The Decentralized Field Service Routing Problem

Edison Avraham, Tal Raviv and Eugene Khmelnitsky

Department of Industrial Engineering, Tel Aviv University, Israel

July 2017

The final version of this paper appeared in Transportation Research Part B, 104 (2017) 290–316

Abstract

Companies that provide service at geographically dispersed locations face the problem of determining the technician that will serve each location as well as setting the best route for each technician. Such a scenario is known as the field service routing problem. Large companies often outsource their field service tasks to several contractors. Each contractor may serve several companies. Since the contractors cannot share the information about the tasks of their other clients, the most common practice involves allocating the tasks to the contractors heuristically based on geographical considerations. In this approach, the tasks for which the contractors have already been committed to other companies are not considered. As a result, the allocation of new tasks can be inefficient. This study develops 2-stage task allocation mechanisms that cope with the problem and result in nearly optimal allocations.

In the first stage, a feasible allocation of tasks to contractors is generated. We consider two possible allocation procedures: sequential combinatorial auctions and sequential negotiations. The sequential combinatorial auctions procedure implements the Generalized Vickrey auction, which is a strategy-proof mechanism for the allocation of multiple goods among several competing agents. A sequential negotiation method is suggested as an alternative task allocation mechanism. The method automates a multi-lateral negotiation process in which the company is the leader, and the contractors are followers. In the second stage, the contractors are allowed to exchange the tasks among themselves so as to decrease their operational costs. The exchanges may or may not include money transfers.

We found that the first-stage procedures yield fairly efficient allocations and the second stage further improves them. The obtained allocations are considerably more efficient than the solutions generated by a reasonable benchmark heuristic. Moreover, the allocations' costs are close to a lower bound established by the optimal allocation of a central planner. That is, the price of decentralization is shown to be small.

1. Introduction

Field service organizations operate in a dynamic, ever-changing world that combines long-range planning with emergency responses. Both temporal and spatial aspects are pivotal in the operation

of field service systems. The former is important because the working hours of field service personnel cannot be stored and used later, and the latter is important because the locations of the demand points are typically spread over a large geographical area.

The standard approach in practice and the main research direction in the literature are based on a centralized view. A central planner who is assumed to have access to all the relevant information determines the schedules and routes of all the field crews in order to optimize an objective function, which represents the total cost of the operation. Commercial software packages support this approach.

In recent decades, many organizations have outsourced their field service operation to contractors. In particular, the field service of a single company is outsourced to multiple contractors who, in turn, may serve other companies, and their geographic coverage areas may overlap. In such a market configuration, planning of the field service operation of a company cannot be carried out centrally. Moreover, since the company has no access to the information about the tasks of other companies served by the contractors, it cannot allocate its service tasks efficiently. In practice, the allocation of service tasks to the contractors is carried out based on some simple arbitrary rules. A smarter coordination mechanism that results in a more efficient operation, profitable to both the company and the contractors, is needed. This observation is based on personal communications with researchers and managers in a leading software company in the domain of Field Service Management, Beniaminy (2013) and Kolka (2017).

In this paper, we consider the problem of a service company that wishes to allocate service tasks to contractors in a manner that allows for an efficient solution of the routing problem later faced by the contractors. We refer to this problem as the Decentralized Field Service Routing Problem (DFSRP). The input of the problem consists of a set of tasks and a set of contractors. Each task is characterized by its location and service time. In addition, each contractor is pre-committed to his own set of tasks, which are not revealed to the company. Each contractor is interested in maximizing his revenue net of the labor and routing costs. The company, on the other hand, aims to minimize its payments to the contractors. However, because the business relations between the parties are of a long-term nature, the company also wishes to maintain the profits of the contractor at a reasonably high level.

Centralized versions of the Field Service Routing and Scheduling (FSRS) problem have been the prime focus of many studies carried out in the recent decades. We discuss several representative papers that highlight important aspects of the state of the art.

Beniaminy et al. (2009) introduce several solution methods for the FSRS problem. First, a genetic algorithm, in which each gene in the chromosome is a pair (R_i, D_j) implying that demand D_j is assigned to technician R_i, is presented. Elective chromosomes along with crossovers and mutations comprise the next generation of the population. Next, the authors present an Ant Colony Optimization (ACO) algorithm in which separate pheromone tables for each resource are created.

This algorithm is amendable for parallelization. Finally, a Greedy Randomized Adaptive Search Procedure (GRASP), modified to include pheromone tables, is suggested. The genetic algorithm and the ACO are found to perform well for a complex instance with 620 demands and 88 resources with different skills.

Cortés et al. (2014) consider a real-life setting of the FSRS problem. The planning horizon is one working day. Each customer has a soft time window and a different priority related with that window. Scheduling a customer to be served in the next day is allowed but bares a penalty. The objective function is minimizing a weighted sum of travel cost, lateness cost and the penalties incurred by postponing the service to the next day. The authors develop a Branch-and-Price algorithm that utilize a constraint programming framework in the column generation phase. Numerical experiments are used to demonstrate the applicability of the method.

Kovacs et al. (2012) also study a variant of the FSRS problem. The problem consists of a set of geographically dispersed tasks that needs to be served during a single day by a set of technicians. Each technician possesses certain skills. Each skill has several levels of expertise. Each task requires one or more skills at some level. Each task may or may not have a related time window. Outsourcing of tasks at a given cost is allowed. The authors aim at finding the allocation of the tasks to the technicians that minimizes the total cost of routing and outsourcing. The authors also consider a version of the problem where, for some tasks, no single technician possess all the required skills. In this case, the problem also deals with grouping the technicians to appropriate teams. The two versions of the problem are solved by ALNS algorithms.

Zamorano and Stolletz (2017) study a similar problem and consider a planning horizon with multiple periods. They aim at finding weekly schedules that minimize the total cost of routing, customer waiting and overtime. A weekly schedule is obtained by solving three allocation problems for each day of the week: (1) The daily grouping of technicians to teams (2) The allocation of teams to tasks (3) The creation of daily routes. Each task has a set of possible time windows. The authors present a mixed integer linear programming (MILP) formulation of the problem and two versions of a Branch-and-Price solution strategies.

Pillac et al. (2013) solve a dynamic version of the FSRS where the service requests over time. The service requests are characterized by time windows, required skills, tools and spare parts. These requests may be rejected at a given penalty. Technicians may replenish their spare parts and tools at a central depot at any point of the day. The objective function is to minimize the total routing and penalty costs. The authors argue that the sequence by which requests are served as well as which requests are rejected should be determined dynamically. They present two solution methods for the problem based on parallel adaptive large neighborhood search (pALNS) and on a multiple plan approach (MPA). Numerical experiments show that the pALNS outperforms the MPA method considerably.

Souyris et al. (2013) present a robust optimization method for the FSRS with stochastic service times and soft time windows. The objective function is minimizing the sum of total travel time, total lateness time and total penalty incurred if service is postponed to a later period. The authors explore a version of the problem where service times are correlated and the total service time does not exceed some threshold. The authors present a set-partitioning model for the problem and its robust counterpart which they solve using a Branch and Price algorithm. This counterpart minimizes the cost incurred for the worst case and therefore yields robust solutions. Numerical experiments carried out show that the benefits from applying the robust method increase as the variance in service times increases and as lateness cost increases.

Binart et al. (2016) address the multi-depot FSRS problem with stochastic service and travel times. Two types of customers, i.e. mandatory and optional, are present. Mandatory customers have to be served at given time windows while optional customers may or may not be served at any time of the working day. The objective function is maximizing the number of the optional customers served while minimizing the travel time. The authors devise a two-stage solution method for the problem. Namely planning and execution stages. The planning stage begins with building routes that serve mandatory customers only and then inserting some of the optional customers between the mandatory ones. Both phases are formulated as a MILP. In the execution stage, a dynamic program is used to decide, on-line, upon skipping additional optional customers, if necessary. A similar solution method for a closely related routing problem was presented by Delage (2010).

The literature regarding decentralized algorithms for the FSRS is still sparse. However, such algorithms have been developed for the closely related VRP. Zhenggang et al. (2009) presented an algorithm of this sort for the centralized problem. They solve a capacitated VRP with time windows by implementing an agent-based algorithm. In the proposed framework, a single *scheduling* agent is responsible for allocating orders among a set of *vehicle* agents. The allocation of the orders is done sequentially and follows these steps: Each order is announced by the *scheduling* agent. Next, the relevant *vehicle* agents calculate and send their proposed bids. Finally, the agent whose bid is the lowest wins the order. The authors show that excluding the geographically farthest agents from the bidding process of each task can save computational effort.

In this paper, we consider a decentralized version of the field service routing problem that has not yet been studied. Hence, the centralized algorithms available in the literature do not provide a proper framework for its solution. Furthermore, although agent-based algorithms seem to simulate the structure of the decentralized settings, they do not consider the different objectives of the various parties. In addition, in a decentralized situation the parties are reluctant to share private business information which is not the case for the centralized situation. Next, we discuss several related decentralized logistic problems, a domain in which the literature is still sparse.

Sandholm (1993) develops a 2-stage algorithm that solves a decentralized transportation problem in which several independent distribution centers operate in overlapping geographical

areas. First, each center, seeking to minimize its transportation costs, allocates tasks to the vehicles. Next, the vehicles negotiate between themselves in order to receive or transfer clients in exchange for payments. This mechanism generates solutions that are better than a benchmark solution based on a real-life case study.

Caplice and Sheffi (2006) discuss a problem in the field of freight transportation in which a *shipper* (typically a large retailer) has to buy freight transportation services from a large number of potential *carriers*. Decisions are made by conducting a single shot, first price, reverse combinatorial auction in which the shipper is the auctioneer and the carriers are the bidders. Regularly, a third-party agent manages the auction on the auctioneer's behalf. A shipper typically conducts an auction every two years (on average), and the bidding process usually takes several months.

Huang and Xu (2013) are the first to suggest that the VCG mechanism may be used in the field of transportation procurement. The authors argue that this mechanism is suitable either when a single shipper and multiple competing carriers exist or when many shippers compete on the services of a single carrier. In these two cases, the mechanism induces truthful bidding and maximizes efficiency. The authors allow one, indivisible, bid for each bidder. Next, the authors relax the indivisibility constraint and present three bilateral auction mechanisms. In these mechanisms, carriers as well as shippers are allowed to submit their bids, one bid for each participant. Afterwards, the auctioneer clears the market according to some rules. The authors show that these mechanisms induce truthful bidding and maximize efficiency when the number of carriers and shippers is very large.

In a later research, Xu and Huang (2014) study a similar problem and allow each bidder to submit multiple bids for bundles of items. The authors state that the VCG mechanism may be used to solve the problem and induces truthful bidding as well as system efficiency. However, in practice, the VCG is rarely applied due to the reluctance of the bidders to disclose private information, as well as the mechanism's complexity. Next, the authors suggest that the problem may be solved to optimality by a Primal-dual Vickrey auction (PDV) that induces truthful bidding and pays Vickrey payments under some conditions.

To the best of our knowledge, this research is the first to study the decentralized version of the field service routing problem (DFSRP). We suggest that a variant of the VCG mechanism is applicable to the context of field service operations and present a novel decision-making mechanism that is computationally tractable and thus may be used in practice. The application of the mechanism requires formulating new optimization problems that support the decisions of the involved parties. The main contributions of this study lie in:

1. Introducing and defining the DFSRP as a multi-agent allocation model in which service tasks that belong to a company are to be executed by a set of contractors.

2. Designing a novel 2-stage allocation mechanism for the decentralized problem that does not require the contractors to reveal their private information. The first stage offers an allocation method based on a series of combinatorial auctions that are resolved by the strategy-proof Vickrey-Clarke-Groves (VCG) mechanism. The resulted allocation is near-optimal. In the second stage, the allocation is improved, from the contractors' perspective, by a series of anonymous exchanges of tasks between the contractors.

The rest of this paper is organized as follows: Section 2 states the DFSRP model. Section 3 presents several variations of the task allocation mechanism. In Section 4, we benchmark these mechanisms and study their properties. Section 5 concludes the paper and suggests directions for future research.

2. Problem Definition

In this section, the decentralized field service routing problem is stated and discussed. We model the independently acting agents that make up the problem and develop features of a desired solution. Two performance measures of the suggested mechanism are presented: One is associated with contractors' profitability and the other with the system's efficiency.

2.1 Problem setting

The Decentralized Field Service Routing Problem (DFSRP) is stated as follows: A set of N service tasks should be carried out by a company. Each task is characterized by its service time and location. The company provides the service by outsourcing the tasks to K contractors. Each contractor is pre-committed to additional service tasks allocated by other companies. The parameters of the contractors' tasks constitute private information and cannot be revealed to the other parties. Similarly, before presenting the tasks to the contractors, their parameters are stored privately at the company.

We denote the set of tasks owned by the company by I^C and the sets of the pre-committed tasks by I_k for $k = 1, \dots, K$. Each contractor provides a single field service team, which is initially located at a given location, herein referred to as the contractor's depot.

The goal of the company is to minimize the cost of outsourcing the tasks, whereas the goal of each contractor is to maximize the payments obtained from the company net of its variable operational costs. These costs are associated with the regular time and overtime fee to his field service team and with the mileage its vehicle traveled.

The traveling cost between a pair of locations i and j is denoted by c_{ij} and the traveling time by t_{ij}. The service time at location i is denoted by s_i. The regular time hourly salary of the field service team is denoted by f_1. In overtime, the hourly salary increases by f_2. The planning horizon is a single working day that consists of up to $\hat{L}$ hours, among which the first L are regular hours and the last $\hat{L} - L$ are overtime hours.

Although there is a clear conflict of interests between the company and the contractors, it is not a zero-sum game. Efficient allocation of tasks to contractors may benefit all the parties. Moreover, because the business relations between the company and the contractors are of a long-term nature, an allocation mechanism can be agreed upon in advance and the parties are unlikely to abandon the agreement as long as their expected long-run benefits are greater than the conceivable alternatives.

The goal of this study is to devise an automatic allocation mechanism that satisfies the following requirements

1. The mechanism allows for the allocation of tasks to contractors such that the total operational cost is as close as possible to that of a hypothetic omniscient central planner whose objective is to serve all the required tasks at a minimum cost.
2. The mechanism does not require the contractors to reveal their private information regarding their pre-committed tasks.
3. The reward obtained by each contractor is large enough to motivate him to accept the agreement.
4. Implementation of the mechanism is computationally tractable.

2.2 Performance measures

In order to evaluate the allocation mechanisms proposed in Section 3, we define two performance measures that reflect the attractiveness of the method from different points of view.

From the contractor's point of view, the greater the difference between the reward it obtains from the company and its marginal operational cost for serving the company, the better the solution is. We refer to this measure as the *profitability of contractors.* We are interested not only in a high total profit of the contractors but also in a sufficiently high profit for each one of them.

From the society's point of view, the lower the total operational costs of the contractors, the better the solution is. The quality of the allocation from this perspective is evaluated by comparing it with an optimal allocation obtained by a hypothetic central planner who considers the pre-committed tasks and the other operational constraints. The gap between these two solutions is referred to as *the price of decentralization*. Clearly, the lower the price of decentralization, the better the decentralized solution is. We note that if the companies outsource their service tasks heuristically, without taking the contractors' pre-committed tasks into account, the generated solutions may be inefficient, and the price of decentralization may be very high.

Calculating the abovementioned measures requires one to solve the routing problem of the central planner, as well as the optimization problems faced by each contractor. Although these two problems are not the focus of this study, we formulate and solve them for the sake of evaluating and benchmarking the proposed allocation mechanism. The problems are closely related but not identical to the field service and routing problems discussed in the literature. In the next two

sections, we formulate the first problem as a mixed integer program and show that the second problem is a special case of the first one.

2.2.1 The problem of the central planner

As noted, an optimal allocation of the company's tasks requires minimizing the total cost of all contractors while meeting certain constraints. In particular, the pre-committed tasks must be served by the contractor who owns them, and each of the company's tasks must be served by one of the contractors.

The set of all locations, I, consists of three subsets: depots, company's tasks and pre-committed tasks, i.e.,

$I = \{1..K\} \cup I^C \cup \bigcup_{k\in\{1,..,K\}} I_k$.

Decision Variables

$$x_{ijk} = \begin{cases} 1 & if\ contractor\ k\ travels\ from\ i\ to\ j \\ 0 & otherwise \end{cases}$$

u_{jk} – arrival time of contractor k at location j

T_k – total working time of contractor k

E_k – overtime of contractor k

Model

$$Min\left\{ \sum_{\substack{i,j\in I \\ k\in\{1..K\}}} c_{ij}x_{ijk} + f_1 \sum_{k\in\{1..K\}} T_k + f_2 \sum_{k\in\{1..K\}} E_k \right\} \quad (1)$$

$Subject\ to$

$$\sum_{i\in I} x_{kik} = 1 \quad \forall k \in \{1..K\} \quad (2)$$

$$\sum_{\substack{j\in I \\ k\in\{1..K\}}} x_{ijk} = 1 \quad \forall i \in I\backslash\{1..K\} \quad (3)$$

$$\sum_{j\in I} x_{ijk} = 1 \quad \forall i \in I_k, \forall k \in \{1..K\} \quad (4)$$

$$\sum_{j\in I} x_{ijk} = \sum_{j\in I} x_{jik} \quad \forall i \in I, \forall k \in \{1..K\} \quad (5)$$

$$u_{jk} \geq u_{ik} + t_{ij} + s_i - \hat{L}\left(1 - x_{ijk}\right) \qquad \forall i \in I \ , \forall j \in I \ , k \in \{1..K\} \quad (6)$$

$$T_k = \sum_{i,j \in I} x_{ijk} s_i + \sum_{i,j \in I} x_{ijk} t_{ij} \quad \forall k \in \{1..K\} \quad (7)$$

$$E_k \geq T_k - L \quad k \in \{1..K\} \quad (8)$$

$$T_k \leq \hat{L} \quad \forall k \in \{1..K\} \quad (9)$$

$$x_{ijk} \in \{0,1\} \quad \forall\, k \in \{1..K\}, \qquad i,j \in I_k \cup I^C \cup \{k\} \quad (10)$$

$$E_k \geq 0 \quad \forall k \in \{1..K\} \quad (11)$$

$$u_{jk} \geq 0 \quad \forall\, k \in \{1..K\}, \qquad j \in I_k \cup I^C \cup \{k\} \quad (12)$$

This model minimizes the total cost (traveling, overtime and regular time) of all contractors, while requiring that contractor k begins his route at depot k (see constraint (2)). Constraint (3) ensures that all the company's tasks are served by the contractors. Constraint (4) states that the pre-committed tasks are served by their corresponding contractor. Constraint (5) is a flow conservation equality. Constraint (6) relates the variable u_{jk} that keeps track of the arrival time of the contractor at each task to the routing variable x_{ijk} and is used to eliminate sub-tours, i.e., cycles that do not pass through depots. Constraint (7) calculates the total working time for the k'th contractor by taking into account both the traveling time and the service time. Constraint (8) assigns the correct overtime value to each contractor and (9) limits the total overtime of contractor k. Finally, Constraints (10), (11) and (12) define the decision variables.

Constraint (6) is based on a well-known sub-tours elimination technique in a variety of vehicle routing problems originally introduced by Miller et al., (1960) in the context of the traveling salesman problem. The advantage of this formulation over other alternatives is that it eliminates the exponential number of sub-tours using a small set of constraints. Hence, it can be easily implemented in a commercial solver and modeling language. However, the LP relaxation of the formulation based on this technique is known to yield weak lower bounds, and hence, these formulations are typically hard to solve to optimality. Note that we are interested in the optimal solution of an omniscient central planner only as a reference point to the results obtained by the proposed allocation mechanisms. For this reference to be valid, one cannot use a sub-optimal solution. Therefore, in the numerical experiments reported in section 4 we present lower bounds for the values of the optimal solutions of our problem instances. These are values of super-optimal solutions obtained by using a state-of-the-art solver.

2.2.2 Total cost for a single contractor

The contractor's problem is a special case of the central planner's problem, in which a single vehicle serves all tasks associated with the contractor. As a result, constraint (4) is omitted. All the other constraints are adjusted to consider a single vehicle. Note that if the traveling cost is proportional

to the traveling time (between each pair of locations), the problem is reduced to TSP, which is much easier to solve.

3 The Solution Mechanism

A 2-stage allocation mechanism is developed and implemented in order to cope with the decentralized problem formulated in the previous section. In Stage A, a feasible allocation of all the company's tasks to contractors is created. In Stage B, the contractors are allowed to exchange tasks among themselves. The exchange procedure reduces the total operational cost of the contractors and improves the solution obtained in the first stage.

We consider two variations of the first stage:

a. Sequential sealed-bid combinatorial auctions.
b. Sequential negotiation protocol.

The second-stage procedure is studied in two different settings, with and without cash transfers between the contractors.

3.1 Stage A – Allocation of tasks to contractors

The next two subsections discuss the sealed-bid combinatorial auctions mechanism and the sequential negotiation protocol, respectively.

3.1.1 Sealed-bid partial combinatorial auctions (SBPCA)

A complete sealed-bid combinatorial auction can be used to allocate the tasks to the contractors as follows: The company's tasks are announced to the contractors, and each contractor is required to quote a bid for serving each subset of the tasks. The company then applies the strategy-proof VCG (Vickrey (1961), Clarke (1971) and Groves (1973)) mechanism and determines the winner of each task and the payments to the contractors that minimize systems' cost. The allocation obtained in this manner is optimal from the society's point of view.

The applicability of the method is limited due to its high complexity. Indeed, when calculating the cost of serving all the subsets of N tasks, each contractor has to solve $2^N - 1$ instances of TSP, one for each subset of the company's task along with all its pre-committed tasks.

In order to reduce the computational effort, the proposed heuristic procedure groups the company's tasks into a number of clusters. The clusters are announced to the contractors one by one. The contractors compute the bid for all the subsets of each cluster sequentially. This process requires solving a much smaller number of routing problems. Based on the bids for each cluster, the company then applies the VCG mechanism and determines the winner of each task in the cluster and the reward to each contractor. Thereafter, the allocated tasks are considered as if they were pre-committed by their contractors. The process repeats, with a new cluster, until all the tasks are allocated.

The clusters may have different sizes, with n_{max} denoting the maximum one. n_{max} should be small enough to allow for a moderate number of TSP instances. The success of this method lies in the intelligent partition of tasks into clusters and in determining the order in which the clusters are presented.

The steps of the proposed procedure are as follows:

Step 1. Divide the tasks into clusters (done by the company).

Step 2. Determine the order the clusters are communicated to the contractors (done by the company).

Step 3. Compute the bids for all subsets of the tasks in the clusters (done by the contractors).

Step 4. Apply the VCG procedure to resolve the auction. Allocate tasks and rewards to the contractors (done by the company).

The four steps of the algorithm are as follows.

Step 1 - Clustering

The goal is to produce clusters that contain tasks with geographically close locations because such clusters may be attractive (i.e., not expensive to serve) for the contractors that are either based in this area or have pre-committed tasks there.

The clustering procedure is initiated when the *N* tasks are divided into *N* clusters, each containing a single task. At each step, the procedure scans all the pairs of clusters that contain jointly up to n_{max} tasks and identifies the closest pair. The distance between a pair of clusters is defined by the minimal distance between two contained tasks. These two clusters are merged into a new cluster. The procedure stops when the number of clusters drops to a pre-specified number, $\overline{K}$. Clearly, $\overline{K}$ should be selected such that it is greater than the number of remaining tasks divided by n_{max}. This procedure is inspired by the clustering method developed by Kruskal (1956).

Step 2 - Determining the order the clusters are communicated to the contractors

The order in which the clusters are presented to the contractors is set as follows. First, the midpoint of each cluster is determined as a geometric center of the locations of the tasks that make up the cluster. Then, the distance between a pair of clusters is calculated as the distance between their midpoints. The most isolated cluster, i.e., the one whose distance to its nearest neighbor is the greatest, is set first in the order. The order of the rest of the clusters is determined iteratively, when at each iteration, the closest neighbor to the previously selected cluster is presented.

This method is based on the reasoning that offering nearby clusters one by one may enable a contractor that has won some tasks in one cluster to also win some tasks in the next nearby cluster. This procedure rationalizes the contractors' routes.

Alternatively, the structure and order of the clusters can be imposed based on practical considerations. For example, if the tasks are originated sequentially by a call service, the company may manage a new auction whenever the number of accumulated tasks exceeds a threshold of, e.g., n_{max} tasks.

Step 3 - Computing the bids

Given a cluster offered by the company, each of the contractors calculates the extra cost incurred by serving each subset of these tasks in addition to its pre-committed tasks. This cost can be calculated by subtracting the cost of a TSP solution over its pre-committed tasks from the TSP solution that include the pre-committed tasks and the company's tasks in the subset. If serving some subset of tasks in the cluster is infeasible with respect to the working day length constraint, the bid is ∞. Recall that under the VCG mechanism the best response of the contractors is to bid their actual costs honestly. The mechanism also guarantees that the contractors' reward will exceed their costs by some margin. Note, however, that the heuristic partition of the auction into clusters enables some speculative profits for the contractors if they are able to forecast, at some level of accuracy, the locations of the customers that will be offered in the next stages of the auction.

Step 4- Determining winners and payments

Once all the bids for a cluster are received, the company applies the VCG mechanism and allocates all the tasks of the cluster to contractors. This procedure is performed by selecting the set of bids (at most one from each contractor) that covers the entire cluster. This is a simple variant of the min cost set covering problem. The optimization problem (13)-(16) outlined below determines the winners and the payments for the clusters of more than one task. Although this problem is NP-Hard, relatively large instances of it can be solved in short time using a commercial solver.

Parameters

$\mathcal{C}$ - the set of tasks in the cluster. $P(\mathcal{C})$ is the collection of all subsets of this set (the power set).

$p_{k,S}$ - the cost communicated by contractor k with respect to subset S.

Decision variables

$$x_{k,S} = \begin{cases} 1 & if\ the\ contractor\ k\ serves\ subset\ S \\ 0 & otherwise \end{cases}$$

The winner determination problem

$$Z^*_{\{1,..,K\}} = min \sum_{\substack{S \in P(\mathcal{C}) \\ k \in \{1..K\}}} p_{k,S} x_{k,S} \qquad (13)$$

$$Subject\ to$$

$$\sum_{S \in P(\mathcal{C})} x_{k,S} \leq 1 \qquad \forall k \in \{1..K\} \qquad (14)$$

$$\sum_{\substack{S \in P(\mathcal{C}): i \in S \\ k \in \{1..K\}}} x_{k,S} = 1 \qquad \forall i \in \mathcal{C} \qquad (15)$$

$$x_{k,S} \in \{0,1\} \qquad \forall k \in \{1..K\}, \quad S \in P(\mathcal{C}) \quad (16)$$

The objective function (13) minimizes the total sum of the bids (equivalent to the additional cost incurred by the contractors). Constraint (14) assumes that at most one bid of each contractor is selected. Constraint (15) stipulates that each task of the cluster is served.

Once all tasks are allocated, the company rewards the contractors. According to the VCG mechanism, a contractor is rewarded for his bid plus the marginal saving incurred by its offer. Let Z_R^* be the value of the optimal solution of the winner determination problem for a set (or subset) of contractors R, and let $x_{k,S}^*$ be the optimal solution of this program for the set of all contractors $\{1, \dots, K\}$. The reward of contractor k is formally given by:

$$P_k = (Z^*_{\{1..K\}\backslash k} - Z^*_{\{1,\dots,K\}}) + \sum_{S \in P(C)} p_{k,S} x^*_{k,S} \quad (17)$$

The calculation of the rewards for all the contractors requires solving (13)-(16) $K + 1$ times, once for the set of all the contractors and once for the set $\{1..K\}\backslash k$, for each k. Clearly, because $Z^*_{\{1..K\}\backslash k} \geq Z^*_{\{1,\dots,K\}}$, the reward for each contractor is at least as high as his bid, which is represented by the second term of (17).

We note that in practice the proposed mechanism can be applied without obtaining bids for every possible combination of tasks from each contractor. For example, the contractors may choose to submit bids only for combinations that (heuristically) suit their geographical locations and the locations of their pre-committed tasks. By doing so, they save computational effort on their side and reduce the dimension of the winner determination problem.

Illustrative example

A simple instance of the DFSRP is presented in Figure 1.

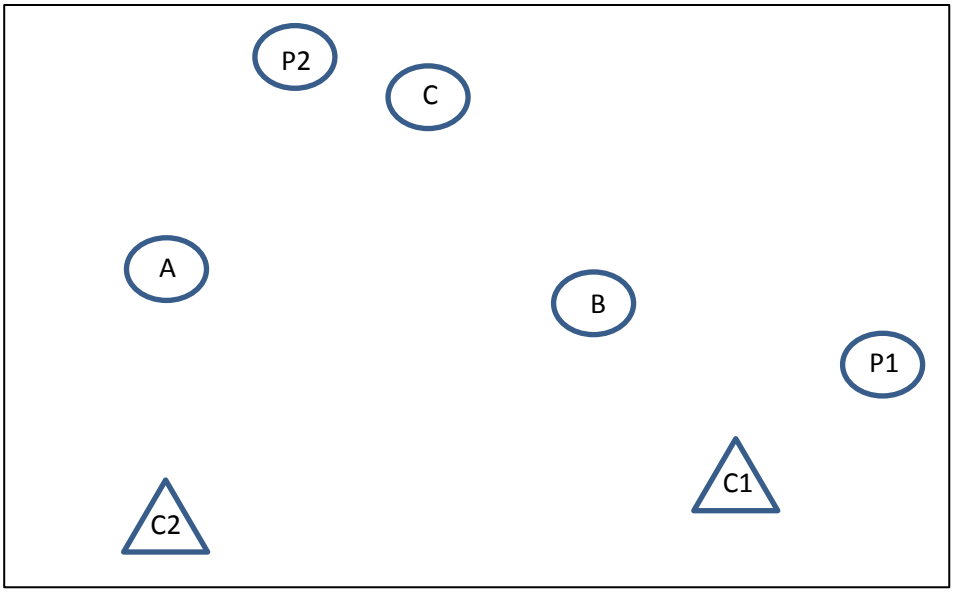


Figure 1: Schematic representation of the DFSRP

The problem consists of two contractors, C1 and C2, and of 3 service tasks, A, B, and C. The three service tasks, A, B, and C should be allocated among the two contractors as to minimize the total added cost for them. Contractor C1 has a pre-committed task P1. Similarly, P2 is a pre-committed task that belongs to contractor C2. The service cost and service time are assumed zero. The length of the working day is 20 time units and the overtime cost (per time unit) is one.

The travel costs are presented in Table 1. Note that the value NA indicates that traveling from origin i to destination j is not allowed.

Table 1: Travel Costs

	C1	C2	A	B	C	P1	P2
C1	0	NA	7	4	8	2	NA
C2	NA	0	4	7	8	NA	7
A	7	4	0	6	5	7	4
B	4	7	6	0	4	3	5
C	8	8	5	4	0	6	1
P1	2	NA	7	3	6	0	NA
P2	NA	7	4	5	1	NA	0

C1 and C2 are required to calculate their bids for each possible subset of the service tasks. These bids are calculated by solving an instance of a TSP that considers the pre-committed tasks of the contractor as well as the appropriate subset. The bids are presented in Table 2

Table 2: Bids

	A	B	C	AB	AC	BC	ABC
C1	12	5	12	14	16	12	18
C2	1	5	2	6	3	4	6

Let us demonstrate the process of bid calculation on C1's bid for the bundle ABC. The TSP route that serves the tasks in the bundle is as follows: C1 – A – C – B – P1 – C1. The total travel time is 7 + 5 + 4 + 3 + 2 = 21. Recall that the length of a regular working day is 20, so the overtime cost is positive and equals 1. The total cost of the route is therefore 22. The original route that contains only pre-committed tasks is C1 – P1 – C1 with a total net cost of 4 units. Therefore, the bid for bundle ABC is 22 – 4 = 18.

Note that P2, for which C2 is pre-committed, is close to tasks A and C. Therefore, C2's bids for A and C are considerably lower than C1's bids for these tasks. This demonstrates the desired effect of considering the pre-committed tasks. Additionally, although the cost of serving task B alone for contractor C2 is high, serving it along with A and C reduces its marginal cost. This highlights the positive effect that bidding on bundles of tasks may yield.

The possible allocation of the service tasks to the contractors and the corresponding additional total cost are presented in Table 3

Table 3: Possible allocations of the service tasks

A	B	C	Total added cost
C1	C1	C1	18
C1	C1	C2	16
C1	C2	C1	21
C1	C2	C2	16
C2	C1	C1	13
C2	C1	C2	8
C2	C2	C1	18
C2	C2	C2	6

The Optimal Solution is to allocate A, B, and C to C2. The cost is 6.

The payment for contractor C2 is the sum of two components. The first component is his cost in the optimal solution, i.e. 6. The second component is the difference between the optimal cost without the bids of C2 and the current optimal cost. In this particular case, without C2'2 bids, all the tasks are allocated to contractor C1, and the total cost is 18. Thus, the contribution of C2 to reducing the total cost is 16 – 6 = 12. The total payment for contractor C2 is therefore 6 + 12 = 18. Note that the optimal solution of a central planner is also to allocate all the three tasks to C2.

An approximation property of the SBPCA solution

Next, we show that gap between the value of the solution obtained by SBPCA mechanism and the optimal central planner solution is theoretically bounded by an additive constant, at least when the

service dominates the travel time. To this end, we consider a special case of the problem (1)-(12) where all depots and tasks are associated with the same (or very close) geographical location(s), so that the traveling times and costs can be assumed zero, $c_{ij} = 0$ and $t_{ij} = 0$ for all (i, j). In such a case, the objective function (1) is simplified; it becomes linearly proportional to the total overtime of the contractors,

$$Z = Min \sum_{k\in\{1..K\}} E_k.$$

This follows since the first term in (1) vanishes and the second term is constant

$$\sum_{k\in\{1..K\}} T_k = \sum_{i\in I} s_i,$$

i.e., it does not depend on the allocation of the tasks to the contractors.

Despite the simplification, the problem remains NP-hard, since it is equivalent to the problem of scheduling a set of jobs on parallel machines with the objective of minimizing the total tardiness about a common due date. The machines are the contractors, and the common due date is L.

For the discussion and proof below let us define the following notation with respect to a fixed allocation of the tasks obtained by SBPCA. Let C_k be the completion time of contractor (machine) k and and let s_k be the duration of the last task allocated to contractor k. We define the following sets of contractors.

$$A = \{k: C_k > L\} \text{ and } B = \{k: C_k \leq L\} \quad (18)$$

Clearly, A is the set of contractors that finish the work after the regular working hours, and thus incur overtime cost while B is the set of the rest of the contractors. Since the SBPCA procedure schedule all the tasks in a cluster optimally with respect to the already decided tasks the following property holds regardless of the cluster size (i.e., even if it one).

Property 1: With respect to the SBPCA allocation mechanism, if B is not empty then any late contractor has exactly one tardy task, and the start time of the tardy task is not greater that the completion time of any early contractor, i.e.,

$$C_k - s_k \leq C_{k'} \quad \text{for all} \quad k \in A \text{ and } k' \in B, \quad (19)$$

The above property follows from the local optimality of the allocated tasks by the VCG mechanism. Indeed, if (19) is violated one of the allocated tasks can be moved from a contractor in A to one in B to reduce the total over time.

The next proposition proves an upper bound of the gap between the optimum allocation of the tasks and the allocation resulted from the Sealed-bid combinatorial auction described above.

Proposition 1.

$$(Z^{SBA} - Z^*) \leq \frac{K-1}{2} s_{max}.$$

Proof. Note that $Z^{SBA} = \sum_{k\in A}(C_k - L)$. If A is empty then $Z^{SBA} = Z^* = \sum_{i\in I} s_i - kL$ and if B is empty then $Z^{SBA} = Z^* = 0$. Therefore, we consider only the case where both A and B are not empty. By Property 1 the inequality (19) holds for any pair of late and early contractors. Therefore, it holds also for the averages,

$$\frac{1}{|A|}\sum_{k\in A}(C_k - s_k) \leq \frac{1}{|B|}\sum_{k\in B} C_k. \tag{20}$$

Inequality (20) is equivalently re-written as

$$\sum_{k\in A}(C_k - L) \leq \frac{|A|}{|B|}\sum_{k\in B}(C_k - L) + \sum_{k\in A} s_k. \tag{21}$$

The proof of the proposition is divided into two cases. Consider first the case where

$$\sum_{k=1}^{K} C_k \leq K \cdot L. \tag{22}$$

By rearranging terms in (22),

$$\sum_{k\in A}(C_k - L) \leq \sum_{k\in B}(L - C_k), \tag{23}$$

moreover, by summing (21) and (23),

$$\sum_{k\in A}(C_k - L) \leq \frac{|B|-|A|}{2|B|}\sum_{k\in B}(L - C_k) + \frac{1}{2}\sum_{k\in A} s_k. \tag{24}$$

Since by definition, $s_k \leq s_{max}$ for all k, and also $L - C_k \leq s_{max}$ for all $k \in B$, inequality (24) is further developed,

$$Z^{SBA} = \sum_{k\in A}(C_k - L) \leq \frac{|B|-|A|}{2|B|}|B|s_{max} + \frac{1}{2}|A|s_{max} = \frac{|B|}{2}s_{max} \leq \frac{K-1}{2}s_{max}.$$

Now since $Z^* \geq 0$, this proves the proposition in the case of (22).

In the opposite case,

$$\sum_{k=1}^{K} C_k > K \cdot L, \tag{25}$$

the cost of the optimal allocation is bounded from below as

$$Z^* \geq \sum_{k=1}^{K} C_k - K \cdot L,$$

or, equivalently,

$$\sum_{k\in A}(C_k - L) - Z^* \leq \sum_{k\in B}(L - C_k). \tag{26}$$

By summing (21) and (26),

$$2\sum_{k\in A}(C_k - L) - Z^* \leq \frac{|B|-|A|}{|B|}\sum_{k\in B}(L - C_k) + \sum_{k\in A} s_k.$$

Similarly to the previous case, the latter is developed into,

$$\sum_{k\in A}(C_k - L) - Z^* \leq \frac{K-1}{2} s_{max},$$

and thus,

$$(Z^{SBA} - Z^*) \leq \frac{K-1}{2} s_{max}$$

which completes the proof of the proposition. ■

Note that while the total gap between the SBPCA solution and the central planner optimal solution grows with the number of contractors. The average gap per contractor is thus bounded by $\frac{1}{2} s_{max}$.

3.1.2 Sequential Negotiation (SN)

The Sequential negotiation method is an alternative task allocation mechanism that automates a multi-lateral negotiation process in which the company is the leader, and the contractors are the followers. The company sets a reward (prize) for each task. In each negotiation session, all the remaining unallocated tasks and corresponding rewards are presented to one of the contractors. The contractor is then allowed to select a subset of the tasks to serve and collect the corresponding reward. The same offer is presented to the contractors according to some arbitrary sequence. After all the contractors have received and reacted to the offer, the rewards are increased, and a new negotiation round is initiated. The process ends when all tasks are allocated.

The negotiation process described above is not a strategy-proof one. Indeed, a contractor may refuse to accept an offer that is profitable for him in expectation for a better offer later in the process, if he believes that no other contractor will take the offer sooner. However, as the number of contractors increases, the monopolistic power of the contractors diminishes, and it is more likely that they will agree to accept offers that exceed their marginal costs only slightly. Moreover, the marginal costs of each contractor change from day to day and even between the rounds of the negotiation process because they are affected by the location of the pre-committed tasks and the offered tasks. Therefore, each contractor may have very little knowledge about the costs of his competitors, a fact that also reduces the opportunities for speculative gains. The following discussion assumes that no speculative considerations are made by contractors. The validity and the implications of this assumption should be verified in a behavioral experiment, possibly in the field.

Three factors determine the dynamics of the negotiation: the order in which the contractors are negotiated, the initial prizes, and the method by which the prizes are incremented.

Contractors could be negotiated according to a fixed order (round robin). However, doing so would be unfair because the contractors who are negotiated first are favored over those who are negotiated last.

Another option would be fixing a certain negotiation order for the first iteration and then reversing this order repeatedly until all tasks are allocated. However, under such ordering, a contractor may know the number and even the identity of the contractors that will be negotiated before his next negotiation session. As an extreme example, the last contractor should never accept the first (or any odd numbered) offer because he will be the first to obtain the first offer in the next round.

In contrast to the previous options, the generation of a random order in each negotiation round appears to be a reasonable choice. Using this approach, no contractor obtains an a priori advantage, and it is more difficult for the contractors to successfully apply speculative considerations that may result in awarding higher prizes.

We propose setting the initial prizes of the tasks by a function of the service time of the task, $r_i = g(s_i)$. The higher the service time is, the higher the prize offered for the task will be.

In order to maintain the correlation between the service time of a task and the prize of that task, the prize is multiplied by a constant b in each subsequent round. The value of the parameter b is greater than 1. Thus, the prize for each task grows exponentially with the number of iterations.

The set of tasks that contractor *k* commits to in the current iteration is obtained by solving the following variant of the prize collecting routing problem.

The prize collecting problem of contractor k

Additional parameters

$\widehat{I_k}$ - set of tasks that contractor k is committed to (either pre-committed to other companies or committed to the same company in pervious negotiation rounds). Contractor *k*'s depot is also included in the set.

I^o – set of tasks the company offers contractor k

$I = \widehat{I_k} \cup I^o$ – set of all geographical locations in the problem

Decision variables

$$x_{ij} = \begin{cases} 1 & if\ contractor\ k\ travels\ from\ i\ to\ j \\ 0 & otherwise \end{cases}$$

u_j – arrival time of contractor at location *j*

T– total working time of contractor

E– Duration of overtime

Model

$$Max\left\{\sum_{i,j\in I^c} r_i x_{ij} - \sum_{i,j\in I} c_{ij}x_{ij} - Tf_1 - Ef_2\right\} \quad (27)$$

$$Subject\ to$$

$$\sum_{j\in I} x_{ij} \le 1 \quad \forall i \in I^o \quad (28)$$

$$\sum_{j\in I} x_{ij} = 1 \quad \forall i \in \widehat{I_k} \quad (29)$$

$$\sum_{j\in I} x_{ij} = \sum_{j\in I} x_{ji} \quad \forall i \in I \quad (30)$$

$$u_j \ge u_i + t_{ij} + s_i - \hat{L}\left(1 - x_{ij}\right) \quad \forall i \in I \quad \forall j \in I\backslash\{k\} \quad (31)$$

$$T = \sum_{i,j\in I} x_{ij}s_i + \sum_{i,j\in I} x_{ij}t_{ij} \quad (32)$$

$$E \ge T - L \quad (33)$$

$$T \le \hat{L} \quad (34)$$

$$x_{ij} \in \{0,1\} \quad \forall i,j \in I \quad (35)$$

$$E \ge 0 \quad (36)$$

$$u_j \ge 0 \quad \forall j \in I \quad (37)$$

The model maximizes the sum of prizes net of costs (regular time, over time and traveling). Constraint (28) allows the contractor to skip some offered tasks, and constraint (29) ensures that he serves all the previously committed tasks. Constraint (30) is a flow conservation equation. Constraint (31) relates the variable u_j that keeps track on the arrival time of the contractor at each task to the routing variable x_{ij}. Constraint (32) calculates the total working time. Constraints (33)-(34) limit and calculate the overtime. The three last constraints define the decision variables. The problem results in obtaining the set of new tasks the contractor is ready to commit to, assuming the value of the optimal solution of this model is greater than the (negative) value of the cost of serving only the previously committed tasks

$$I^s = \left\{i|\exists j, x_{ij} = 1,\ \ i \in I^0\right\} \quad (38)$$

3.2 Exchanging tasks between the contractors

Stage A uses the mechanisms described in Section 3.1 to produce allocations of the company's tasks to the contractors. Stage B, developed in this section, is intended to further decrease the total operational cost of the contractors by allowing them to exchange the company's tasks among themselves. An *Exchange* of task i with task j is defined from the point of view of the contractor who serves task i and means removing task i from his route and inserting task j into the route instead. Note that the payments received from the company in Stage A are not exchanged.

We note that the company has nothing to gain or lose directly from the outcomes of Stage B. However, because the profitability of the contractors may be improved and the relationships between the company and contractors are of a long-term nature, it is in the interest of the company to allow it. The exchange process can be facilitated by the company itself or by a third-party middleman who need not be aware of the location and content of the tasks. Thus, the contractors do not need to disclose sensitive business information to the company. We refer to the manager of the exchange process as *the automatic controller*.

We offer the following protocol for Stage B:

- The company communicates the locations and durations of its service tasks to all the contractors.
- Each contractor calculates the potential profit that he could gain when exchanging task i (that it serves) with task j (that it does not serve). The profit is denoted by a_{ij}.
- The value of a_{ij} is communicated to the automatic controller.
- The automatic controller searches for a set of exchanges that maximizes the sum of profits for the contractors.
- The exchanges are carried out, and possibly money is transferred between the controller and the contractors.
- The process repeats until no set of exchanges that yields positive net profit for all the contractors exists.

The information communicated by the contractors regarding a_{ij} is likely to affect the performance of the protocol. Two levels of information are considered in this study:

1. Complete information (CI) – the amount and the sign (positive/negative) of the profit.
2. Partial information (PI) – the sign (positive/negative) of the profit only.

For each of these levels of information, we propose a different exchange mechanism. In particular, for CI we propose a mechanism that involves money transfers between the contractors, which can be performed with the mediation of the automatic controller. For the PI case, we propose a simpler exchange mechanism that does not require monetary transfers.

3.2.1 Calculation of the profits

Each contractor is required to calculate the profit associated with the exchange of each task i (that it serves) with each task j (that is served by another contractor). To this end, the contractor calculates his cost in the current state (with i in the route and without j), as well as his cost in the alternative status (without i and with j). The two costs are obtained by solving the problem (1)-(12) with appropriately modified input. The value of a_{ij} is obtained by subtracting the latter cost from the former one. For the case of CI, the profit is communicated to the automatic controller (amount and sign). If an exchange is infeasible, then $a_{ij} = -\infty$ is communicated.

When the only communicated information is the sign of the profit, the automatic controller sets $a_{ij} = 1$ for the profitable exchanges and $a_{ij} = -\infty$ for the unprofitable ones.

3.2.2 Finding best sets of feasible exchanges

Once a_{ij} is calculated for all pairs of the company's tasks $i, j \in I^C$, the automatic controller identifies the best set of feasible exchanges. Let us define a complete weighted directed graph $D = (N, A)$, where each node represents a task, and let a_{ij} be the weight of arc (i, j). A feasible set of exchanges constitutes a directed circle on this graph. The total weight of the circle represents the net cost reduction for all the involved contractors. Thus, the problem of finding the best set of exchanges can be cast as the problem of finding a maximal weight set of disjoint circles in the graph. However, recall that each arc (i, j) in the graph is associated with a contractor who is the original owner of task i. Clearly, the profit a_{ij} is calculated assuming only i and j are exchanged and all the other tasks of the contractor remain the same. Hence, for the weight of the circles to truly represent the profit gained from the exchange, we allow the selection of at most one arc of each contractor.

Notations

$\widetilde{I_k}$ – set of company's tasks allocated to contractor k

$I^C = \bigcup_k \widetilde{I_k}$ - set of all company's tasks

For simplicity and without loss of generality, we assume $a_{ij} = 0$ for all pairs of tasks i, j allocated to the same contractor.

Decision Variables

$$y_{ij} = \begin{cases} 1 & if\ task\ i\ is\ exchanged\ with\ task\ j \\ 0 & otherwise \end{cases}$$

Model

$$Max\left\{\sum_{i,j\in I^c} a_{ij}y_{ij}\right\} \qquad (39)$$

$$Subject\ to$$

$$\sum_{j\in I^C} y_{ij} = \sum_{j\in I^C} y_{ji} \quad \forall i \in I^C \qquad (40)$$

$$\sum_{j\in I^C} y_{ij} \leq 1 \quad \forall i \in I^C \qquad (41)$$

$$\sum_{\substack{i\in \widetilde{I_k} \\ j\in I^C}} y_{ij} \leq 1 \quad k \in \{1..K\} \qquad (42)$$

$$y_{ij} \in \{0,1\} \qquad (43)$$

The problem maximizes the sum of weights of the circles while ensuring that a removed task appears in the routes of another contractor (constraint (40)). Each task is exchanged at most once (constraint (41)) and each contractor exchanges no more than one task (constraint (42)).

In the case of PI, in which a_{ij} is either 1 or minus infinity, solving (39)-(43) simply means maximizing the number of performed exchanges. All exchanges chosen by the model are profitable for the corresponding contractors.

In the case of CI, the solution of the model, in fact, maximizes the total gain of all the contractors. However, the optimal exchanges can be unprofitable for some contractors. Consider, for example, three tasks with a_{ij} given in Table 4.

Table 4: The values of a_{ij} for three tasks

	1	2	3
1		-2	$-\infty$
2	$-\infty$		8
3	8	3	

The possible exchanges are represented by the graph given in Figure 2.

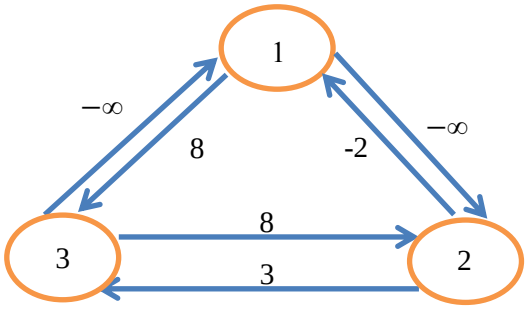


Figure 2: Graph for the example of Table 4

In the example, the maximum total profit is 14. Note that obtaining this profit suggests exchanging task 1 with task 2, which results in a negative profit of 2 for the owner of task 1. To make this arrangement acceptable by this contractor, money transfers should be applied.

3.2.3 Money transfers

A situation in which some contractors have negative profits when the most profitable set of exchanges is applied can be resolved by money transfers, when contractors whose profits are large subsidize the contractors whose profits are low (or negative). Recall that the total profit gain from the exchange of all the contractors is positive. This net profit can be distributed among all contractors such that each contractor is ensured a positive net profit. Deciding how to distribute the total profit derives the corresponding subsidies for all contractors.

Let $\tilde{C}$ be a set of tasks that constitute a circle in the obtained solution and $\tilde{S}$ be the set of contractors that own the tasks in $\tilde{C}$ (each contractor owns one task). The total profit gained by implementing the exchanges suggested by $\tilde{C}$ is $\sum_{i,j\in\tilde{C}} a_{ij}$. If contractor $k \in \tilde{S}$ exchanges task i with task j, then his profit is a_{ij}. If the desired profit of contractor k is π_k, then the corresponding subsidy equals $\pi_k - a_{ij}$. Clearly, if $\sum_k \pi_k = \sum_{i,j\in\tilde{C}} a_{ij}$, then the sum of subsidies is 0, and no extra money is required to perform the transfers.

Because the values of a_{ij} are known, what is left to determine is π_k – the desired profit for contractor k. We suggest that the profit of each contractor is equal, i.e., $\pi_k = \pi = \frac{\sum_{i,j\in\tilde{C}} a_{ij}}{|\tilde{C}|}$. The corresponding subsidy for contractor is $\frac{\sum_{i,j\in\tilde{C}} a_{ij}}{|\tilde{C}|} - a_{ij}$. We note that this distribution constitutes a solution to a Nash's bargaining game for more than two players (Nash 1950).

3.2.4 Stage B – Discussion

Stage B ensures that all contractors are better off or indifferent in each iteration of the method. This claim holds both for the case of CI and for the case of PI. However, truthfully reporting a_{ij}

values may not be an optimal strategy for the contractors. Consider the scenario illustrated in Figure 3.

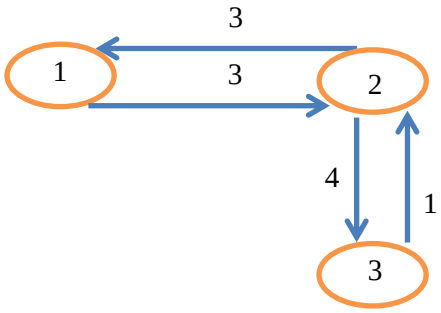


Figure 3: Example graph of truthful bidding

The maximal total gain for the contractors equals 6. The gain is obtained when the contractor who owns task (1) and the contractor who owns task (2) exchange their tasks. Each of the contractors gains 3 units of money from the exchanges, and no money is transferred. Hence, each contractor is better off by 3 units. The contractor who owns (3) is not a part of the solution and therefore is not affected.

Note that the contractor who owns (3) can report false values of his potential profits and by doing so becomes a part of the set of exchanges. This set will be sub-optimal from the perspective of the system. Consider the alternative graph displayed in Figure 4.

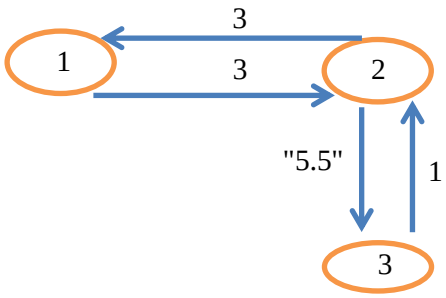


Figure 4: Example graph of untruthful bidding

From the perspective of the automatic controller, the maximal profit equals 6.5 and is obtained when the contractor who owns task (2) and the contractor who owns task (3) exchange tasks between them. The profit of each contractor should be $\frac{6.5}{2} = 3.25$. Thus, the contractor who owns task (3) pays $5.5 - 3.25 = 2.25$ units of money as a subsidy to the contractor who owns (2). The former's real net profit equals 4-2.25 = 1.75. The system's real net profit equals 4+1 = 5. In

other words, a contractor can successfully manipulate the mechanism to output a sub-optimal solution that is better for him.

Successful manipulations carried out by dishonest contractors are also possible in the case of PI.

Consider the example shown in Figure 5.

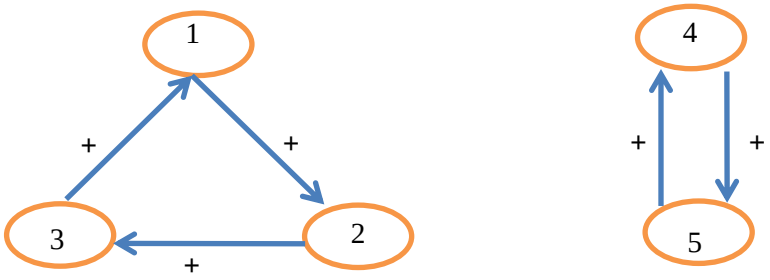


Figure 5: Example graph of truthful reporting

Suppose that contractor k_1 owns tasks (1) and (4), contractor k_2 owns tasks (2) and (5) and contractor k_3 owns task (3). Because all possible exchanges are profitable, reporting truthfully leads to executing the exchanges implied in the circle that contains the highest number of arcs, that is, the circle that contains 3 arcs. All contractors are better off.

Now, suppose that the exchanges implied in the 2-arc circle yield the highest profit for contractor k_2. The latter can report a (false) negative profit from executing the exchanges implied by the 3-arc circle, thus making it unfeasible in the eyes of the automatic controller. Figure 6 illustrates this report.

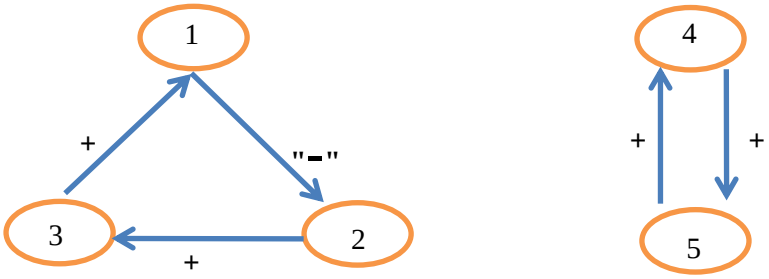


Figure 6: Example graph of untruthful reporting

By reporting untruthfully, contractor k_2 manipulated the controller and made him choose the set of exchanges preferred by him, although they are sub-optimal from the perspective of the society.

To conclude, the mechanisms presented for Stage B are not strategy-proof. However, we propose including the mechanism as an improvement heuristic for the solution generated in Stage A. This recommendation is based on the following practical and theoretical reasons.

First, in practice, for a contractor to successfully manipulate the system, he needs to have information regarding the a_{ij} of the other contractors, which is not likely. Without this information, each contractor can secure a non-negative profit by truthfully reporting his a_{ij}. Second, none of the contractors can be worse off as a result of the exchanges due to one of the abovementioned mechanisms.

Proposition: The proposed exchange mechanism does not reduce the profit of each of the contractors in either version of Stage B regardless of the information available to the contractors and the action of the other contractors.

Proof: We first show that, under both mechanisms and regardless of the actions of the other contractors, a contractor can secure a non-negative improvement in his profit by bidding truthfully. Thus, no contractor submits a bid that results in a loss.

For the version of Stage B without money transfers, the claim is trivial. In this case, only exchanges that are reported by the contractors as profitable are carried out. Thus, any truthful bidder can improve his profit.

For the version of Stage B with money transfers, consider a contractor who owns task i_1 but does not own task j_1 and let $\bar{a}_{i_1 j_1}$ be the true (possibly negative) profit of this contractor from trading i_1 for j_1. Now, assuming he bids truthfully, that is, $a_{i_1 j_1} = \bar{a}_{i_1 j_1}$, and that the exchange is carried out, he will receive the direct profit $a_{i_1 j_1}$ and the subsidy $= \frac{\sum_{i,j \in \tilde{C}} a_{ij}}{|\tilde{C}|} - \bar{a}_{i_1 j_1}$. That is, he will receive a total of $\frac{\sum_{i,j \in \tilde{C}} a_{ij}}{|\tilde{C}|}$. This is a positive value because regardless of the truthfulness of the bids of the other contractors, the circle $\tilde{C}$ is selected by the mechanism only if $\sum_{i,j \in \tilde{C}} a_{ij}$ is positive. This concludes the proof. ■

4. Numerical Experiments

We have conducted numerical experiments to evaluate the effectiveness of the mechanisms developed in the previous section. The results of the experiments are presented and analyzed with respect to the performance measures defined in Section 2.

4.1 Problem Instances

An instance of the problem is characterized by the locations and durations of all service tasks (those of the company and those pre-committed by the contractors) as well as by the locations of the contractors. The locations of the contractors remain fixed for all the generated instances of the problem, whereas the locations of the service tasks change across the problem instances.

The number of tasks in each instance is set to 40, half of which are pre-committed tasks and the rest are company's tasks. The number of contractors is set to four such that the size of the generated instances is large enough to demonstrate the outcomes of the proposed mechanisms. Realistic instances of the problem are likely to be larger. However, testing the mechanisms with larger instances would require developing and implementing stronger solution methods (exact or heuristic) for the underlying routing problem, which is out of the scope of this study. Recall that our focus here is to study the ability of the mechanisms to induce efficient allocation of the tasks to the contractors.

For each instance, the locations of the tasks were randomly generated on a 100×100 square using a continuous uniform distribution ($x_i \sim U(0,100), y_i \sim U(0,100)$). The locations of the contractors' depots were set to $(25,25)$, $(25,75)$, $(75,75)$ and $(75,25)$. The durations of service tasks (in minutes) were generated using a normal distribution with an expectancy of 30 and a standard deviation of 10 ($s_i \sim N(30, 10^2)$).

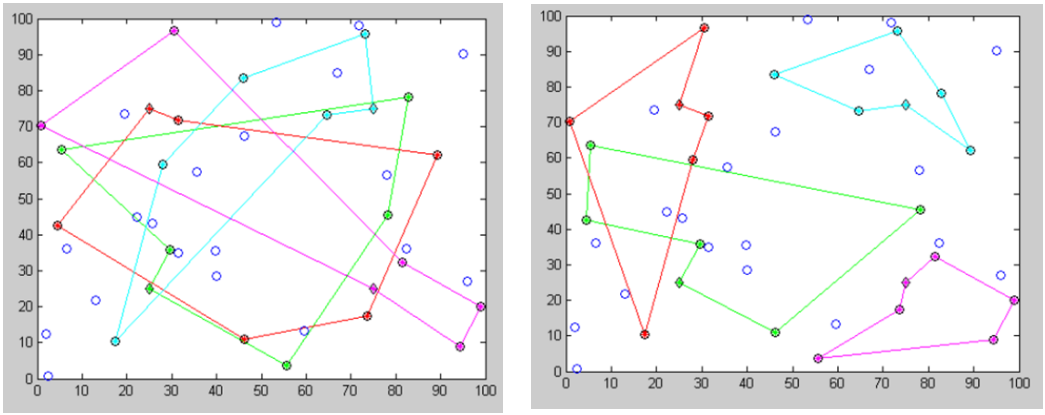
The length of a regular working day is set to $L = 480$ minutes (eight hours), and the maximum length of a working day, including overtime, is set to $\hat{L} = 720$ minutes (twelve hours). The cost of regular working time is set to $f_1 = 3.33$ money units per minute (i.e., 200 per hour). The additional cost of overtime is set to $f_2 = 1.67$ (i.e., extra 50%). The traveling cost and traveling time between each two points are set to the Euclidian distance between the points. This definition implies that the cost of traveling is 1 unit of money per unit of distance (e.g., km) and that the traveling speed is one (e.g., km/min).

The tasks are randomly partitioned to the company's tasks and pre-committed ones. We used a parametric probabilistic procedure that relates pre-committed tasks to the contractors with a probability that decreases with the traveling time from their locations. This approach is practical because a task is more likely to be served by a nearer contractor. For pre-committed task i, we defined the probability that the task belongs to contractor k as

$$\Pr(i \in I_k) = \frac{\left(\frac{1}{t_{ik}}\right)^{\alpha}}{\sum_k \left(\frac{1}{t_{ik}}\right)^{\alpha}} \qquad (44)$$

where t_{ik} is the traveling time between the locations of task i and contractor k. Parameter α controls the clustering structure of the instance. The larger the value of α is, the higher is the probability of relating a task to its nearest contractor. However, because we wish to create balanced instances, we fixed the number of tasks to be related to each of the four contractors to five (exactly one quarter of the number of pre-committed tasks).

We generated 20 datasets of 40 service tasks locations and times. For each dataset, the locations of the 20 company tasks were randomly selected. The rest of the tasks (pre-committed) were related to contractors using (44) for $\alpha = 1$ and for $\alpha = 3$. That is, we created 40 instances of the problem in total. The input for these instances is available upon request from the corresponding author.



(a) $\alpha = 1$ (b) $\alpha = 3$

Figure 7: Optimal routes serving the pre-committed tasks only, for $\alpha = 1$ and $\alpha = 3$

Figure 7 demonstrates two such instances, with the routes of the contractors serving their pre-committed tasks only. The diamonds represent the contractors' depots. The route and pre-committed locations of each contractor are shown in different colors. Whereas in the case of $\alpha = 3$ (Figure 7b) most of the pre-committed tasks are related to their nearest contractor (or at least – second nearest), for $\alpha = 1$ (Figure 7a) the relation appears almost random. Consequently, the initial routes for $\alpha = 3$ are much shorter than the initial routes for $\alpha = 1$.

4.2 A heuristic allocation used as a baseline

In practice, companies that outsource their service tasks to several contractors typically allocate the tasks using simple geographic considerations. A service zone is defined for each contractor, and service calls are allocated based on this zoning. If a contractor is overloaded such that he cannot handle tasks from his zone, the task is allocated to a contractor in a neighboring zone. To benchmark our method, we imitate such heuristic allocation and compare its performance measure with the allocation obtained by our mechanism.

We use the following heuristic method: The tasks are first scanned in increasing order of the distance to their nearest contractor. Then, the tasks are iteratively allocated to the nearest contractor. In order to balance the work load of each contractor, the number of the company's tasks

allocated to a single contractor is limited to $\left\lceil \frac{N}{K} \right\rceil$ tasks (5 in our case). Once a contractor has received 5 tasks, he is removed from further consideration.

After applying this procedure, the shortest tour for each contractor is found, and the total working time (service and routing) is calculated. The total cost of this allocation is the sum of the total costs of the contractors. Below, we refer to this procedure as HE.

4.3 Mechanism parameters

In this section, we present the values of the parameters for the suggested versions of Stage A of the allocation mechanism. Recall that there are no parameters to define in Stage B.

For the SBPCA, n_{max} (maximal size of each cluster) must be defined. As noted, the effectiveness of the solution increases with n_{max}. However, the value of n_{max} is limited due to the complexity of the combinatorial auction and the need to conduct it in a reasonable time. This arises the need to evaluate the tradeoff between the duration of the procedure and the obtained results. We therefore selected two values for n_{max} i.e. $n_{max} = 1$ and $n_{max} = 8$. When $n_{max} = 1$, the number of clusters $\overline{K} = 20$ and the auction reduces to a simple Vickrey auction and is carried out relatively fast. When $n_{max} = 8$, choosing $\overline{K} = K = 4$ seems adequate and the procedure requires much larger computation time. Comparing the obtained results for the two values of n_{max} enables us to derive insights regarding the added value of the SBPCA procedure.

For the SN mechanism, we considered two values for $g(s_i)$, i.e. $g(s_i) = 100\% \times f_1 \times s_i = 100\% \times \frac{10}{3} \times s_i$ and $g(s_i) = 150\% \times f_1 \times s_i = 150\% \times \frac{10}{3} \times s_i$ for r_i. Recall that s_i is the service time in minutes and f_1 is the cost of one minute of regular working time. Note that $r_i = 100\% \times \frac{10}{3} \times s_i$ is the minimal prize for which it may be possible to have a contractor who is willing to serve task i. This is the case only if the task is located exactly on the route of a contractor and the contractor has enough spare regular time to serve it. Allocating the task to the contractor in such a case incurs neither additional traveling cost nor overtime cost. Even in this case, the net profit from serving the task is zero. Therefore, it is reasonable from the company's point of view to begin the negotiation offering this prize.

However, offering $r_i = 100\% \times \frac{10}{3} \times s_i$ may result in low profits for the contractors. If the company wishes to maintain the profits of the contractors at a reasonably high level, it may choose to begin the negotiation with higher values for r_i. We selected $r_i = 150\% \times \frac{10}{3} \times s_i$ as a benchmark for such values. Clearly, increasing the value of the initial prizes may increase the profitability for the contractors. However, this may also result in less efficient solutions.

The increase rate of the prize was set to $b = 1.1$ for the two values of $g(s_i)$.

4.4 Computational environment

Experiments were carried out using an Intel® core ™ i7-2600 CPU @ 3.40 Ghz processor and 16 GB of RAM. The MILP models were solved using IBM CPLEX 12.5.1.

The TSP (presented in section 2.2.2) and the prize-collecting problem of the contractor (presented in section 3.1.2) were solved directly using CPLEX with time limits of 10 seconds and 1800 seconds, respectively. Both of these time limits were typically sufficient to obtain optimal solutions for the two problems. The winner determination problem in the combinatorial auction and the problem of finding the best set of exchanges (see sections 3.1.1 and 3.2.2 respectively) were solved to optimality in less than one second.

4.5 Experimental procedure

As previously noted, 40 instances were generated. For each instance, all variations of Stage A were applied, and the obtained allocations were kept. Next, the two variations of Stage B were applied for each allocation of Stage A.

Note that Stage B is also considered for the allocation generated heuristically. This consideration is valuable because applying Stage B, performed between the contractors themselves, is possible even without applying collaborative mechanisms designed to generate the initial allocation.

4.6 Results

We now present the results of the experiments for the procedure described above. The results are analyzed using the performance measures defined in Section 2. A solution to the decentralized problem is defined by an allocation of the company's tasks to the contractors and by the sum of the prizes that the company pays the contractors.

Total cost for the contractors

The total cost incurred by the contractors is compared with the cost of the optimal allocation that a central planner with complete information could have found. Because the optimal solution of the central planner problem is not always available, due to the complexity issue, the comparison is carried out relative to the best lower bound that the solver provides. The lower bounds are presented in Appendix A. Note that the lower bounds for $\alpha = 3$ are tighter, and in many cases the model is solved to optimality. This approach is taken because the central allocation problem is easier to solve when the pre-committed tasks of each contractor are more clustered.

We examine the suggested mechanisms by considering the total cost of the allocations that they generate and comparing this cost with that of the optimal allocation of the central planner (or a lower bound whenever the optimal solution cannot be found). We report the gap between the

cost of the allocation obtained by the mechanisms and the optimal cost. This gap represents the *price of decentralization* and is calculated as follows: $100\% \times \left(1 - \frac{Central\ planner\ cost}{Mechnism\ Cost}\right)$. The abovementioned gap is presented in Table 5 for the variations of Stage A as well as for the heuristic method as a baseline. For each method, we present the average gap (over the 20 instances) and the standard deviation. The results are reported for the outcome of Stage A only as well as with both variations of Stage B, i.e., with partial information and no money transfers (PI) and with money transfers and complete information (CI). For the detailed results of all instances, see Appendixes B and C.

Table 5: Average optimality gaps and standard deviations for generated allocations

		$\alpha = 1$			$\alpha = 3$		
		Stage A only	Stage B without money transfers	Stage B With money transfers	Stage A only	Stage B without money transfers	Stage B With money transfers
SBPCA $n_{max} = 1$ (Vickrey)	Average	4.57%	4.45%	3.96%	1.63%	1.56%	1.40%
	Std.	0.58%	0.56%	0.49%	0.37%	0.36%	0.33%
SBPCA $n_{max} = 8$	Average	3.96%	3.93%	3.69%	1.41%	1.37%	1.26%
	Std.	0.45%	0.46%	0.48%	0.30%	0.30%	0.30%
SN $r_i = 100\% \times \frac{10}{3} \times s_i$	Average	4.14%	4%	3.76%	1.74%	1.60%	1.38%
	Std.	0.52%	0.50%	0.47%	0.31%	0.31%	0.28%
SN $r_i = 150\% \times \frac{10}{3} \times s_i$	Average	5.31%	5.13%	4.06%	3.39%	2.78%	1.79%
	Std.	0.50%	0.50%	0.43%	0.44%	0.36%	0.28%
HE	Average	9.57%	6.5%	4.29%	5.70%	3.52%	1.61%
	Std.	0.71%	0.63%	0.49%	0.76%	0.50%	0.31%

For $\alpha = 1$, the total cost generated by applying the SBPCA with $n_{max} = 8$ is significantly lower than the total cost generated by SBPCA with $n_{max} = 1$ (p-value=0.005). However, this result is not significant for the case of $\alpha = 3$ (p-value=0.1). For $\alpha = 3$, the initial routes are clustered and routes, do not overlap. In this case, allocating each task to the lowest bidder separately and sequentially is more likely to result in a good solution. Additionally, the next announced task in the sequence is the one closest to the previous one. This approach also helps to build efficient solutions at each step in the sequence. The added value from considering many tasks at the same auction is greater when the routes overlap. This is the case when $\alpha = 1$.

In the SN method, the gap of the obtained solution increases with the initial prize r_i. This result is statistically significant (p-value<0.001). This can be explained by the fact that higher prizes may attract contractors to accept tasks that they are less fit to serve. For both values of α, the total cost of the allocations generated by applying the SBPCA or the SN are significantly lower than the total cost of the allocations generated by the baseline heuristic approach, the comparisons with all other methods are significant with p-value<0.02.

The improvement gained by applying Stage B with money transfers is always significant. The advantage of Stage B with money transfers over the version without transfers is apparent and statistically significant for most of the combinations of α and Stage A methods. The detailed output of the statistical tests is presented in Appendix D.

The average gaps for $\alpha = 3$ are lower than for $\alpha = 1$. Therefore, it appears that all variations of Stage A perform better for $\alpha = 3$. This result can be an indication of a weakness of the mechanism for $\alpha = 1$. Another possible explanation for the higher optimality gap is the fact that the lower bounds for the $\alpha = 1$ instances are looser. In our view the latter explanation is more likely because while many of the $\alpha = 3$ instances of the central planner problem were solved to optimality, there was only one optimal solution among the $\alpha = 1$ instances.

The contribution of Stage B is more significant if the allocation of Stage A is far from the central planner optimum. The experiments that used the heuristic allocation demonstrated it best. In fact, applying the heuristic method and then Stage B results in a fairly good allocation for both values of α. Indeed, one may argue that one viable protocol is to first allocate the tasks heuristically based on their geographical locations and then apply Stage B with money transfers. Such an approach will yield an allocation that is only slightly less efficient than the ones obtained by the other alternatives, and it is much easier to implement. In addition, it allows generating feasible allocations immediately and then improving them gradually until the beginning of the working day.

Profitability of the contractors

We observed that the suggested mechanism yields good allocation of tasks from a social point of view and results in a low price of decentralization. Next, we discuss the profitability of the allocation from the contractors' point of view. The net profit (sum of prizes net of additional costs) of the contractors for $\alpha = 1$ and for $\alpha = 3$ is presented in Table 6.

In this section, we consider profits for contractors generated after Stage A. Stage B reduces the contractors' costs and thus further improves the allocation and the contractors' profits. However, we do not address this aspect because our emphasis is on the interaction between the contractors and the company.

Table 6: Net profit for the contractors for $\alpha = 1$ and for $\alpha = 3$

	$\alpha = 1$				$\alpha = 3$			
Instance number	SBPCA $n_{max} = 1$ (Vickrey)	SBPCA $n_{max} = 8$	SN $r_i = 100\% \times \frac{10}{3} \times s_i$	SN $r_i = 150\% \times \frac{10}{3} \times s_i$	SBPCA $n_{max} = 1$	SBPCA $n_{max} = 8$	SN $r_i = 100\% \times \frac{10}{3} \times s_i$	SN $r_i = 150\% \times \frac{10}{3} \times s_i$
1	1772.92	913.09	149.43	597.80	1347.79	1134.17	154.04	659.23
2	1433.94	1027.03	157.02	757.56	2342.59	1371.40	130.92	630.40
3	2034.95	1251.09	141.29	715.58	1723.01	1227.91	150.36	707.02
4	823.84	662.86	142.42	481.27	2035.87	1375.68	145.90	570.36
5	464.95	809.16	150.61	658.21	2377.78	1189.86	170.73	645.89
6	2397.3	1187.74	127.86	532.40	1798.2	1059.35	160.79	520.30
7	612.18	606.85	171.51	773.06	730.76	740.02	175.90	698.42
8	1618.38	1158.19	179.42	639.50	2082.3	1453.00	156.11	818.94
9	512.57	456.28	161.70	666.73	1927.06	1294.44	139.52	575.97
10	1147.84	1015.29	175.18	617.72	1562.36	1456.81	173.57	652.09
11	1265.72	812.38	123.39	713.12	2454.05	1518.07	105.11	487.60
12	1221.1	821.37	166.35	427.45	1718.86	974.15	193.06	454.74
13	821.41	938.66	162.98	572.50	927.62	976.97	105.39	672.72
14	681.73	652.75	148.10	545.65	988.6	1063.06	157.14	545.75
15	858.05	915.98	139.52	1011.72	2323.33	1360.43	115.43	1025.33
16	1688.22	950.68	184.64	590.53	2444.54	1464.86	200.33	1061.23
17	907.52	375.43	147.10	518.58	1436.15	1177.07	158.93	739.83
18	894.99	868.99	131.11	592.53	1822.73	1017.98	143.57	502.44
19	698.38	700.23	169.65	909.51	1325.61	1010.55	152.72	739.97
20	852.64	1103.27	164.62	903.88	1807.31	1415.64	144.39	705.02

The results above show that the suggested mechanism is always profitable for the contractors (even before applying Stage B). The profit for the contractors is the highest in the SBPCA with $n_{max} = 1$. In this case, the prize for each task is the second lowest bid. This bid is calculated based on the marginal cost of that task. That is, the sum of prizes is the sum of the marginal costs. Typically, this sum is higher than the additional cost incurred when serving a bundle of tasks. The latter being the basis for the bid calculation in SBPCA with $n_{max} = 8$. It is also the basis for the cost estimation of the set of tasks to be served in the sequential negotiation.

The profit for the contractors is the lowest for the SN with $r_i = 100\% \times \frac{10}{3} \times s_i$. Note, however, that our evaluation of the profits of the contractors is based on the assumption that the contractors do not apply strategic considerations in their bids. This assumption is fairly realistic for the "nearly" strategy-proof SBPCA procedure but not necessarily for the SN procedure. In practice, the contractors may gain higher profit by speculatively differing their offers in the SN procedure.

The detailed output of the statistical tests we conducted is presented in Appendix E. Summary of several key features of the various allocation methods is presented in Table 7.

Table 7: Summary of features of the various allocation methods

	SBPCA $n_{max} = 1$ (Vickrey)	SBPCA $n_{max} = 8$	SN $r_i = 100\% \times \frac{10}{3} \times s_i$	SN $r_i = 150\% \times \frac{10}{3} \times s_i$	HE
Computational complexity	Medium	High	Medium	Medium	Low
Efficiency of allocation	Close to SBPCA with $n_{max} = 8$	Usually the best	Very close to SBPCA with $n_{max} = 8$	Medium	Poor. However, can be significantly improved by Stage B
Profits for contractors	Very high	Medium	Low	Medium	NA

No allocation method is dominant over all the other methods in all aspects. In practice, the appropriate allocation method may vary from one organization to another.

5. Conclusions and future research

The Decentralized Field Service Routing Problem (DFSRP) is faced by companies that provide service to a set of spatially dispersed clients by outsourcing their tasks to contractors. The contractors are interested in maximizing their net profit, which is affected by the location of other tasks for which they are pre-committed. The contractors are reluctant to reveal information about these pre-committed tasks since this is a sensitive private information.

The game theoretic literature provides us with a strategy-proof auction mechanism that guarantees a socially optimal allocation of items to buyers (or tasks to service providers in our case). However, this mechanism is computationally intractable because it requires an exponential number of bids from each agent. Moreover, in the context of the DFSRP, the calculation of the value of each bid is an NP-hard problem. In this paper, we present several mechanisms that allow for a nearly optimal allocation of the tasks among the contractors and the determination of the compensation that is worthy for all the contractors. While none of the mechanisms presented in the paper are strategy-proof, it appears that the speculative power of the contractors is limited and decreases with the number of parties. We note that if the number of tasks and contractors is very large, the speculative power of the contractors is likely to diminish. In the future, it may be interesting to study the exact effect of these possible speculations on the allocation of the tasks.

The important message of this study is that, with a proper allocation mechanism, it is possible to reach a very efficient assignment of tasks to contractors even without a central planner that is aware of the exact cost structure of all the involved parties. In particular, we demonstrate that a sealed-bid partial combinatorial auction (SBPCA) leads to a very efficient solution. The computational complexity of this procedure may be excessively large, especially in a real-time setting. In such a case the company may reduce the size of the clusters it posts. Indeed, we see that even when the auction is reduced to a simple Vickrey auction (when the company posts one task at a time), the outcome is still relatively efficient. Another attractive option is to use a sequential negotiation mechanism, which is less demanding from a computational point of view and leaves the company with a fairly good control of the division of the profits between it and the contractors. In such a case, a higher level negotiation process between the parties is required to determine the parameters of the mechanism in the long run. When considering both computational viability, efficiency and degree of control of the company none of the offered mechanisms dominate the others. In practice, the companies should select a method and parameters that fit its situation the best. The mechanism itself may be an issue for a long run negotiation between the company and the contractors.

The DFSRP offers several interesting directions to pursue in future research. From a computational point of view, scaling up the solution methods presented in this paper, possibly by the introduction of heuristic methods, is required for real-world applications of the mechanisms. The introduction of additional features into our model is also required, e.g., adding time windows for the tasks. Another interesting detection is to formulate an on-line version of the problem in which requests for service arrive over time. The task exchange mechanisms presented in Section 3.2 can be adapted to this on-line setting. Finally, considering the operation of "a field service market" where many companies and contractors interact is also of interest.

References

[1] I. Beniaminy, D. Yelin, U. Zahavi and M. Zardin. When the rubber meets the road: Bio-inspired field service scheduling in the real world. In: F.B. Pereira and J. Tavares (Eds.) *Bio-inspired Algorithms for the Vehicle Routing Problem*, SCI 161: 191-213, Springer, Heidelberg, 2009.

[2] I. Beniaminy, Personal communication, 2013

[3] S. Binart, P. Dejax, M. Gendreau and F. Semet. A 2-stage method for a field service routing problem with stochastic travel and service times. *Computers & Operations Research*, 65: 64-75, 2016.

[4] C. Caplice and Y. Sheffi. Combinatorial Auctions for Truckload Transportation. In: P. Cramton et al. (Eds.) *Combinatorial Auctions*, Cambridge, MA, 1016-1069, 2006.

[5] C. E. Cortés, M. Gendreau, L. M. Rousseau, S. Souyris and A. Weintraub. Branch-and-price and constraint programming for solving a real-life technician dispatching problem. *European Journal of Operational Research*, 238, 300-312, 2014.

[6] E. Delage. Re-optimization of technician tours in dynamic environments with stochastic service time. Master's thesis, Ecole des Mines de Nantes, 2010.

[7] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 11: 17-33, 1971.

[8] T. Groves. Incentives in teams. *Econometrica*, 41: 617-631, 1973.

[9] G. Q. Huang and S. X. Xu. Truthful multi-unit transportation procurement auctions for logistics e-marketplaces. *Transportation Research Part B,* 47: 127-148, 2013.

[10] A. A. Kovacs, S. N. Parragh, K. E. Doerner and R. F. Hartl. Adaptive large neighborhood search for service technician routing and scheduling problems. *Journal of Scheduling*, 15(5): 579-600, 2012.

[11] O. Kolka, personal communication, 2017

[12] J. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7: 48-50, 1956.

[13] C.E. Miller, A.W. Tucker and R.A. Zemlin. Integer programming formulations and traveling salesman problems. J. Association for computing machinery, 7: 326-329, 1960.

[14] J.F. Nash. The Bargaining Problem. *Econometrica*, 18: 155-162, 1950.

[15] V. Pillac, C. Guéret, and A.L. Medaglia. A parallel matheuristic for the technician routing and scheduling problem. *Optimization Letters*, 7(7), 1525-1535, 2013.

[16] T. Sandholm. An implementation of the contract net protocol based on marginal cost calculations. *Proc. AAAI-93, Washington, DC*, pages 256–262, 1993.

[17] S. Souyris, C. E. Cortés, F. Ordóñez, and A. Weintraub. A robust optimization approach to dispatching technicians under stochastic service times. *Optimization Letters*, 7(7), 1549-1568, 2013.

[18] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16: 8-37, 1961.

[19] S. X. Xu and G. Q. Huang. Efficient auctions for distributed transportation procurement. *Transportation Research Part B,* 65: 47-64, 2014.

[20] E. Zamorano and R. Stolletz. Branch-and-price approaches for the Multiperiod Technician Routing and Scheduling Problem. *European Journal of Operational Research*, 257: 55-68, 2017.

[21] D. Zhenggang, C. Linning, and L. Zheng. Improved multi-agent system for the vehicle routing problem with time windows, *Tsinghua Science Technology*, 14(3): 407-412, 2009.

Appendix A: Lower bounds for the optimal cost for the system with outsourcing

Table 8: Lower bounds for optimal cost for the system with outsourcing

Instance	$\alpha = 1$	$\alpha = 3$
1	7,779.69	7,824.62
2	8,049.80	7,221.82*
3	6,947.65*	6,997.63
4	8,446.17	7,380.88*
5	8,779.09	7,742.64*
6	7,092.55	7,275.30
7	7,186.92	6,983.97
8	8,144.45	7,472.00*
9	7,498.11	6,747.02*
10	8,258.01	7,714.24*
11	7,360.57	6,498.73*
12	8,615.89	8,382.35
13	7,236.78	7,513.61
14	8,444.39	7,508.69
15	7,084.75	7,040.39*
16	8,243.94	7,804.34*
17	7,941.98	7,301.47*
18	8,652.95	7,820.74
19	8,217.27	7,839.24
20	8,143.69	8,025.45*

(*) optimal value

Appendix B: Total costs and price of decentralization for the Stage A allocations

Table 9: Total cost for the system where $\alpha = 1$ – Stage A

Instance number	SBPCA $n_{max} = 1$	SBPCA $n_{max} = 8$	SN $r_i = 100\% \times \frac{10}{3} \times s_i$	SN $r_i = 150\% \times \frac{10}{3} \times s_i$	HE	Lower Bound
1	8286.5	**8,236.02**	8,256.81	8266.58	8,731.69	7,779.69
2	8247.68	**8,233.19**	8,280.96	8334.31	8,788.40	8,049.80
3	**6947.65 ***	**6,947.65 ***	**6,947.65 ***	7309.8	7,734.98	6,947.65
4	**8754.4**	8,845.24	8,788.95	8906.53	9,559.13	8,446.17
5	9502.72	**9,398.03**	9,419.45	9439.62	10,040.73	8,779.09
6	**7135.19**	7,212.65	7,163.86	7214.22	7,743.36	7,092.55
7	7752.48	**7,632.09**	7,686.71	7839.27	8,234.74	7,186.92
8	8382.21	**8,301.97**	8,359.60	8487.59	8,669.23	8,144.45
9	8229.64	**8,073.02**	8,152.05	8179.99	8,392.12	7,498.11
10	**8375.01**	8,442.16	**8,375.01**	8454.86	8,704.11	8,258.01
11	7591.17	**7,564.64**	7,603.17	7590.47	7,854.78	7,360.57
12	8959.377	**8,845.67**	8,865.63	9049.52	9,377.40	8,615.89
13	7778.82	**7,645.04**	7,739.59	7753.32	8,464.54	7,236.78
14	8746.91	**8,698.67**	8,735.79	8821.91	9,386.67	8,444.39
15	7585.07	**7,429.68**	7,570.47	7871.63	7,626.25	7,084.75
16	8854.39	**8,849.32**	8,852.16	8850.17	9,738.38	8,243.94
17	8483.17	8,276.70	**8,147.31**	8404	8,236.28	7,941.98
18	8948.65	8,994.01	**8,945.85**	8999.72	9,639.17	8,652.95
19	8761.5	8,714.72	**8,704.65**	8802.92	9,312.26	8,217.27
20	8480.37	**8,400.98**	8,440.42	8428.7	8,835.62	8,143.69

Table 10: Total cost for the system where $\alpha = 3$ – Stage A

Instance number	SBPCA $n_{max} = 1$	SBPCA $n_{max} = 8$	SN $r_i = 100\% \times \frac{10}{3} \times s_i$	SN $r_i = 150\% \times \frac{10}{3} \times s_i$	HE	Lower Bound
1	8,071.56	**7,985.52**	8,025.02	8,027.67	8,401.74	7,824.62
2	*** 7,221.82 ***	**7,221.82 ***	7,250.86	7,287.23	7,912.04	7,221.82
3	7,089.41	7,179.43	7,158.86	**7,082.64**	7,781.13	6,997.63
4	7,404.30	**7,380.88 ***	7,404.30	7,465.77	8,027.10	7,380.88
5	**7,764.26**	7,821.84	7,796.30	7,894.23	8,189.30	7,742.64
6	7,447.29	**7,376.49**	7,550.65	7,592.08	7,646.72	7,275.30
7	7,232.81	**7,137.19**	7,171.33	7,239.81	7,814.13	6,983.97
8	7,510.27	**7,563.11**	7,595.83	7,914.25	7,896.91	7,472.00
9	**6,771.11**	6,780.42	6,792.17	6,890.50	7,026.31	6,747.02
10	7,792.16	7,821.31	7,897.59	8,071.67	**7,744.66**	7,714.24
11	**6,506.42**	6,542.59	6,538.46	6,773.85	6,972.02	6,498.73
12	8,594.27	**8,566.44**	8,607.65	8,714.79	9,001.12	8,382.35
13	7,847.28	**7,817.55**	7,845.06	7,823.25	8,526.62	7,513.61
14	**7,552.96**	7,571.87	7,576.37	7,598.47	7,847.23	7,508.69
15	*** 7,040.39 ***	**7,040.39 ***	7,079.17	7,616.42	7,136.32	7,040.39
16	7,852.01	**7,830.63**	7,837.26	8,315.46	8,093.60	7,804.34
17	7,357.13	7,353.42	**7,328.63**	7,381.37	7,380.08	7,301.47
18	8,133.10	**8,056.60**	8,070.49	8,244.09	8,095.92	7,820.74
19	8,281.50	8,216.80	**8,202.17**	8,218.23	8,560.44	7,839.24
20	8,201.28	**8,027.63**	8,076.99	8,255.83	8,178.86	8,025.45

Table 11: Prices of decentralization where $\alpha = 1$ – Stage A

Instance number	SBPCA $n_{max} = 1$	SBPCA $n_{max} = 8$	SN $r_i = 100\% \times \frac{10}{3} \times s_i$	SN $r_i = 150\% \times \frac{10}{3} \times s_i$	HE
1	6.12%	5.54%	5.78%	5.89%	10.90%
2	2.40%	2.23%	2.79%	3.41%	8.40%
3	0.00%	0.00%	0.00%	4.95%	10.18%
4	3.52%	4.51%	3.90%	5.17%	11.64%
5	7.61%	6.59%	6.80%	7.00%	12.57%
6	0.60%	1.67%	1.00%	1.69%	8.40%
7	7.30%	5.83%	6.50%	8.32%	12.72%
8	2.84%	1.90%	2.57%	4.04%	6.05%
9	8.89%	7.12%	8.02%	8.34%	10.65%
10	1.40%	2.18%	1.40%	2.33%	5.13%
11	3.04%	2.70%	3.19%	3.03%	6.29%
12	3.83%	2.60%	2.82%	4.79%	8.12%
13	6.97%	5.34%	6.50%	6.66%	14.50%
14	3.46%	2.92%	3.34%	4.28%	10.04%
15	6.60%	4.64%	6.42%	10.00%	7.10%
16	6.89%	6.84%	6.87%	6.85%	15.35%
17	6.38%	4.04%	2.52%	5.50%	3.57%
18	3.30%	3.79%	3.27%	3.85%	10.23%
19	6.21%	5.71%	5.60%	6.65%	11.76%
20	3.97%	3.06%	3.52%	3.38%	7.83%
Average	**4.57%**	**3.96%**	**4.14%**	**5.31%**	**9.57%**
Standard deviation	0.58%	0.45%	0.52%	0.50%	0.71%

Table 12: Prices of decentralization where $\alpha = 3$ – Stage A

Instance number	SBPCA $n_{max} = 1$	SBPCA $n_{max} = 8$	SN $r_i = 100\% \times \frac{10}{3} \times s_i$	SN $r_i = 150\% \times \frac{10}{3} \times s_i$	HE
1	3.06%	2.01%	2.50%	2.53%	6.87%
2	0.00%	0.00%	0.40%	0.90%	8.72%
3	1.29%	2.53%	2.25%	1.20%	10.07%
4	0.32%	0.00%	0.32%	1.14%	8.05%
5	0.28%	1.01%	0.69%	1.92%	5.45%
6	2.31%	1.37%	3.65%	4.17%	4.86%
7	3.44%	2.15%	2.61%	3.53%	10.62%
8	0.51%	1.20%	1.63%	5.59%	5.38%
9	0.36%	0.49%	0.66%	2.08%	3.97%
10	1.00%	1.37%	2.32%	4.43%	0.39%
11	0.12%	0.67%	0.61%	4.06%	6.79%
12	2.47%	2.15%	2.62%	3.81%	6.87%
13	4.25%	3.89%	4.23%	3.96%	11.88%
14	0.59%	0.83%	0.89%	1.18%	4.31%
15	0.00%	0.00%	0.55%	7.56%	1.34%
16	0.61%	0.34%	0.42%	6.15%	3.57%
17	0.76%	0.71%	0.37%	1.08%	1.07%
18	3.84%	2.93%	3.09%	5.14%	3.40%
19	5.34%	4.59%	4.42%	4.61%	8.42%
20	2.14%	0.03%	0.64%	2.79%	1.88%
Average	**1.63%**	**1.41%**	**1.74%**	**3.39%**	**5.70%**
Standard deviation	0.37%	0.30%	0.31%	0.44%	0.76%

Appendix C: Prices of decentralization for the Stage B allocations

Table 13: Prices of decentralization for $\alpha = 1$ – after Stage B

Instance number	SBPCA + PI $n_{max} = 1$	SBPCA+ CI $n_{max} = 1$	SBPCA+ PI $n_{max} = 8$	SBPCA+ CI $n_{max} = 8$	SN+ PI $r_i = 100\% \times \frac{10}{3} \times s_i$	SN+ CI $r_i = 100\% \times \frac{10}{3} \times s_i$	SN+ PI $r_i = 150\% \times \frac{10}{3} \times s_i$	SN+ CI $r_i = 150\% \times \frac{10}{3} \times s_i$	HE + PI	HE + CI
1	6.12%	4.98%	5.54%	5.35%	4.85%	4.77%	5.35%	4.24%	6.78%	4.24%
2	2.40%	2.40%	2.23%	2.23%	2.79%	2.79%	3.41%	2.50%	4.16%	3.76%
3	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	4.95%	3.05%	10.00%	2.10%
4	3.52%	3.52%	4.51%	3.92%	3.90%	3.52%	5.17%	4.43%	4.45%	3.50%
5	7.38%	6.48%	6.59%	6.44%	6.80%	6.42%	6.73%	6.32%	8.96%	8.30%
6	0.60%	0.60%	1.67%	1.45%	1.00%	1.00%	1.69%	1.69%	3.00%	1.84%
7	6.98%	5.84%	5.83%	5.66%	6.00%	5.55%	8.32%	7.13%	7.89%	5.81%
8	2.84%	2.84%	1.90%	1.90%	2.57%	2.57%	4.04%	3.07%	4.69%	3.39%
9	8.89%	7.94%	7.12%	7.07%	8.02%	7.43%	8.34%	6.48%	8.75%	7.57%
10	1.40%	1.40%	1.90%	1.43%	1.40%	1.40%	2.33%	1.63%	4.28%	1.43%
11	3.04%	2.58%	2.70%	1.24%	2.75%	2.75%	2.77%	1.99%	4.37%	2.31%
12	3.83%	3.83%	2.60%	2.01%	2.82%	2.74%	4.79%	2.92%	5.92%	4.48%
13	6.47%	6.47%	5.34%	5.08%	6.14%	5.83%	6.66%	6.14%	9.80%	5.50%
14	2.95%	2.66%	2.53%	2.53%	3.34%	2.41%	4.28%	2.57%	6.70%	2.71%
15	6.60%	4.75%	4.64%	4.64%	6.42%	5.32%	9.67%	6.12%	6.21%	4.93%
16	6.89%	6.89%	6.84%	6.84%	6.87%	6.85%	6.85%	6.83%	13.77%	8.83%
17	5.86%	3.50%	4.04%	3.60%	2.52%	2.52%	3.28%	2.59%	3.57%	3.20%
18	3.30%	3.30%	3.79%	3.70%	3.27%	3.27%	3.85%	3.38%	5.57%	3.62%
19	6.21%	5.54%	5.71%	5.71%	5.60%	5.12%	6.65%	5.13%	7.64%	5.83%
20	3.78%	3.71%	3.06%	3.01%	2.98%	2.91%	3.38%	3.01%	3.40%	2.38%
Average	**4.45%**	**3.96%**	**3.93%**	**3.69%**	**4.00%**	**3.76%**	**5.13%**	**4.06%**	**6.50%**	**4.29%**
Standard deviation	0.56%	0.49%	0.46%	0.48%	0.50%	0.47%	0.50%	0.43%	0.63%	0.49%

Table 14: Prices of decentralization for $\alpha = 3$ – after Stage B

Instance number	SBPCA + PI $n_{max} = 1$	SBPCA+ CI $n_{max} = 1$	SBPCA+ PI $n_{max} = 8$	SBPCA+ CI $n_{max} = 8$	SN+ PI $r_i = 100\% \times \frac{10}{3} \times s_i$	SN+ CI $r_i = 100\% \times \frac{10}{3} \times s_i$	SN+ PI $r_i = 150\% \times \frac{10}{3} \times s_i$	SN+ CI $r_i = 150\% \times \frac{10}{3} \times s_i$	HE + PI	HE + CI
1	3.06%	2.08%	2.01%	2.01%	1.70%	1.70%	2.53%	2.22%	5.34%	1.84%
2	0.00%	0.00%	0.00%	0.00%	0.40%	0.40%	0.90%	0.74%	6.10%	1.39%
3	1.29%	1.29%	2.53%	2.53%	2.25%	1.40%	1.20%	1.05%	3.80%	1.10%
4	0.32%	0.32%	0.00%	0.00%	0.32%	0.32%	1.14%	1.14%	3.49%	2.55%
5	0.00%	0.00%	0.42%	0.42%	0.69%	0.37%	1.92%	1.63%	5.45%	4.37%
6	2.31%	2.11%	1.37%	1.37%	3.65%	2.44%	4.17%	3.93%	3.87%	1.91%
7	3.44%	2.74%	2.15%	1.71%	2.61%	2.61%	3.53%	3.04%	4.42%	1.62%
8	0.51%	0.51%	1.20%	0.98%	1.63%	1.63%	5.59%	0.88%	4.12%	0.45%
9	0.36%	0.29%	0.49%	0.49%	0.66%	0.58%	2.08%	0.23%	2.40%	0.32%
10	1.00%	1.00%	1.37%	0.65%	2.32%	1.15%	3.31%	1.79%	0.39%	0.39%
11	0.12%	0.12%	0.67%	0.67%	0.61%	0.61%	1.05%	1.05%	5.19%	0.39%
12	1.74%	1.74%	2.15%	2.15%	2.47%	1.70%	3.81%	1.65%	2.50%	1.95%
13	4.25%	3.92%	3.59%	3.36%	3.45%	3.45%	3.96%	3.19%	8.57%	3.19%
14	0.59%	0.59%	0.83%	0.27%	0.39%	0.39%	1.18%	1.18%	1.19%	0.75%
15	0.00%	0.00%	0.00%	0.00%	0.55%	0.55%	4.38%	0.65%	0.59%	0.59%
16	0.61%	0.61%	0.34%	0.34%	0.42%	0.42%	1.13%	0.41%	2.85%	1.54%
17	0.76%	0.76%	0.71%	0.71%	0.37%	0.37%	1.08%	1.08%	1.07%	0.08%
18	3.84%	3.84%	2.93%	2.93%	3.09%	3.09%	5.14%	3.50%	3.40%	2.78%
19	4.79%	4.68%	4.59%	4.59%	4.42%	4.41%	4.61%	4.59%	5.37%	4.69%
20	2.14%	1.31%	0.03%	0.03%	0.00%	0.00%	2.79%	1.76%	0.26%	0.26%
Average	**1.56%**	**1.40%**	**1.37%**	**1.26%**	**1.60%**	**1.38%**	**2.78%**	**1.79%**	**3.52%**	**1.61%**
Standard deviation	0.36%	0.33%	0.30%	0.30%	0.31%	0.28%	0.36%	0.28%	0.50%	0.31%

Appendix D: Prices of decentralization for the Stage B allocations

Table 15: P-values for paired t-tests for the $\alpha = 1$ instances – Stage A

This (i, j) item in the table is the $P - value$ for $H_0: \mu_i - \mu_j = 0,\ H_1: \mu_i - \mu_j < 0$

	SBPCA $n_{max} = 8$	SN $r_i = 100\% \times \frac{10}{3} \times s_i$	SBPCA $n_{max} = 1$	SN $r_i = 150\% \times \frac{10}{3} \times s_i$	HE
SBPCA $n_{max} = 8$		0.15	$5 * 10^{-3}$	$2 * 10^{-4}$	$2.7 * 10^{-9}$
SN $r_i = 100\% \times \frac{10}{3} \times s_i$			0.02	$3 * 10^{-4}$	$4.5 * 10^{-9}$
SBPCA $n_{max} = 1$				0.01	$2.1 * 10^{-7}$
SN $r_i = 150\% \times \frac{10}{3} \times s_i$					$1.4 * 10^{-6}$

Table 16: P-values for paired t-tests for the $\alpha = 3$ instances – Stage A

This (i, j) item in the table is the $P - value$ for $H_0: \mu_i - \mu_j = 0,\ H_1: \mu_i - \mu_j < 0$

	SBPCA $n_{max} = 8$	SBPCA $n_{max} = 1$	SN $r_i = 100\% \times \frac{10}{3} \times s_i$	SN $r_i = 150\% \times \frac{10}{3} \times s_i$	HE
SBPCA $n_{max} = 8$		0.1	0.01	$3 * 10^{-4}$	$1.2 * 10^{-6}$
SBPCA $n_{max} = 1$			0.32	10^{-3}	$4.9 * 10^{-6}$
SN $r_i = 100\% \times \frac{10}{3} \times s_i$				$8 * 10^{-4}$	$5.6 * 10^{-6}$
SN $r_i = 150\% \times \frac{10}{3} \times s_i$					0.01

Table 17: P-values for paired t-tests for the α = 1 instances – SBPCA and $n_{max} = 8$

$H_0: \mu_{SBPCA+CI} - \mu_{SBPCA+PI} = 0$ $H_1: \mu_{SBPCA+CI} - \mu_{SBPCA+PI} < 0$ $P - value = 3 * 10^{-3}$	$H_0: \mu_{SBPCA+CI} - \mu_{SBPCA} = 0$ $H_1: \mu_{SBPCA+CI} - \mu_{SBPCA} < 0$ $P - value = 2 * 10^{-3}$	$H_0: \mu_{SBPCA+PI} - \mu_{SBPCA} = 0$ $H_1: \mu_{SBPCA+PI} - \mu_{SBPCA} < 0$ $P - value = 0.08$

Table 18: P-values for paired t-tests for the α = 3 instances – SBPCA and $n_{max} = 8$

$H_0: \mu_{SBPCA+CI} - \mu_{SBPCA+PI} = 0$ $H_1: \mu_{SBPCA+CI} - \mu_{SBPCA+PI} < 0$ $P - value = 0.02$	$H_0: \mu_{SBPCA+CI} - \mu_{SBPCA} = 0$ $H_1: \mu_{SBPCA+CI} - \mu_{SBPCA} < 0$ $P - value = 7 * 10^{-3}$	$H_0: \mu_{SBPCA+PI} - \mu_{SBPCA} = 0$ $H_1: \mu_{SBPCA+PI} - \mu_{SBPCA} < 0$ $P - value = 0.09$

Table 19: P-values for paired t-tests for the α = 1 instances – SBPCA and $n_{max} = 1$

$H_0: \mu_{SBPCA+CI} - \mu_{SBPCA+PI} = 0$ $H_1: \mu_{SBPCA+CI} - \mu_{SBPCA+PI} < 0$ $P - value = 2.3 * 10^{-3}$	$H_0: \mu_{SBPCA+CI} - \mu_{SBPCA} = 0$ $H_1: \mu_{SBPCA+CI} - \mu_{SBPCA} < 0$ $P - value = 1.3 * 10^{-3}$	$H_0: \mu_{SBPCA+PI} - \mu_{SBPCA} = 0$ $H_1: \mu_{SBPCA+PI} - \mu_{SBPCA} < 0$ $P - value = 8.2 * 10^{-3}$

Table 20: P-values for paired t-tests for the α = 3 instances – SBPCA and $n_{max} = 1$

$H_0: \mu_{SBPCA+CI} - \mu_{SBPCA+PI} = 0$ $H_1: \mu_{SBPCA+CI} - \mu_{SBPCA+PI} < 0$ $P - value = 0.02$	$H_0: \mu_{SBPCA+CI} - \mu_{SBPCA} = 0$ $H_1: \mu_{SBPCA+CI} - \mu_{SBPCA} < 0$ $P - value = 3 * 10^{-3}$	$H_0: \mu_{SBPCA+PI} - \mu_{SBPCA} = 0$ $H_1: \mu_{SBPCA+PI} - \mu_{SBPCA} < 0$ $P - value = 0.054$

Table 21: P-values for paired t-tests for the α = 1 instances – SN and $r_i = 100\% \times \frac{10}{3} \times s_i$

$H_0: \mu_{SN+CI} - \mu_{SN+PI} = 0$ $H_1: \mu_{SN+CI} - \mu_{SN+PI} < 0$ $P - value = 2 * 10^{-3}$	$H_0: \mu_{SN+CI} - \mu_{SN} = 0$ $H_1: \mu_{SN+CI} - \mu_{SN} < 0$ $P - value = 2 * 10^{-4}$	$H_0: \mu_{SN+PI} - \mu_{SN} = 0$ $H_1: \mu_{SN+PI} - \mu_{SN} < 0$ $P - value = 0.02$

Table 22: P-values for paired t-tests for the α = 3 instances – SN and $r_i = 100\% \times \frac{10}{3} \times s_i$

$H_0: \mu_{SN+CI} - \mu_{SN+PI} = 0$ $H_1: \mu_{SN+CI} - \mu_{SN+PI} < 0$ $P - value = 0.02$	$H_0: \mu_{SN+CI} - \mu_{SN} = 0$ $H_1: \mu_{SN+CI} - \mu_{SN} < 0$ $P - value = 8 * 10^{-4}$	$H_0: \mu_{SN+PI} - \mu_{SN} = 0$ $H_1: \mu_{SN+PI} - \mu_{SN} < 0$ $P - value = 0.02$

Table 23: P-values for paired t-tests for the $\alpha = 1$ instances – SN and $r_i = 150\% \times \frac{10}{3} \times s_i$

$H_0: \mu_{SN+CI} - \mu_{SN+PI} = 0$	$H_0: \mu_{SN+CI} - \mu_{SN} = 0$	$H_0: \mu_{SN+PI} - \mu_{SN} = 0$
$H_1: \mu_{SN+CI} - \mu_{SN+PI} < 0$	$H_1: \mu_{SN+CI} - \mu_{SN} < 0$	$H_1: \mu_{SN+PI} - \mu_{SN} < 0$
$P - value = 6.2 * 10^{-3}$	$P - value = 5.2 * 10^{-6}$	$P - value = 0.06$

Table 24: P-values for paired t-tests for the $\alpha = 3$ instances – SN and $r_i = 150\% \times \frac{10}{3} \times s_i$

$H_0: \mu_{SN+CI} - \mu_{SN+PI} = 0$	$H_0: \mu_{SN+CI} - \mu_{SN} = 0$	$H_0: \mu_{SN+PI} - \mu_{SN} = 0$
$H_1: \mu_{SN+CI} - \mu_{SN+PI} < 0$	$H_1: \mu_{SN+CI} - \mu_{SN} < 0$	$H_1: \mu_{SN+PI} - \mu_{SN} < 0$
$P - value = 2 * 10^{-3}$	$P - value = 2 * 10^{-3}$	$P - value = 0.04$

Table 25: P-values for paired t-tests for the $\alpha = 1$ instances – heuristic allocation

$H_0: \mu_{HE+CI} - \mu_{HE+PI} = 0$	$H_0: \mu_{HE+CI} - \mu_{HE} = 0$	$H_0: \mu_{HE+PI} - \mu_{HE} = 0$
$H_1: \mu_{HE+CI} - \mu_{HE+PI} < 0$	$H_1: \mu_{HE+CI} - \mu_{HE} < 0$	$H_1: \mu_{HE+PI} - \mu_{HE} < 0$
$P - value = 1.4 * 10^{-5}$	$P - value = 2 * 10^{-9}$	$P - value = 6.25 * 10^{-7}$

Table 26: P-values for paired t-tests for the $\alpha = 3$ – heuristic allocation

$H_0: \mu_{HE+CI} - \mu_{HE+PI} = 0$	$H_0: \mu_{HE+CI} - \mu_{HE} = 0$	$H_0: \mu_{HE+PI} - \mu_{HE} = 0$
$H_1: \mu_{HE+CI} - \mu_{HE+PI} < 0$	$H_1: \mu_{HE+CI} - \mu_{HE} < 0$	$H_1: \mu_{HE+PI} - \mu_{HE} < 0$
$P - value = 4.58 * 10^{-5}$	$P - value = 2.7 * 10^{-6}$	$P - value = 4.84 * 10^{-5}$

Appendix E: Payments to the contractors

Table 27: P-values for paired t-tests for the $\alpha = 1$ instances – Stage A

This (i, j) item in the table is the $P - value$ for $H_0: \mu_i - \mu_j = 0,\ H_1: \mu_i - \mu_j > 0$

	SBPCA $n_{max} = 1$	SBPCA $n_{max} = 8$	SN $r_i = 150\% \times \frac{10}{3} \times s_i$	SN $r_i = 100\% \times \frac{10}{3} \times s_i$
SBPCA $n_{max} = 1$		$3.6 * 10^{-3}$	10^{-3}	$6.3 * 10^{-8}$
SBPCA $n_{max} = 8$			$1.4 * 10^{-3}$	$2 * 10^{-11}$
SN $r_i = 150\% \times \frac{10}{3} \times s_i$				$2.7 * 10^{-12}$

Table 28: P-values for paired t-tests for the $\alpha = 3$ instances – Stage A

This (i, j) item in the table is the $P - value$ for $H_0: \mu_i - \mu_j = 0,\ H_1: \mu_i - \mu_j > 0$

	SBPCA $n_{max} = 1$	SBPCA $n_{max} = 8$	SN $r_i = 150\% \times \frac{10}{3} \times s_i$	SN $r_i = 100\% \times \frac{10}{3} \times s_i$
SBPCA $n_{max} = 1$		$2.8 * 10^{-6}$	$5.2 * 10^{-9}$	$1.1 * 10^{-11}$
SBPCA $n_{max} = 8$			$6.5 * 10^{-10}$	$3.8 * 10^{-15}$
SN $r_i = 150\% \times \frac{10}{3} \times s_i$				$4.2 * 10^{-12}$