POLITECNICO DI TORINO
Repository ISTITUZIONALE

A Progressive Hedging Method for the Optimization of Social Engagement and Opportunistic IoT Problems

Original
A Progressive Hedging Method for the Optimization of Social Engagement and Opportunistic IoT Problems / Fadda, Edoardo; Perboli, Guido; Tadei, Roberto. - In: EUROPEAN JOURNAL OF OPERATIONAL RESEARCH. - ISSN 0377-2217. - STAMPA. - 277:2(2019), pp. 643-652. [10.1016/j.ejor.2019.02.052]

Availability:
This version is available at: 11583/2710071 since: 2020-02-09T19:07:26Z

Publisher:
Elsevier

Published
DOI:10.1016/j.ejor.2019.02.052

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

12 October 2024

A Progressive Hedging method for the optimization of Social Engagement and Opportunistic IoT problems

Fadda, Edoardo
edoardo.fadda@polito.it

Perboli, Guido
guido.perboli@polito.it

Tadei, Roberto
roberto.tadei@polito.it

May 23, 2018

1 Abstract

Due to the spreading of the social engagement paradigm, several companies are asking people to perform tasks in exchange for a reward. The advantages of this business model are savings in economic and environmental terms. In previous works, it has been proved that the problem of finding the minimum amount of reward such that all tasks are performed is hard to solve even for medium size realistic instances (if more than one type of people are considered). In this paper, we propose a customized version of the Progressive Hedging algorithm able to provide good solutions for large realistic instances. The method developed reaches the goal to define a procedure that can be used in real environments.

2 Introduction

Social engagement is a new model of business that several companies are adopting. It consists on the request, by the company to the people, to perform business tasks in exchange for a reward. This new business model has been applied in logistics and data gathering giving rise to the so called crowd shipping and opportunistic Internet of Thing (oIoT), respectively. In particular, crowd shipping is a new way to perform last-mile logistics: the company asks people to take packages from a point to another of the city. By doing so, the company can save part of the cost of the standard workforce. Furthermore, people willing to accept to transport the freight are likely to have already planned a similar trip for personal reasons. Hence, crowd shipping decreases pollution. A real example of application of this new business model is in the e-grocery domain:

Walmart (a grocery retailer) asks in-store customers to carry packages to online customers in exchange for discounts. The oIoT application asks the users to share their mobile phone connection with sensors collecting data in the urban environment. In this way, it is possible for the sensors to send the collected data to a central unit that can use them to provide useful information. This application saves the company from building a huge network infrastructure that covers the whole city. It also saves from the usage of standard workforce for doing basic tasks that everybody is able to do. OIoT is applied in the project Coiote by TIM (Telecom Italia Mobile) and the ICT for City Logistics and Enterprises Lab of the Politecnico di Torino [TIM Jol Swarm(2016)]. The goal of the project is to develop an application able to ask in real time users to share their Internet connection with smart dumpsters in exchange for a reward. In this way, a central unit can be aware of the amount of waste in every dumpster and it can organize the waste collection in an optimal way. A mathematical optimization model describing the problem is given in [E. Fadda et al.(2017)]. The objective of that model is to minimize the total cost of the rewards that the company must pay while performing all the tasks. Computational experiments consider only small and medium size instances. Nevertheless, the number of applications using social engagement is growing and the optimization methods are needed in order to achieve good results. In this paper, our goal is to fill the gap in the literature by proposing a heuristic able to solve large instances. In [E. Fadda et al.(2017)], the authors use the Loss of Reduced Costs-based Variable Fixing heuristic for solving the problem. Nevertheless, the heuristic does not manage to obtain good solutions for large instances in a reasonable amount of time. Since the number of scenario is the second greatest dimension of the problem (the first one is the number of nodes of the network), we consider relaxation techniques for dealing with it. For this reason, we use a customized version of the progressive hedging (PH). The article is organized as follows: in Section 3 we review the literature related to the problem, in Section 4 we describe the stochastic model, in Section 5 we briefly describe the PH algorithm and its customized version. In Section 7, we present the computational results of the proposed methodology on several instances and finally, in Section 8, we conclude the work and we depict future developments.

3 Literature review

The main topic of this paper is the application of optimization techniques to social engagement for large size problems.

One study dealing with the optimization of this business model is [E. Fadda et al.(2017)]. In this article, the authors split the review of the literature into two branches, one concerning the cross fertilization between IoT and optimization and the other the underlined optimization model (identified as a multi-period stochastic assignment). By following the same approach, we split the literature into two branches, one considering social engagement and the other the optimization model.

The first book related to social engagement is [R. Algar(2007)] and it dates back to 2007. Then, around 2010 several papers describing the business model appear, one of the most important paper in this period is [R. Botsman et al.(2011)]. In this paper, the authors classify several similar approaches of the social engagement paradigm. Then, around 2015 other works appear, all of them analyze particular aspects of the social engagement business model (see e.g. [J. Hamari et al.(2015)]). All of them underline that the main strength of the model is the exploitation of resources that otherwise would be lost and describe the problem from a business perspective.

From the point of view of the optimization, we consider the same model as the one proposed in [E. Fadda et al.(2017)]. The literature about the multi-period stochastic assignment problem (MPSAP) is not so developed yet. The most similar models considered in the literature are treated in the papers [Klibi et al.(2010)Klibi, Lasalle, Martel and [Pironet Thierry(2015)]. Both these articles minimize, by using heuristics, the assignment cost of a fleet of vehicles to a set of different tasks which number is stochastic. The difference between these models and the approach in [E. Fadda et al.(2017)] is that the source of uncertainty is in the number of resources, while in those two papers it is in the number of tasks. Another difference is that the time horizon is now finite. In this paper, we improve the computational results of the paper [E. Fadda et al.(2017)] by properly defining a customized version of the PH heuristic. The PH heuristic first defined in [R.T. Rockafellar et al.(1991)] and in [A. Lokketangen et al.(1996)] consists in a decomposition of the problem considering the augmented Lagrangian relaxation of the non-anticipative constrains. Several works apply the PH heuristic to different problems, some examples are [G. Perboli et al.(2017)] or [A. Lamghari et al.(2016)]. The first paper applies progressive hedging to a logistic optimization problem while the second applies it to scheduling problems. To the authors knowledge, PH has never been applied to the MPSAP problem.

4 The mathematical model

The mathematical model that describes the problem has been introduced in [E. Fadda et al.(2017)]. It considers a strongly connected graph, where in each node there are people available to do tasks or tasks that need to be done but not both together. This assumption simplifies the problem but it does not produce any loss of generality because people in a node can do the tasks in the same node for a negligible reward.

The mathematical model uses four sets of indexes:

- the set of time indexes $\mathcal{T}$ with cardinality T,
- the set of nodes containing resources (i.e. people willing to perform activities) $\mathcal{I}$ with cardinality I (we call them sources),
- the set of nodes with task to be performed $\mathcal{J}$ with cardinality J (we call them sinks),

- the set of users types $\mathcal{M}$ with cardinality M. This set models different kinds of people, each one characterized by its availability to perform more tasks and the related price.
- the set $\mathcal{V} = \mathcal{I} \cup \mathcal{J}$ as the set of all nodes, its cardinality is V.
- the set $\mathcal{S}$ the set of the scenarios, its cardinality is S. the probability of each scenario is p_s.

The model uses the following data:

- c_{ij}^{tm} is the cost of the reward for one person of type m in node i at time t that goes into node j to execute n_m tasks;
- q_{ij}^{stm} is the cost of the reward for one person of type m in node i at time t, during the second stage, that goes in node j to execute n_m tasks. These costs are greater than the first stage costs;
- N_j is the number of tasks that must be performed in the operational node j between time 1 and time T;
- n_m is the number of tasks performed by one person of type m;
- θ_i^{stm} is the number of people of type m in node i during the second stage of time step t in scenario s;
- $\hat{\theta}_i^{tm}$ is the expected number of people of type m in node i during time step t; in particular, $\hat{\theta}_i^{0m}$ is the numbers of people in node i of type m when the algorithm starts.

The variables used are:

- x_{ij}^{tm}: number of people of type m in node i at time t that are asked to perform n_m tasks in node j, ;
- y_{ij}^{stm}: number of people of type m in node i at time t in scenario s that are asked to perform n_m tasks in node j;

The associated linear program is

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

subject to

$$\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{i=1}^{I} n_m (x_{ik}^{tm} + y_{ij}^{stm}) \geq N_j \quad \forall \ j \in \mathcal{J} \quad \forall \ s \in \mathcal{S} \tag{2}$$

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

$$\sum_{j=1}^{J}(x_{ij}^{tm} + y_{ij}^{stm}) \leq \theta_i^{stm} \quad \forall \ i \in \mathcal{I} \ t \in \mathcal{T} \ m \in \mathcal{M} \ s \in \mathcal{S} \tag{4}$$

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

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

The objective function of the problem is the sum of the first stage rewards and the expected rewards of the second stage. Constraints (2) impose that all tasks must be performed and constraints (3) limit the first-stage users while constraints (4) limit the second-stage users. The telecommunication experts involved in the project suggest that the time step of the model should be equal to 30 minutes in order to be more responsive to the people flow in the city. From this requirement follows the constraint to solve the problem in just a few minutes.

We do not discuss more the model, but the interested reader is referred to [E. Fadda et al.(2017)] for details.

Remark 4.1. *Without loss of generality we can assume that the constraints* (2), (3) *and* (4) *hold with equality. This result can be achieved by properly adding fictitious nodes in the network.*

It is possible to prove the following theorem.

Theorem 4.2. *[E. Fadda et al.(2017)] Problem* (1)-(4) *can be solved in polynomial time for* $M = 1$

The proof of Theorem 4.2 can be found in [E. Fadda et al.(2017)]. The main result of the proof is that if $M = 1$ the continuous version of the problem has integer optimal solution because the constraint matrix is totally unimodular (TU) and the known vector has integer components.

5 Progressive Hedging

PH is a heuristic based on the augmented Lagrangian relaxation of the non-anticipative constraints of the stochastic problem. Problem (1) - (4) has the non-anticipative constraints incorporated in the choice of the variables. Hence, in order to apply the PH we exploit them by defining the new variables x_{ij}^{stm} that replace variables x_{ij}^{tm} and by adding non-anticipative constraints, i.e.

$$x_{ij}^{stm} = x_{ij}^{s'tm} \ \forall \ s \neq s', i, j, t, m. \tag{5}$$

Constraints (5) enforce that the first stage variables must be equal under each possible scenario. It is worth noting that constraints (5) are equivalent to

$$x_{ij}^{stm} = \sum_{s=1}^{S} p_s x_{ij}^{stm} \ \forall s, i, j, t, m. \tag{6}$$

In the rest of the paper, we consider constraints (6) instead of (5) because these are used in the standard form of the PH. From the relaxation of constraints (6), we obtain the following problem

$$\begin{aligned}\text{minimize} \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{m=1}^{M} c_{ij}^{tm}x_{ij}^{tm} + \sum_{s=1}^{S}\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{m=1}^{M} q_{ij}^{stm}y_{ij}^{stm} + \\ + \sum_{s=1}^{S}\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{m=1}^{M} w_s|x_{ij}^{stm} - \bar{x}_{ij}^{tm}|\end{aligned} \tag{7}$$

subject to

$$\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{i=1}^{I} n_m(x_{ik}^{tm} + y_{ij}^{stm}) \geq N_k \quad \forall\ k \in \mathcal{I} \quad \forall\ s \in \mathcal{S} \tag{8}$$

$$\sum_{j=1}^{J} x_{ij}^{tm} \leq \hat{\theta}_i^{tm} \quad \forall\ i \in \mathcal{I}\ \ t \in \mathcal{T}\ \ m \in \mathcal{M} \tag{9}$$

$$\sum_{j=1}^{J}(x_{ij}^{tm} + y_{ij}^{stm}) \leq \theta_i^{stm} \quad \forall\ i \in \mathcal{I}\ \ t \in \mathcal{T}\ \ m \in \mathcal{M}\ \ s \in \mathcal{S} \tag{10}$$

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

where

$$\bar{x}_{ij}^{tm} = \sum_{s=1}^{S} p_s x_{ij}^{stm}.$$

It is worth noting that problem (7)-(10) can be split into S independent problems, one for each scenario s. In order to increase the speed of convergence of the Lagrangian method, the PH heuristic adds a quadratic penalty term that yields to an augmented Lagrangian method. Then, by removing constant terms, problem (7)-(10) becomes

$$\begin{aligned}\text{minimize} \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{m=1}^{M} c_{ij}^{tm}x_{ij}^{tm} + \sum_{s=1}^{S}\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{m=1}^{M} q_{ij}^{stm}y_{ij}^{stm} + \\ + \sum_{s=1}^{S}\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{m=1}^{M} w_s x_{ij}^{stm} + \frac{\rho}{2}\sum_{s=1}^{S}\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{m=1}^{M}(x_{ij}^{stm} - \bar{x}_{ij}^{tm})^2 \\ \text{subject to (8) (9) (10)}\end{aligned} \tag{11}$$

By exploiting the independence of the scenarios, it is possible to define the augmented Lagrangian algorithm shown in Algorithm 1, namely the PH. In

Algorithm 1 Original Progressive Hedging

k:=0
$g^{(k)} = +\infty$
for each scenario $s \in \mathcal{S}$ **do**
    Solve:

$$x_s^{(k)} := \text{argmin}_{x,y_s}\, c^T x + f_s^T y_s \;:\; (x, y_s) \in \mathcal{Q}_s \tag{12}$$

$\bar{x}^{(k)} := \sum_{s\in\mathcal{S}} p_s x_s^{(k)}$
$w_s^{(k)} := \rho(x_s^{(k)} - \bar{x}^{(k)})\ \forall\ s\ \in\ \mathcal{S}$
while $g^{(k)} < \epsilon$ **do**
    k:=k+1
    for each scenario $s \in \mathcal{S}$ **do**
        Solve:

$$x_s^{(k)} := \text{argmin}_{x,y_s}\, c^T x + w_s^{(k-1)} x + \frac{\rho}{2}\left\|x - \bar{x}^{(k-1)}\right\|_2^2 + f_s^T y_s \;:\; (x, y_s) \in \mathcal{Q}_s \tag{13}$$

    $\bar{x}^{(k)} := \sum_{s\in\mathcal{S}} p_s x_s^{(k)}$
    $w_s^{(k)} := w_s^{(k-1)} + \rho(x_s^{(k)} - \bar{x}^{(k)})\ \forall\ s\ \in\ \mathcal{S}$
    $g^{(k)} = \sum_{s\in\mathcal{S}} p_s \|x_s^{(k)} - \bar{x}_s^{(k)}\|$

the pseudo-code we use the general notation $x_s^{(k)}$ to define the solution of the scenario problem s in the k-th iteration.

The algorithm is composed by two phases: the first one computes the solutions of all the sub problems, the second one uses these solutions to update $\bar{x}_{ij}^{tm}$. These two phases are repeated iteratively until convergence i.e. the norm of the difference between the $\bar{x}_{ij}^{tm}$ and the x_{ij}^{stm} is smaller than a threshold ϵ for all the scenarios $s = 1, \ldots, S$.

Definition 5.1. *If it holds that*

$$x_{ij}^{stm} = \bar{x}_{ij}^{tm} \quad \forall\ s\ \in\ \mathcal{S}, \tag{14}$$

we say that variable x_{ij}^{stm} has reached the consensus.

The PH algorithm provably converges in linear time if the decision variables are continuous. Instead, for integer problems some instances have proved that the convergence to the optimal solution is not guarantee (see [J.-P.. Watson et al.(2011)]) for this reason, we add to Algorithm 1 a stopping criteria based on a maximum limit of CPU time and a maximum number of iterations. If these limits are overcome, we then solve the original problem by fixing the values of all the variables that reach consensus in the PH.

6 The Customized Progressive Hedging

The PH solves the problem of dealing with several scenarios. Nevertheless, if the computation of the optimal solutions of the sub-problems takes a long time (as it is the case with $I \geq 500$), then also the PH takes too much time for being used in a real environment. For this reason, we propose a customized version of the PH by applying three modifications.

The first modification that we apply is to use a heuristic in order to solve the single scenario problem (i.e. we modify point 10 in Algorithm 1).

The second modification is related to the evolution of the penalty term ρ. The idea of updating the penalty parameter over the iterations was first introduced in [J.M. Mulvey et al.(1991)]. In the paper, the authors notice that the convergence rate was extremely sensitive to the penalty parameter value. For a literature review about the topic the reader is referred to [Zehtabian, et al.(2016)]. From all the updating policies presented in [Zehtabian, et al.(2016)], we find to be particularly efficient the one proposed in [Zéphyr et al.(2014)].

In this work, the penalty parameter is updated by means of an adaptive learning update that considers $g^{(k)}$ and $\delta^{(k)} = g^{(k)} + \sum_{s \in \mathcal{S}} p_s ||x_s^{(k)} - x_s^{(k-1)}||$ where $x_s^{(k)}$ is the solution of problem scenario s during the k-th iteration of the algorithm. Intuitively, $g^{(k)}$ represents a gap related to the satisfaction of the non anticipative constraints at iteration k. Instead, we can interpret $\delta^{(k)}$ as a sort of optimality gap if we consider that for the optimal solution this difference is null. For the sake of completeness we present the updating rule of parameter ρ in iteration k.

$$
\begin{aligned}
\tau^{(k+1)} &= \frac{\delta^{(k+1)}}{\delta^{(k)}} \\
\gamma^{(k+1)} &= \max[0.1, \min[0.9, \tau^{(k+1)} - 0.6]] \\
\sigma^{(k+1)} &= (1 - \gamma^{(k+1)})\sigma^{(k)} + \gamma^{(k+1)}\tau^{(k+1)} \\
\zeta^{(k+1)} &= \sqrt{1.1\sigma^{(k+1)}} \\
\alpha^{(k+1)} &= 0.8\alpha^{(k)} + 0.2\frac{g^{(k+1)}}{\delta^{(k+1)}} \\
b^{(k+1)} &= 0.98b^{(k)} + 0.02\alpha^{(k+1)} \\
c^{(k+1)} &= \max[0.95, \frac{1 - 2b^{(k+1)}}{1 - b^{(k+1)}}] \\
h^{(k+1)} &= \max[c^{(k+1)} + \frac{1 - c^{(k+1)}}{b^{(k+1)}}\alpha^{(k+1)}, 1 + \frac{\alpha^{(k+1)} - \beta^{(k+1)}}{1 - b^{(k+1)}}] \\
q^{k+1} &= (\max[\zeta^{(k+1)}, h^{(k+1)}])^{\frac{1}{1+0.01(k-1)}} \\
\rho^{(k+1)} &= \max[0.01, \min[100, q^{(k+1)}\rho^{(k)}]].
\end{aligned}
\tag{15}
$$

We choose to use this updating strategy since, by using an updating rule of the form $\rho^{(k)} = q^{(k)}\rho^{(k-1)}$, it is possible to drive the evolution by considering two indicators: one considering the average contraction rate of the optimality

gap, and one considering an average proportion of the optimality gap accounted for by the non-anticipativity gap.

The third modification is to use the term

$$\frac{\rho}{2}\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{m=1}^{M}|x_{ij}^m - \bar{x}_{ij}^m|,$$

instead of the quadratic term in the objective function. This choice transform problem (13) in a linear problem.

The final procedure is shown in Algorithm 2.

Algorithm 2 Customized Progressive Hedging

k:=0
for each scenario $s \in \mathcal{S}$ **do**
    Solve Heuristically:

$$x_s^{(k)} := \text{argmin}_{x,y_s}\, c^T x + f_s^T y_s \;:\; (x, y_s) \in \mathcal{Q}_s \tag{16}$$

$\bar{x}^{(k)} := \sum_{s\in\mathcal{S}} p_s x_s^{(k)}$
$w_s^{(k)} := \rho(x_s^{(k)} - \bar{x}^{(k)})\ \forall\ s\ \in\ \mathcal{S}$
$k = 1$
while $g^{(k)} < \epsilon$ **do**
    k:=k+1
    Update ρ according to (15)
    for each scenario $s \in \mathcal{S}$ **do**
        Solve Heuristically:

$$x_s^{(k)} := \text{argmin}_{x,y_s}\, c^T x + w_s^{(k-1)} x + \frac{\rho}{2}\left\|x - \bar{x}^{(k-1)}\right\|_1 + f_s^T y_s \;:\; (x, y_s) \in \mathcal{Q}_s \tag{17}$$

    $\bar{x}^{(k)} := \sum_{s\in\mathcal{S}} p_s x_s^{(k)}$
    $w_s^{(k)} := w_s^{(k-1)} + \rho(x_s^{(k)} - \bar{x}^{(k)})\ \forall\ s\ \in\ \mathcal{S}$
    $g^{(k)} = \sum_{s\in\mathcal{S}} p_s \|x_s^{(k)} - \bar{x}_s^{(k)}\|$

In the rest of the paper, we refer to Algorithm 1 as original PH, and we refer to Algorithm 2 as customized PH.

For the sake of simplicity and without loss of generality we present the heuristic for solving the single scenario problem without using the index s. Furthermore, we do not consider the time index t since it can be removed by replacing every node with T nodes, one for each time step.

The single scenario heuristic is based on the observation that if it is known how many people of type m are necessary to satisfy the demand of sink j, then we can solve M independent problems with only one type of customer (and these problems can be solved by solving their continuous relaxation as stated by Theorem 4.2). In particular, the heuristic is composed by three phases. In

the first phase, we compute for each type m the number of resources of that type to allocate in each sink j. In the second phase, we solve m continuous independent linear problems, from these solutions it is possible to generate a solution of the whole problem. It is worth noting that even if we know how many customers of type m must be used to satisfy the demand of sink j we have no information about the sources to use to satisfy each sink. Finally, in the third phase of the heuristic, we update the information used in the first phase by using the information extrapolated by the solution found in the second phase. In the following subsections, we describe the three phases of the proposed heuristic.

6.1 Phase I

The goal of the first phase is to define the number of people of type $m \in \mathcal{M}$ necessary to satisfy the number of tasks to do in sink $j \in \mathcal{J}$. For achieving this goal, we solve a linear integer problem that considers the variables $w_j^m \in \mathbb{N}, \forall\, j \in J,\ m \in M$. These variables represent the number of people of type m that the solution uses for satisfying the request of sink j. The model uses the parameters $\hat{q}_j^m$ that are randomly assigned in the first iteration of the algorithm, while in the following iterations they are updated by using the procedure described in Subsection 6.3. The mathematical model solved in the first phase is

$$\text{minimize} \sum_{j=1}^{J} \sum_{m=1}^{M} \hat{q}_j^m w_j^m \tag{18}$$

subject to

$$\sum_{m=1}^{M} n_m w_j^m = N_j \quad j \in \mathcal{J} \tag{19}$$

$$\sum_{j=1}^{J} w_j^m = n_m \sum_{i=1}^{I} \theta_i^m \quad m \in \mathcal{M} \tag{20}$$

$$w_j^m \in \mathbb{N} \quad \forall\ j \in \mathcal{J} \quad m \in \mathcal{M}$$

The objective of this model is to minimize a linear approximation (with respect to variables w_j^m) of the objective function of problem (7)-(10). Constraints (19) impose that all the tasks must be done, while constraints (20) impose that the total number of people does not exceed θ_i^m. The constraint matrix of this problem is not TU, nevertheless it has dimension smaller than the single scenario problem. It is worth noting that (19) and (20) have the equal sign because of Remark 4.1.

6.2 Phase II

In the second phase of the proposed heuristic, we split constraints (8) of model (7)-(10) into M constraints, one for each type of people, by using the optimal solution obtained in the previous step ($w_j^{*m}\ \forall\ j \in \mathcal{J}, m \in \mathcal{M}$) instead of the

demand N_j. We can then split problem (11) into M independent problems. It is worth noting that in order to remove the absolute value, we add in the problem slack variables z_{ij}^{tm} such that

$$z_{ij}^{tm} > x_{ij}^{stm} - \bar{x}_{ij}^{tm} \tag{21}$$

and

$$z_{ij}^{tm} > \bar{x}_{ij}^{tm} - x_{ij}^{stm}. \tag{22}$$

These constraints cancel the TU structure of the constraint matrix because it inserts sub-matrices such as:

$$\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \tag{23}$$

which determinant is not in $0, \pm 1$. For this reason, we dynamically enforce these constraints in the solver by using the callback function of the solver. In particular, we first consider only constraints (21), then if the solution does not respect constraint (22), we add it and we remove (21). By using this procedure in few iterations the solver converges even if from a theoretical point of view there is no guarantee that this should happen.

We then recall the following lemma:

Lemma 6.1. *Given a matrix A, with the TU property and a vector v with all zero but one nonzero being ± 1, then by adding v as a row or as a column of the matrix A, the TU property is preserved.*

Proof. See [A. Schrijver (1986)]. □

It is worth noting that if there are not both (21) and (22) for the same i, j, t, m, then the constraint matrix is

$$\left(\begin{array}{cccc|cccc} \multicolumn{4}{c|}{\text{A}} & \multicolumn{4}{c}{0} \\ \hline \pm 1 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & \pm 1 & \dots & 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \pm 1 & 0 & 0 & \dots & 1 \end{array}\right). \tag{24}$$

It is worth noting that we can build this matrix by adding row vectors $[0, \dots, \pm 1, \dots, 0]$ to the matrix A and then by juxtaposing column vectors $[0, \dots, 1, \dots, 0]$ to the obtained matrix. By doing so, and by using Lemma 6.1, we have that the constraint matrix is TU. Then, by approximating variables $\bar{x}_{ij}^m$ to the nearest integer, the continuous relaxation of these problems provide integer solutions.

6.3 Phase III

In the final phase of the proposed heuristic, we use the solution computed in the previous step in order to update the costs $\hat{q}_j^m$ used in the mathematical model (18)-(20). The updating rule is the following

$$\hat{q}_j^m \leftarrow \Big(\xi \hat{q}_j^m + \frac{\sum_{i=1}^I c_{ij}^m x_{ij}^{*m} + \sum_{i=1}^I q_{ij}^m y_{ij}^{*m}}{\sum_{i=1}^I x_{ij}^{*m} + \sum_{i=1}^I y_{ij}^{*m}}\Big)(1+\eta) \tag{25}$$

where $\xi = 0.01$, c_{ij}^m and q_{ij}^m are the costs in problem (11), x_{ij}^{*m} and y_{ij}^{*m} are the optimal solutions obtained in the second phase and η is a random variable uniformly distributed between $[-0.1, 0.1]$. The first term of Equation (25) considers the old estimation. Instead, the second term is the unitary resource cost of the solution that uses resources of type m in sink j. The summation of these two terms is multiplied by $(1 + \eta)$ in order to increase the exploitation of the solution space.

7 Numerical Results

In this section we present the numerical performance of the proposed method. We split the description of the experiments into two subsections. In Subsection 7.1 we present the performance of the single scenario heuristic and in Subsection 7.2 we present the performance of the original PH and of the customized PH.

In this section, we consider several different instances characterized by different numbers of nodes (V), time steps (T), and different ratios between the number of sources and number of sinks (ν). All the experiments are performed on an Intel Core i7-5500U CPU @2.40 GHz with 8 GB of RAM and Microsoft Windows 10 installed, all the code has been developed in C++ and the commercial solver Gurobi [1] (version 7.0), is used in order to compute exact solutions. Since the code has been developed in the context of the Coiote project, TIM experts have validated all instances and solutions.

In all instances we consider three types of people. This choice has been derived by the standard segmentation that a telecommunication company performs from data of the phone cell, without violating the privacy regulations [European Commission(2012)]. The people types considered are the standard people ($m = 0$, they are available to perform one task); the business people ($m = 1$, they are available to perform three tasks) and the workers of the company ($m = 2$, they are available to perform ten tasks). Driven by these characteristics, we define the cost of the reward for a person of type m that is asked to go from node i to node j during time t for executing n_m tasks to be

$$c_{ij}^{tm} = |\frac{i-j}{4} + 1|C \log(2(m+1)), \tag{26}$$

where C is a random variable uniformly distributed between $[5, 10]$. For simulating the urban network, we consider different relative quantity between number of sources and number of sinks. For this reason, we define $\nu = I/V$, in the experiment we consider $\nu = 0.4$ and $\nu = 0.8$. The first value models applications such as crowd shipping in which there are more sinks than sources, the second one models applications such as oIoT in which there are more sources than sinks. We simulate the number of people in each node of the network, by rounding a realizations of normal distributions (see [E. Fadda et al.(2017)]). Finally, in order to be sure to have a feasible solution we verify that the following condition

[1] http://www.gurobi.com

holds

$$\sum_{i=1}^{I}\sum_{m=1}^{M}\sum_{t=1}^{T} n_m \theta_i^{tm} = \sum_{j=1}^{J} N_j. \tag{27}$$

If Equation (27) is not verified, we randomly increase the values of θ_i^{tm} until it holds. Since we are considering stochastic programs, it is necessary to correctly set the number of scenarios in order to reach stability of the solution (for a discussion about stability we refer to [Kaut et al.(2007)Kaut, Vladimirou, Wallace, and Zenios]). We test out of sample stability since it is the most difficult condition to satisfy and, in general, it requires more scenarios than the in-sample stability. Since for the instances with $V > 100$, $T > 1$, the number of scenarios that guarantees stability cannot be tested by using exact algorithms, we perform the stability analysis by using Algorithm 1 if possible and, for the largest instances, by using Algorithm 1. The results of these experiments are shown in Table 1. All the results are averaged over 100 iterations.

Table 1: Number of scenario (n Scenarios) that leads to out of sample stability and algorithm used (Algorithm) for different combination of the parameters (columns V, M, T and ν). Experiments have been repeated 100 times, in brackets the standard deviations of the values.

V	M	T	ν	n Scenarios	Algorithm
30	3	1	0.4	22.2 (1.4)	Exact
30	3	1	0.8	21.7 (1.6)	Exact
30	3	10	0.4	21.4 (1.5)	Exact
30	3	10	0.8	20.3 (1.5)	Exact
30	3	20	0.4	25.7 (2.2)	Exact
30	3	20	0.8	24.6 (2.1)	Exact
100	3	1	0.4	24.9 (2.8)	Exact
100	3	1	0.8	26.1 (2.8)	Exact
100	3	10	0.4	29.4 (3.1)	Original PH
100	3	10	0.8	27.8 (3.8)	Original PH
100	3	20	0.4	27.7 (4.1)	Original PH
100	3	20	0.8	29.4 (3.9)	Original PH
300	3	1	0.4	28.6 (3.8)	Original PH
300	3	1	0.8	24.7 (4.0)	Original PH
500	3	1	0.4	28.5 (3.9)	Original PH
500	3	1	0.8	31.6 (4.1)	Original PH

As the reader can notice, Table 1 shows for each algorithm the maximum size of the instances that it is able to solve. In particular, we have that until $I = 100$, $M = 3$ and $T = 1$ the exact method is able to solve the problem and the original PH is able to solve problems until $I = 500$, $M = 3$ and $T = 1$. If original PH is used to test stability instead of exact method the resulting number of scenarios increases, on average, of the 8.5% with a standard deviation of the 2.4%. Instead, if the customized PH is used to test stability instead of

exact method the resulting number of scenarios increases, on average, of the 12.7% with a standard deviation of the 3.1%. This is reasonable since we are considering the performance of an algorithm that is approximating the real problem.

7.1 The single scenario heuristic

In this section, we test the performances of the proposed single scenario heuristic. The results are described in Table 2. All the values are obtained by averaging on 100 runs. Since in each iteration of the original PH the objective function changes, we test the approaches by considering the objective function (17). The results are shown in Table 2.

Table 2: Average time of the Heuristic (Time Heu. [s]), average time of the exact method (Time Ex. [s]) and gap between the solutions (Gap[%]) for different combination of the parameters (columns V, M, T and ν). The value *n.p.* means not present and it is reported for the instances in which Gurobi produces an out-of-memory exception. Experiments have been repeated 100 times, in brackets the standard deviations of the values.

V	M	T	ν	Time Ex. [s]	Time Heu. [s]	Gap[%]
30	3	1	0.4	0.04 (0.00)	0.21 (0.02)	0.13 (0.06)
30	3	1	0.8	0.02 (0.00)	0.19 (0.05)	0.26 (0.08)
30	3	10	0.4	0.05 (0.01)	0.41 (0.09)	0.14 (0.09)
30	3	10	0.8	0.07 (0.01)	0.34 (0.10)	0.42 (0.09)
30	3	20	0.4	0.12 (0.03)	0.65 (0.13)	0.55 (0.10)
30	3	20	0.8	0.13 (0.04)	0.64 (0.12)	0.25 (0.09)
100	3	1	0.4	1.14 (0.08)	1.18 (0.24)	0.76 (0.11)
100	3	1	0.8	1.14 (0.09)	1.12 (0.22)	0.18 (0.09)
100	3	10	0.4	2.41 (0.11)	1.85 (0.32)	0.72 (0.10)
100	3	10	0.8	2.42 (0.15)	1.86 (0.28)	0.27 (0.10)
100	3	20	0.4	9.41 (1.56)	3.05 (0.31)	0.86 (0.09)
100	3	20	0.8	10.02 (2.01)	3.16 (0.34)	0.36 (0.11)
300	3	1	0.4	69.42 (12.34)	7.82 (0.47)	0.56 (0.11)
300	3	1	0.8	72.35 (14.52)	8.10 (0.62)	0.27 (0.12)
500	3	1	0.4	129.12 (13.44)	9.23 (0.74)	0.79 (0.08)
500	3	1	0.8	126.23 (15.32)	15.34 (0.82)	0.86 (0.11)

It is important to notice that the gap between the values of the two solutions is really small. If we call $\hat{w}_j^m$ the number of people of type m that satisfies the request of sink j found by the proposed heuristic and we call w_j^{*m} the same quantity in the optimal solution ($w_j^{*m} = \sum_{i=1}^{I} \sum_{t=1}^{T} x_{ij}^{tm}$). The average $\sum_{j=1}^{J} \sum_{m=1}^{M} |\hat{w}_j^m - w_j^{*m}|$ is about 4.8 with a standard deviation of 1.4. In order to understand this result, we prove the following theorem.

Theorem 7.1. *Given a matrix $A \in \mathbb{Z}^{m \times n}$ such that the sum on the elements of each rows and the sum on the elements in each column is fixed i.e. $\sum_{i=1}^{n} a_{ij} = \bar{a}_j \ \forall j = 1, \dots, m$ and $\sum_{j=1}^{m} a_{ij} = \bar{a}_i \ \forall i = 1, \dots, n$, then the smallest change with respect to the norm $||A||_1 = \sum_{i,j} |a_{ij}|$ has the form:*

$$\Sigma_{ij}^{i_1 j_1} = \left[\begin{array}{c|c|c|c|c} 0 & \pm 1 & \dots & \mp 1 & \dots \\ \hline \dots & \dots & \dots & \dots & \dots \\ \hline 0 & \mp 1 & \dots & \pm 1 & \dots \end{array} \right]. \tag{28}$$

Where the couples of indexes i, j and i_1, j_1 are the positions of the upper left and of the lower right non null elements.

Proof. Since $A \in \mathbb{Z}^{m \times n}$, the smallest change of one element is to add ± 1. We call a_{ij} the element considered. Since $a_{ij} = a_{ij} \pm 1$, we have to add ∓ 1 to one element in the same row and to one element in the same column in order to keep constant the sum over the row and columns. Let us call these elements a_{kj} and a_{il}. Nevertheless, by changing these elements we have to add ± 1 to the element a_{kl} in order to maintain the value of the summation over the rows and over the columns constant. □

Remark 7.2. *From Theorem 7.1, it is possible to conclude that, given a feasible solution W of problem* (18)-(20)*. Then, the possible changes of matrix W leading to another feasible solution can be described as $A = \sum_{j=1}^{J} \sum_{m=1}^{M} \alpha_{jm} \Sigma_{jm}^{j_1 m_1}$ for suitable integer values α_{jm} and suitable indexes j_1 and m_1.*

As the reader can notice, for instances with $I \leq 100$ and $T = 1$, the proposed heuristic requires more time than the exact method (on average it takes 4.5 times more). Instead, for largest instances, the proposed heuristic perform much better, in particular in the instances with $I = 300$, the heuristic takes the 12% of the time of the exact solver. While for instances with $I = 500$ it takes the 10% of the time. This behavior is due to the time consumed by the solution of the integer model in the first part of the heuristic that, in small instances, does not produce computational gain. The gaps that the proposed heuristic reaches are very good: on average the gaps are smaller than the 1% in all the different settings. Furthermore, in all runs the largest gap obtained is 2.34%. It is worth noting that the parameter ν does not influence neither the time nor the gap. Hence, we can claim that the proposed approach does not depend on the characteristic of the network.

7.2 Customized Progressive Hedging Results

In this section, we test the performances of the customized PH and the one of the original PH. The results of these experiments are shown in Table 3. Each experiment is repeated 100 times. We set the maximum CPU time to be one hour and the maximum number of iterations of the PH to be 200 iterations (these bounds are never reached because convergence is reached before).

As the reader can notice, for small instances the customized PH uses more time than the original one. The reason of this behavior is due to the time

Table 3: Average time and optimality gap of the original PH (Time PH Or [s], Gap PH Or [%]), and of the customized PH (Time PH Cust.[s], Gap PH Cust.[%]) for different combination of the parameters (columns V, M, T and ν). The value *n.p.* means not present and it is reported for the instances in which Gurobi produces an out-of-memory exception. Experiments have been repeated 100 times, in brackets the standard deviations of the values.

V	M	T	ν	Time PH Or. [s]	Time PH Cust.[s]	Gap PH Or. [%]	Gap PH Cust. [%]
30	3	1	0.4	18.01 (6.5)	20.12 (2.1)	0.77 (0.01)	0.78 (0.02)
30	3	1	0.8	12.43 (2.3)	21.23 (3.4)	0.56 (0.05)	0.63 (0.07)
30	3	10	0.4	24.12 (4.6)	37.12 (2.4)	0.75 (0.04)	0.82 (0.05)
30	3	10	0.8	33.25 (4.2)	36.97 (5.3)	0.98 (0.09)	1.04 (0.04)
30	3	20	0.4	62.32 (13.5)	45.27 (6.1)	0.98 (0.04)	1.01 (0.04)
30	3	20	0.8	66.36 (14.7)	47.43 (6.2)	0.96 (0.04)	1.03 (0.05)
100	3	1	0.4	72.23 (21.5)	30.21 (5.3)	1.08 (0.08)	1.23 (0.05)
100	3	1	0.8	70.12 (12.3)	34.42 (6.5)	1.01 (0.01)	1.26 (0.02)
100	3	10	0.4	208.12 (22.4)	62.23 (9.8)	*n.p.*	*n.p.*
100	3	10	0.8	210.64 (21.6)	61.23 (8.3)	*n.p.*	*n.p.*
100	3	20	0.4	623.54 (42.6)	125.24 (21.4)	*n.p.*	*n.p.*
100	3	20	0.8	653.53 (44.2)	153.12 (22.8)	*n.p.*	*n.p.*
300	3	1	0.4	154.12 (33.5)	54.64 (14.5)	*n.p.*	*n.p.*
300	3	1	0.8	124.32 (37.3)	55.32 (12.4)	*n.p.*	*n.p.*
500	3	1	0.4	532.23 (42.5)	123.21 (22.3)	*n.p.*	*n.p.*
500	3	1	0.8	542.12 (45.6)	124.43 (21.5)	*n.p.*	*n.p.*

consumed by the single scenario heuristic that for small instances consumes more time than the exact method. Nevertheless, as the dimension of the network grows the performance of the customized PH outperforms the ones of the original PH. In particular, customized PH reaches the consensus faster than original PH. This is due to the updating rule of the parameter ρ. Without this modification, the customized PH never achieves consensus.

The differences between the optimal solutions and the solutions found by the original PH and the customized PH are related to an increment in the number of second stage resources used. This effect is produced by the values of the coefficients w_s in both the original and the customized PH that increase the costs of the first stage variables.

In order to compare the performance of the original PH and of the customized PH we compute for the different values of the parameters V the gaps between the solutions provided by the original PH and the customized PH. The results are shown in Figure 1. As it is reasonable to guess, the gap increase as the number of nodes in the network increases. Nevertheless, this gap is of the order of 1.5% for instances of 500 nodes. These errors are due to differences in the choices of the first stage resources used by the two solutions. It is important to point out that in the runs with the $V = 500$, in the 2% of the instances, we record a better solution of the customized PH with respect to the original PH.

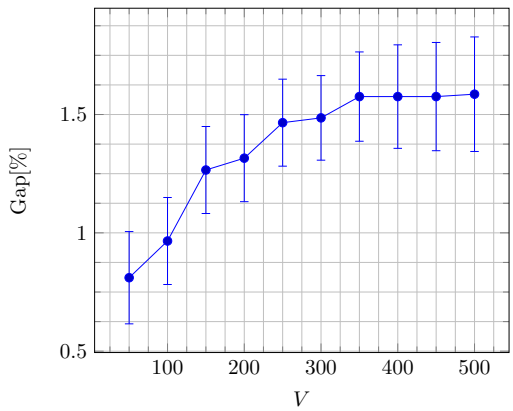

Figure 1: Gaps between the original PH and the customized PH for different values of V, $T = 1$ and $M = 3$.

As for the single scenario heuristic, also the performance of the customized PH are not influenced by the parameter ν. Hence, the proposed approach does not depend on the characteristic of the network and it can be used in every setting.

In order to test the customized PH in the real setting, we run it with instances up to 1500 nodes, 3 types of people and 20 time periods. It manages to converge in approximate 20.76 minutes with a standard deviation of 2.59 minutes (these data are obtained by averaging on 100 runs). Since the exact method and the original PH are not able to solve such instances we have no data in order to compare the optimal solution found.

8 Conclusion

In conclusion, we can claim that the customized PH is able to achieve good performances and to increase the maximum size of instances that can be considered. Furthermore, it can be used in the real field due to the computational times that it requires. Thanks to this heuristic, it is possible to reduce the costs that the companies using social engagement have to bear. Furthermore, since the costs of the rewards are proportional to the path that the people have to do for reaching the task, finding a good solution to the problem means to use people that are near to the task. Then, using this heuristic generates environmental gains by reducing pollution and traffic congestion.

References

[R. Algar(2007)] Algar, R. (2007). Collaborative consumption, Leisure Report, 4, 72-83.

[Aickin and Gensler(1996)] 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.

[V. Barbu et al.(2012)] Barbu, V. and Precupanu, T.,2012. Convexity and Optimization in Banach Spaces. Springer Netherlands, Dordrecht,67–151.

[Billingsley(1995)] Billingsley, P., 1995. Probability and Measure, 3rd Edition. John Wiley & Sons, Inc., New York, NY, USA.

[Birge and Louveaux(1997)] Birge, J. R., Louveaux, F., 1997. Introduction to stochastic programming. Springer series in operations research. Springer, New York.

[R. Botsman et al.(2011)] Botsman, R., Rogers, R. (2011). What's mine is yours: how collaborative consumption is changing the way we live, Collins London.

[Cagliano et al.(2014)Cagliano, Gobbato, Tadei, and Perboli] 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.

[European Commission(2012)] European Commission, 2012. General Data Protection Regulation.

[E. Fadda et al.(2017)] Fadda, E., Perboli, G. and Tadei Roberto (2017). Customized Multi-period Stochastic Assignment Problem for Social Engagement and Opportunistic IoT. Computers and Operation Research. 93, 41–50

[A. Ghouila-Houri (1962)] Ghouila-Houri,A., 1962. Caracterisation des matrices totalement unimodulaires. Académie des Sciences Paris 254,1192–1194.

[B. Guo et al.(2012)] Guo, B., Yu, Z., Zhou, X.,Zhang, D., 2012. Opportunistic IoT: Exploring the social side of the internet of things, Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 925–929.

[B. Guo et al.(2013)] Guo, B., Zhang, D.,Wang, Z., Yu, Z., Zhou, X.,2013. Opportunistic IoT: Exploring the harmonious interaction between human and the internet of things, Journal of Network and Computer Applications 36(6), 1531 – 1539.

[J. Hamari et al.(2015)] Hamari, J., Sj oklint, M., Ukkonen, A. 2017. The sharing economy: Why people participate in collaborative consumption. Journal of the Association for Information Science and Technology 67 (9).

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

[Kaut et al.(2007)Kaut, Vladimirou, Wallace, and Zenios] Kaut, M., Vladimirou, H., Wallace, S., Zenios, S., 2007. Stability analysis of portfolio management with conditional value-at-risk. Quantitative Finance 7 (4), 397–409.

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

[N. Kong et al.(2013)] Kong N., Schaefer, A. J., Shabbir A.,2013. Totally unimodular stochastic programs. Mathematical Programming 138 (1), 1–13.

[A. Lamghari et al.(2016)] Lamghari, A., Dimitrakopoulos, A., 2016. Progressive hedging applied as a metaheuristic to schedule production in open-pit mines accounting for reserve uncertainty. European Journal of Operational Research 253 (3)843 - 855.

[D. Lindley(1991)] Lindley, D., 1991. Making Decisions, 2nd Edition. John Wiley & Sons, Inc., New York, NY, USA.

[A. Lokketangen et al.(1996)] Lokketangen, A., Woodruff,D. L. 1996. Progressive hedging and tabu search applied to mixed integer (0,1) multistage stochastic programming. Journal of Heuristics 2, 111-128.

[F. Maggioni et al.(2017)Maggioni, Crainic, Perboli, and Rei] Maggioni, F., Crainic, G., Perboli, G., Rei, W., 2017. Reduced cost-based variable fixing in stochastic programming. CIRRELT-2017-10.

[S. Martello et al.(1990)] Martello, S., Toth, S., 1990. Knapsack problems Algorithms and Computer Implementation. John Wiley & Sons.

[J.M. Mulvey et al.(1991)] Mulvey, J. M. and Vladimirou, H., 1991. Applying the progressive hedging algorithm to stochastic generalized networks. Annals of Operation Research, 31 (1), 399–424

[D. A. L. Nuevo et al.(2015)] Nuevo, D. A. L., Valles, D. R., Medina E. M., Pallares, R. M., 2015. OIoT: A Platform to Manage Opportunistic IoT Communities,2015 International Conference on Intelligent Environments, 104–111.

[Perboli et al.(2016)Perboli, Fadda, Gobbato, Rosano, and Tadei] 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.

[G. Perboli et al.(2017)] Perboli, G., Gobbato, L., Maggioni, F. 2017. A progressive hedging method for the multi-path travelling salesman problem with stochastic travel times. Journal of Management Mathematics 28 (1), 65-86.

[Perboli et al.(2012)] Perboli, G., Tadei, R., Baldi, M.M., 2012. The stochastic generalized bin packing problem. Discrete Applied Mathematics 160, 1291–1297.

[V. Pilloni et al.(2017)] Pilloni, V., Atzori, L.,2017. Consensus-based resource allocation among objects in the internet of things. Annals of Telecommunications.

[Pironet Thierry(2015)] 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.

[R. Pozza(2016)] Pozza R., Georgoulas S., Moessner K., Nati M., Gluhak A., Krco S., 2016. An Arrival and Departure Time Predictor for Scheduling Communication in Opportunistic IoT. Zanella A, Mahmoodi T, eds. Sensors (Basel, Switzerland).

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

[R.T. Rockafellar et al.(1991)] Rockafellar, R.T., Wets, R.J.B., 1991. Scenarios and policy aggregation in optimization under uncertainty. Mathematics of Operation Research, 119-147.

[A. Schrijver (1986)] Schrijver,A. 1986. Theory of Linear and Integer Programming. John Wiley & Sons, Inc., New York, NY, USA.

[A. Shabbir(2010)] Shabbir, A.,2010. Two-Stage Stochastic Integer Programming: A Brief Introduction, John Wiley & Sons, Inc., Wiley Encyclopedia of Operations Research and Management Science.

[R. Tadei et al.(2017)Tadei, Perboli, Perfetti] Tadei, R., Perboli, G., Perfetti, F., 2017. The multi-path Traveling Salesman Problem with stochastic travel costs. EURO Journal on Transportation and Logistics 6 (1), 3-23.

[B. Tan et al.(2017)] Tan, B., Cui, H., Wu, J., Chen, C. W., 2017. An Optimal Resource Allocation for Superposition Coding Based Hybrid Digital-Analog System. IEEE Internet of Things Journal (99), 1–1.

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

[J.-P.. Watson et al.(2011)] Watson, J.-P., Woodruff,D. L. 2011. Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. Computational Management Science 8 (4), 355-370.

[Zanella et al.(2014)] Zanella, A., Bui, N., Castellani, A., Vangelista, L., Zorzi, M., 2014. Internet of things for smart cities. IEEE Internet of Things Journal 1 (1), 22–32.

[Zéphyr et al.(2014)] Zéphyr, L., Lang, P., and Lamond, B. F. (2014). Adaptive monitoring of the progressive hedging penalty for reservoir systems management. Energy Systems, 5(2):307-322.

[Zehtabian, et al.(2016)] Zehtabian, S. and Bastin, F. (2016) Penalty Parameter Update Strategies in Progressive Hedging Algorithm. CIRRELT (Collection)