52941.00	0.02	7244.98
3000	8	60822.00	6.42	3625.17	60420.00	5.36	3698.49
3000	9	69511.00	5.44	3713.16	67210.00	9.65	2490.31
3000	10	73794.00	6.90	3634.41	76503.00	3.22	3707.81
4000	5	35074.03	0.54	3466.77	35217.29	0.00	392.08
4000	6	47618.57	3.60	3077.22	47775.86	2.40	3639.14
4000	7	60909.47	1.80	3298.60	61625.78	0.51	3657.86
4000	8	68880.90	2.00	3320.58	69291.86	1.23	3736.40
4000	9	73333.07	2.55	2900.66	72950.00	2.96	3760.41
4000	10	75300.18	8.88	2680.76	78983.52	4.84	3920.63
5000	5	38076.75	1.22	3012.52	38335.89	0.00	3642.92
5000	6	53694.98	2.99	2937.08	54255.35	1.18	3644.94
5000	7	64448.06	1.89	2770.72	64620.83	1.17	3514.81
5000	8	NA	NA	2399.58	72933.64	1.14	3550.47
5000	9	75576.59	6.51	2068.63	NA	NA	2412.45
5000	10	NA	NA	NA	78153.83	7.48	4199.89
Avg.:			3.59	3238.77		2.56	3185.44
Max:			13.11	6977.37		12.22	7244.98

Table 23: Stabilization results for $|S| = 400$ and $costfactor = 10^6$

$ K_s $	b^w	BAC			BAC + Stabilization		
		Obj	g2(%)	time	Obj	g2(%)	time
1000	5	15468.48	0.00	89.70	15468.48	0.00	94.86
1000	6	26177.68	0.00	133.76	26177.68	0.00	84.18
1000	7	39646.27	0.00	748.31	39646.27	0.00	134.53
1000	8	49871.45	0.00	490.65	49871.45	0.00	580.62
1000	9	57873.95	0.00	7238.90	57874.01	0.00	5997.19
1000	10	65806.07	0.00	1840.20	65806.07	0.00	1268.76
2000	5	27757.71	0.00	2547.61	27757.71	0.00	2134.50
2000	6	44451.73	0.00	3644.63	44482.72	0.00	4661.86
2000	7	56668.85	1.06	3650.43	56138.81	1.76	3631.52
2000	8	65823.83	0.00	3658.38	65823.83	0.00	3357.37
2000	9	75697.92	2.12	3719.06	75794.91	1.85	3543.95
2000	10	81483.93	3.61	3779.63	93906.00	1.24	3570.63
3000	5	35990.00	0.04	4662.27	36003.00	0.00	607.38
3000	6	55450.00	0.08	3155.86	55466.00	0.00	2101.62
3000	7	67525.00	0.04	3659.51	67527.00	0.00	3561.99
3000	8	77187.00	1.85	3687.13	77305.00	1.86	3698.79
3000	9	87713.00	1.44	2871.67	88115.00	0.59	3705.00
3000	10	94131.00	1.28	3407.46	93906.00	1.24	3570.63
4000	5	48848.23	0.00	4031.82	48848.67	0.00	1594.02
4000	6	63588.63	0.00	6874.92	63595.10	0.00	660.57
4000	7	76664.31	1.44	3125.14	76652.12	1.38	3659.63
4000	8	85618.65	0.54	3006.59	85766.92	0.23	3718.84
4000	9	89879.41	2.73	2241.09	90547.20	1.47	3713.99
4000	10	NA	NA	3763.53	NA	NA	3657.22
5000	5	52313.42	0.00	4787.01	52313.64	0.00	5117.03
5000	6	70004.18	0.00	1639.99	70004.18	0.00	771.57
5000	7	79570.60	3.30	1712.44	81736.42	0.09	3594.09
5000	8	85447.95	5.97	4066.78	89404.37	1.26	3338.63
5000	9	NA	NA	1551.55	96370.38	0.07	3944.20
5000	10	NA	NA	3704.26	NA	NA	3760.16
Avg.:			0.94	3116.34		0.47	2794.51
Max:			5.97	7238.90		1.86	5997.19

Table 24: Stabilization results for $|S| = 500$ and $costfactor = 10^5$

$ K_s $	b^w	BAC			BAC + Stabilization		
		Obj	g2(%)	time	Obj	g2(%)	time
1000	5	5111.15	0.00	73.27	5111.15	0.00	261.12
1000	6	13234.80	0.14	7140.04	13250.98	0.00	1680.42
1000	7	25518.07	0.00	1400.21	25518.07	0.00	418.05
1000	8	35489.21	0.00	6506.83	35484.78	0.03	7056.08
1000	9	43616.39	0.00	2740.76	43616.39	0.00	1847.90
1000	10	51038.57	0.00	2683.34	51038.57	0.00	2162.15
2000	5	17250.71	0.00	3039.81	16722.73	3.96	3605.19
2000	6	28902.75	16.20	3623.33	32106.70	0.86	3625.97
2000	7	41916.82	6.58	3634.91	43447.82	0.91	3627.63
2000	8	50315.81	6.10	3684.24	51227.87	2.72	3671.57
2000	9	60599.88	3.42	3750.86	60601.91	3.13	3725.49
2000	10	NA	NA	3632.11	61928.94	13.51	3729.19
3000	5	24748.00	0.60	3618.49	24760.00	0.36	3622.86
3000	6	39676.00	9.48	3631.95	40284.00	6.28	3629.92
3000	7	52894.00	0.37	3662.27	52964.00	0.00	3653.13
3000	8	58188.00	10.76	3542.66	61259.00	3.83	3706.88
3000	9	71309.00	2.27	3785.88	64028.00	15.04	3742.24
3000	10	74348.00	6.40	3584.84	74998.00	5.35	3936.76
4000	5	35217.06	0.00	911.54	35217.06	0.00	396.75
4000	6	48641.22	0.00	2577.13	47830.73	2.46	3640.39
4000	7	61137.12	0.00	2280.34	60624.10	2.18	3694.54
4000	8	54403.02	0.00	1800.12	65438.70	7.39	3289.97
4000	9	74625.03	0.52	3951.75	72940.40	3.01	3824.05
4000	10	78729.95	2.82	4252.01	78352.84	4.06	4293.13
5000	5	38041.99	1.43	3600.00	37900.62	1.55	3622.64
5000	6	53600.25	3.35	2676.05	53689.09	2.46	3431.46
5000	7	64291.25	2.33	2432.70	64798.89	0.89	3700.32
5000	8	68877.06	8.84	1952.44	70553.23	4.83	3233.14
5000	9	76172.02	3.60	4097.20	77120.40	3.54	4036.87
5000	10	78505.64	6.89	4369.60	76300.33	11.58	3827.19
Avg.:			3.18	3287.89		3.33	3289.77
Max:			16.20	7140.04		15.04	7056.08

Table 25: Stabilization results for $|S| = 500$ and $costfactor = 10^6$

$ K_s $	b^w	BAC			BAC + Stabilization		
		Obj	g2(%)	time	Obj	g2(%)	time
1000	5	15468.30	0.00	53.96	15468.30	0.00	92.67
1000	6	26177.23	0.00	153.55	26177.23	0.00	97.33
1000	7	39645.63	0.00	175.34	39645.63	0.00	1651.14
1000	8	49870.84	0.00	1154.23	49870.84	0.00	186.32
1000	9	57872.74	0.00	TL	57873.10	0.00	TL
1000	10	65805.56	0.00	1423.28	65805.56	0.00	1830.14
2000	5	27757.71	0.00	1947.88	27691.72	0.28	TL
2000	6	44484.72	0.00	TL	44484.72	0.00	TL
2000	7	56920.82	0.00	TL	56920.82	0.00	970.66
2000	8	65823.83	0.00	2695.83	65823.83	0.00	1160.57
2000	9	74863.89	3.15	TL	69449.98	11.04	TL
2000	10	80925.97	4.27	TL	80566.97	4.90	TL
3000	5	36003.00	0.00	977.55	36003.00	0.00	638.82
3000	6	54947.00	1.92	TL	55463.00	0.02	TL
3000	7	66767.00	2.37	TL	67143.00	1.12	TL
3000	8	77627.00	1.63	TL	77290.00	1.63	TL
3000	9	86517.00	2.92	TL	87403.00	1.54	TL
3000	10	94269.00	0.93	TL	93500.00	1.92	TL
4000	5	48848.42	0.00	1336.59	48848.42	0.00	308.06
4000	6	63594.86	0.00	1253.94	63594.86	0.00	724.81
4000	7	77638.73	0.00	TL	77017.95	0.88	TL
4000	8	85724.10	0.25	TL	85223.95	0.90	TL
4000	9	90952.24	1.82	TL	90621.31	1.50	TL
4000	10	NA	NA	TL	NA	NA	TL
5000	5	52306.14	0.00	TL	52307.88	0.00	TL
5000	6	69864.52	0.25	TL	69791.24	0.46	TL
5000	7	81736.81	0.08	TL	81616.56	0.27	TL
5000	8	90500.98	0.00	TL	90470.78	0.04	TL
5000	9	96072.79	0.40	TL	NA	NA	TL
5000	10	NA	NA	TL	NA	NA	TL
Avg.:			0.71	3161.01		0.98	2861.56
Max:			4.27	7271.27		11.04	5667.87