A progressive hedging method for the multi-path travelling salesman problem with stochastic travel times

IMA Journal of Management Mathematics Page 1 of 21
doi:10.1093/imaman/dri000

A Progressive Hedging method for the multi-path Traveling Salesman Problem with stochastic travel times

Guido Perboli
DAUIN, Politecnico di Torino, Turin, Italy and CIRRELT, Montreal, Canada [1]
Luca Gobbato
DAUIN, Politecnico di Torino, Turin, Italy
Francesca Maggioni
Department of Management, Economics and Quantitative Methods, University of Bergamo, Bergamo, Italy e-mail: francesca.maggioni@unibg.it

In this paper we consider a recently introduced problem specifically designed for Smart Cities and City Logistics applications: the multi-path Traveling Salesman Problem with stochastic travel costs (mpTSP$_s$). The mpTSP$_s$ is a variant of the TSP problem where a set of paths exists between any two nodes and each path is characterized by a random travel time. We propose a two stage stochastic programming formulation where tour design makes up the first stage, while a recourse decisions related to the choice of the path to follow are made in the second stage. To solve this formulation we propose an heuristic method inspired by the Progressive Hedging algorithm of Rockafellar & Wets. We then benchmark the solution method by solving model instances derived from the traffic speed sensor network of the city of Turin. Furthermore, the impact of the stochastic travel time costs on the problem solution is examined, showing the benefits of the proposed methodology in both solution quality and computational effort when compared to solving the deterministic equivalent formulation using a commercial solver.

Keywords: TSP, multiple paths, stochastic programming, stochastic travel times, Progressive Hedging.

1. Introduction

In this paper we consider the multi-path Traveling Salesman Problem with stochastic travel times (mpTSP$_s$), a recently introduced variant of the standard TSP related to Smart City and City Logistics applications (Tadei *et al.*, 2014). More precisely, given a graph characterized by a set on nodes connected by arcs, in the mpTSP$_s$ we consider that, for every pair of nodes, we have multiple paths between the two nodes. Each path is characterized by a random travel time which can be decomposed in the sum of a deterministic term and a stochastic term, which represents the travel time oscillation due to the path congestion. In practice, that travel time oscillations randomly depend on time realizations in different dates and time-slots and are very difficult to measure. Similarly to the standard TSP, the aim of the problem is to define the Hamiltonian cycle minimizing the expected total cost.

The problem was defined in a smart city application, the PIE_VERDE project. PIE_VERDE, funded by the European Regional Development Fund (ERDF), aims at developing new planning tools for freight delivery in urban areas by means of environmentally-friendly light duty vehicles. One of the aims of this project is to plan and manage a two-echelon delivery service. Trucks are not allowed to directly enter the city and the freight is consolidated in small peripheral depots. The goods are finally delivered using hybrid vehicles (Perboli *et al.*, 2011). The planning of a hybrid vehicle tour requires the determination of the sequence of clients to visit and the selection of the powertrain configuration during the tour. In fact, hybrid vehicles can change the powertrain configuration during a route, impacting their GHG emissions, and their energy and fuel consumption. These vehicles can be fueled by thermal, thermal-electric or exclusively electric engine thanks to a rechargeable energy storage system able to supplement fossil

[1] To be considered for the ICSP 2013 special issue, guest edited by Patrizia Beraldi, Vittorio Moriggia, and Asgeir Tomasgard

fuel energy for vehicle propulsion. Additionally, some hybrid vehicles use a co-generative thermal engine that exploits braking power to generate electricity while traveling. Hence, the gain in terms of reduction of GHG emissions obtained by a hybrid vehicle varies according to how the different powertrain configurations have been selected during each route. An intelligent planning of the powertrain selection is a key factor in the efficient use of a hybrid vehicle. In (Tadei *et al.*, 2014) the mpTSP_s was introduced for the PIE_VERDE project in order to create a model able to optimize both the freight vehicle tour and the powertrain configuration selection for hybrid vehicles in parcel and courier deliveries. The model is designed to meet several current needs in the context of freight transportation planning. Firstly, there is a need for new routing models that consider the stochasticity of generalized cost functions, which include both operational and environmental aspects. Secondly, new freight distribution business models such as parcel delivery and e-commerce freight delivery allow highly limited computational time for vehicle fleet planning: about 30 minutes for planning the full fleet. Hence, the tours must be computed with a very small effort. Thirdly, in real cases the distribution of the stochastic parameters is unknown and it is not always possible to derive the distribution of these variables from real data. This is because of the small number of hybrid vehicles currently used for freight distribution, the difficulty or impossibility of accessing sensor data and the difficulty of obtaining data from the vehicles' control unit. In fact, one of the main issues when facing real settings is that the solution should also be applicable in cities where a network of sensors and a traffic flow forecaster are not available. In these cases, the vehicle routing is achieved using commercial vehicular navigation systems, which use aggregated mean values for the vehicle speed. Thus, only a very limited knowledge of the dynamic behaviour of the speed on the roads can be assumed to be known (Deflorio *et al.*, 2012).

Beyond this specific case study, other applications of the mpTSP_s arise in the City Logistics context. Cities, today, provide garbage collection, periodic delivery of goods in urban grocery distribution and bike sharing services, among others. These services require the definition of fixed tours to be used for a time span of one to several weeks (see, e.g., CITYLOG Consortium & CITYLOG (2010)). However, within urban areas, paths are affected by the uncertainty of travel time. Travel time distributions differ from one path to another and they are time dependent. Even an approximated knowledge of travel time distribution may be difficult due to the large size of the data involved. The use of the travel time mean (or other measures of expectation) may imply relevant errors when the variance is high (Cagliano *et al.*, 2014; Perboli *et al.*, 2014).

The first objective of this paper is to describe the problems cited above from the stochastic point of view. We will focus on the sources of uncertainty and the effects of the recourse actions, by introducing two two-stage stochastic models with recursions. Both models share the same two-stage decision structure. In the first stage, the design decisions are taken, i.e. the order of nodes in the Hamiltonian cycle is determined, while the choice of the actual path between a pair of nodes is decided in the second stage. The two models originate from different deterministic routing problems and, from the model point of view, consider two different conceptual representations of the mpTSP_s. The first model is an extension of the standard sub-tour elimination model (Cook, 2012), while the second one derives from vehicle routing literature, and in particular from the model of the Two-Echelon VRP problem (Perboli *et al.*, 2011). In particular, the last model uses flow-based constraints to forbid the presence of subtours.

The second contribution of this paper is to propose an effective and efficient approach for solving the mpTSP_s. The flow-based model is more suitable than the other model for resolution by means of commercial MIP solvers. However, we will show by extensive computational results how the size of the two-stage formulations increases rapidly with the number of nodes and that they are only usable for very small instances. Thus, to consider the instances arising in real applications, more advanced methods are needed. For this reason, starting from the sub-tour elimination formulation, we introduce a meta-heuristics based on the Progressive Hedging algorithm by Rockafellar & Wets (1991). Our method first decomposes the problem by scenarios and then reflects the differences between the different scenario solutions in terms of sequence of nodes in the Hamiltonian cycle. The differences in terms of the Hamiltonian cycle design are thus reflected as penalties in the objective function. In this way each scenario problem becomes addressable by means of an exact solver while providing information on how to use the local solution of a subproblem to push the overall algorithm to convergence. Extensive computational results on realistic instances show how the developed method is able to reproduce optimal solutions, while also addressing instances which are unaddressable by commercial solvers due to their size.

The paper is organized as follows. In Section 2 the main literature on the mpTSP$_s$ is given. In Section 3 the two-stage models with recursion are presented, while the Progressive Hedging algorithm is introduced in Section 4. The computational results of the recourse models and the Progressive Hedging method are then given in Section 5. Finally, the lessons learned are summarized in Section 6, where future directions for the research on the mpTSP$_s$ are also outlined.

2. Literature review

While different stochastic and/or dynamic variants of TSP (and more in general of vehicle routing problem) are present in the literature (Gendreau *et al.*, 1996; Golden *et al.*, 2008; Pillac *et al.*, 2013), the mpTSP$_s$ has been recently defined by Tadei *et al.* (2014). For this reason, we also consider some relevant literature on similar problems, highlighting the main differences with the problem faced in this paper.

In Tadei *et al.* (2014) the authors introduce the problem and derive a deterministic approximation. In particular, under a mild hypothesis on the unknown probability distribution of the travel time for the different paths, the deterministic approximation becomes a TSP problem where the minimum expected total travel time is equivalent to the maximum of the logarithm of the total accessibility of the Hamiltonian tours to the path set. The quality of the deterministic approximation is then evaluated by comparing it with the Perfect Information results obtained by means of a Monte Carlo method. The comparison shows a good accuracy of the deterministic approximation, with a reduction of the computational times of two orders of magnitude.

One of the search directions indicated in Tadei *et al.* (2014) was the extension of the instance sets in the literature in order to better represent real City Logistic settings. Maggioni *et al.* (2014b) make a first attempt in this direction. In details, the authors introduce a standard methodology to build realistic multi-path instances starting from the real data collected by a network of flow and speed sensors and apply their methodology to the mpTSP$_s$. The results show how the basic data extraction methodology is valid, but more work is needed to fully catch the peculiar setting of urban freight delivery.

In the literature several stochastic variants of the TSP problems can be found. In these problems a known distribution affecting some problem parameters is given and the theoretical results are strongly connected with the hypotheses on such distribution. The main sources of uncertainty are related to the arc costs (Leipala, 1978; Toriello *et al.*, 2012) and the subset of cities to be visited with their location (Jaillet, 1988; Goemans & Bertsimas, 1991). A different type of stochastic behaviour has been considered in Bertazzi & Maggioni (2014a,b), where a stochastic capacitated traveling salesmen location problem with uncertainty given by the location of the customers has been solved with different solution approaches.

If we consider general routing problems, different types of uncertainty and dynamics can be considered. The most studied variants are related to the online arrival of customers, with the requests being both goods (Hvattum *et al.*, 2006, 2007; Ichoua *et al.*, 2006; Mitrović-Minić & Laporte, 2004) and services (Beaudry *et al.*, 2010; Bertsimas & Van Ryzin, 1991; Gendreau *et al.*, 1999; Larsen *et al.*, 2004). Only in recent years the dynamics related to travel times has been considered in the literature (Chen *et al.*, 2006; Fleischmann *et al.*, 2004; Güner *et al.*, 2012; Kenyon & Morton, 2003; Tagmouti *et al.*, 2011; Taniguchi & Shimamoto, 2004), while, to the best of our knowledge, service time has not been explicitly studied. The last variants of vehicle routing problems are related to the dynamically revealed demands of a known set of customers (Novoa & Storer, 2009; Secomandi & Margot, 2009) and the vehicle availability (Li *et al.*, 2009a,b; Mu *et al.*, 2011). For a recent review, the reader can refer to Pillac *et al.* (2013).

All the papers presented in this survey deal with uncertainty and/or dynamic aspects of the routing problems where the magnitude of the uncertainty is limited and the parameter values are revealed in a time interval compatible with the operations optimization. Then the multi-path aspects can be ignored, being possible an a priori choice of the path connecting the two nodes. In our case, the mpTSP$_s$ is thought to be used for planning a service. Thus, the enlarged time horizon as well as strong dynamic changes in travel times due to traffic congestion and other nuisances typical of the urban transportation induce the presence of multiple paths connecting every pair of nodes, each one with its stochastic cost. This is, to our knowledge, an aspect of the transportation literature considered only in transshipment problems, where the routing aspect is heavily relaxed (Baldi *et al.*, 2012; Maggioni *et al.*, 2009; Tadei *et al.*, 2012).

When we consider other routing problems, a large literature is available for the stochastic Shortest Path. One of the few papers directly dealing with multiple path is due to Eiger *et al.* (1985). In their paper the authors consider an extension of the classical shorted route problem where multiple arcs interconnect the nodes and the costs are uncertain. In particular, they show how, when the preferences between the arcs are linear or exponential distributed, a Dijkstra-type algorithm using the mean of the distributions finds an optimal path. Unfortunately, the results are strictly related to the specific problem, the shortest path, and to the presence of a Dynamic Programming solution method. Moreover, differently from our case, the preferences must be exponential distributed, while we assume that only the right tail converges to an exponential distribution. Psaraftis & Tsitsiklis (1993) introduce a variant of the stochastic Shortest Path where the arc costs are stochastic and dynamic, in the sense that the arc cost is a known function of a certain environment variable which depends on the origin node i of the arc the time in which we are leaving from node i. Differently to our case, not only there is only one path associated to each node, but the environment variable associated to each arc is an independent stochastic process associated to a finite-state Markov process with a known transition probability matrix. Thus, this approach is not suitable to urban transportation, where the estimation of the Markov process could be not usable in practice. Finally, Jaillet & Melvyn Sim (2013) recently propose criteria to design shortest paths when deadlines are imposed to the nodes and the goal of the problem is to minimize the deviation of the actual arrival time with respect to the desired one. They also show that the stochastic shortest path with deadlines under uncertainty can be solved in polynomial time when there is stochastic independence between the arc travel times. Even in this case between any pair of node only one arc exists.

3. Two-Stage stochastic models with recourse for the mpTSP$_s$

To include the random nature of the travel time process in a urban context, we consider a two-stage stochastic linear program with recourse (see e.g. (Birge & Louveaux, 2011)). The travel time oscillation Δ_{ij}^{k} by using path k between nodes i and j, is then described by a stochastic process represented using a discrete random variable. All possible discrete values that the random variable can assume, is represented by a finite set of vectors, named scenarios. We represent each realization (scenario) of random travel time oscillation process by Δ_{ij}^{ks}. We denote with S the set of time scenarios, and the probability of each scenario $s \in S$ by p_s. In two-stage stochastic programming, we explicitly classify the decision variables according to whether they are implemented before or after an outcome of the random travel time variable of each path is observed. In the multi-path traveling salesman problem, in the *first stage* the decision maker does not have any information about the travel time oscillation. However, the routing among the nodes should be determined before the complete information is available. Thus, the first-stage decision variable y_{ij} is represented by the selection of nodes i and j to be visited in a tour. In the second stage, travel time oscillations are available and the paths k between each couple of nodes i and j under scenario s, x_{ij}^{ks} can be calculated. The objective of the two-stage stochastic model with recourse for the mpTSP$_s$ is the minimization of the total cost due to paths congestion.

A two-stage modeling, implies a strong simplification in the learning process, i.e. the path is chosen when the tour starts, with full knowledge of travel times. Hence, no learning takes place as the vehicle moves along. This type of approximation is meaningful from a practical point of view for two reasons.

First, the distribution of stochastic travel times in a multi-stage setting is unknown: it is not easy and not always possible to derive them from real data because of the small number of hybrid vehicles used for freight distribution, the difficulty or impossibility to access data from the vehicles' control unit and sensor data. Second, the model should be applicable also in cities where a network of sensors and a traffic flow forecaster are partially implemented or, more often, completely unavailable. In these cases, the vehicle routing is achieved using commercial vehicular navigation systems, which use aggregated mean values for the vehicle speed. Thus, only a very limited knowledge of the dynamic behaviour of the speed in the roads can be assumed to be known. Then, by considering only a two-stage model we can explicitly introduce the uncertainty of the travel times without limiting to much the possibility to apply our model to real last mile applications (Cagliano *et al.*, 2014; Perboli *et al.*, 2014; Deflorio *et al.*, 2012).

In the next subsections we consider two different models for this problem:

1. A sub-tour elimination based two-stage stochastic model with recourse, which is an is an extension of the classical deterministic sub-tour elimination model (Cook, 2012);

2. A flow-based two-stage stochastic model with recourse derived from the MIP model of the Two-Echelon VRP problem by Perboli *et al.* (2011).

We introduce two models to express the same problem because each has its own characteristic in terms of structure and size. In fact, the flow-based model is more suitable to be solved by commercial MIP solvers like CPLEX, whilst the sub-tour elimination model has characteristics which make it ideally solved by the Progressing Hedging approach we propose.

3.1 *A sub-tour elimination based two-stage stochastic model with recourse for the mpTSP$_s$*

First we consider a sub-tour elimination based two-stage stochastic model with recourse.

Let N and U respectively be the finite set of nodes of the graph and a subset of nodes in N, while K_{ij} defines the set of paths between the pair of nodes $i, j \in N$. Let S be the set of time scenarios, each scenario being associated with a probability p_s. Each path $k \in K_{ij}$ between nodes $i, j \in N$ is characterized by a non-negative estimation of the mean unit travel time cost $\bar{c}_{ij}$ and a non-negative unit random travel time cost c_{ij}^{ks} under the time scenario $s \in S$. Let $\Delta_{ij}^{ks} = c_{ij}^{ks} - \bar{c}_{ij}$ be the error on the travel time cost estimated for the path $k \in K_{ij}$ under time scenario $s \in S$.

Let the first-stage variables be $y_{ij} = 1$ if node $j \in N$ is visited just after node $i \in N$ and 0 otherwise. Whilst, x_{ij}^{ks} is equal to 1 if path $k \in K_{ij}$ between nodes $i, j \in N$ is selected at the second-stage and 0 otherwise.

The deterministic equivalent formulation of the sub-tour elimination based two-stage stochastic model with recourse is as follows:

$$\min_{\{y,x\}} \left[\sum_{i\in N} \sum_{j\in N} \bar{c}_{ij} y_{ij} + \sum_{s\in S} p_s \sum_{i\in N} \sum_{j\in N} \sum_{k\in K_{ij}} \Delta_{ij}^{ks} x_{ij}^{ks} \right] \tag{3.1}$$

subject to

$$\sum_{j \in N: j \neq i} y_{ij} = 1 \qquad i \in N \tag{3.2}$$

$$\sum_{i \in N: i \neq j} y_{ij} = 1 \qquad j \in N \tag{3.3}$$

$$\sum_{i \in U} \sum_{j \notin U} y_{ij} \geqslant 1 \qquad \forall\, U \subset N \tag{3.4}$$

$$\sum_{k \in K_{ij}} x_{ij}^{ks} = y_{ij} \qquad i \in N, \quad j \in N, \quad s \in S \tag{3.5}$$

$$x_{ij}^{ks} \in \{0,1\} \qquad k \in K_{ij}, \quad i \in N, \quad j \in N, \quad s \in S \tag{3.6}$$

$$y_{ij} \in \{0,1\} \qquad i \in N, \quad j \in N \tag{3.7}$$

Problem (3.1)-(3.7) is a large-scale binary problem. The first sum in the objective function (3.1) represents the first-stage travel cost, while the second sum represents the recourse action, consisting in choosing the best path $k \in K_{ij}$ under time scenario realization $s \in S$. Notice that, due to the two-stage nature of the model, and the consequent revelation of the uncertainty, the recourse action in (3.1) reduces to:

$$\sum_{s \in S} p_s \sum_{i \in N} \sum_{j \in N} \sum_{k \in K_{ij}} \Delta_{ij}^{ks} x_{ij}^{ks} = \sum_{s \in S} p_s \sum_{i \in N} \sum_{j \in N} \min_{k \in K_{ij}} \Delta_{ij}^{ks} x_{ij}^{ks}. \tag{3.8}$$

Constraints (3.2) and (3.3) are the standard first-stage assignment constraints and (3.4) represents the first-stage sub-tour elimination constraint. Constraint (3.5) guarantees that path k between nodes i and j can be chosen at stage 2 only if nodes i and j were part of the tour fixed at stage 1. Finally, the integrality constraints (3.6)-(3.7) define the second- and first-stage decision variables of the problem. By solving problem (3.1)-(3.7), one finds a single tour y_{ij}, $\forall i, j \in N$, with minimum travel time cost overall scenarios included in S. The main problem when dealing with this formulation is represented by constraints (3.4), which are exponential in their number and are suitable to dynamic cut generation methods only (Applegate *et al.*, 2007). This makes difficult to incorporate such a model in a MIP solver. On the other hand, as we will show in Section 4, this model makes quite easy to define an efficient Progressive Hedging-based algorithm.

3.2 *A flow-based two-stage stochastic model with recourse for the mpTSP$_s$*

We consider now a flow-based two-stage stochastic model with recourse for the mpTSP$_s$. The notation adopted is as in the previous subsection. A first stage real variable ϕ_{ij} associated to the flow on the arc (i, j) is also introduced.

The deterministic equivalent formulation of the flow-based two-stage stochastic model with recourse is as follows:

$$\min_{\{y,x\}} \left[\sum_{i \in N} \sum_{j \in N} \bar{c}_{ij} y_{ij} + \sum_{s \in S} p_s \sum_{i \in N} \sum_{j \in N} \sum_{k \in K_{ij}} \Delta_{ij}^{ks} x_{ij}^{ks} \right] \tag{3.9}$$

subject to

$$\sum_{j\in N:j\neq i} y_{ij} = 1 \qquad i \in N \tag{3.10}$$

$$\sum_{i\in N:i\neq j} y_{ij} = 1 \qquad j \in N \tag{3.11}$$

$$\sum_{i\in N:i\neq j} \phi_{ij} - \sum_{k\in N:k\neq j} \phi_{jk} = 1 \qquad \forall j \in N \setminus \{1\} \tag{3.12}$$

$$\sum_{i\in N:i\neq 1} \phi_{i1} - \sum_{k\in N:k\neq 1} \phi_{1k} = 1 - |N| \tag{3.13}$$

$$\sum_{k\in N:k\neq 1} \phi_{1k} = |N| \tag{3.14}$$

$$\phi_{ij} \leqslant |N| y_{ij} \qquad i \in N, \quad j \in N \tag{3.15}$$

$$\sum_{k\in K_{ij}} x_{ij}^{ks} = y_{ij} \qquad i \in N, \quad j \in N, \quad s \in S \tag{3.16}$$

$$x_{ij}^{ks} \in \{0,1\} \qquad k \in K_{ij}, \quad i \in N, \quad j \in N, \quad s \in S \tag{3.17}$$

$$y_{ij} \in \{0,1\} \qquad i \in N, \quad j \in N \tag{3.18}$$

While the meaning of the objective function (3.9) and the constraints (3.10) and (3.11) is the same of (3.1), (3.2), and (3.3) in the previous model, the sub-tour elimination constraints (3.4) are rewritten by means of the constraints (3.12), (3.13), (3.14), and (3.15). Without loosing in generality, let us consider the node 1 as the starting point of our tour. Constraints (3.13) and (3.14) force the node 1 to have an outbound flow equal to to the number of nodes $|N|$ and an inbound with value 1. Thus, the subtours are forbidden by constraint (3.12), which obliges every node to reduce by 1 the outbound flow if compared to its inbound one. If a subtour exists, this constraint is violated by at least one of the nodes in the subtour (see Perboli *et al.* (2011) for further details). Finally, constraint (3.15) links the binary variables y_{ij} to the existence of a flow in an arc. This constraint, with the constraints (3.10) and (3.11), force to have exactly one arc with a non zero flow both as inbound and outbound.

This flow-based model is more suitable to be solved by means of MIP solvers, involving $O(N)$ constraints instead of the exponential number of constraints (3.4).

4. A Progressive Hedging algorithm for the mpTSP$_s$

This section briefly introduces a variant of the Progressive Hedging (PH) algorithm by Rockafellar & Wets (1991) to the sub-tour formulation (3.1) - (3.7).

The setup of the algorithm is the same of that presented by Crainic *et al.* (2011) for the stochastic network design. Starting from the sub-tour elimination based model, we first define a copy of first stage variables $y_{ij}^s \in \{0,1\}$, $\forall i, j \in N$ for each scenario $s \in S$. The objective function (3.1) is now expressed as

$$\min_{\{y,x\}} \sum_{s\in S} p_s \left[\sum_{i\in N} \sum_{j\in N} \bar{c}_{ij} y_{ij}^s + \sum_{i\in N} \sum_{j\in N} \sum_{k\in K_{ij}} \Delta_{ij}^{ks} x_{ij}^{ks} \right]. \tag{4.1}$$

To maintain the consistency of the problem, we also introduce non-anticipativity constraints (4.2), which makes sure the decisions on the tour are not tailored according to the scenarios considered in S. All the scenario tour variables must be equal to each other to produce a single implementable tour $\bar{y_{ij}} \in \{0,1\}$, $\forall i, j \in N$.

$$y_{ij}^s = \bar{y}_{ij} \qquad i, j \in N, \quad s \in S \tag{4.2}$$

We can note that relaxing constraints (4.2) the problem becomes scenario separable. So, we apply a scenario decomposition technique based on augmented Lagrangian relaxation to separate the stochastic problem following the scenarios of the scenario tree. The Lagrangian multipliers are used to penalize a lack of implementability due to differences in the first-stage variables values among scenario subproblems. The results of this decomposition is a series of deterministic scenario subproblems that can be reduced to a classical TSP and, thus, solved by means of a specialized method.

Then, the PH proceeds in two phases. Phase 1 aims to obtain consensus among the subproblems. At each iteration, the subproblems are first solved separately. Concorde (Applegate *et al.*, 2007; Cook, 2012) is used to address these problems, which, embedding a cutting-plane algorithm within a branch-and-bound search, is currently one of the best procedures for the TSP. By using the average function, their solutions are then aggregated into the overall tour $\bar{y}$. It is very important to note that the overall tour does not necessarily produce an overall feasible tour. However, the search process is gradually guided toward scenario consensus by adjusting the Lagrangian multipliers based on the deviations of the scenario solutions from the overall tour. It should be noted that although a feasible reference solution can be produced with a constructive heuristic, whenever the consensus is low, it may wrongly bias the search process of penalization of the non-consensus scenario solutions.

The search process continues until the consensus is achieved or one of the termination criteria is met (i.e. the maximum limit of CPU time of 1 hour or 200 iterations of the algorithm). When consensus is not achieved in the first phase, phase 2 solves the restricted mpTSP$_s$ obtained by fixing the first-stage variables for which consensus has been reached, e.g., the arcs used in all the scenario subproblems.

Finally, given the straightforward parallelization of the PH algorithm, parallel computation has been used to address the mpTSP$_s$. In the current version of the algorithm, a simple master-slave synchronous strategy is implemented, where the master controls the search, computes the global design, and performs the parameters updates, while the slave processors modify the costs associated to the arcs of the graph and solve the resulting scenario subproblems. Moreover, the master can assist slaves by solving scenario subproblems when the workload is unbalanced. Synchronization is performed at the end of each iteration.

5. Computational results

In this section, we present and analyze the results of the computational experiments. The goal is to evaluate the effectiveness of the methodology proposed for the mpTSP$_s$.

We compare the PH-based method solutions with the solutions obtained by the recourse problem (*RP*), solved by means of CPLEX 12.5 MIP solver (IBM ILOG, 2012). Moreover, we analyze the loss in terms of quality of the solution due to the use of the expected value approaches (EV) for mpTSP$_s$ by measuring the *Value of Stochastic Solution* (VSS) and the importance of the stochasticity by the *Expected Value of Perfect Information* ($EVPI$) (Birge, 1982), (Maggioni & Wallace, 2012) and (Maggioni *et al.*, 2014a). In particular, we use the aforementioned measures both to qualify our method and to discuss some economically relevant issues from the *last mile* point of view.

Subsection 5.1 introduces the instance sets and test environment. A discussion on the size of the scenario tree is discussed in Subsection 5.2, while a validation of the PH algorithm is presented in Subsection 5.3. Finally, considerations on the importance of using a stochastic model for the mpTSP$_s$ are given in Subsection 5.4.

5.1 *Instance sets and test environment*

No real-life instances are present in the literature for this stochastic version of the TSP problem. For this reason, we generate an instance set based on the real traffic sensor network of Turin, a medium sized city in Italy. The instance set considered allows to better understand also other real cases of City Logistics applications.

According to the guidelines presented in (Tadei *et al.*, 2014), instances are characteristics by the following inputs:

- Instance size. We considered instances with a number of nodes up to 200. This number is of the same order of magnitude of a day trip of the main parcel and courier delivery services as TNT and DHL. In particular, we split those instances into three sets: instances with up to 50 nodes (N50), up to 100 nodes (N100) and up to 200 nodes (N200).

- Nodes. Given a square of 14 km edge, which is equivalent to a medium sized city like Turin, nodes are mapped in such portion of plane and then partitioned into two subsets:
  - Central nodes: the nodes belonging to city center, which are the nodes in the circle with the center coincident with the geometric center of the 14 km square and a radius equals to 7 km;
  - Suburban nodes: the nodes which are not central.

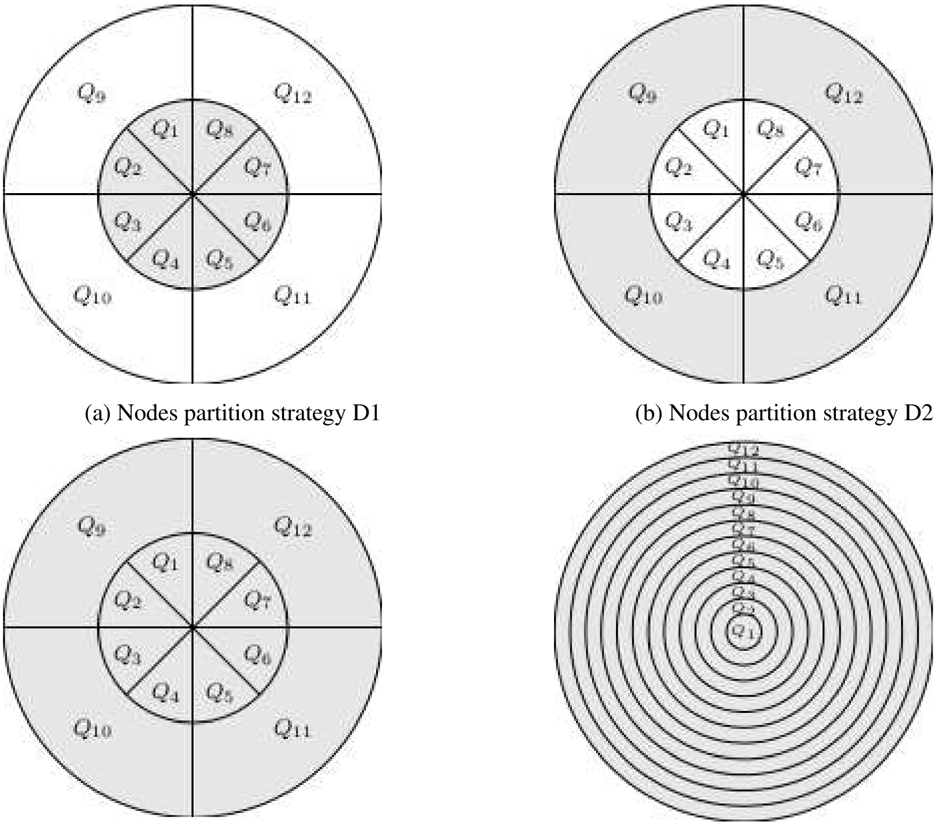
- Nodes distribution strategies. In order to define the spatial distribution of nodes,we divided the portion of plane in 12 neighborhoods (Q1 to Q12), 8 in the city center and 4 in the suburban area. For each neighborhood, nodes are randomly generated according with the following distribution strategies:
  - D1: the nodes are distributed only in the city center;
  - D2: the nodes are distributed only in the suburban area of the city;
  - D3: the nodes are distributed in all neighborhoods both central and suburban in ratio 3:1 respectively;
  - D4: the nodes are distributed in all neighborhoods both central and suburban in ratio 1:1 respectively.

  Figure 1 shows the subdivision of the 14 km square in neighborhoods and, for each distribution strategy, the neighborhoods involved.

- Multiple paths. The number of paths between any pair of nodes is set to 3. The choice of this number is related to hybrid vehicles applications, where the typical number of power train modes is 3 (Tadei *et al.*, 2014).

- Pair of nodes types: the pairs of nodes can be homogeneous or heterogeneous.
  - Homogeneous: they are pairs of nodes where the starting node i and the destination node j are both central or suburban. In this case all the multiple paths between the nodes present the empirical speed profile of a central or suburban speed sensor, respectively.
  - Heterogeneous: they are pairs of nodes where the starting node i and the destination node j belongs to a different subset. In this case the multiple paths between the nodes present the empirical speed profile of a central speed sensor for $1/3$ of the paths and a suburban one for the $2/3$ of them.

- Speed data. We build central and suburban speed profiles from real data on the traffic of Turin available at the website *http://www.5t.torino.it/5t/*. The data of the mean vehicle speed, expressed in kilometers per hour (km/h), are accessible with an accuracy of 5 minutes. We aggregated them into blocks of 30 minutes, for a total of 48 observations per day. The instances refer to 50 central speed sensors locations and 100 suburban ones in the period since 13 to 17 February 2013 (see the two circles in Figure 2, giving the distribution of the actual sensors).



(a) Nodes partition strategy D1

(b) Nodes partition strategy D2

(c) Nodes partition strategy D3

(d) Nodes partition strategy D4

FIG. 1: Subdivision in neighborhoods and admissible areas (grey) for each distribution strategy

- Scenario tree generation. Scenario generation is an important part of the modeling process, since a bad scenario tree can lead to a not meaningful solution of the optimization problem. In most practical applications, the distributions of the stochastic parameters have to be approximated by discrete distributions with a limited number of outcomes. The discretization is usually called a *scenario tree*.

  We assume that the random variable *travel time oscillation* has a finite number of possible outcomes at the end of the period considered. All possible discrete values that the random variable can assume, are represented by a finite set of scenarios and are assumed to be exogenous to the problem. Consequently, the probability distribution is not influenced as well by decisions. Making these assumptions we can represent the stochastic process travel time oscillation using a scenario tree which contains a root and finite set of leaves. Several methods are adopted adopted in literature for discretizing the distribution and generating scenarios. Among the most common we list: conditional sampling, sampling from specied marginals and correlations, moment matching, path-based methods and optimal discretization. For a short overview of scenario generation methods see (Kaut & Wallace, 2007).

  In the problem under consideration, the random vector is high dimensional, and presents complicated dependencies among arcs and paths. These factors makes the uncertainty very difficult to represent. Due to the limited availability of dynamically specified real data, we decided to generate empirical velocity profile distributions v_{ij}^{ks} associated to the path k between i and j under scenario s

as inverse of the Kaplan-Meier estimate of the cumulative distribution function (also known as the empirical cdf) of the real data on the speed of traffic of Turin.

Due to the two-stage nature of the model and the simplification of the information structure, all the velocities on an Hamiltonian path refer to the same time block and the correlations between the velocity on an arc and the next one in the path are not considered. An investigation of a more suitable scenario generation method which takes into account correlations among arcs and paths both in multi-period and in multi-stage fashions will be addressed in a future work, when real data will be available.

From the empirical speed distribution, several scenarios trees of increasing size are generated both for the central and the suburban areas. A discussion on the influence of size of the scenario tree with tests on the quality on the solution of the optimization model is presented in Subsection 5.2.

- Time blocks. In order to represent different traffic flows cases, we use data corresponding to 8.00, 12.00 and 16.00, which represent the hours of maximum oscillation of the travel times due to traffic congestions. Note that this does not imply that the tours must be shorter than 4 hours.

- Path travel times. The travel time c_{ij}^{ks} is a function of the Euclidean distance between nodes i and $j \in N$, EC_{ij}, the type of pair of nodes, and the empirical velocity profile distributions v_{ij}^{ks} associated to the path k between i and j under scenario s. In details, this travel time has been computed as

$$c_{ij}^{ks} = \frac{EC_{ij}}{v_{ij}^{ks}} \tag{5.1}$$

and

$$\bar{c}_{ij} = \mathbb{E}_{s \in S} \frac{EC_{ij}}{\mathbb{E}_{k \in K_{ij}} \left[v_{ij}^{ks} \right]} \tag{5.2}$$

is the average travel time over all scenarios $s \in S$ when an average empirical velocity is considered for all path $k \in K_{ij}$ between nodes i and j, where $\mathbb{E}_{s \in S}$ and $\mathbb{E}_{k \in K_{ij}}$ are the expectation operators. The random travel time oscillations are then computed as

$$\Delta_{ij}^{ks} = c_{ij}^{ks} - \bar{c}_{ij} = \frac{EC_{ij}}{v_{ij}^{ks}} - \mathbb{E}_{s \in S} \frac{EC_{ij}}{\mathbb{E}_{k \in K_{ij}} \left[v_{ij}^{ks} \right]} \tag{5.3}$$

For each combination of the parameters mentioned above (i.e., 3 graph's sizes, 4 nodes distribution strategies, and 3 time blocks) we define 5 graphs. Moreover, for each graph, 10 different sets of scenarios S are generated according with the scenario tree generation method. This gives us a final set of 1800 instances.

We compare the PH results with the ones of the two-stage stochastic models. Being the two models equivalent and following the computational results presented by Perboli *et al.* (2011), we consider the flow-based two-stage stochastic model, using CPLEX 12.5 as MIP solver. We give now a short note about the usage of commercial solvers with two-stage stochastic models. We made some tests on other commercial and open-source solvers and the two leading ones were CPLEX and Gurobi. The latter is usually quicker than CPLEX in finding the optimal solution, but its computational effort is larger than CPLEX when having to prove the optimality of the solutions. Thus, we can say that if one is interested in finding a good solution Gurobi is the best choice, but if one needs the proven optimality CPLEX represents the best choice. A time limit of 2 hours of computational time and 8 CPU was given to CPLEX.

PH was implemented in C++ and compiled by means of Visual Studio 2012. The Concorde TSP solver is used in the PH to optimally solve the TSP subproblems instances (Applegate *et al.*, 2007). All

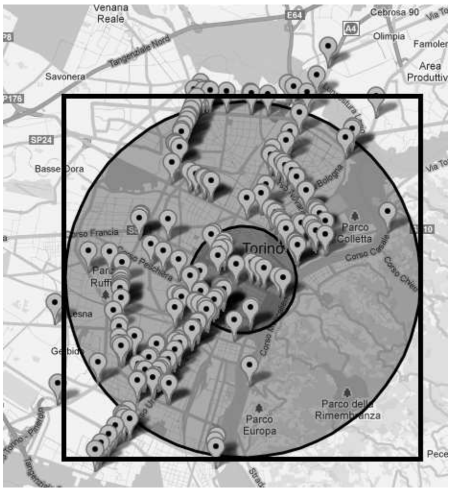


FIG. 2: Distribution of central (dark gray circle) and suburban (light gray circle) speed sensors in the city of Turin in Italy. Figure taken from (Tadei *et al.*, 2014).

the tests were performed on an Intel I7 2 GHz workstation with 8 GB of RAM. Concerning the parallel computation, CPLEX and Progressive Hedging are executed with a limit of 8 CPU. PH has an additional limit of 200 iterations of the algorithm.

5.2 *Determine the size of the scenario tree*

In order to qualify the results of both the PH and the two-stage models, we first perform a tuning of the number of scenarios needed in the tree to obtain stable results. The first finding is that the two-stage models, from a purely computational perspective, cannot reach the optimality with a reasonable time limit (5 hours) for instances with 100 customers when the number of scenarios $S \geqslant 50$ (e.g. the average optimality gap after 5 hours is greater than 3%). Moreover, even with such a small number of scenarios, they become impracticable with more than 100 customers. This is mainly due to the size of the resulting model instances in terms of number of variables. For these reasons, we performed a sensitivity analysis of the objective function in terms of the number of scenarios S using the PH. In order to perform this task we used a subset of 180 instances defined by selecting a set of scenarios for each combination of the parameters mentioned in Section 5.1. In order to tune the right number of scenarios to include in the tree, we computed both in-sample and out-of-sample stability measures as described in (Kaut & Wallace, 2007). Table 1 and Figure 3 show in-sample values for increasing sizes of the scenario tree. This is obtained by comparing the optimal objective function values using scenario trees of increasing size. However, we have to remember that in-sample values are not directly comparable. To be able to estimate the effect of using a larger scenario tree, we have to compare the out-of-sample costs. For this purpose, we declare a scenario tree with $S = 500$ to be the true representation of the real world and we use it as a benchmark to evaluate the cost of the optimal solutions obtained using scenario trees with a smaller size. The results are presented in Figure 4, where in the horizontal axis we report the number of scenarios considered and in the vertical axis the percentage gap of the objective functions over 10 runs with respect to the true cost associated to the benchmark scenario tree with $S = 500$. According to the numerical results, out-of-sample stability is reached with a sufficient precision (under 1%) already around 100 scenarios. This value is satisfactory, considering that the raw data for the empirical distributions come from sensors, which are introducing a sensing error larger than 1%.

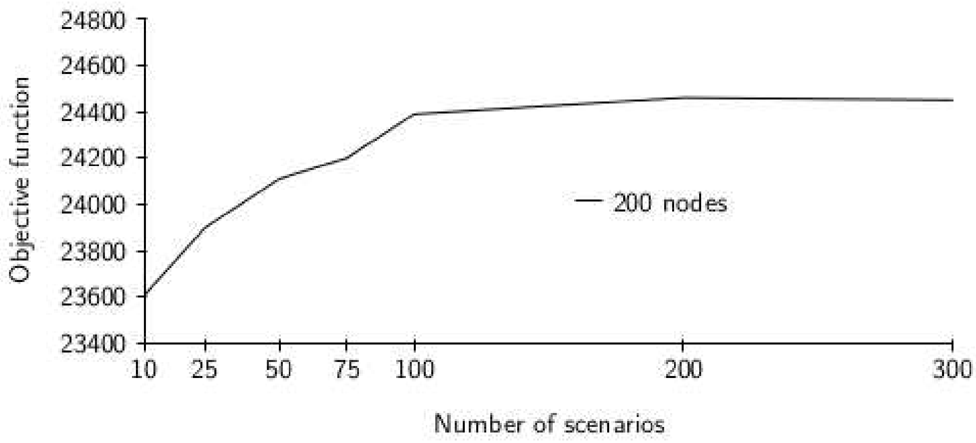


FIG. 3: Convergence of the function value for an increasing number of scenarios

TABLE 1 *Convergence of the function value for an increasing number of scenarios*

	$S = 10$	$S = 25$	$S = 50$	$S = 75$	$S = 100$	$S = 200$	$S = 300$
N50	11550.58	11641.97	11693.72	11785.25	11852.24	11850.14	11855.31
N100	16341.24	16602.04	16649.19	16751.29	16871.24	16898.28	16870.87
N200	23608.15	23900.48	24110.57	24200.97	24390.31	24460.42	24450.86

Another factor influencing the choice of a suitable size of the scenario tree arises from computational considerations. The computational time required to solve the mpTSP$_s$ increases very rapidly with the size of the scenario tree. By increasing the number of scenarios, the number of subproblems to be solved at each iteration increases as well as the number of PH iterations needed to find a consensual solution. More in detail, let $T100$, $T200$ and $T300$ be the computational time of PH with a scenario tree of 100, 200 and 300 scenarios, respectively. Although the computational time increases linearly for scenario trees with less than 100 scenarios, $T200$ is in average 3 times $T100$, while $T300$ is in average greater than 6 times $T100$. Then, we considered a scenario tree with $S = 100$ ensuring in-sample and out-of-sample stability with a reasonable computational effort.

5.3 *Validation of the Progressive Hedging algorithm*

This set of tests aims to qualify the usage of the PH method as a solution method for the mpTSP$_s$. We compare the PH results with the ones of the two-stage models solved by CPLEX (see Table 2). This is performed on $N50$ instance set only, being the only one where CPLEX is able to solve and to prove optimality for all the instances in the time limit. We also tried to increase the time limit for CPLEX, but, while the number of nodes increases, the optimality gap is not reached. This is mostly due to the large number of variables involved in the two-stage formulation. The first two columns reports the instance set and the node distribution strategy. Column 3 and 4 give the percentage gap $Gap_{\%}$ and the standard deviation σ_{Gap} between the optimal solution given by CPLEX and the PH one. Finally columns 5 and 6 provide the computational time in seconds of the two solution methods, respectively.

The computational results show how the PH is accurate, with a gap from CPLEX of less than 0.1% and stable with all the node distribution strategy. Standard deviation is less than 0.038% for all instances considered. By considering the computational effort, CPLEX is more efficient in distribution strategies

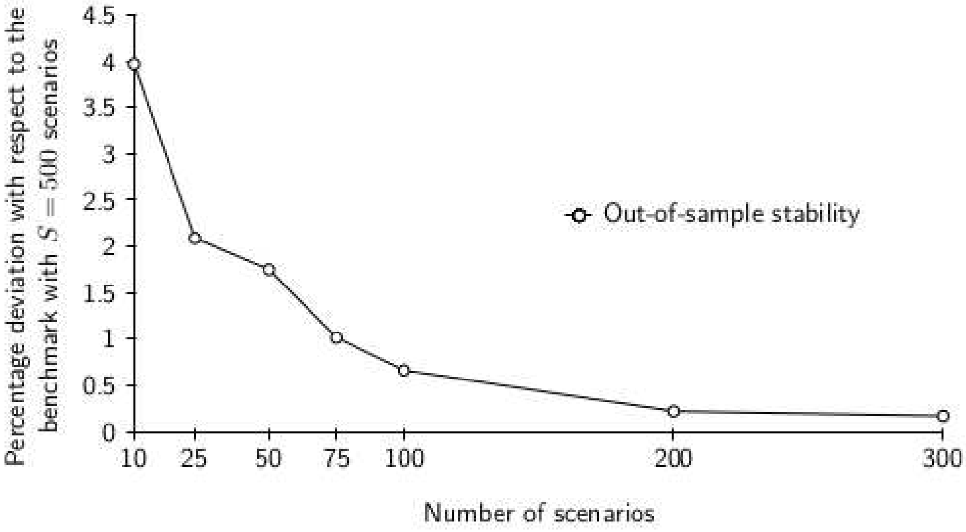


FIG. 4: Out-of-sample stability with respect to the size of the scenario tree

TABLE 2 *Comparison of the results between CPLEX and the Progressive Hedging algorithm*

N	D	$Gap_{\%}$	σ_{Gap}	T_{CPLEX} [s]	T_{PH} [s]
N50	D1	0.04	0.031	60.6	170.8
	D2	0.03	0.024	55.9	179.3
	D3	0.04	0.037	138.8	171.9
	D4	0.05	0.038	162.4	181.9

$D1$ and $D2$, but the results are similar to PH in $D3$ and $D4$. Notice that from the computational efficiency point of view, the PH pays an implementation of the parallel strategy which is a simple (and often inefficient) master-slave one. More complex parallel paradigms might be implemented in order to speed up the PH.

When we increase the number of nodes, the computational time of the PH increases almost linearly with the number of nodes, while CPLEX is unable to solve to optimality the instances in set $N100$ in the given time limit. Moreover, in these instances the solution found by CPLEX is of the same quality or worst than the one found by the PH.

Up to $N100$ the PH never reaches the time limit and the iteration one, while the iteration limit is reached in some of the $N200$ instances. From a pure computer architecture point of view, the PH uses less resources than CPLEX. In fact, CPLEX requires up to 2GB for $N50$ and more than 4GB for $N100$ instances, thus imposing the use of a 64 bit machine. On the contrary, the PH algorithm never uses more than 300 MB even when it addresses $N200$ instances. Due to the accuracy and the efficiency at dealing with larger size instances, we use for the results presented in the next sections the PH as the solver for the mpTSP$_s$.

In our experimental work, we found that PH scales up nicely, as the computational complexity increases almost linearly with the number of nodes. On the other hand, the speedup achievable via parallelization is almost linear in the number of worker threads. For N50 we achieved a speedup of 5.6 using 8 parallel threads; for larger instances (e.g. N100 and N200) the average speedup is 4.2.

5.4 *Analyzing the impact of uncertainty*

This section is devoted to qualify the mpTSP$_s$ by showing the benefits of using the two-stage with recursion models when compared to the Perfect Information case (the so-called *Wait and see* approach WS) and the *Expected Value Problem* EV. This is done by considering the well known *Expected Value of Perfect Information* ($EVPI$) and the *Value of Stochastic Solution* (VSS) measures (Birge & Louveaux, 2011). The Expected Value of Perfect Information is defined by the difference between the objective values of the stochastic and wait-and-see solutions where the realizations of all the random parameters are known at the first stage, given by:

$$EVPI := RP - WS \tag{5.4}$$

Besides, the Value of Stochastic Solution indicates the expected gain from solving the stochastic model rather than its deterministic counterpart, in which the random parameters are replaced with their expected values and is defined as follows:

$$VSS := EEV - RP \tag{5.5}$$

where EEV denotes the solution value of the RP model, having the first-stage decision variables fixed at the optimal values obtained by using the expected value of coefficients. The results presented in this section are all computed by means of the PH, and are presented in an aggregated form in Table 3. We report, for each combination of instance size N (Column 1) and time block $Hour$ (Column 2), node distribution strategy D (Column 3), the average computational time T_{PH} in seconds of PH (Column 4), the average and the maximum percentage $EVPI$ computed as $EVPI/RP \cdot 100$ (Column 5 and 6) and the average and the maximum percentage VSS computed as $VSS/RP \cdot 100$ (Column 7 and 8).

First of all we can highlight how the computational effort is stable with respect to the time block and the node distribution strategy and it increases almost linearly with the instance size. The overall computational effort is limited, being of the order of magnitude of 10 minutes for the 200 customers instances. This makes the PH a strategic tool that can be incorporated in larger Decision Support Systems.

The percentage $EVPI$ present values of about 30%, showing the relevance, for a decision maker, to have the information about the future in advance. Notice that the $EVPI$ is stable regardless the instance parameters N, $Hour$, and D.

When considering the VSS, we can see how it is increasing both in mean and in maximum values while the size of the instance increases. In particular, the gap between the expected value solution and the stochastic solution becomes relevant when the number of nodes is between 100 and 200, which is the typical size of the day tour of a single vehicle in the parcel delivery and courier services in a medium and large city. Even when considering small instances (50 nodes) the VSS is relevant, with values of the maximum gap up to 6% and it shows the losses obtained by following the tour suggested by the deterministic solution. When analyzing the results with respect to the node distribution strategy, the most critical ones are $D4$ and $D3$. This finding is relevant, being the latter the most representative of the distribution of customers in a city (Maggioni *et al.*, 2014b). Notice that, due to the combinatorial nature of the problem, measures of the quality of deterministic solution (Maggioni & Wallace, 2012) like *loss using the skeleton solution* $LUSS$, obtained fixing at zero all first stage variables which are at zero in the expected value solution and then solving the stochastic program, correspond to VSS. The same for the *loss of upgrading the deterministic solution* $LUDS$, obtained by considering the expected value solution variables as a stating point to the stochastic model; the reason can be explained as follows: if an arc has been opened in the EV solution, then it must be used also in the stochastic setting, on the contrary if an arc is closed in the EV solution it can be opened in the stochastic one. But since the EV solution is a cycle and the stochastic solution cannot add or remove any arc because of the subtour elimination constraints and of the $LUDS$ condition, then $LUDS = VSS$.

An important point when comparing the solutions of the Recourse and the Expected Value problems is to determine how much the first stage decisions, i.e., the sequence of the nodes to visit, differ in the two problem solutions. We analyzed this issue, seeing how the two decisions differ of more than 15%.

An example of this is given in Figure 5, where the first stage decisions are highlighted for an instance with 50 customers. In particular, Figure 5c overlaps the two solutions, presenting the arcs differing in the two solutions with a long-dotted line for the Expected Value Problem solution and a short-dotted line for the Recourse Problem one, respectively. From the pictures we can see how the gap in terms of objective functions (and recourse actions in particular) is determined by a change of a relevant part of the central area tour, with 11 arcs involved, corresponding to about 20% of the first stage decisions. A specific point is the suburban arc in the South-West portion which is inserted in the Expected Value Problem solution, unfortunately misleading due to he high variance in cost oscillations along different scenarios. On the contrary, the Recourse Problem solution reduces the overall cost of 3.89%.

We now analyze the data in Table 3 from a 3PL (Third Party Logistic Service Provider) economic point of view. The $EVPI$, which represents the value of the objective function achievable by having perfect information regarding the resolution of the uncertainty, gives us a measure of the economical gain that a 3PL could obtain by having such information. This gain is about 30%, while the profitability of parcel delivery is typically between 15% and 20%. Hence, building an accurate forecaster of traffic flow and traveling times in a medium sized city as Turin, might drastically increase the profit of any delivery company. Moreover, a measure of the value of such type of forecaster gives us an insight of the value of data provided for free by the users of online services such as Google Maps to the business owners. In fact, using their users as a network of sensors, they can build an instant traffic database which is able to partially fulfill the gap between the current state of things (no forecast available) and total information on the future. In this way they will be relocating their services from *nice-to-have* to *must-to-have* for all the parcel delivery companies.

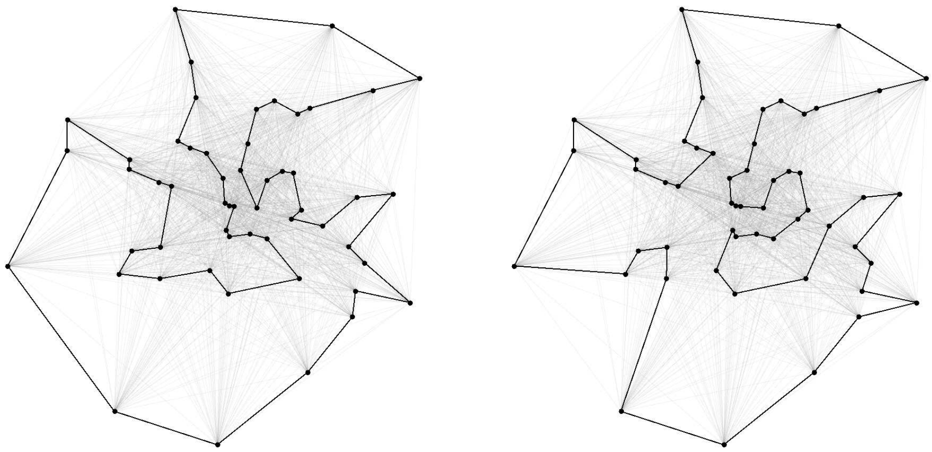
The VSS gives us another insight about the operational issues of 3PL involved in parcel delivery. The average number of customers served by a single vehicle in a day is typically between 80 and 90, which means that the N100 instances considered before are representative of the typical working day of a 3PL driver. For the N100 instances, the VSS is around 7%, meaning that the profitability of parcel delivery is cut in half due to the 3PLs' operational practice of considering average travel times. Due to the aggressive delivery price policies of Amazon and other on-line shops, in the next years 3PLs will be forced to increase their margins by refining their planning procedures. The approach of taking into account various uncertainties including the traveling times, as we propose in this paper, seems certainly very promising, especially because instant traffic information will be more and more available and computational power is ever increasing, making it easier to react quickly to changing situations.

6. Conclusions and future directions

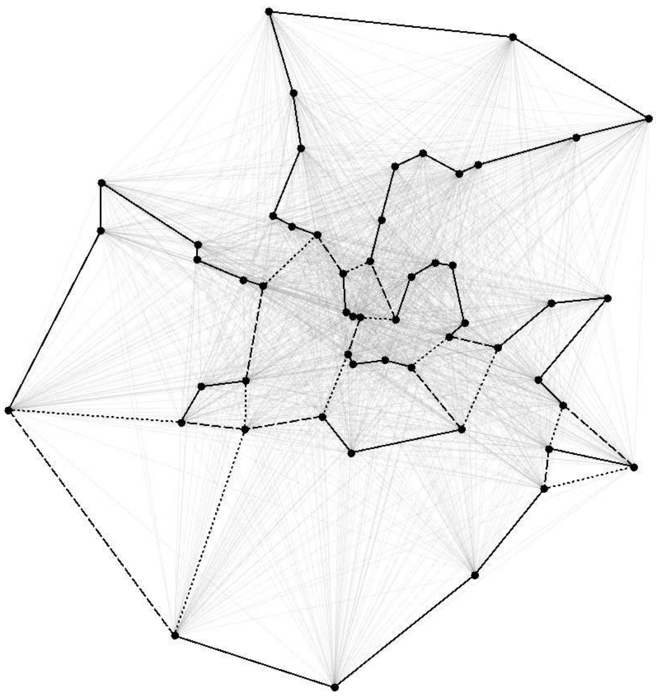
In this paper we have addressed the multi-path Traveling Salesman Problem with stochastic travel times, which consists in finding an expected minimum Hamiltonian tour connecting all nodes, where each pair of nodes is connected by several paths characterized by a stochastic travel time. In detail, we introduced two recourse models able to solve to optimality small-sized instances and a Progressive Hedging method able to cope with larger instances. The latter was shown to be very efficient, both from the computational point of view and quality of solutions, when compared to a direct solution approach obtained using commercial solvers. A simple parallel scheme has been also used to solve larger instances.

Future works will include an investigation of more sophisticated asynchronous parallel paradigms in order to consider instances with more than 200 nodes. From a modeling point of view, additional technology constraints related to the state of charge level of the battery of hybrid vehicles should be considered. This issue becomes of particular interest, being both the stochastic oscillation of the costs and the state-of-charge level of the battery originated by the same sources of uncertainty (e.g. travel times, traffic congestion, number of Stop&Go).

Multi-stage variants of the problem capturing the dynamic nature of velocity over the arcs will be also



(a) Expected Value Problem solution (b) Recourse Problem solution



(c) Solutions comparison

FIG. 5: Comparison of the first-stage Expected Value Problem solution (a) and the Recourse Problem solution (b). Figure (c) shows the common arcs (solid line), arcs used only in the Expected Value Problem solution (long-dotted line) and arcs used only in Recourse Problem one (short-dotted line).

addressed in future works.

References

APPLEGATE, D. L., BIXBY, R. E., CHVÁTAL, V. & COOK, W. J. (2007) *The Traveling Salesman Problem: A Computational Study*. Princeton, NJ, USA: Princeton University Press.

BALDI, M. M., GHIRARDI, M., PERBOLI, G. & TADEI, R. (2012) The capacitated transshipment location problem under uncertainty: A computational study. *Procedia - Social and Behavioral Sciences*, **39**, 424–436.

BEAUDRY, A., LAPORTE, G., MELO, T. & NICKEL, S. (2010) Dynamic transportation of patients in hospitals. *OR Spectrum*, **32**, 77–107.

TABLE 3 *Full instance set results: EVPI and VSS values comparison*

N	Hour	D	T_{PH} [s]	EVPI$_{\%}$	max EVPI$_{\%}$	VSS$_{\%}$	max VSS$_{\%}$
N50	8.00	D1	174.71	32.67	38.81	2.50	4.92
		D2	174.06	30.62	35.48	3.09	5.66
		D3	174.49	27.41	32.07	2.94	6.43
		D4	178.49	28.04	31.70	2.76	8.52
	12.00	D1	159.31	31.58	36.92	2.81	4.82
		D2	188.16	30.18	34.57	3.73	6.43
		D3	168.3	26.58	30.17	2.60	5.11
		D4	189.60	26.60	30.98	2.32	3.81
	16.00	D1	178.42	32.32	38.45	3.21	6.28
		D2	175.68	29.97	33.29	3.70	5.83
		D3	172.97	29.40	30.89	2.69	5.60
		D4	171.40	26.65	30.77	2.56	4.71
N100	8.00	D1	335.91	29.40	34.44	6.82	11.77
		D2	336.09	30.09	34.68	6.80	10.13
		D3	338.38	29.27	33.30	6.70	9.77
		D4	368.45	29.00	33.71	7.20	14.07
	12.00	D1	357.84	29.15	32.96	5.67	8.80
		D2	361.18	30.42	33.47	6.16	8.57
		D3	393.69	29.57	32.52	6.07	10.20
		D4	382.42	29.06	31.84	6.28	12.33
	16.00	D1	332.12	28.71	33.23	6.06	8.46
		D2	367.57	29.99	34.45	7.53	13.84
		D3	336.48	28.62	32.81	6.47	10.41
		D4	387.26	28.85	32.92	6.63	14.51
N200	8.00	D1	704,13	31,76	38,19	7,66	10,16
		D2	694,85	31,87	36,53	8,12	12,87
		D3	727,09	28,85	34,59	7,61	11,51
		D4	702,16	28,89	34,17	7,88	11,67
	12.00	D1	635,44	30,63	37,39	7,80	11,55
		D2	654,46	30,61	35,21	7,69	11,87
		D3	640,05	26,70	30,77	7,50	11,16
		D4	629,21	26,93	31,69	7,60	10,69
	16.00	D1	667.12	31,70	38,76	7,87	12,99
		D2	633.10	30,55	35,16	7,76	10,63
		D3	646,27	26,46	31,94	7,65	11,36
		D4	695,66	27,00	31,75	7,22	10,50

BERTAZZI, L. & MAGGIONI, F. (2014a) Solution approaches for the stochastic capacitated traveling salesmen location problem with recourse. *Journal of Optimization Theory and Applications*. DOI: 10.1007/s10957-014-0638-z.

BERTAZZI, L. & MAGGIONI, F. (2014b) The Stochastic Capacitated Traveling Salesmen Location Problem: a computational comparison for a United States instance. *PROCEDIA - Social and Behavioral Sciences*, **108**, 47–56.

BERTSIMAS, D. & VAN RYZIN, G. (1991) A stochastic and dynamic vehicle-routing problem in the Euclidean plane. *Operations Research*, **39**, 601–615.

BIRGE, J. (1982) The value of the stochastic solution in stochastic linear programs with fixed recourse. *Mathematical Programming*, **24**, 314–325.

BIRGE, J. R. & LOUVEAUX, F. (2011) *Introduction to stochastic programming*. Princeton, NJ, USA: Springer.

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.

CHEN, H.-K., HSUEH, C.-F. & CHANG, M.-S. (2006) The real-time time-dependent vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, **42**, 383–408.

CITYLOG CONSORTIUM & CITYLOG (2010). http://www.city-log.eu/. Last access 2015/02/28.

COOK, W. J. (2012) *In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation*. Princeton, NJ, USA: Princeton University Press.

CRAINIC, T. G., FU, X., GENDREAU, M., REI, W. & WALLACE, S. W. (2011) Progressive hedging-based metaheuristics for stochastic network design. *Networks*, **58**, 114–124.

DEFLORIO, F., GONZALES-FELIU, J., PERBOLI, G. & R., T. (2012) The influence of time windows on the costs of urban freight distribution services in city logistics applications. *European Journal of Transport and Infrastructure Research*, **12**, 256–274.

EIGER, A., MIRCHANDANI, P. B. & SOROUSH, H. (1985) Path preferences and optimal paths in probabilistic networks. *Transportation Science*, **19**, 75–84.

FLEISCHMANN, B., GNUTZMANN, S. & SANDVOSS, E. (2004) Dynamic vehicle routing based on online traffic information. *Transportation Science*, **38**, 420–433.

GENDREAU, M., LAPORTE, G. & SÉGUIN, R. (1996) Stochastic vehicle routing. *European Journal of Operational Research*, **88**, 3–12.

GENDREAU, M., GUERTIN, F., POTVIN, J.-Y. & TAILLARD, E. (1999) Parallel tabu search for real-time vehicle routing and dispatching. *Transportation Science*, **33**, 381–390.

GOEMANS, M. X. & BERTSIMAS, D. (1991) Probabilistic analysis of the Held and Karp lower bound for the Euclidean traveling salesman problem. *Mathematics of Operations Research*, **16**, 72–89.

GOLDEN, B. L., RAGHAVAN, S. & WASIL, E. A. (2008) *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer.

GÜNER, A. R., MURAT, A. & CHINNAM, R. B. (2012) Dynamic routing under recurrent and non-recurrent congestion using real-time ITS information. *Computers & Operations Research*, **39**, 358–373.

HVATTUM, L. M., LOKKETANGEN, A. & LAPORTE, G. (2006) Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. *Transportation Science*, **40**, 421–438.

HVATTUM, L. M., LOKKETANGEN, A. & LAPORTE, G. (2007) A branch-and-regret heuristic for stochastic and dynamic vehicle routing problems. *Networks*, **49**, 330–340.

IBM ILOG (2012) *CPlex User's Manual Version 12 Release 5*. IBM Corporation.

ICHOUA, S., GENDREAU, M. & POTVIN, J.-Y. (2006) Exploiting knowledge about future demands for real-time vehicle dispatching. *Transportation Science*, **40**, 211–225.

JAILLET, P. (1988) A priori solution of a traveling salesman problem in which a random subset of the customers are visited. *Operations Research*, **36**, 929–936.

JAILLET, P. & MELVYN SIM, J. Q. (2013) Routing optimization with deadlines under uncertainty. *"Tech. Rep."*. MIT.

KAUT, M. & WALLACE, S. W. (2007) Evaluation of scenario-generation methods for stochastic programming. *Pacific Journal of Optimization*, **3**, 257–271.

KENYON, A. S. & MORTON, D. P. (2003) Stochastic vehicle routing with random travel times. *Transportation Science*, **37**, 69–82.

LARSEN, A., MADSEN, O. B. G. & SOLOMON, M. M. (2004) The a priori dynamic traveling salesman problem with time windows. *Transportation Science*, **38**, 459–472.

LEIPALA, T. (1978) On the solutions of stochastic traveling salesman problems. *European Journal of Operational Research*, **2**, 291–297.

LI, J.-Q., MIRCHANDANI, P. B. & BORENSTEIN, D. (2009a) A Lagrangian heuristic for the real-time vehicle rescheduling problem. *Transportation Research Part E: Logistics and Transportation Review*, **45**, 419–433.

LI, J.-Q., MIRCHANDANI, P. B. & BORENSTEIN, D. (2009b) Real-time vehicle rerouting problems with time windows. *European Journal of Operational Research*, **194**, 711–727.

MAGGIONI, F., KAUT, M. & BERTAZZI, L. (2009) Stochastic optimization models for a single-sink transportation problem. *Computational Management Science*, **6**, 251–267.

MAGGIONI, F., ALLEVI, E. & BERTOCCHI, M. (2014a) Bounds in multistage linear stochastic programming. *Journal of Optimization Theory and Applications*, **163**, 200–229.

MAGGIONI, F., PERBOLI, G. & TADEI, R. (2014b) The multi-path traveling salesman problem with stochastic travel costs: Building realistic instances for city logistics applications. *Transportation Research Procedia*, **3**, 528–536.

MAGGIONI, F. & WALLACE, S. (2012) Analyzing the quality of the expected value solution in stochastic programming. *Annals of Operations Research*, **200**, 37–54.

MITROVIĆ-MINIĆ, S. & LAPORTE, G. (2004) Waiting strategies for the dynamic pickup and delivery problem with time windows. *Transportation Res. B*, **38**, 635–655.

MU, Q., FU, Z., LYSGAARD, J. & EGLESE, R. (2011) Disruption management of the vehicle routing problem with vehicle breakdown. *Journal of the Operational Research Society*, **62**, 742–749.

NOVOA, C. & STORER, R. (2009) An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. *European Journal of Operational Research*, **196**, 509–515.

PERBOLI, G., TADEI, R. & VIGO, D. (2011) The two-echelon capacitated vehicle routing problem: Models and math-based heuristics. *Transportation Science*, **45**, 364–380.

PERBOLI, G., DE MARCO, A., PERFETTI, F. & MARONE, M. (2014) A new taxonomy of smart city projects. *Transportation Research Procedia*, **3**, 470–478.

PILLAC, V., GENDREAU, M., GUÉRET, C. & MEDAGLIA, A. L. (2013) A review of dynamic vehicle routing problems. *European Journal of Operational Research*, **225**, 1–11.

PSARAFTIS, H. N. & TSITSIKLIS, J. N. (1993) Dynamic shortest paths in acyclic networks with markovian arc costs. *Operations Research*, **41**, 91–101.

ROCKAFELLAR, R. T. & WETS, R. J.-B. (1991) Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, **16**, 119–147.

SECOMANDI, N. & MARGOT, F. (2009) Reoptimization approaches for the vehicle routing problem with stochastic demands. *Operations Research*, **57**, 214–230.

TADEI, R., PERBOLI, G., RICCIARDI, N. & BALDI, M. M. (2012) The capacitated transshipment location problem with stochastic handling utilities at the facilities. *International Transactions in Operational Research*, **19**, 789–807.

TADEI, R., PERBOLI, G. & PERFETTI, F. (2014) The multi-path traveling salesman problem with stochastic travel cost. *EURO Journal on Transportation and Logistics*, **forthcoming**, doi:10.1007/s13676–014–0056–2.

TAGMOUTI, M., GENDREAU, M. & POTVIN, J.-Y. (2011) A dynamic capacitated arc routing problem with time-dependent service costs. *Transportation Res. C*, **19**, 20–28.

TANIGUCHI, E. & SHIMAMOTO, H. (2004) Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel times. *Transportation Res. C*, **12**, 235–250.

TORIELLO, A., HASKELL, W. B. & POREMBA, M. (2012) A dynamic traveling salesman problem with stochastic arc costs. *Technical Report*. Optimization Online.