Customized multi-period stochastic assignment problem for social engagement and opportunistic IoT

Customized Multi-period Stochastic Assignment Problem for Social Engagement and Opportunistic IoT

Edoardo Fadda

ICT for City Logistics and Enterprises Lab - DAUIN, Politecnico di Torino, Turin, Italy

Guido Perboli

ICT for City Logistics and Enterprises Lab - DAUIN, Politecnico di Torino, Turin, Italy and CIRRELT, Montreal, Canada

Roberto Tadei

DAUIN, Politecnico di Torino, Turin, Italy

Abstract

An enormous number of devices are currently available to collect data. One of the main applications of these devices is in the urban environment, where they can collect data useful for improving the management of different operations. This is the main goal of smart cities. To gather these data from devices, companies can build expensive networks able of reaching every part of the city or they can use cheaper alternatives as opportunistic connections, i.e., use the devices of selected people (e.g., mobile users) as mobile hotspots in exchange for a reward. In this paper, we consider this second choice and, in particular, we solve the problem of minimizing the sum of the rewards while providing the connectivity to all sensors. We show that the stochastic approach must be considered since deterministic solutions produce considerable waste. Finally, to reduce the computational time we apply the loss of reduced costs-based variable fixing (LRCVF) heuristic and we compare, by means of computational tests, the performances of the heuristic and a commercial solver. The results prove the effectiveness of the LRCVF heuristic.

Keywords: `Internet of Things`, `Multi-period Stochastic Assignment Problem`

1. Introduction

Many companies currently use business models based on social engagement, i.e., they use smartphone applications to ask people to perform tasks to reach a business goal. In exchange for this service, the companies offer a reward to the people that perform the tasks. Some examples of this model are crowd shipping and the opportunistic Internet of things (IoT). Crowd shipping is a new business model to perform last-mile logistics: the companies that use this delivery method ask people to take a package from one point of a city to another. In this way, the company can save a lot of money because it can decrease the number of vehicles used and the number of drivers to hire. Instead, opportunistic IoT tries to solve the problem of gathering data from a distributed network of sensors in an urban area in the case of insufficient coverage by hubs and hotspots. In this case, it is possible to collect data by using the devices of selected people (e.g., the mobile users) as mobile hotspots. This application is more critical than the previous application. In fact, while crowd shipping reduces the costs of logistics, gathering data from these devices is nearly impossible without the proposed opportunistic connections because it would require a huge network.

The opportunistic IoT was the inspiration for the Coiote project by TIM (Telecom Italia Mobile) and the ICT (Information and Communication Technology) for City Logistics and Enterprises Lab of Politecnico di Torino (TIM Jol Swarm, 2016). The goal of this project is to develop a mobile phone application that enables TIM to ask users to perform some tasks in relation to the mobile phone cell where the users are located. The task that the users are asked to perform is to share their Internet connection with the dumpsters. In this way, the dumpsters can transmit to the central unit the data regarding the amount of waste that they have collected and the company in charge of the waste collection can plan the operations in an optimal way. In exchange for the Internet connection that the users share with the dumpster, TIM offers a reward. The characteristic of asking people to execute tasks in exchange for a reward is common to all social engagement business models. As an example, in the e-grocery domain, Walmart (a grocery retailers) asks in-store shoppers to carry packages to on-line shoppers in exchange for a discount. The main objective of this paper is to define a mathematical model suitable to help companies that use social engagement. The objective of this model is to minimize the total costs of the rewards that the company must pay while maximizing the number of tasks performed by the users. Therefore, the topic of this paper is far more general than the Coiote project and it embraces many social engagement business models. To the best of the authors knowledge, this is the first time that such a problem has been presented. The model that describes the problem is a multi-period stochastic assignment problem (MPSAP). The computational experiments that we perform show that the exact method has poor performance in real test instances. For this reason, we use the loss of reduced costs-based variable fixing (LRCVF) heuristic.

This article is organized as follows. In Section 2, we review the literature about the IoT and the MPSAP. In Section 3, we present the stochastic model and we describe the more suitable stochastic distributions that should be used. In Section 4, we use a set of benchmark instances to study the stability of the problem, the value of stochastic solution (VSS), and the performances of the heuristic LRCVF. Finally, in Section 5, we outline the main results achieved in this paper.

2. Literature Review

The main topic of this paper is the application of optimization techniques to social engagement. Since, to the best of the authors' knowledge, there are no previous studies on this topic, we split the literature review in two: the first part considers the application of optimization techniques to the IoT applications, the second considers the optimization problem class that is more similar to our problem, multi-period stochastic assignment.

IoT can be thought as the enrichment of objects with sensors and with the capacity to exchange data. If these objects are also enriched with actuators, then the range of applications increases and encompasses also smart grids, intelligent transportation, and smart cities (for a survey of these applications the user is referred to Kaur and Kalra (2016), Qureshi and Abdullah (2013), and Zanella et al. (2014), respectively). Optimization plays an important role in IoT technology. For example, in Cagliano et al. (2014) the authors propose an intelligent transportation system that uses information from a network of sensors to improve route planning for vehicles and in Perboli et al. (2016) the authors describe an heuristic that optimizes waste collection operations using data regarding the waste production collected from vehicles. These two articles are examples of the multitude of applications.

The second part of the analysis considers optimization problems. In particular, the optimization problem that we consider is a special case of the assignment problem. All assignment problems have two features in common: tasks to be performed and resources to be allocated to each task. In our setting the tasks are the collection of data regarding the amount of waste in each dumpster, while the resources are the application users. The stochastic behavior of the problem comes from the uncertainty related to the amount of available resources. Finally, since we consider the problem in various time periods, this is also a multi-period problem. For these reasons, the problem that we consider is a MPSAP. Unfortunately, the literature on this problem is not very developed. The main references to similar problems are Klibi et al. (2010) and Pironet Thierry (2015). In both articles, the authors minimize the assignment cost of a fleet of vehicles to different tasks, the number of which is stochastic. As the reader may notice, the main difference between these papers and our application is that in our setting, the amounts of resources are uncertain and the number of tasks is deterministic, while in those papers the opposite is true. Furthermore, we consider a finite time horizon while the other papers consider an infinite horizon. Finally, we model the problem as a two-stage problem while the other papers consider a multi-stage problem.

The procedure that we follow to justify the solution of the stochastic model as well as the stability concepts is explained by Birge and Louveaux (1997) and Kaut et al. (2007). The LRCVF heuristic that we propose has been described for the first time in Maggioni et al. (2017). It consists of solving the continuous relaxation of the integer problem, and then fixing to the lower bound all the variables that have a reduced cost that is higher than a threshold. Usually this threshold is a quantile of the reduced costs. In this way, it is possible to control the number of variables that need to be fixed.

3. Stochastic Mathematical Problem

In this section, we present the mathematical model of the aforementioned problem. We present the general problem and we model it a as stochastic problem. We prove that it is important to consider this formulation and not the standard formulation in Section 4.2.

Let us consider a company that wants to collect data from a network of sensors distributed in a city. Owing to the high number of sensors and their low density, building a physical infrastructure to collect the data is not a feasible solution. For this reason, the company wants to use opportunistic IoT, i.e., to use the connection that some users share with the dumpsters. In exchange for this service, the company offers a small reward to the participants. Our objective is then to minimize the total amount of reward offered by visiting all the dumpsters.

Before introducing the mathematical model, we describe the flows of time and information that characterize our problem. In the following, we use the term *stage* when we are referring to the information flow, while we use the term *time step* when we are referring to the time flow. During the first stage, the company is asked to send messages to all target application users to ask them to connect to one or more sensors. In the following stage, the company observes the number of users that accept the tasks and can ask other users to connect to the sensors that were not covered in the first stage. This is the recourse action of the company. For each time step we have the same problem and we have to connect to all sensors before the time limit T. The mathematical model that describes the problem uses three sets: $\mathcal{T}$, the set of all time indexes, the cardinality of which is T; $\mathcal{I}$, the set of all cells, the cardinality of which is I; and $\mathcal{M}$, the set of all users types, the cardinality of which is M. Here $\mathcal{M}$ models the user availability to perform more tasks and the related price. For example, we can model a type of user that performs one task in exchange for a reward with a low probability or another type that for a slightly higher reward performs two tasks with more reliability, and so on. This set is useful to model standard workers, i.e., the more expensive customer can perform many tasks that are certainly available.

The model uses the following parameters (some are random variables):

- n_m is the number of tasks that a customer of type m executes;
- c_{ij}^{tm} is the cost of the reward for a customer of type m in cell i at time t that goes into cell j to execute n_m tasks;
- q_{ij}^{tm} is the cost of the reward for a customer of type m in cell i at time t, during the second stage, that goes into cell j to execute n_m tasks;
- N_k is the number of tasks that must be performed in the operational cell k before time T;
- $\theta_i^{tm}(\omega)$ is the number of customers of type m in cell i during the second stage of time step t;
- $\hat{\theta}_i^{tm}$ is the expected number of customers of type m in cell i during time step t; in particular, if the computational time is lower than the time step considered in the optimization,

$\hat{\theta}_i^{tm}|_{t=0}$ are the numbers of people in each cell when the algorithm is started.

The variables used are:

- x_{ij}^{tm}, the number of customers of type m that are asked to perform n_m tasks in cell j, starting from i at time t;
- $y_{ij}^{tm}(\omega)$, the number of customers of type m that are asked to perform n_m tasks in cell j, starting from i at time t;
- $z_{ij}^{tm}(\omega)$, a general variable that has the role of adjusting the first stage request x_{ij}^{tm} if it is greater than the real value $\theta_i^{tm}(\omega)$.

The general stochastic model is

$$\begin{aligned}\text{minimize} \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{m=1}^{M} c_{ij}^{tm} x_{ij}^{tm} + \\ \mathbb{E}\Big[\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{m=1}^{M} q_{ij}^{tm} y_{ij}^{tm}(\omega)\Big]\end{aligned} \tag{1}$$

subject to

$$\begin{aligned}\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{i=1}^{I} n_m(x_{ik}^{tm}+y_{ij}^{tm}(\omega) - z_{ij}^{tm}(\omega)) \geq N_k \\ \forall\ k \in \mathcal{I} \ \ \forall\ s \in \mathcal{S}\end{aligned} \tag{2}$$

$$\sum_{j=1}^{J} x_{ij}^{tm} \leq \hat{\theta}_i^{tm} \quad \forall\ i \in \mathcal{I}\ t \in \mathcal{T}\ m \in \mathcal{M} \tag{3}$$

$$\begin{aligned}\sum_{j=1}^{J}(x_{ij}^{tm} + y_{ij}^{tm}(\omega) - z_{ij}^{tm}(\omega)) \leq \theta_i^{tm}(\omega) \\ \forall\ i \in \mathcal{I}\ t \in \mathcal{T}\ m \in \mathcal{M}\end{aligned} \tag{4}$$

$$x_{ij}^{tm} \in \mathbb{N} \quad \forall\ i \in \mathcal{I}\ j \in \mathcal{J}\ t \in \mathcal{T}\ m \in \mathcal{M}$$

$$y_{ij}^{tm}(\omega) \in \mathbb{N} \quad \forall\ i \in \mathcal{I}\ j \in \mathcal{J}\ t \in \mathcal{T}\ m \in \mathcal{M}\ \forall\ \omega$$

$$z_{ij}^{tm}(\omega) \in \mathbb{N} \quad \forall\ i \in \mathcal{I}\ j \in \mathcal{J}\ t \in \mathcal{T}\ m \in \mathcal{M}\ \forall\ \omega$$

As in the stochastic model, TIM is only supposed to implement the decisions of the first stage of the first time step ($x_{ij}^{0m}\ \ \forall\ i, j, m$). Once these decisions are implemented, TIM has to run the computation again to implement the new decisions of the first stage of the first time step, and continue in this manner. This fact is crucial because then the model can avoid considering some stochastic behavior that otherwise would produce an explosion in the dimension of the problem (see Remark 3.2).

Remark 3.1. *We consider a two-stage stochastic model instead of a multi-period model even if the information flow is not consistent with this choice. This simplification is due to the fact that in the real world the company implements only the solution of the first time step and both the two-stages and the multi-stage structures are used to account for the future. Further, it is common knowledge that multi-stage problems are more difficult to solve than two-stage problems because they need more variables. Hence, because our problem already needs many variables, we consider the two-stage model to be a reasonable approximation of the future. In other words, we prefer a more precise estimation of an incorrect future than a very imprecise estimation of the real future.*

Remark 3.2. *In the model, we consider that all requests are accepted and that all users perform the assigned tasks. This assumption simplifies the model, but is acceptable. In fact, as stated above, the real decisions that the company implements are described by the first-period, first-stage variables. Hence, all the information about the number of tasks performed will be considered by the problem solved in the next run of the algorithm.*

To define the model, we still have to describe the distribution of $\theta_i^{tm}(w)$. If we consider each single person, for each time instant we can define a set of random variables X_{ip}, such that each

$$X_{ip} = \begin{cases} 1, & \text{if person } p \text{ is in the cell } i, \\ 0, & \text{otherwise.} \end{cases}$$

We consider these variables to be independent because it is a reasonable simplification. In fact, TIM can filtrate users with similar behavior; furthermore, we are not considering the entire population, only the app users. From these assumptions, it follows that $\theta_i^{\cdot\cdot}(w) = \sum_p X_{ip}$ and the distribution of $\theta_i^{\cdot\cdot}(w)$ is a Poisson binomial distribution (the distribution of a sum of Bernoulli random variables with different probabilities). In an urban context, such as that considered in this study, the number of people can be huge, hence we can use some asymptotic result by using a version of the central limit theorem (CLT) for non-identically distributed random variables. In particular, it is possible to apply the Lyapunov CLT (Billingsley, 1995). To introduce this result we need the following definition.

Definition 3.3. *If a sequence of independent random variables* $\{X_1, X_2, \ldots\}$ *is such that* $\mathbb{E}[X_i] = \mu_i < \infty$, $\mathbb{E}[(X - \mu_i)^2] = \sigma_i^2 < \infty$ *and for some* $\delta > 0$,

$$\lim_{n\to\infty} \frac{1}{s_n^{2+\delta}} \sum_{i=1}^{n} \mathbb{E}\left[\,|X_i - \mu_i|^{2+\delta}\,\right] = 0 \tag{5}$$

where $s_n^2 = \sum_{i=1}^{n} \sigma_i^2$*, then the Lyapunovs condition holds for* $\{X_1, X_2, \ldots\}$.

This definition is useful for the following theorem.

Theorem 3.4. *If the Lyapunovs condition holds, it can be proved that*

$$\frac{1}{s_n} \sum_{i=1}^{n} (X_i - \mu_i) \xrightarrow{d} \mathcal{N}(0, 1).$$

We omit the proof of this result (Billingsley, 1995). By this theorem we can derive the following corollary.

Corollary 3.5. *If* $\{X_1, X_2, \ldots\}$ *is a set of Bernoulli's random variables* $X_k \sim \mathcal{B}(p_k)$ *such that* $p_k \neq 0, 1$ *for all* k*, then*

$$\frac{\sqrt{n}\frac{1}{n}\sum_{i=1}^{n}(X_i - p_i)}{\sqrt{\sum_{i=1}^{n} p_i(1-p_i)/n}} \xrightarrow{d} \mathcal{N}(0, 1).$$

Proof We observe that for each $X_k \sim \mathcal{B}(p_k)$, it holds that

$$1 \geq p_k(1 - p_k) = \mathbb{E}[(X_k - p_k)^2] \geq \mathbb{E}[(X_k - p_k)^{2+\delta}].$$

Hence,

$$\frac{1}{s_n^{2+\delta}} \sum_{k=1}^{n} \mathbb{E}[(X_k - p_k)^{2+\delta}] \leq \frac{1}{s_n^{2+\delta}} \sum_{k=1}^{n} \mathbb{E}[(X_k - p_k)^2] \leq \frac{1}{s_n^{\delta}}.$$

If p_k is not zero or one (it is guaranteed by the hypothesis), then $s_n^\delta \to +\infty$, the Lyapunov's condition holds and we can apply Lyapunov's CLT.

Remark 3.6. *In Corollary 3.5, we have considered that* $p_k \neq 0, 1\ \forall k$*. This assumption is not strict because if* $p_k = 1$*, we are considering a person that is certainly in cell* k*, while if* $p_k = 0$ *we are considering a person that is certainly not in cell* k*. Both cases are not good model choices because of Cromwell's rule (see Lindley, 1991).*

Owing to this result and since we are considering a crowded environment, we can simulate the number of people in a node by using a normal distribution. This result gives us a distribution to use for the simulation of the number of people in a cell. Furthermore, given data about the number of people in a cell in a certain hour, we can fit these values by using a normal distribution. Finally, the properties of the normal distribution have an advantage in chance constrained models.

3.1. Linear stochastic model

The stochastic problem (1)–(4) can be expressed as a large-scale linear program by using the set of scenarios $\mathcal{S}$, the cardinality of which is S. The problem is then

$$\text{minimize} \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{m=1}^{M} c_{ij}^{tm} x_{ij}^{tm} + \sum_{s=1}^{S}\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{m=1}^{M} q_{ij}^{tm} y_{ij}^{stm} \tag{6}$$

subject to

$$\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{i=1}^{I} n_m(x_{ik}^{tm} + y_{ij}^{stm} - z_{ij}^{stm}) \geq N_k \quad \forall\ k\ \in \mathcal{I}\ \forall\ s\ \in \mathcal{S} \tag{7}$$

$$\sum_{j=1}^{J} x_{ij}^{tm} \leq \hat{\theta}_i^{tm} \quad \forall\ i\ \in \mathcal{I}\ t\ \in \mathcal{T}\ m\ \in \mathcal{M} \tag{8}$$

$$\sum_{j=1}^{J} (x_{ij}^{tm} + y_{ij}^{stm} - z_{ij}^{stm}) \leq \theta_i^{stm} \quad \forall\ i\ \in \mathcal{I}\ t\ \in \mathcal{T}\ m\ \in \mathcal{M}\ s\ \in \mathcal{S} \tag{9}$$

$$x_{ij}^{tm} \in \mathbb{N} \quad \forall\ i\ \in \mathcal{I}\ j\ \in \mathcal{J}\ t\ \in \mathcal{T}\ m\ \in \mathcal{M}$$

$$y_{ij}^{stm} \in \mathbb{N} \quad \forall\ i\ \in \mathcal{I}\ j\ \in \mathcal{J}\ t\ \in \mathcal{T}\ m\ \in \mathcal{M}\ \forall\ s\ \in \mathcal{S}$$

$$z_{ij}^{stm} \in \mathbb{N} \quad \forall\ i\ \in \mathcal{I}\ j\ \in \mathcal{J}\ t\ \in \mathcal{T}\ m\ \in \mathcal{M}\ \forall\ s\ \in \mathcal{S}$$

The objective function of the problem is the sum of the first stage rewards and the expected rewards of the second stage. Constraints (7) impose that all tasks must be performed. Instead, constraints (8) limit the first-stage users and (9) limit the second-stage users. Note that in each cell we consider that there are tasks to perform or there are users, but not both together. This hypothesis is due to the fact that the price for assigning tasks in the same cell of the users can be considered negligible. We call all the cells in which $N_k = 0$ sources and we call all the cells in which $N_k > 0$ sinks. The problem is a two-stage multi-period linear integer stochastic problem. Since the recourse matrix does not depend on the scenario realization, the problem is a fixed recourse problem. Further, the z_{ij}^{stm} variables ensure that for every possible choice of x_{ij}^{tm} such that $\sum_{j=1}^{J} x_{ij}^{tm} \leq \hat{\theta}_i^{tm}$, the second-stage problem has at least one feasible solution. Hence, by using these variables we have a complete recourse problem, i.e., for each feasible solution of the first-stage variables, there exists at least a second stage feasible solution. Note that if we remove z_{ij}^{stm}, then constraints (8) are useless, because to have an always feasible second stage, $x_{ij}^{tm} \leq \min_{s\in\mathcal{S}} \theta_i^{stm}$. Furthermore, at the optimum the solution must be such that

$$z_{ij}^{stm} y_{ij}^{stm} = 0. \tag{10}$$

The choice of the number of scenarios is related to the stability of the solution of the problem. In particular, to decide the number of scenarios to use, we compute two values: in-sample stability and out-of-sample stability (for an in-depth discussion about these values the reader is referred to Birge and Louveaux (1997) and Kaut et al. (2007)).

In-sample stability checks whether, given two scenario trees ($\mathcal{T}_i$ and $\mathcal{T}_j$), the optimal values of the objective functions of the problems that consider that scenarios are nearly the same, i.e., $\hat{f}(x_j^*, \mathcal{T}_j) \approx \hat{f}(x_i^*, \mathcal{T}_i)$, where $\hat{f}(x_j^*, \mathcal{T}_j)$ is the expected value of the objective function computed by using the scenario tree $\mathcal{T}_j$ ($\hat{f}(x_j^*, \mathcal{T}_j) = \mathbb{E}_{\mathcal{T}}[f(x_j^*)]$), f is the objective function, and x_i^* is the optimal solution of the problem that considers scenario tree $\mathcal{T}_i$ (note that x_i is the generic solution of an optimization problem and it is not related to the x_{ij}^{tm} of our model). To have a relative measure of $\hat{f}(x_j^*, \mathcal{T}_j) - \hat{f}(x_i^*, \mathcal{T}_i)$ we consider the in-sample relative stability to be

$$\frac{\hat{f}(x_j^*, \mathcal{T}_j) - \hat{f}(x_i^*, \mathcal{T}_i)}{(\hat{f}(x_j^*, \mathcal{T}_j) + \hat{f}(x_i^*, \mathcal{T}_i))/2}. \tag{11}$$

Out-of-sample stability checks whether different solutions have the same performance when tested with the real distribution, i.e., $\hat{f}(x_j^*, \omega) \approx \hat{f}(x_i^*, \omega)$ where $\hat{f}(x_i^*, \omega) = \mathbb{E}_P[f(x_i^*)]$ and P is the real probability distribution. Since the test of this quantity for all possible realizations is not feasible we compute these values by using 1000 different scenarios. As above, we obtain a relative measure of $\hat{f}(x_j^*, \omega) - \hat{f}(x_i^*, \omega)$ by considering the out-of-sample relative stability as

$$\frac{\hat{f}(x_j^*, \omega) - \hat{f}(x_i^*, \omega)}{(\hat{f}(x_j^*, \omega) + \hat{f}(x_i^*, \omega))/2}. \tag{12}$$

As the reader may notice, the use of the scenarios tree increases the number of variables and the complexity of the model. Hence, we have to justify our effort to solve the stochastic version of the problem instead of some easier formulations. The easiest method to prove that the effort is reasonable is to compute the VSS; see Birge and Louveaux (1997) for more details.

Definition 3.7. *The value of stochastic solution is computed as $VSS = EVS - RP$, where RP is the value of the recursive solution and EVS is the value of the expected value solution as described in Birge and Louveaux (1997).*

The VSS value represents how much the expected problem is worse than the solution of the stochastic problem. We define the relative VSS as

$$VSS_r = \frac{EVS - RP}{RP}.$$

If this value is below a threshold, then solving the stochastic problem does not produce any advantage. If this value is above a threshold it justifies the effort in solving the stochastic problem.

We compute these quantities for some benchmark instances in Section 4.

3.2. LRCVF-based Heuristic

Let us consider the following mathematical model, which represents a general formulation of a stochastic program in which a decision maker needs to determine x in order to minimize (expected) costs or outcomes (Maggioni et al., 2017):

$$\min_{x\in X} E_{\boldsymbol{\xi}} z(x, \boldsymbol{\xi}) = \min_{x\in X} \left\{ f_1(x) + E_{\boldsymbol{\xi}} \left[h_2(x, \boldsymbol{\xi}) \right] \right\}, \tag{13}$$

where x is a first-stage decision vector restricted to the set $X \subseteq \mathbb{R}^n_+$, $\mathbb{R}^n_+$ is the set of non-negative real vectors of dimension n, and $E_{\boldsymbol{\xi}}[\cdot]$ denotes the expectation with respect to a random vector $\boldsymbol{\xi}$, defined on some probability space $(\Omega, \mathscr{A}, p)$ with support Ω and given probability distribution p on the σ-algebra $\mathscr{A}$. The function h_2 is the value function of another optimization problem defined as

$$h_2\left(x, \xi\right) = \min_{y \in Y(x,\xi)} f_2\left(y; x, \xi\right), \tag{14}$$

which is used to reflect the costs associated with adapting to information revealed through a realization ξ of the random vector $\boldsymbol{\xi}$. The term $E_{\boldsymbol{\xi}}\left[h_2\left(x, \boldsymbol{\xi}\right)\right]$ in (13) is referred to as the recourse function. In this paper we assume that functions f_1 and f_2 are linear in their unknowns. The solution x^* obtained by solving problem (13) is called the *here and now solution* and

$$RP = E_{\boldsymbol{\xi}} z(x^*, \boldsymbol{\xi}), \tag{15}$$

is the optimal value of the associated objective function.

Let $\mathcal{J} = \{1, \dots, J\}$ be the set of indices for which the components of the expected value solution $\bar{x}(\bar{\xi})$ are at zero or at their lower bound (non-basic variables). Then, let $\hat{x}$ be the solution of

$$\begin{aligned} \min_{x \in X} \quad & E_{\boldsymbol{\xi}} z\left(x, \boldsymbol{\xi}\right) \\ \text{s.t.} \quad & x_j = \bar{x}_j(\bar{\xi}),\ j \in \mathcal{J}. \end{aligned} \tag{16}$$

We then compute the *expected skeleton solution value*

$$ESSV = E_{\boldsymbol{\xi}}\left(z\left(\hat{x}, \boldsymbol{\xi}\right)\right). \tag{17}$$

Then, the LRCVF heuristic can be summarized in the following steps:

- solve the (continuous relaxation of the) deterministic version of the original problem;

- divide the resulting reduced costs into N intervals and fix in the stochastic first-stage formulation the variables belonging to the third class only, i.e., the out-of-basis variables with highest reduced costs;

- if feasibility issues appear, consider the removed interval and split it again into three intervals; then fix to zero in the stochastic first-stage formulation only the variables belonging to the new third class.

For the discussion about tuning of the parameter N, the reader can refer to Maggioni et al. (2017). Actually, during the numerical simulations we found it was very effective to use this heuristic dependently on the first-stage variables and on the second-stage variables. For this reason, in the latter we use the notation $LRCVF_{\alpha;\beta}$ to consider a heuristic fixing to fix their lower bound to $\alpha\%$ of variables in the first stage and $\beta\%$ of the variables in the second stage.

4. Numerical Simulations

In this section, we analyze the numerical behavior of our problem. We compute the stability of the problem and the VSS for some benchmark instances with different numbers of cells (I), customers types (M), time steps (T), and different ratios between the number of sources and number of sinks. To compute the VSS and to study the stability of the system, we need the optimal solution of the problem. For this reason, we use the commercial solver gurobi (http://www.gurobi.com).

In the simulations, we consider the following types of users.

- The standard users, $m = 0$. They perform one task with standard reliability.

- The business users, $m = 1$. They perform three tasks, with the same reliability as the previous type.

- The workers of the company, $m = 2$. They perform 10 tasks with high probability. Further, the number of this users is known in advance.

Owing to their characteristics, we define the costs of each type to be

$$c_{ij}^{tm} = |\frac{i-j}{4} + 1| C \log(2(m+1)), \tag{18}$$

where C is a realization of a uniform random variable uniformly distributed between $C_{\min}$ and $C_{\max}$. Furthermore, to generate the urban network, we consider different ratios between sources and sinks, and we call this parameter ρ. In the experiment, we consider $\rho = 0.4$ and 0.8, and these values model situations with different dispersions of tasks in the city. Once we have determined sources and sinks, we define the number of people in each cell, by rounding realizations of normal distributions (as suggested by Corollary 3.5). We randomly choose means and variances of the normal because we do not have any data to use to fit the distributions. Once that we have generated the quantity of people in each cell, we ensure that there exists a feasible solution by verifying that

$$\sum_i \sum_m \sum_t n_m \theta_i^{tm} \geq \sum_i N_i. \tag{19}$$

If this is not the case, we randomly add people to a set of cells θ_i^{tm} to satisfy (19). All the following experiments are performed on an Intel Core i7-5500U CPU @2.40 GHz with 8 GB of RAM and Microsoft Windows 10 installed.

4.1. Stability

In this section, we tune the number of scenarios and we check the stability of our model. Table 1 reports the smallest number of scenarios such that the relative in-sample (11) and the relative out-of-sample stability (12) are below 1%.

Table 1 does not report results for instances greater than 100 cells because the solver runs out of memory for those instances. From Table 1, it follows that the smallest number of scenarios necessary to reach the convergence for all the instances with 30 cells is 25 scenarios, while for the instances with 100 cells we need 28 scenarios. All the out-of-sample stabilities are smaller than the in-sample stability. Confidence intervals are reported in Table 1 by using their means and standard deviations. In most cases the 0.53 confidence interval (i.e., $[\mu - \frac{\sigma}{\sqrt{n}}, \mu + \frac{\sigma}{\sqrt{n}}]$ with mean μ and standard deviation σ) contains the value 0, which can be considered a satisfactory result if compared with the level of accuracy of the other parameters in the problem, which are considered as deterministic. Thus, the number of scenarios was set to 30.

4.2. Value of Stochastic Solution

As discussed in Section 3, to justify the need to solve the stochastic version of the problem, we compute the VSS_r for each combination of parameters. The results are shown in Table 2. The effort to solve the stochastic version of the problem is reasonable: with some combination of parameters, the VSS_r is more than 100% and in all experiments it is no smaller than 48%. As the reader may notice, the more variables we have, the larger VSS_r is.

The confidence intervals require us to consider a high quantile to contain the value 0 (i.e., it is unlucky to have a null VSS_r). This result is even more valuable if we consider that it holds for each experiment (see Aickin and Gensler (1996) for a deeper discussion). To better understand the stochastic nature of the problem we compare the solutions of the stochastic problem and the solutions of the expected value problem. The main difference of the two solutions is the number of third user type used, i.e., the users with the highest price, but with the highest performances. This implies that in terms of reduction of the total stochastic cost, it is structurally better to engage a portion of the costly users to reduce the recursion due to the uncertainty of the users' availability. While the stochastic solutions use them in the first stage, the expected value solutions do not use them. Furthermore, while the stochastic solutions try to alert more users in the first time periods and in the first stage, the expected value solutions have not such a pattern. Finally, on average the objective values of the first stage account for 43% of the total costs. This underlines that, for the problem, the on-line management of the planning is more important.

4.3. LRCVF-based Heuristic

As shown previously, exact methods can deal with small and medium-sized instances. In fact realistic instances of the problem range between 1000 and 3000 cells. Thus, heuristic methods are required. On the other hand, we noticed how commercial solvers can deal efficiently with limited size problems. For this reason, we decided to use the LRCVF approach. In fact, we can use the original MIP formulation while reducing its size in

Table 1: Average and standard deviation of the in-sample and out-of-sample stability for each combination of the number of parameters (columns I, M, T, and ρ). Experiments have been repeated 50 times.

Instance parameters				In-sample		Out-of-sample		# Scenarios
I	M	T	ρ (%)	Mean	Std Dev	Mean	Std Dev	
30	3	1	0.4	0.0421724	0.0479721	0.0662104	0.0261267	22
30	3	1	0.8	0.066621	0.0904609	0.0967704	0.0358688	20
30	3	10	0.4	0.00626975	0.0129024	0.0951536	0.0961721	21
30	3	10	0.8	0.0114897	0.00912324	0.0121968	0.00735419	20
30	3	20	0.4	0.0194175	0.007261	0.0139308	0.00794815	25
30	3	20	0.8	0.0983748	0.009983	0.0253745	0.00817391	24
100	3	1	0.4	0.0937461	0.001736	0.0629179	0.00026482	28
100	3	1	0.8	0.0918378	0.001232	0.0616498	0.00029837	27

Table 2: Average VSS (column "Mean") and the standard deviation (column "Std Dev") for each combination of the number of parameters (columns I, M, T and ρ). Experiments have been repeated 50 times.

Instance parameters				VSS	
I	M	T	ρ (%)	Mean	Std Dev
30	3	1	0.4	0.482051	0.245137
30	3	1	0.8	0.444769	0.188483
30	3	10	0.4	0.983473	0.222917
30	3	10	0.8	0.851779	0.109818
30	3	20	0.4	3.84889	1.6708
30	3	20	0.8	3.80675	1.36582
100	3	1	0.4	4.1426	0.47008
100	3	1	0.8	5.4469	0.55

terms of variables and increasing computational efficiency by one order of magnitude while preserving the solution quality (Maggioni et al., 2017).

In this section, we apply the results of the LRCVF heuristic to the problem to decrease the computational time and to study how it performs. The results of these experiments are shown in Table 3. For the instances reported in the table, the heuristic manages to obtain the optimal solution and to decrease the computational time by about 53%. For all experiments we use the number of scenarios in Table 1.

In Table 3 the values $n.p.$ (not present) are set in the cells that cannot be computed because the solver runs out of memory. As the reader may notice, the objective value function decreases if the number of time steps increases. This is reasonable because the more time steps we consider to solve the problem, the greater the probability that low-cost people will perform the task. Nevertheless, the computational time to find a solution increases due to the increased number of variables. From the computation point of view, not only does the heuristic drastically reduce times between 2 and 5 without loss in solution quality, it also allows solutions to be found to instances where the full MIP is unable to even compute an initial solution. It is worth noticing that the parameter ρ has an influence on the computational time of the algorithm. To better understand the influence of this parameter on the computational complexity, we have solved the problem with 30 cells for different values of ρ. The resulting graph is reported in Figure 1. As the reader may notice, the peak of complexity is for extreme values of ρ.

Table 3: Computational time and the optimal objective function value for each combination of the number of parameters (columns I, M, T and ρ). The value $n.p.$ means not present and it is reported for the instances in which gurobi produces an out-of-memory exception. Each row is a different instance.

Instance parameters				gurobi		$LRCVF_{0.6;0.9}$		$LRCVF_{0.6;0.6}$	
I	M	T	ρ (%)	Time (s)	Solution	Time (s)	Solution	Time (s)	Solution
30	3	1	0.4	7.45	10.51	2.69	10.51	3.17	10.51
30	3	1	0.8	23.06	13.08	2.89	13.08	13.27	13.08
30	3	10	0.4	49.13	1.19	12.48	1.19	34.67	1.19
30	3	10	0.8	40.73	2.73	16.62	2.73	29.15	2.73
30	3	20	0.4	72.79	1.05	39.08	1.05	57.48	1.05
30	3	20	0.8	112.23	0.88	45.65	0.88	75.55	0.88
100	3	1	0.4	511.75	22.2	174.66	22.2	238.91	22.2
100	3	1	0.8	514.65	64.80	188.20	64.80	340.88	64.80
100	3	5	0.4	$n.p.$	$n.p.$	379.19	10.86	835.17	10.85
100	3	5	0.8	$n.p.$	$n.p.$	360.02	44.2	383.68	44.2

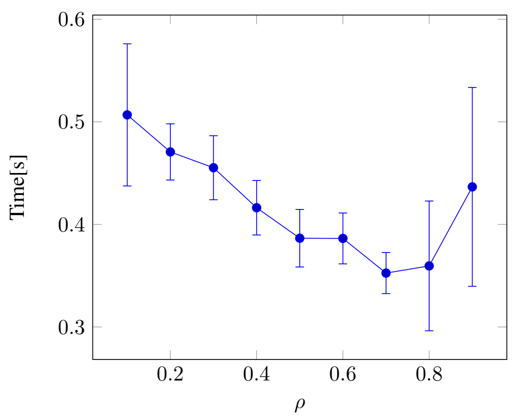

Figure 1: Time to solve the problem with gurobi. The vertical lines represent the standard deviation of the values.

5. Conclusions

In this paper, we have defined a new problem in which the goal is to optimize the activity of a company using opportunistic IoT. We have focused on the Coiote project by TIM, in which the company uses opportunistic IoT to retrieve data from a network of sensors distributed in a wide area. We proved that the number of users in each cell can be approximated by means of a normal distribution. Further, we have shown, by means of numerical examples, that the stochastic version of the problem must be considered due to the high VSS. Finally, we applied the LRCVF heuristic. The results of the numerical experiments show that the heuristic can find the optimal solution in half the computational time used by the commercial software. Despite this analysis, the study of this problem has not yet concluded: in the real world, the number of cells could be greater than 1000 and the computations must complete within 15 minutes. For this reason, a heuristic approach able to deal with larger instances is required. Furthermore, another related problem is the choice of the reward for each type of user. These topics will be addressed in future work.

6. Acknowledgements

Partial funding for this project was provided by Telecom Italia under its TIM Joint Open Labs program and the Munici-

pality of Turin under its “Torino Living Lab” project.

References

Aickin, M., Gensler, H., 1996. Adjusting for multiple testing when reporting research results: the Bonferroni vs Holm methods. American Journal Public Health 86 (5), 726-728.

Billingsley, P., 1995. Probability and Measure, 3rd Edition. Wiley.

Birge, J. R., Louveaux, F., 1997. Introduction to stochastic programming. Springer series in operations research. Springer, New York.

Cagliano, A. C., Gobbato, L., Tadei, R., Perboli, G., 2014. Its for e-grocery business: The simulation and optimization of urban logistics project. Transportation Research Procedia 3, 489-498.

Kaur, M., Kalra, S., 2016. A review on IOT based smart grid. International Journal of Energy, Information and Communications 7 (3), 11-22.

Kaut, M., Vladimirou, H., Wallace, S., Zenios, S., 2007. Stability analysis of portfolio management with conditional value-at-risk. Quantitative Finance 7 (4), 397409.

Klibi, W., Lasalle, F., Martel, A., Ichoua, S., 2010. The stochastic multiperiod location transportation problem. Journal Transportation Science 44 (2), 221–237.

Lindley, D., 1991. Making Decisions, 2nd Edition. Wiley.

Maggioni, F., Crainic, G., Perboli, G., Rei, W., 2017. Reduced cost-based variable fixing in stochastic programming. CIRRELT-2017-10.

Perboli, G., Fadda, E., Gobbato, L., Rosano, M., Tadei, R., 2016. Optimization for networked data in environmental urban waste collection: the onde-uwc project. 28th European Conference on Operational Research, Poznań, Poland, 4-7 July 2016.

Pironet Thierry, C. Y., 2015. Multi-period vehicle assignment problem with stochastic transportation order availability. Odysseus 2015 Sixth International Workshop on Freight Transportation and Logistics, Ajaccio.

Qureshi, K. N., Abdullah, A. H., 2013. A survey on intelligent transportation systems. Middle-East Journal of Scientific Research 15 (5), 629–642.

TIM Jol Swarm, 2016. Coiote project. http://jol.telecomitalia.com/jolswarm/coiote-opportunistically-connecting-the-internet-of-things/.

Zanella, A., Bui, N., Castellani, A., Vangelista, L., Zorzi, M., 2014. Internet of things for smart cities. IEEE Internet of Things Journal 1 (1), 2232.