European Journal of Operational Research 259 (2017) 1–18

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Invited Review

The Traveling Purchaser Problem and its variants

Daniele Manerba[a,*], Renata Mansini[a], Jorge Riera-Ledesma[b]

[a] Department of Information Engineering, University of Brescia, via Branze 38, Brescia 25123, Italy
[b] DEIOC, Facultad de Matematicas, Universidad de La Laguna, La Laguna 38271, Tenerife, Spain

ARTICLE INFO

Article history:
Received 27 May 2016
Accepted 15 December 2016
Available online 19 December 2016

Keywords:
Routing
Traveling Purchaser Problem
Supplier selection
Exact and heuristic methods

ABSTRACT

The Traveling Purchaser Problem (TPP) has been one of the most studied generalizations of the Traveling Salesman Problem. In recent decades, the TPP attracted the attention of both researchers in combinatorial optimization and practitioners, thanks to its double nature of procurement and transportation problem. The problem has been used to model several application contexts and is computationally challenging, dealing at the same time with the suppliers selection, the optimization of the purchasing plan and the routing decisions of the purchaser. For the first time after 50 years from its birth, we survey all the research done on the TPP including the most interesting and best performing solution methods proposed so far. We conclude providing some interesting future developments.

1. Introduction

Procurement problems, optimizing costs and revenues for manufacturing companies or firm retailers, have a long history in the specialized literature. The aim of a procurement problem is, in general, to elaborate a purchasing plan that satisfies the demand for a set of products/raw materials while minimizing the procurement costs. Usually, the plan is formalized in terms of two joint decisions, one concerning which suppliers should be selected, and the other one deciding how much should be ordered from each supplier (Aissaoui, Haouari, & Hassini, 2007). This activity is critical in any organization, considering that procurement expenditure typically accounts for a large portion of a firm total cost. For this reason, nowadays, the procurement logistics is still a vivid stream of research (Manerba, 2015).

The study of *routing/transportation problems* optimizing traveling costs dates even back. A routing problem generally aims at finding one or more optimal tours in order to visit a set of geographical locations (customers, suppliers, etc.) from a central depot. The well-known Traveling Salesman Problem (TSP) and the Vehicle Routing Problem (VRP) belong to this category (see Gutin and Punnen, 2002; Toth & Vigo, 2014).

The joint evaluation of both transportation and procurement problems is a more recent stream of research combining the relevant features of the two previous contexts. The *Traveling Purchaser Problem* (TPP) belongs to this stream. In the TPP, given a list specifying products and quantities required, a purchaser has to find a purchasing plan that exactly satisfies the products demand by visiting a subset of suppliers in a unique tour. The objective of the purchaser is to minimize the combined traveling and purchasing cost. In the classical TPP only a single vehicle is involved, even if multi-vehicle variants have already been studied in the literature.

According to Golden, Levy, and Dahl (1981) and Fischetti, Salazar-Gonzalez, and Toth (2007), the TPP represents, together with the family of orienteering problems, one of the most interesting generalizations of the TSP. The large number of papers published on this problem in the last decade demonstrates that it is still attractive for researchers and practitioners. For these reasons, the purpose of the present paper is to survey, for the first time, the existing literature on the TPP. In particular, we will focus on its modeling aspects and solution methods, also including the analysis of its several variants, as the multi-vehicle ones.

The paper is organized as follows. In Section 2, we formally define the TPP, standardize its classifications, point out some interesting properties, and present its many practical applications. In Sections 3 and 4, we survey different Mixed Integer Linear Programming (MILP) formulations for the TPP and the most important polyhedral results, respectively. Sections 5 and 6 present exact and heuristic approaches, respectively. Section 7 analyzes deterministic, dynamic, and stochastic variants of the TPP, as well as its multi-vehicle extensions. Conclusions and open lines of research

* Corresponding author.
E-mail addresses: daniele.manerba@unibs.it (D. Manerba), renata.mansini@unibs.it (R. Mansini), jriera@ull.es (J. Riera-Ledesma).

http://dx.doi.org/10.1016/j.ejor.2016.12.017
0377-2217/

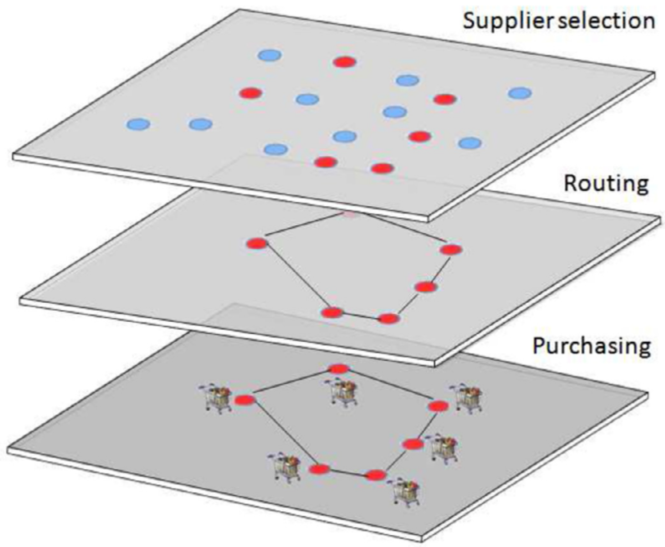

Fig. 1. Components of the TPP in a layered structure.

are drawn in Section 8. Finally, Appendix A presents and compares different sets of benchmark instances.

2. Problem definition and properties

The TPP, in its original form, is a single-vehicle routing and procurement problem defined as follows. Consider a depot 0, a set K of products/items to purchase, and a set M of geographically dispersed suppliers/markets. A discrete demand d_k, specified for each product $k \in K$, can be accomplished in a subset $M_k \subseteq M$ of suppliers at a price $p_{ik} > 0$, $i \in M_k$. Moreover, a product availability $q_{ik} > 0$ is also defined for each product $k \in K$ and each supplier $i \in M_k$. Note that, to guarantee the existence of a feasible purchasing plan with respect to the product demand, the condition $\sum_{i \in M_k} q_{ik} \geq d_k, \forall k \in K$ has to hold. The problem is defined on a complete directed graph $\mathcal{G} = (V, A)$ where $V := M \cup \{0\}$ is the node set, and $A := \{(i, j) \colon i, j \in V, i \neq j\}$ is the arc set. A traveling cost c_{ij} is associated with each arc $(i, j) \in A$. The TPP looks for a simple tour in $\mathcal{G}$ starting and ending at the depot, visiting a subset of suppliers and deciding how much to purchase for each product in each supplier so to satisfy the demand at minimum traveling and purchasing costs.

The great interest in the TPP is probably due to the fact that it challengingly combines supplier selection, routing construction, and product purchase planning. Fig. 1 shows the three components of the problem in a layered framework. It is clear that optimally solving each subproblem separately does not guarantee to achieve the optimal solution for the TPP. The supplier selection (first layer) is a crucial aspect that differentiates TPP from traditional routing problems, linking it to the so-called *routing problems with profits* (Feillet, Dejax, & Gendreau, 2005). Since the main goal of the purchaser is to satisfy products demand, not all the suppliers have to be visited necessarily. In general, the convenience of a visit depends on the trade-off between the additional traveling cost for reaching the supplier and the possible saving obtained by purchasing products at lower prices[1]. The TPP has, in fact, a bi-objective nature, linearly combining in a single objective function the minimization of both traveling and purchasing costs (second and third layer, respectively). This makes the problem of selecting optimal suppliers more complex since the traveling costs optimization pushes the purchaser to select only suppliers that are strictly necessary to satisfy products demand, while the purchasing costs minimization pushes to select a more convenient and potentially larger set of suppliers.

2.1. Common classifications

A first classification comes from the TPP routing nature. As for the TSP, a TPP modeled on a directed graph, where the cost c_{ij} is potentially different from c_{ji}, is named *asymmetric* (ATPP). Otherwise, if $c_{ij} = c_{ji}$ for each arc $(i, j) \in A$, the problem is called *symmetric* TPP (STPP). In the literature, ATPP and STPP are often referred to as *directed* and *undirected* TPP, respectively.

A second common classification concerns the availability of products at the suppliers. If the available quantity of a product $k \in K$ in a supplier $i \in M_k$ is defined as a finite value q_{ik}, potentially smaller than product demand d_k, then the TPP is called *restricted* (R-TPP). The *unrestricted* TPP (U-TPP), instead, considers the case in which supplies are unlimited, i.e., where $q_{ik} \geq d_k$, $k \in K$, $i \in M_k$. Note that U-TPP represents a special case of R-TPP, since having unlimited supplies is equivalent to consider $d_k = 1$ and $q_{ik} = 1, \ \forall k \in K, \forall i \in M_k$. In the literature, several papers refer to R-TPP and U-TPP as *capacitated* and *uncapacitated* TPP, respectively. However, we prefer to adopt the former nomenclature in order to avoid confusion with the concept of vehicle capacity, appearing in the multi-vehicle case. Rarely, these variants are also called *limited-supply* and *unlimited-supply* TPP.

2.2. Complexity

The TPP is $\mathcal{NP}$-hard in the strong sense since it generalizes both the TSP and the Uncapacitated Facility Location Problem (UFLP). This can be proved by the following reductions: (1) the TSP corresponds to an U-TPP where each supplier offers a product that cannot be purchased elsewhere; (2) the UFLP can be seen as an U-TPP where each potential facility location corresponds to a supplier and each customer to a product, $M_k = M$ for all $k \in K$, p_{ik} is the cost of serving customer k from facility i, and $c_{ij} := (b_i + b_j)/2$, $\forall (i, j) \in A$, with b_i the cost of opening facility i.

However, as highlighted by Teeninga and Volgenant (2004), some TPP special cases can be solved trivially, namely (a) when the supplier nearest to the depot sells, for each product, all the required quantity at the lowest price, and (b) when the traveling costs are null. In the latter case, an optimal U-TPP solution can be found by purchasing each product from its cheapest supplier, since any tour connecting these suppliers is optimal (in the R-TPP, instead, the suppliers are first sorted in non-decreasing order of price for each product k, then each product is purchased from its cheapest suppliers in an amount equal to the minimum between the available quantity and the residual demand).

Finally, note that the problem feasibility can be checked polynomially just by inspection of the input data. If a product is not available at any supplier, then no solution exists for the U-TPP. Similarly, for the R-TPP, the infeasibility occurs if it exists a product k such that $\sum_{i \in M_k} q_{ik} < d_k$.

2.3. Applications

In the previous sections, we have presented the TPP as a procurement logistics problem. Interesting enough, its combinatorial structure appears for the first time (in the unrestricted form) in the work by Burstall (1966) to model the scheduling of different jobs on a multi-purpose production line. In that case, products

[1] We remark that suppliers selection in the TPP has to be intended only at an operating level, and depending on the daily product demand, prices, and availabilities. Strategic decisions for selecting the best suppliers based on qualitative criteria (e.g., service quality and reliability) concern another well-studied stream of research (Degraeve, Labro, & Roodhooft, 2000).

correspond to required jobs and suppliers to particular production line configurations. The cost p_{ik} represents the time needed to process job k in configuration i, while cost c_{ij} represents the time needed to changeover from configuration i to j. The objective is to choose the sequence of configurations and jobs so to minimize the overall time to conclude the jobs' batch. In the cited work the application was that of manufacturing steel tubes, but it is easy to imagine how this setting can be generalized to schedule jobs in any application involving general-purpose machineries. E.g., Cattrysse, Beullens, Collin, Duflou, and van Oudheusden (2006) use a TPP model to control a sheet metal bending machine in the press brakes production and show that is possible to save about the 10% in makespan time on real-life work orders taken from several companies.

However, the TPP has started gaining a lot of attention from the operation research community only after its reinterpretation as a vehicle routing problem by Ramesh (1981)[2]. In real procurement settings, in fact, it is very common that a company is directly involved in the purchasing and collection of raw materials, spare parts, or products from some reliable suppliers. With the evolution of more and more efficient solution algorithms, the TPP has become a useful tool for companies' information systems in supporting their procurement operations. A notable example is represented by the commercial web application "*Le Bon Côté des Choses*" (http://www.leboncotedeschoses.fr/) implementing the approach proposed by Cambazard and Penz (2012), in which a purchaser selects his location, the products list, and a maximum number of markets to visit, and receives the most convenient shopping plan (see Section 5.3). Other routing applications include, e.g., the problem of planning the tour needed for a school bus to pick-up students from several stops (see Section 7.3) or, as suggested by Singh and van Oudheusden (1997), warehousing operations for dispatching a vehicle to pick up the ordered items (stored in different locations) and transport them to the shipping area.

Some works have considered the TPP to model operations in maritime transport networks. A ship–truck intermodal freight transportation problem has been modeled as a TPP by Infante, Paletta, and Vocaturo (2009). Here, seaport terminals play the role of exchange hubs in which containers, addressed to hinterland logistic centers, are moved from ship to trucks and vice versa. Schwarze and Voß (2015), instead, propose a two-level hierarchical planning problem to enhance the optimized interaction between maritime shipping and hinterland traffic and show how, under several assumptions, the problem can be reduced to a single or multi-vehicle TPP, or to other standard routing models (TSP, VRP).

The TPP can also be employed in many *network design* applications such as subway/rail lines, irrigation networks, and so on. Not surprisingly, along with the extraordinary worldwide expansion of telecommunication networks in the nineties, Voß (1990) and Ravi and Salman (1999) highlighted the possibility of using TPP to design special configurations in industrial and generic communication networks. Such infrastructures consist of several *local access networks* (LANs) collecting traffic of user nodes at the switching centers and of a *backbone network* that routes high volume traffic among switching centers. Because of its reliability and *self-healing* properties, an optimized network structure requires a ring architecture for the backbone and a star architecture for the LANs. The problem is to determine a tour (the ring backbone) on a subset of the network nodes and connect the remaining nodes to others in the tour (star configuration) minimizing the overall connection cost. This problem, named the *ring–star problem*, is actually a TPP special case where the graph nodes correspond to both the set of suppliers and the set of products.

Finally, to further underline its wide applicability, we remark that even a stand management problem for bio-diversity sustainable forests (Wikström & Eriksson, 2000) have been related to the TPP. The authors want to determine harvest periods and the number of every tree type to be cut in these periods, while maximizing the net present value of harvests after deduction of entry costs for the harvesting periods. Some similarities induced the authors to adapt an heuristic procedure proposed by Voß (1996) for the TPP (see Section 6.1.2) to their problem. In the recent years, along with the study of TPP variants and generalizations, other applications such as home health-care and scheduling of surgeries in operating rooms have appeared. We will describe them in Section 7.

3. Mathematical programming formulations

In this section, we present different MILP formulations for the ATPP and the STPP.

3.1. Asymmetric TPP

Let y_i, $i \in M$, be a binary variable taking value 1 if supplier i is selected, and 0 otherwise. Let x_{ij}, $(i, j) \in A$, be a binary variable taking value 1 if arc (i, j) is traversed, and 0 otherwise. Let z_{ik}, $k \in K$, $i \in M_k$, be a variable representing the number of units of product k purchased from supplier i. Moreover, for any subset V' of nodes, let us define $\delta^+(V') := \{(i, j) \in A : i \in V', j \notin V'\}$ and $\delta^-(V') := \{(i, j) \in A : i \notin V', j \in V'\}$. Then, the ATPP can be formulated as follows:

$$(ATPP) \quad \min \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{k\in K}\sum_{i\in M_k} p_{ik}z_{ik} \tag{1}$$

$$\text{subject to} \quad \sum_{i\in M_k} z_{ik} = d_k \qquad k \in K \tag{2}$$

$$z_{ik} \le q_{ik}y_i \qquad k \in K, i \in M_k \tag{3}$$

$$\sum_{(i,j)\in\delta^+(\{h\})} x_{ij} = \sum_{(i,j)\in\delta^-(\{h\})} x_{ij} = y_h \qquad h \in M \tag{4}$$

$$\sum_{(i,j)\in\delta^-(M')} x_{ij} \ge y_h \qquad M' \subseteq M, h \in M' \tag{5}$$

$$x_{ij} \in \{0, 1\} \qquad (i, j) \in A \tag{6}$$

$$y_i \in \{0, 1\} \qquad i \in M \tag{7}$$

$$z_{ik} \ge 0 \qquad k \in K, i \in M_k. \tag{8}$$

Objective function (1) aims at the joint minimization of the traveling and purchasing costs. Eqs. (2) ensure that each product demand is satisfied exactly. Constraints (3) impose that each supplier has to be visited to purchase a product from it and the purchased quantity should not exceed the corresponding availability. Constraints (4) and (5) rule the visiting tour feasibility. Eqs. (4) impose that, for each visited supplier, exactly one arc must enter and leave the relative node. Inequalities (5) are *connectivity constraints* that prevent the creation of sub-tours not including the depot by imposing that at least one arc must enter each subset M' of suppliers in which at least one supplier h is visited. Finally, constraints (6)–(8) impose binary and non-negative conditions on variables.

[2] Actually, in contrast to what claimed in many papers, the applicability of the TPP in a procurement context has appeared along with the *Decorator's Problem* example discussed by Buzacott and Dutta (1971), ten years before the work by Ramesh (1981).

No integrality conditions are required for z-variables, even if they actually represent the number of units purchased for each product in each supplier. If all input data are integer, in fact, then an optimal solution where all z-variables have integer values always exists.

A first trivial preprocessing can be applied to strengthen model (1)–(8). To this aim, we define $M^* := \{0\} \cup \left\{i \in M : \exists k \in K \text{ such that } \sum_{j \in M_k \setminus \{i\}} q_{jk} < d_k\right\}$ as the node set that must be necessarily part of any feasible TPP solution, and $K^* := \left\{k \in K : \sum_{i \in M_k} q_{ik} = d_k\right\}$ as the product set for which suppliers selection and purchasing plan decisions can be predetermined. Thus, constraints (7) can be replaced by $y_i = 1$ when $i \in M^*$, and constraints (2) by $z_{ik} = q_{ik}$ when $k \in K^*$, $i \in M_k$.

3.1.1. Compact formulations

The formulation (1)–(8) cannot be implemented through a commercial solver even for small-size instances since the number of constraints (5) is exponential in the size of M. We will see in Section 5.2 that it is possible, and often efficient, to initially exclude constraints (5) from the model and then to identify in polynomial time (and add dynamically during a branch-and-bound approach) only those constraints that are necessary for the optimal solution. However, there exist other subtour elimination constraints that yield, expanding the variables subspace, *compact* formulations with a polynomial number of constraints. The *Miller–Tucker–Zemlin* (MTZ) and the *commodity flow* (CF) formulations can be directly inherited from TSP and adapted to the TPP. Unfortunately, in general, compact formulations have weaker continuous Linear Programming (LP) relaxations with respect to formulations with an exponential number of constraints.

Let us introduce a non-negative variable u_i for each supplier $i \in M$ representing the total number of suppliers already visited when leaving supplier i. Then, the MTZ formulation (Miller, Tucker, & Zemlin, 1960) for the TPP can be obtained by substituting inequalities (5) with the following inequalities

$$u_i - u_j + |M| x_{ij} \le |M| - 1 \quad i, j \in M, i \ne j \tag{9}$$

that prevent the creation of subtours by controlling the order of visit of the suppliers.

Another option is to define a non-negative flow variable f_{ij} for each arc $(i, j) \in A$ representing the quantity of a commodity on the vehicle when it leaves supplier i and arrives in j. Then, the *single*-CF formulation for the TPP can be obtained by substituting inequalities (4) and (5) with:

$$\sum_{j \in M} f_{0j} = \sum_{k \in K} d_k \tag{10}$$

$$\sum_{(i,j) \in \delta^+(\{h\})} f_{ij} - \sum_{(i,j) \in \delta^-(\{h\})} f_{ij} = -\sum_{k \in K} z_{hk} \quad h \in M \tag{11}$$

$$f_{ij} \le x_{ij} \sum_{k \in K} d_k \quad (i, j) \in A. \tag{12}$$

The connectivity requirement is imposed by creating, through flow variables, a path stemming from the depot. More precisely, for constraint (10), a single commodity is sent from the depot in an amount equal to the total demand $\sum_{k \in K} d_k$ and, for constraints (11), each visited supplier i absorbs an amount equal to the quantity purchased $\sum_{k \in K} z_{ik}$. Inequalities (12) states that a positive flow f_{ij} can be sent along arc (i, j) only if it is traversed ($x_{ij} = 1$). Other similar compact formulations with a stronger LP relaxation and based on *two* or even *multi-commodity* flow exist (see Gutin & Punnen, 2002) but, for sake of space, we skip their description. Anyway, their application is quite straightforward.

3.1.2. Valid inequalities

The LP relaxation of (1)–(8) can be strengthened by using valid inequalities derived from some of its subproblems. First, constraints (4)–(7) are the ones of the *Cycle Problem* (a TSP generalization in which only a subset of vertices must be visited) thus, e.g., the *lifted cycle* D_k^+ and D_k^- inequalities (Balas & Oosten, 2000) are valid for the ATPP. Second, constraints (2), (3), (7), and (8) define an UFLP with upper bounds on the customer facility variables. When $q_{ik} = d_k$, $k \in K, i \in M_k$, TPP valid inequalities can be obtained from the *Set Covering* polytope (Balas & Ng, 1989).

Some TPP specific valid inequalities also exist. For example, the *zSEC* inequalities (13), introduced in Riera-Ledesma (2002), state that at least one arc must enter into the subset $M' \subseteq M$ whenever some amount of any product k is purchased in a market belonging to M', i.e.:

$$\sum_{(i,j) \in \delta^-(M')} x_{ij} \ge \frac{1}{d_k} \sum_{i \in M' \cap M_k} z_{ik} \quad M' \subseteq M, k \in K. \tag{13}$$

Inequalities (13) can be easily strengthened by replacing d_k with $\min\{d_k, \sum_{i \in M' \cap M_k} q_{ik}\}$.

3.2. Symmetric TPP

The STPP is defined over a complete undirected graph $\mathcal{G}_U = (V, E)$, where $E := \{e = [i, j] : i, j \in V, i < j\}$ is the edge set and a traveling cost c_e is associated with each edge $e \in E$. Let x_e, $e \in E$, be a binary variable taking value 1 if edge e is crossed, and 0 otherwise. Let also $\delta(V') := \{[i, j] \in E\colon i \in V', j \in V \setminus V'\}$ for any subset V' of nodes. Then, the STPP can be defined as follows:

$$(STPP) \quad \min \sum_{e \in E} c_e x_e + \sum_{k \in K} \sum_{i \in M_k} p_{ik} z_{ik} \tag{14}$$

subject to constraints (2), (3), (7), (8), and

$$\sum_{e \in \delta(\{h\})} x_e = 2y_h \qquad h \in M \tag{15}$$

$$\sum_{e \in \delta(M')} x_e \ge 2y_h \qquad M' \subseteq M, h \in M' \tag{16}$$

$$x_e \in \{0, 1\} \quad e \in E. \tag{17}$$

Due to the use of an undirected graph, now in the degree constraints (15) two edges must be incident to each visited vertex, and in the connectivity constraints (16) at least two edges must be incident to each subset of suppliers containing a visited one. Note that this STPP formulation does not allow solutions with less than three vertices, one being the depot. Two-vertex cycles containing the depot and one market can be easily generated and compared to the optimal solution given by the model.

STPP compact formulations can be derived similarly to what we have already seen for the ATPP. Note, however, that MTZ formulations need the orientation of arcs to work correctly. A common workaround when using an undirected graph is to replace each edge $e = [i, j] \in E$ with two directed arcs (i, j) and (j, i) with $c_{ij} = c_{ji} = c_e$, making sure that only one arc linking each couple of vertexes can be used.

Concerning valid inequalities for the STPP, arguments similar to the ones dealt with in Section 3.1.2 can be done (Laporte, Riera-Ledesma, & Salazar-González, 2003), e.g., the symmetric version of the *zSEC* cuts is as follows:

$$\sum_{e \in \delta(M')} x_e \ge \frac{2}{d_k} \sum_{i \in M' \cap M_k} z_{ik} \quad M' \subseteq M, k \in K. \tag{18}$$

4. Polyhedral aspects

This section gathers the main polyhedral findings on the TPP (proved, unless otherwise indicated, in Riera-Ledesma, 2002 or Laporte et al., 2003). We focus on the dimension and facets for the ATPP polytope $\mathcal{P}$, i.e. the convex hull of all the vectors (x, y, z) that satisfy (2)–(8), and for the STPP polytope $\mathcal{S}$, i.e. the convex hull of all the vectors (x, y, z) that satisfy (2), (3), (7), (8), and (15)–(17).

4.1. Dimension of ATPP and STPP polytopes

Let $\bar{\mathcal{P}} := \{(x, y, z) \in \mathcal{P} : y_i = 1, \forall i \in V\}$ be the ATPP polytope in which all suppliers have to be visited, $\bar{\mathcal{P}}_x$ and $\bar{\mathcal{P}}_z$ be the projection of $\bar{\mathcal{P}}$ onto the affine space of x and z variables, respectively. $\bar{\mathcal{P}}_x$ is the asymmetric TSP polytope and $dim(\bar{\mathcal{P}}_x) = |A| - 2|V| + 1$ (Grötschel & Padberg, 1985), whereas $\bar{\mathcal{P}}_z$ is the polytope of an Assignment Problem generalization and $dim(\bar{\mathcal{P}}_z) = \sum_{k\in K\setminus K^*}(|M_k| - 1)$. Since $\bar{\mathcal{P}} := \bar{\mathcal{P}}_x \times \{y : y_i = 1, \forall i \in V\} \times \bar{\mathcal{P}}_z$ it holds that $dim(\bar{\mathcal{P}}) = |A| - 2|V| + 1 + \sum_{k\in K\setminus K^*}(|M_k| - 1)$. To extend this result onto $\mathcal{P}$, an intermediate polytope $\bar{\mathcal{P}}(V') := \{(x, y, z) \in \mathcal{P} : y_i = 1, \forall i \in V \setminus V'\}$ is introduced for all $V' \subseteq V$. It can be shown that $\bar{\mathcal{P}}(\emptyset) = \bar{\mathcal{P}}$, $\bar{\mathcal{P}}(V) = \mathcal{P}$, and, for any given $V' \subseteq V$, $dim(\bar{\mathcal{P}}(V')) = |A| - 2|V| + 1 + \sum_{k\in K\setminus K^*}(|M_k| - 1) + |V'|$. The dimension of $\mathcal{P}$ follows:

Theorem 4.1. $dim(\mathcal{P}) = |A| - |V| + 1 + \sum_{k\in K\setminus K^*}(|M_k| - 1)$.

The dimension of $\mathcal{S}$ only slightly differs from $\mathcal{P}$. The difference comes from the underlying circuit problem that contains a different number of linear independent vectors:

Theorem 4.2. $dim(\mathcal{S}) = |E| + \sum_{k\in K\setminus K^*}(|M_k| - 1)$.

Finally, recalling the meaning of M^* (i.e., the set of necessary suppliers in any feasible TPP solution), one can take it into account in the definition of ATPP and STPP polytopes. In such a case, the dimensions in Theorems 4.1 and 4.2 have to be reduced by $|M^*|$.

4.2. Facet-defining valid inequalities for ATPP and STPP polytopes

Some facets of $\mathcal{P}$ and $\mathcal{S}$ can be obtained from facets of particular projected polytopes through a procedure called *sequential lifting* that allows to derive and prove the following results.

Let us start from some trivial inequalities: inequality $x_{ij} \geq 0$ ($x_e \geq 0$) defines a facet of $\mathcal{P}$ (of $\mathcal{S}$) for every $(i, j) \in A$ (for every $e \in E$); inequality $y_i \leq 1$ defines a facet if and only if $i \in V\setminus M^*$; inequality $z_{ik} \geq 0$ defines a facet for each $k \in K\setminus K^*$ and $i \in M_k$ with $|M_k| \geq 3$ and $d_k < \sum_{j\in M_k\setminus\{i\}} q_{jk}$.

Other facets can be obtained for $\mathcal{P}$. For example, by lifting the subtour elimination constraints it derives that, for each $M' \subset M$,

- $\sum_{(i,j)\in A: i,j\in M'} x_{ij} \leq \sum_{h\in M'\setminus\{l\}} y_h$ defines a facet $\forall l \in M'$ if $M' \cap M^* = \emptyset$, and
- $\sum_{(i,j)\in A: i,j\in M'} x_{ij} \leq \sum_{h\in M'} y_h - 1$ defines a facet if $M' \cap M^* \neq \emptyset$.

In the same spirit, it is possible to derive facet-defining inequalities for $\mathcal{S}$. For example, for any $M' \subset M$ with $2 \leq |M'| \leq |M| - 1$,

- $\sum_{e\in\delta(M')} x_e \geq 2$ defines a facet if $M' \cap M^* \neq \emptyset$ or $|M'| = |M| - 1$, and
- $\sum_{e\in\delta(M')} x_e \geq 2y_i$ defines a facet for any $i \in M'$, otherwise.

Interesting enough, for the particular case with $|M'| = |M| - 1$, constraints $x_{[0,\, i]} \leq y_i$ also defines a facet of $\mathcal{S}$ for all $i \in V$. For $|M'| = 2$ it holds that, for any $e = [i, j] \in E$:

- $x_e \leq 1$ defines a facet if $i, j \in M^*$,
- $x_e \leq y_i + y_j - 1$ defines a facet if $i, j \notin M^*$ and $\exists k \in K$ such that $\sum_{s\in M_k\setminus\{i,j\}} q_{sk} < d_k$,
- otherwise, $x_e \leq y_i$ and $x_e \leq y_j$ define facets when $i \notin M^*$ and $j \notin M^*$, respectively.

Other facets can be obtained from the *Assignment Problem* polytope, e.g., $z_{ik} \leq q_{ik} y_i$ defines a facet $\forall k \in K\setminus K^*$ and $\forall i \in M_k$ if $i \notin M^*$, otherwise $z_{ik} \leq q_{ik}$ defines a facet when $q_{ik} < d_k$.

5. Exact solution approaches

In this section, we review algorithms to obtain TPP exact solutions. Because of its long history, the TPP has experimented the same evolution than other classical combinatorial problems, starting with Dynamic Programming (DP) in the early 70's, passing through the use of Mathematical Programming (MP) techniques in the late 90's, and concluding (at least up to now) with Constraint Programming. For each existing exact approach, Table 1 indicates the problem variant tackled, the solution method adopted, and the size $(|M|, |K|)$ of the biggest instances solved to optimality.

5.1. Early approaches

The first exact algorithm for the U-ATPP was proposed by Buzacott and Dutta (1971) and focused on the problem described by Burstall (1966) in the context of machine scheduling for the manufacture of steel tubes. The size of the instances considered was pretty small. In a first attempt, the authors tried to solve an ILP formulation, concluding that the approach was not very useful because "*a large number of subtour constraints were necessary even for a reasonably small sized problem*". It should be noted that this statement was written long before the spreading of the dynamic separation of constraints started with Padberg and Rinaldi (1991). Hence, following the main principles of that time (see, e.g., the survey on the TSP by Bellmore and Nemhauser, 1968 already suggesting the use of DP for problems with less than 13 cities), Buzacott and Dutta (1971) finally proposed a DP algorithm to solve the problem. In their DP method, at each stage, a configuration-job pair is added to the optimal scheduling sequence. At the end the algorithm provides the minimum cost sequence which processes all required jobs. The algorithm was written in FORTRAN

Table 1
Exact solution approaches.

Reference	Problem	Solution method	$(\lvert M\rvert, \lvert K\rvert)$
Buzacott and Dutta (1971)	U-ATPP	Dynamic Programming	(–, 15)
Ramesh (1981)	U-STPP	Lexicographic search	(12, 10), (8, 22)
Singh and van Oudheusden (1997)	U-ATPP	Branch-and-bound	(20, 100), (25, 50)
	U-STPP	Branch-and-bound	(20, 30)
Laporte et al. (2003)	R-STPP	Branch-and-cut	(200, 200)
Riera-Ledesma and Salazar-González (2006)	R-ATPP	Branch-and-cut	(200, 200)
Gouveia et al. (2011)	U-ATPP[a],[b]	Dynamic Programming	(300, 100)[c]
Cambazard and Penz (2012)	U-STPP[a]	Constraint Programming	(250, 200)

[a] The number of suppliers to visit is bounded.
[b] The number of products to be purchased in each supplier is bounded.
[c] Upper bounds obtained by heuristic algorithm.

IV and executed on a *IBM Model 360/65*. The authors claimed to have achieved, for small and medium sized instances (12–15 jobs and any practical number of configurations) exact solutions or solutions within the 10% of optimality gap.

After a decade, Ramesh (1981) introduced a lexicographic search algorithm for the U-STPP where each solution is represented as a sequence of symbols, and searching for an optimal solution is analogous to search for a specific word's location in a dictionary. Solutions are generated starting from a partial word in some hierarchy which reflects an analogous order in their values. Each partial word defines a block of solutions, and for each block of solutions a lower bound is computed. If this lower bound exceeds the value of the best known solution, the entered block of words is rejected because it does not lead to promising solutions, and the next block is explored. The computational experience, performed on a non specified machine, solves to optimality instances with up to 12 suppliers and 10 products, and 8 suppliers and 22 products. However, the computed lower bound only depends on the traveling costs from different suppliers to the depot and it is independent from the purchasing costs. This aspect clearly influences the quality of the bound, causing poor computational results.

5.2. Branch-and-bound based approaches

The first branch-and-bound algorithm was presented by Singh and van Oudheusden (1997). Their main idea is to break up the set of all possible tours into smaller subsets, and to calculate, for each of them, a lower bound on the sum of the traveling and purchasing costs. This lower bound is computed by solving a UFLP obtained by removing the routing optimization constraints from TPP. The bounds guide the partition of the subsets and allow to identify an optimal solution when a subset that contains a single tour is found (the relative bound has to be, in fact, less than or equal to that of all other subsets). They solved to optimality (on a *IBM 3031*) ATPP instances with up to $|M| = 20$ and $|K| = 100$ or $|M| = 25$ and $|K| = 50$, and STPP instances with up to $|M| = 20$ and $|K| = 30$.

A significant advance in the size of the problems solved to optimality occurred with the idea of exploiting the dynamic separation of constraints (i.e., along the branch-and-bound tree) by the branch-and-cut technique. More precisely, the TPP model is initially relaxed and solved without the subtour elimination constraints, then only those inequalities that cut off the linear relaxation optimal solution of the current branch-and-bound node are added to the model through separation algorithms. Laporte et al. (2003) approached the R-STPP and U-STPP solution by this technique also using some other valid inequalities. This paper proposes a polyhedral study of the problem combining the cycle and set-covering polytopes (that leads to the characterization of the valid inequalities already presented in Sections 3.2 and 4.2), and exact and heuristic separation procedures for the most effective valid inequalities found. Along with the separation of subtour elimination constraints, the authors also introduce the separation of the *zSEC* inequalities, the *2-matching inequalities* (Edmonds, 1965), and other mechanisms such as the dynamic generation and deletion of variables through a simple pricing. With these components the branch-and-cut algorithm (coded in C++ and using *ABACUS* linked to *Cplex 6.0* as a framework) solved to optimality, on a *Pentium 500* MHz, families of instances taken from the literature as well as new random instances up to 200 suppliers and 200 products.

Since the just presented approach has become a consolidated starting point in developing exact methods for TPP-like problems that deals with an exponential number of subtour elimination constraints (see, e.g., Batista-Galván, Riera-Ledesma, and Salazar-González, 2013; Beraldi, Bruni, Manerba, and Mansini, 2016; or Manerba & Mansini, 2015), we briefly present how inequalities (16) and *zSEC* inequalities (18) can be separated efficiently. Both the procedures are exact and are based on the solution of *max-flow/min-cut* problems for which different efficient polynomial-time algorithms can be found in the literature (see, e.g., Goldberg and Tarjan, 1988; Goldberg & Rao, 1998). In the following, we indicate as x^*, y^*, and z^* the value of the x, y, and z variables in the optimal solution of the continuous relaxation problem, respectively.

Separation procedure for (16): Consider a graph $\bar{\mathcal{G}_U} = (V, E)$ where a capacity x^*_e, corresponding to the current linear relaxation optimal solution, is associated with each edge $e \in E$. Then, given a supplier h, such that $y^*_h \neq 0$, the most violated inequality (16) corresponds to the partition (M', $V \backslash M'$) associated with a minimum-capacity cut in $\bar{\mathcal{G}_U}$ separating node 0 from h, with $h \in M'$. This cut can be found in $O(|M|^3)$ by computing a maximum flow in $\bar{\mathcal{G}_U}$ from node 0 to node h, hence the entire procedure takes $O(|M|^4)$ time. The introduction of an inequality (16) is effective (i.e., it excludes some fractional solutions) only if the resulting maximum flow is less than 2.

Separation procedure for (18): An exact separation of *zSEC* inequalities can be obtained in a similar way by solving ad-hoc max-flow problems. The procedure is a little bit trickier, but preserves a polynomial-time complexity. More precisely, given a product k, construct a graph $\mathcal{G}^k_U := (V^k, E^k)$. where the vertex set V^k contains the depot 0, the set of suppliers M_k, and an additional dummy vertex $\bar{v}$, and the edge set E^k contains all the edges $e = [i, j] : i, j \in V^k, i < j$, with capacity x^*_e, plus all the edges $e = [i, \bar{v}], i \in M_k$, with capacity $2z^*_{ik}/d_k$. Then find the partition (M', $V^k \backslash M'$) associated with a minimum-capacity cut in $\mathcal{G}^k_U$ and separating the depot 0 and the dummy vertex $\bar{v}$, with $\bar{v} \in M'$. If the capacity of this cut is at least 2, then the LP solution satisfies all the inequalities (18), for a given k. Otherwise, the set $M' \setminus \{\bar{v}\}$ yields the most violated inequality (18). Since a max-flow problem has to be solved for each product, the entire procedure takes $O(|K||M|^3)$ time.

Few years later, Riera-Ledesma and Salazar-González (2006) extend the described branch-and-cut algorithm to solve asymmetric instances. To this aim they proposed, along with the exact separation procedures for (5) and (13), that are actually analogous to the explained procedures for the symmetric variant, the heuristic separation of specific ATPP valid inequalities. The resulting branch-and-cut approach achieved similar performance with respect to its symmetric counterpart. In fact, it was able to solve to optimality, on a *AMD 1333* MHz, the asymmetric instances proposed by Singh and van Oudheusden (1997), and new random instances up to 200 suppliers and 200 products. In the same paper, the authors also detail a procedure to transform an ATPP instance into a STPP one.

5.3. Recent approaches

The DP paradigm has reappeared as solution method in Gouveia, Paias, and Voß (2011) for a U-ATPP variant in which both the number of suppliers to be visited and the number of products to purchase at each supplier are limited. The authors first test, on a *Pentium IV 3.2* GHz, the performance of *Cplex 11.0* tackling a compact ILP formulation of the problem and discover that exact solutions could be achieved in reasonable time only for instances with up to 100 suppliers. Hence, they decide to approach the larger size instances through a complex DP algorithm applied to a Lagrangian relaxation that uses a subgradient optimization procedure to compute the bounds. Due to the expected exponentially sized state space of the proposed DP, a *state space relaxation* method is used to provide a lower bound on the cost of the optimal solution. Moreover, a Lagrangian greedy heuristic that attempts to transform relaxed solutions into feasible ones is also proposed. Computational results for instances with up to 300 suppliers show reasonably small gaps between best upper and lower bound values on the optimal solutions (except for few cases). Other tests show, however, that the method does not lead to the same performance for regular TPP benchmark instances. It seems in fact that considering side-constraints strongly increase the efficiency of the DP process.

Cambazard and Penz (2012) study an U-STPP, similar to the one just described, where only the number of suppliers to visit is bounded. They propose a new Constraint Programming approach that takes advantage of three key problems related to the TPP, namely, the TSP, the *p-median*, and the *hitting* problem. The hitting problem finds the minimum subset per product containing the suppliers in which it is available, the *p*-median problem corresponds to the cheapest way of buying all the products in a subset of markets, and, finally, the TSP finds the best tour visiting the selected suppliers. Global constraints are introduced to simultaneously handle these core structures and the propagation algorithms are based again on Dynamic Programming and Lagrangian relaxation. Eventually, they test, on a *Dual Quad Core Xeon CPU 2.66* GHz, two versions of the resulting approach solving ad hoc modified instances of similar size than the ones proposed in Laporte et al. (2003). Although this approach was initially designed for solving instances with a small maximum number of supplier visits, it proves to be surprisingly competitive (differently from what happened with the work by Gouveia et al., 2011) when applied to the unbounded case. In fact, optimal solutions including up to 25 markets have been found and, among some instances not previously solved, 10 solutions have been improved. In particular, the successful idea seems to exclude the routing optimization part from the propagation search space (encapsulating the TSP inside a constraint) and to perform an exponential-time propagation only depending on the bound of the visits.

6. Heuristic algorithms

This section is devoted to present and categorize all the heuristic approaches proposed for the TPP. Section 6.1 concerns constructive and simple local search methods, Section 6.2 discusses metaheuristic frameworks, and Section 6.3 presents the only existing approximation algorithm.

6.1. Basic heuristics

Basic heuristics for the TPP include constructive procedures, simple local search methods without a defined higher-level strategy to recover from local optima, and a last class concerning heuristics based on decomposition that exploits the presence of different subproblems. In the presentation of the main results, we follow the refining process these methods have experienced in the literature.

6.1.1. Constructive methods and variants

All constructive heuristics for the TPP are based on the concept of *saving* to measure the convenience in terms of total decrease of purchasing costs net of the possible traveling costs increase, when inserting a new supplier in a solution. The first saving algorithm for the U-TPP was proposed by Golden et al. (1981). The algorithm, called *Generalized Savings Heuristic* (GSH), is a greedy procedure that adds to the current visiting cycle the most convenient supplier at each iteration. The very first added supplier is the one that sells more products at the cheapest prices (ties are solved by choosing the supplier that yields the minimal total price), and a cycle starting and ending at the depot and visiting only this supplier is created. Then, at each iteration, an unselected supplier h is added to the existing cycle between the two adjacent suppliers i and j that allow its cheapest insertion provided that this choice maximizes the saving, and if such a saving is strictly positive. More precisely, given the current cycle τ, the saving Δ^- for each tuple (i, j, h) such that $i, j \in \tau, h \notin \tau$ is calculated as:

$$\Delta^-(i, j, h) = c_{ij} - c_{ih} - c_{hj} + \sum_{k \in K} \max\{0, p_{min}(k, \tau) - p_{hk}\}$$

where $p_{min}(k, \tau) = \min_{i \in \tau}\{p_{ik}\}$. After the insertion, the purchasing plan for each visited supplier is updated according to the saving. The algorithm terminates when no supplier satisfies the positivity condition on savings (the procedure works correctly only if we consider, for each product k not available in a supplier i, a fictitious purchasing cost $p_{ik} \gg \max\{\max_{i \in M, k \in K}\{p_{ik}\}, \max_{(i,j) \in A}\{c_{ij}\}\}$.). Note that the method does not stop when a feasible solution is reached, but when the insertion fails to improve the solution. Hence, the saving (and insertion) principle is more powerful than a simple constructive heuristic and more similar to a local search operator, as highlighted in the following section.

GSH is easy to implement, and requires $O(\max\{|K|, |V|\} \cdot |V|^2)$ operations in the worst case. However, it works badly if the problem contains a supplier which sells most of the products and is located far apart from the others. To prevent this drawback, Ong (1982) proposes the *Tour Reduction Heuristic* (TRH): the algorithm starts from a tour involving a subset of suppliers satisfying the products demand, and iteratively drops the supplier yielding the maximum reduction of total costs (measured as the reduction of traveling costs net of the possible increase in purchasing costs) until a reduction is possible. Effectiveness and complexity of TRH both depend upon which procedure is used to select the initial set of suppliers and to create the relative tour (that actually corresponds to solve a TSP). Ong (1982), for example, suggests to incorporate TRH into GSH, applying the former as soon as a tour containing all products is generated by the latter.

Another constructive heuristic that considers the products one by one, called *Commodity Adding Heuristic* (CAH), is proposed for the U-TPP by Pearn (1991). CAH starts with an initial solution containing only the depot and the supplier that minimizes the total cost for purchasing the first product. At each iteration, a new product is considered and the convenience to purchase the product in one of the suppliers already included in the current solution or to add another supplier is evaluated. The algorithm terminates when all the products have been considered.

During the years, many improvements have been proposed for the basic version of GSH, TRH, and CAH (Boctor, Laporte, & Renaud, 2003; Ong, 1982; Pearn & Chien, 1998; Teeninga & Volgenant, 2004). A common practice, which seems to be quite effective, is to frequently re-sequence the order of the visited suppliers by using a TSP heuristic. Classical *cheapest-insertion* and the *Lin–Kern* heuristics (Lin & Kernighan, 1973) have been preferred to this aim. Other variants take into account a different starting solution, resolve the ties by considering other factors besides the costs, calculate the savings using some parametric weights for the two components of the objective function, or, again, use several different sequential or random orders of the products to generate several complete solutions and then select the best one. Boctor et al. (2003) also extend CAH for solving the more general R-TPP including all the variants just described.

Finally, Laporte et al. (2003) describe a *market adding heuristic* (MAH) for the R-TPP that gradually extends a cycle by inserting at each step a new supplier selling a product the demand of which is not fully satisfied. MAH determines in which supplier each product is available at the lowest price, and among these minima, adds to the tour (according to a standard maximum-saving rule) the supplier corresponding to the highest product price. Once a feasible cycle has been obtained, it is post-optimized by iteratively acting on the set of suppliers in the solution, the assignment of products, and the routing cost to visit them. The authors also apply MAH to fractional LP solutions (LP-MAH), choosing as initial cycle the one containing the edges whose associated variables have the largest values. A similar heuristic is proposed by Infante et al. (2009). In their MAH constructive phase, the farthest supplier from the ones in the current tour is added, whereas in a following improvement phase, markets are eliminated in a TRH fashion according to saving in purchasing or traveling costs.

6.1.2. Local search operators

Some of the above constructive methods, if applied to a known feasible solution, can be seen as local search moves potentially embeddable in a metaheuristic framework. In particular, GSH and TRH are formalized in Voß (1986) as local search operators based on the addition (ADD) and the deletion (DROP) of a supplier in a solution, respectively. The same author also proposes a combined use of adding and dropping moves. In DROP-ADD, an iteration is defined by a drop step followed by a number of consecutive add steps that operates on a possible unfeasible solution until no more improvement of the objective function is possible. The drop step is characterized by the exclusion of the supplier that gives the best improvement of the objective function value, if any is possible, or its smallest increase, otherwise. The procedure terminates when any supplier previously removed is added again to the solution. In ADD-DROP, each iteration consists of an add step followed by a sequence of drop steps.

ADD and DROP methods follow the maximum saving criterion presented in Section 6.1.1. However, similar neighborhoods but using different evaluation criteria have been proposed by Ochi, Silva, and Drummond (2001). AddGeni and DropGeni select the node to add or remove by using GENIUS heuristic (Gendreau, Hertz, & Laporte, 1992), whereas AddRandom and DropRandom randomly select the supplier from a candidate list.

Boctor et al. (2003) propose to explore classical TSP as well as TPP tailored neighborhoods. Concerning the latter, some new operators are introduced: double market drop considers each pair of visited suppliers and remove the one yielding the largest cost reduction, market exchange substitutes a visited supplier with a non-visited one, and double market exchange drops two consecutive suppliers in solution and substitute them with a single non-visited one. The authors combine the above procedures in different ways creating three *perturbation heuristics* (PH): UPH1 and UPH2 apply to the U-TPP, RPH to the R-TPP. More precisely, the methods are post-optimization schemes in which an improvement procedure is applied to a perturbed solution. The interest of working with a perturbed solution, although not formalized in a metaheuristic, aims at helping the search process to escape from local minima. Computational results on symmetric instances by Laporte et al. (2003) show that both UPH1 and UPH2 produce solution values within 0.75% of the optimum for $|M| \leq 200$ and close to 1% of the best-known solution value for $250 \leq |M| \leq 350$. Moreover, all PH also produce smaller optimality gaps than any other heuristic used in the comparison (CAH, MAH). This is a clear indication of the quality of these algorithms although the perturbation phase is highly time consuming.

Few years later, Riera-Ledesma and Salazar-González (2005a) generalize the supplier-exchanging neighborhood through the l-ConsecutiveExchange procedure. The method aims first at reducing the length of a feasible visiting cycle by removing l consecutive suppliers, and at restoring the feasibility, if lost, by adding suppliers not belonging to the solution that result convenient according to the classical saving criterion and by re-optimizing the purchase. An ad hoc data structure, presorted according to the products price, is used to speed up the insertion of a new supplier in a partial solution. The authors show the effectiveness of dynamically resizing the neighborhood by reducing l as soon as a new local optimum is achieved. The main idea is readapted by Mansini and Tocchella (2009a) for a R-TPP bounded version in the local search scheme called *EJEct and MOve* (EJEMO). Actually, EJEMO is an *enhanced* local search where the neighborhood is varied during the search by dynamically changing the value of parameter l and a simple tabu structure is used to avoid cycling.

Other operators based on dropping suppliers can be found. Teeninga and Volgenant (2004) propose a DropOpt(l) heuristic that removes from the solution a path consisting of l consecutive suppliers and refits it into the tour if the total cost decreases. Instead, Bontoux and Feillet (2008) introduce a multi-supplier deletion procedure called dropstar that determines, by keeping the ordering of the current tour, the optimal subsequence of suppliers (consecutive or not) to drop. Since finding this subsequence corresponds to an $\mathcal{NP}$-hard set covering problem, the authors (inspired by the algorithm proposed for the elementary shortest path problem with resource constraint in Feillet, Dejax, Gendreau, & Gueguen, 2004) search the neighborhood through a DP recursion applied to a graph obtained from the original tour.

Finally, the only local search method explicitly based on products' operations, instead of on suppliers' ones, is proposed by Mansini, Pelizzari, and Saccomandi (2005) and called ProductNeighborhood. Given a current feasible solution, this neighborhood consists of $O(|K|)$ solutions, each one obtained by setting to zero the quantities purchased of a given product k in the currently selected suppliers and by satisfying the product's demand through the introduction of non-selected ones. If the total demand for a product cannot be satisfied by the new suppliers, the ones belonging to the current solution can be used too.

In Table 2, we summarize the local search operators described in this review, indicating their basic principle and the work in which they have been presented.

6.1.3. Heuristics based on problem decomposition

Some procedures base their effectiveness on decomposing the TPP by exploiting its multi-problem nature (see Fig. 1). Once a subset of suppliers has been chosen, the TPP can be split into a products assignment problem on the selected suppliers (to minimize the purchasing costs), and a problem of finding a visiting order for them (to minimize the traveling costs). The products assignment is actually an easy problem that can be optimally solved by simple inspection or by means of an LP solver. The tour definition corresponds instead to a TSP, that is $\mathcal{NP}$-hard itself, but a plethora of exact and heuristic methods (with well-studied efficiency and effectiveness) is available in the literature for its solution. Hence, the supplier selection phase appears the most challenging, given the intractability of exhaustively exploring all the possible subsets of suppliers. In the following, we present decomposition algorithms proposed in the literature differing for (a) the method used to generate promising subsets of suppliers, and (b) the specialized procedures adopted to efficiently solve the two subproblems.

Very early, Burstall (1966) introduced a *reduction procedure* (RED) for the U-TPP that, combined to a branching tree, is able to generate all the smallest subsets of suppliers satisfying products demand. The final solution is obtained by solving the two relative subproblems for all the generated subsets, and comparing the costs. This method performs well under the strong assumption that the price differences for a given product in various suppliers are small compared to the traveling costs between them. Lomnicki (1966) commented this work showing that a simpler method based on Boolean algebra can be used to the same purpose.

Recently, Beraldi et al. (2016) propose a *Beam Search* (BS) strategy to explore a tree where each node represents a particular subset of selected suppliers. More precisely, the root node represents the selection of all the suppliers, and each child node has a se-

Table 2
Local search operators.

Principle	Name	Introduced by
Single-supplier insertion	`ADD`	Voß (1986,1996)
	`ADDGeni`	Ochi et al. (2001)
	`ADDRandom`	Ochi et al. (2001)
Single-supplier deletion	`DROP`	Voß (1986,1996)
	`DROPGeni`	Ochi et al. (2001)
	`DROPRandom`	Ochi et al. (2001)
Double-supplier deletion	`double market drop`	Boctor et al. (2003)
Multi-supplier deletion	`dropstar`	Bontoux and Feillet (2008)
Suppliers exchange (1-dropped, 1-added)	`market exchange`	Boctor et al. (2003)
Suppliers exchange (2-consecutive dropped, 1-added)	`double market exchange`	Boctor et al. (2003)
Suppliers exchange (l-consecutive dropped)	`l-ConsecutiveExchange`	Riera-Ledesma and Salazar-González (2005a)
	`EJEMO`	Mansini and Tocchella (2009a)
Changing position of l-consecutive suppliers	`DropOpt(l)`	Teeninga and Volgenant (2004)
Product purchasing exchange	`ProductNeighborhood`	Mansini et al. (2005)

Table 3
Metaheuristic approaches.

Metaheuristic	Reference	Problem
Tabu Search (TS)	Voß (1996)	U-TPP[a]
	El-Dean (2008)	U-TPP
	Mansini et al. (2005)	R-TPP
Simulated Annealing (SA)	Voß (1996)	U-TPP[a]
General Random Adaptive Search Procedure (GRASP)	Ochi et al. (2001)	U-TPP
Variable Neighborhood Search (VNS)	Ochi et al. (2001)	U-TPP
	Mansini and Tocchella (2009a,2009b)	TPP-B[b]
Ant Colony Optimization (ACO)	Bontoux and Feillet (2008)	U-ATPP
Late Acceptance Hill-Climbing (LAHC)	Goerler et al. (2013)	U-STPP
Genetic Algorithms (GA)	Ochi et al. (1997)	U-ATPP
	Goldbarg et al. (2009)	U-STPP
	Almeida et al. (2012)	2TPP[c]

[a] Considers a fixed cost for each visited supplier.
[b] Stands for *TPP with budget constraint* (see Section 7.1.2).
[c] Stands for *bi-objective TPP* (see Section 7.1.1).

lected supplier less than its father. In order to reduce the exploration, at each level of the tree, only a promising subset of nodes is taken into account for generating children. Promising nodes have to satisfy products demand and are chosen according to the joint cost evaluation of the TSP on the selected suppliers solved with a recent implementation of the *Lin–Kern* heuristic (Helsgaun, 2000) and the product assignment problem. Actually, the method is used for solving the deterministic counterpart of a stochastic R-TPP (see Section 7.2.1).

Finally, Cattrysse et al. (2006) decompose the TPP in two subproblems, one corresponding to an UFLP and the other to a TSP on the subset of suppliers previously selected. In the UFLP, the opening costs for the different plants are computed by averaging the routing costs. Then, a TSP on the open plants and using the real traveling costs is solved heuristically by means of a *guided local search*. The heuristic, in order to escape from local optima, penalizes those arcs in a current solution which are unlikely to be incorporated into a good tour. However, this decomposition method looks appropriate only when routing costs are small with respect to the purchasing ones or when their variance is limited.

6.2. Metaheuristics

A *metaheuristic* is a master strategy guiding one or more heuristics to find solutions beyond local optimality. In the following, we present the metaheuristics used to solve the TPP and the obtained computational results. Table 3 summarizes all these methods, reporting the corresponding reference and the problem variant tackled.

6.2.1. Tabu Search (TS)

Voß (1996) proposes a TS for a U-TPP generalization taking into account a fixed cost for visiting each supplier. `ADD-DROP` and `DROP-ADD` procedures (see Section 6.1.2) are used as local search. The author identifies a solution as a vector of binary values partitioned into three sets representing, respectively, forbidden suppliers, uncertain suppliers for which a decision has not been taken yet, and surely visited ones. When changing solution, the attributes may be the set of all possible changes in the assignment of values to the binary variables. Thus the approach is multi-attribute, and since the number of suppliers added and removed in a move during the search is not static, the multi-attribute move is dynamic. During the short-term memory, the tabu list management exploits dynamic interdependencies among memory attributes according to a *cancellation sequence method* (`CSM`) and a *reverse elimination method* (`REM`). These interdependencies are aspects usually neglected in the tabu literature. While `CSM` allows to diversify by imposing a tabu status to more attributes than needed, `REM` allows intensification by better searching solutions possibly overlooked by `CSM`. The author also makes recourse to a simplified *strategic oscillation* allowing for intermediate infeasibilities when dropping a supplier. The experimental analysis, run over 198 instances, show that the TS based on a `DROP-ADD` neighborhood and a `CSM` dynamic strategy provides slightly better results.

A TS for the R-STPP is also described in the unpublished work by Mansini et al. (2005). Neighborhood exploration is based on procedure `ProductNeighborhood`. Suppliers added to a solution become tabu and cannot be dropped for a given number of iterations. The authors introduce two variants for the TS, one where long term memory features are taken into account and tabu tenure is managed dynamically, and a basic one implementing only short term memory search and the tabu list is constant. *Frequency-based* memory information is used to drive the search in possibly unexplored regions of the solution space, endowing the suppliers with a counter which represents the number of times each one has been part of a feasible solution. Feasible solutions are post-optimized in terms of traveling costs by applying `GENIUS` algorithm. Proposed TS provides, on average, higher quality solutions with respect to `l-ConsecutiveExchange` heuristic on the instances by Laporte et al. (2003).

Finally, also El-Dean (2008) proposes a general TS approach. Unfortunately, no details on the move applied and on the local search neighborhood used are provided. Results consists of a few examples with up to 9 suppliers and 5 products.

6.2.2. Simulated Annealing (SA)

A straightforward application of SA to the TPP is due to Voß (1996). `DROP-ADD` and `ADD-DROP` are used again as local search.

The method is outperformed by the TS proposed in the same paper.

6.2.3. General Random Adaptive Search Procedure (GRASP)

Ochi et al. (2001) develop a GRASP-based distributed parallel algorithm for the U-TPP. The parallel implementation for GRASP is straightforward being each iteration independent from the other. The authors implement 48 variants of GRASP by combining six greedy construction methods and eight local search procedures. Some of them are new adaptations of adding and dropping procedures exposed in Section 6.1.2. The algorithms have been compared with 48 VNS variants and with two implementations of the TS proposed by Voß (1996) (considering procedure REM and CSM, respectively) on 36 medium-size instances (see Appendix A). Sequential GRASP already obtains better solutions than TS. The authors test a static load balance implementation of a parallel GRASP where the number of iterations of the sequential algorithm are equally divided among the existing processors, and a dynamic load balance where the charge of each processor changes during the search so that faster processes receive a higher number of iterations. Each processor executes either the same GRASP or a different one giving rise to two variants. Each parallel algorithm is run for 3 trials with 2000 iterations on 8 and 4 processors using five additional instances of larger size. Results show that static load balance model works better especially when the number of processors increases. Quality of the solutions improves with the number of processors, while efficiency is higher in variants implementing a different algorithm for each processor.

6.2.4. Variable Neighborhood Search (VNS)

Ochi et al. (2001) also implement a distributed parallel VNS exploring a quite new area of research. Each sequential VNS is a combination of six construction methods and eight local searches used for the GRASP, duly adapted to work in a standard VNS framework. The distributed parallel model applied to the VNS and to a hybrid variant combining GRASP with the VNS as local search, is akin to the one described for the GRASP. Computational results show that the combined GRASP+VNS, also in parallel implementation, provides the best performance in terms of efficiency and gets the best solution values in most of the cases.

Other VNS approaches can be found for the TPP with budget constraint (see Section 7.1.2).

6.2.5. Ant Colony Optimization (ACO)

Bontoux and Feillet (2008) address the U-ATPP solution with an ACO approach. A set of *traveling ants* leave the depot in parallel. When an ant returns to the depot, a new one leaves it. An ant constructs its tour by using information provided by past experience of other ants. The amount of pheromone deposited on the arcs of the tours depends on the quality of the solution. Arcs with higher level of pheromone will be preferred thus increasing the probability to visit their suppliers. Since the length of the generated tours is different, ants are not synchronized and this allows a better pheromone update (30 different levels of pheromone are considered). Four parameters measure the attraction of the ant to the pheromone, its independence in following its own path, its avidity in minimizing purchasing costs with respect to traveling costs and the reverse. These parameters define the probability to select the supplier to visit next from the one the ant is currently located. The ant population evolves by killing ants that provide (also partial) tours with a cost larger than the value of the best solution found multiplied by a given coefficient depending on the instance number of the suppliers and products. Initially each ant receives a dote that is reduced when the ant is cut and it has to restart its tour from the depot. When the dote is totally consumed, the ant is definitely cut down, and a new ant, defined as a clone of the ant having found the best-known solution, leaves the depot. Best ants are promoted by increasing their dote points, while ants visiting uninteresting parts of the solution space are eliminated. A similar reserve of points is assigned to each level of pheromone. When a pheromone level reserve has been exhausted, the level is deleted. This allows to concentrate on the more promising levels and to preserve diversity. Deletion is stopped when the number of levels goes down to a predefined value. The method sequentially applies different local search procedures to the tour obtained when an ant goes back to the depot. In particular, `dropstar` local search (see Section 6.1.2) has been developed to this aim. Unfortunately, the experimental results do not allow its complete performance evaluation. The proposed ACO is tested on the Euclidean the U-TPP instances by Laporte et al. (2003), running with a time limit of 1 hour and for 5 trials. It improves the best-known values for 48 out of 51 non closed instances.

6.2.6. Late Acceptance Hill-Climbing (LAHC)

LAHC is a recent metaheuristic based on the idea to delay the comparison between neighborhood solutions (late acceptance), to escape from local optima. Goerler, Schulte, and Voß (2013) apply LAHC to the U-STPP. The initial solution is constructed with a nearest neighborhood algorithm where suppliers are inserted ignoring purchasing costs, then once a tour is obtained the method decides which suppliers to drop maintaining feasibility and reducing traveling and purchasing costs. Then, an hill climbing procedure is applied, storing visited solutions in a list of predefined length, called *fitness array*. Contrary to a pure hill climbing, LAHC compares a candidate not to its direct previous current solution but to the last element of the fitness array. If accepted the candidate is inserted in the list and the last element removed. Before applying comparison the method uses two local search methods, the l-ConsecutiveExchange and the random addition of suppliers if the solution is not feasible. Then a TSP heuristic is applied to improve traveling costs. Algorithm is tested on *Class 1* instances (see Appendix A). Results are interesting: 25 instances with 50–250 products have been solved to optimality, whereas in 25 instances with a number of products from 300 up to 500, the algorithm was able to produce new best known upper bounds on the optimal value.

6.2.7. Genetic Algorithms(GA)

Ochi, Drummond, and Figueiredo (1997) propose for the U-ATPP a parallel GA called GENPAR, based on the *island model*. The population is partitioned into several subpopulations which evolve in parallel and periodically get in touch by migration of individuals among islands. A basic component of the method is the permutation consisting in a purchasing order represented by a vector where the i-component indicates the supplier where product i is purchased. The first permutation is generated using GSH and then distributed among all the processors. To guarantee a better search in the solution space each processor modifies the initial permutation by generating additional ones through the change of a segment (*window*) of the initial purchasing order. Each processor can only change the elements associated with its window and retains the remaining part of the permutation. The number of new permutations generated by a processor depends on the size of the assigned window. Each processor can also update permutations by applying the windows switching. The central part of the algorithm consists in 3 operators, namely, the selection, the crossover and the mutation (Ochi, Santos, Montenegro, & Maculan, 1995). In particular p parents generate $\frac{p}{2}$ children. The worst parent is substituted with the child that that better fits. In case the child is worse it can be accepted and the switch implemented with a given probability depending on the number of implemented substitutions parent/child. A 2-opt heuristic is then applied to the best

Table 4
Deterministic variants.

Problem name	Objective	Side-constraints	Reference
2TPP	$(f_1(\sigma), f_2(\sigma))$		Ravi and Salman (1999)
2TPP	$(f_1(\sigma), f_2(\sigma))$		Riera-Ledesma and Salazar-González (2005b)
TPP-B	$f_1(\sigma) + f_2(\sigma)$	$f_2(\sigma) \leq b_2$	Mansini and Tocchella (2009a,2009b)
–	$f_1(\sigma) + f_2(\sigma)$	$f_3(\sigma) \leq b_3, f_4(\sigma) \leq b_4$	Gouveia et al. (2011)
–	$f_1(\sigma) + f_2(\sigma)$	$f_4(\sigma) \leq b_4$	Cambazard and Penz (2012)
TPP-TQD	$f_1(\sigma) + f_2(\sigma)$	TQD(σ)*	Manerba and Mansini (2012a)
TPP-MSD**	$f_1(\sigma^P) + f_1(\sigma^D) + f_2(\sigma^D)$	MS-LIFO(σ^P, σ^D)***	Batista-Galván et al. (2013)

* Stands for *Total Quantity Discount* policy (Section 7.1.4).
** σ^P represent the pick-up route and σ^D the delivery one.
*** Stands for *Multiple-Stack Last In-First Out* policy (Section 7.1.3).

generated individual to improve solution value. The migration process for each processor consists in sending/receiving the best solution to/from all the other subpopulations. Tests have been run on a 8-processor SP/2 using instances with up to $|M| = 500$ and $|K| = 500$. Results show that, the parallel method is only slightly better than its sequential variant, but the CPU time improvement is quite impressive.

Finally, Goldbarg, Bagi, and Goldbarg (2009) propose a *transgenetic algorithm* (TA) inspired by two major evolutionary forces, namely, the *horizontal gene transfer* (the acquisition of foreign genes by organisms) and the *endosymbiosis* (the mutually beneficial relationship between organisms which live one, the symbiont, within another, the host). TA searches the solution space resembling the information sharing process between a host and a population of endosymbionts, in which each one represents a sequence of visited suppliers. Implicitly, products are purchased in the visited suppliers that offer them at the lowest price. The population of candidate solutions evolves by means of mutation operators that smartly use already presented local search methods (suppliers add and drop, saving criteria, TSP re-optimization). The algorithm outperforms, on instances with up to 300 suppliers and 200 products, the heuristics presented in Riera-Ledesma and Salazar-González (2005a) and Bontoux and Feillet (2008), and is able to find 26 new best known values.

6.3. Approximation algorithms

Ravi and Salman (1999) propose the only existing approximation algorithm with a performance guarantee for the *metric* TPP special case, i.e. for a STPP in which all edge costs fulfill the triangle inequality. Their poly-logarithmic worst-case ratio algorithm finds, in polynomial time, a solution whose cost is $\max\{(1+\epsilon), (1+\epsilon)O(\log^3 |V| \log\log |V|\}$ times the optimal TPP cost, for any $\epsilon > 0$. The algorithm is based on rounding procedures for the LP relaxation solution of a bi-criteria version of the TPP, and uses known results on the *Group Steiner Tree* problem. The authors also produced a constant-factor approximation algorithm for the TPP special case with metric and proportional costs (that models the ring–star network problem presented in Section 2.3).

7. Main TPP variants

As for other problems, the basic setting of the TPP can be complicated creating interesting variants or including additional constraints. In Section 7.1, we present all the TPP deterministic variants, whereas in Section 7.2, works concerning the introduction of uncertainty in the problem data are analyzed. Finally, contributions about the multi-vehicle TPP are presented in Section 7.3.

7.1. Deterministic variants

We survey the TPP as a bi-objective problem, some variants involving the introduction of side-constraints, and some others where the changes in the TPP structure are more significant. To make the definitions of these variants more clear we denote by $\sigma = (V(\sigma), A(\sigma))$ a feasible TPP solution visiting the vertices $V(\sigma) \subseteq V$ and traversing the arcs in $A(\sigma) \subseteq A$, and by Γ the set of all feasible solutions for a given TPP instance. We also define, for a given solution $\bar{\sigma}$, the following functions:

$$f_1(\bar{\sigma}) = \sum_{(i,j)\in A(\bar{\sigma})} c_{ij}, \quad f_2(\bar{\sigma}) = \sum_{k\in K} p_k^*,$$

$$f_3(\bar{\sigma}) = \max\{|K_i(\bar{\sigma})| : i \in V(\bar{\sigma})\}, \quad f_4(\bar{\sigma}) = |V(\bar{\sigma}) \setminus \{0\}|$$

where $K_i(\bar{\sigma})$ is the set of products purchased in supplier i and where, for each product $k \in K$,

$$p_k^* = \min \sum_{i\in M_k\cap V(\bar{\sigma})} p_{ik} z_{ik} : \sum_{i\in M_k\cap V(\bar{\sigma})} z_{ik} = d_k, z_{ik} \leq q_{ik}, i \in M_k \cap V(\bar{\sigma}).$$

Functions f_1 and f_2 represent the routing and the purchasing cost associated with a feasible solution $\bar{\sigma}$, respectively. Hence, TPP $= \min\{f_1(\sigma) + f_2(\sigma) : \sigma \in \Gamma\}$. Functions f_3 and f_4 represent instead the maximum number of products purchased in a supplier and the number of supplier visited in $\bar{\sigma}$, respectively. Using this notation, in Table 4 we summarize the described deterministic variants in terms of objective function and additional side-constraints.

7.1.1. The bi-objective TPP (2TPP)

In the basic TPP it is assumed that f_1 and f_2 are summed up in a single objective function. However, this sum may not make a sense in certain applications when the two cost functions represent incomparable entities (e.g., f_1 may represent distance or time and f_2 money), or when the priorities of minimizing the functions are different. For this reason, some works been have focused explicitly on the bi-objective TPP, that is 2TPP:$= \min\{(f_1(\sigma), f_2(\sigma)) : \sigma \in \Gamma\}$. The approximation algorithm by Ravi and Salman (1999) already described (see Section 6.3) applies to the 2TPP. Riera-Ledesma and Salazar-González (2005b) approach the same problem proposing an exact algorithm that explores by a binary search the objective space determining Pareto optimal solutions. Each step of the algorithm solves, by a variation of the branch-and-cut proposed in Laporte et al. (2003), the problem $\min\{\omega f_1(\sigma) + (1-\omega) f_2(\sigma) : \sigma \in \Gamma, f_1(\sigma) \leq f_1(\sigma'), f_2(\sigma) \leq f_2(\sigma')\}$ in which σ' is a currently Pareto-optimal solution and ω a dynamically generated weighting parameter. During each resolution the set of dynamic constraints generated is stored in a cut pool to be used in further stages, speeding up the global procedure. The algorithm is able to solve instances up to $|M| = 100$ and $|K| = 200$ using a 500 MHz *Pentium* computer. Finally, Almeida, Gonçalves, Goldbarg, Goldbarg, and Delgado (2012) study the application of trans-genetic algorithms (TAs), already shown to be effective for the single-objective case (see Section 6.2.7), to the 2TPP. Two novel trans-genetic multi-objective algorithms, called NSTA and MOTA/D, are proposed hybridizing state-of-the-art multi-objective evolutionary frameworks (based only on Pareto dominance) with decomposition-based TAs. The methods are validated on 365 U-TPP instances by applying Pareto compliant indicators and statistical tests. The results show the overall MOTA/D superiority, which better integrates the diversification mechanisms based on problem decomposition and the intensification of TA operators.

7.1.2. TPP with upper bound restrictions

The most studied variant of this type, the *TPP with budget constraint* (TPP-B), restricts the total purchasing cost by a constant threshold b_2, i.e., TPP-B $:= \min\{f_1(\sigma) + f_2(\sigma) : \sigma \in \Gamma, f_2(\sigma) \le b_2\}$. The TPP-B is inspired by real applications in telecommunications network design. Both the works described in Section 7.1.1 use this problem as an intermediate step to solve the 2TPP. In particular, Riera-Ledesma and Salazar-González (2005b) show that the weak LP-relaxation induced by the budget constraint produces, in their branch-and-cut algorithm, branching trees with a conspicuous number of nodes. Mansini and Tocchella (2009a), 2009b) propose then a *Multi-start VNS* (MVNS) using a modification of EJEMO algorithm as local search scheme for both the U-TPP-B and the R-TPP-B in which only the traveling costs are minimized.

Other TPP variants with similar restrictions exist. Gouveia et al. (2011) study the ATPP where the number of visited suppliers and the number of products bought per supplier are limited by two constants b_3 and b_4. This problem corresponds to $\min\{f_1(\sigma) + f_2(\sigma) : \sigma \in \Gamma, f_3(\sigma) \le b_3, f_4(\sigma) \le b_4\}$. These constraints are motivated from a production planning case study involving a furnace (the multi-purpose machine) in which jobs have to be treated at different temperatures (configurations). In the work by Cambazard and Penz (2012) a bound is only established on the number of visited suppliers (see Section 5.3).

7.1.3. TPP with multiple stacks and deliveries (TPP-MSD)

The *double TSP with multiple stacks* (DTSPMS) is a routing problem in which a capacitated vehicle has to pickup products before the deliveries (Petersen & Madsen, 2009). The TPP-MSD studied by Batista-Galván et al. (2013) generalizes the DTSPMS considering that each product is offered in several pickup locations (markets) at different prices, hence not all of them need to be visited, whereas all the delivery locations (customers), each requiring a product, must be visited. The problem has to: (a) select a subset of pickup locations; (b) determine a tour visiting them taking into account the order in which products are loaded; (c) design a Hamiltonian tour which visits the delivery locations. A branch-and-cut exploiting some new and known cuts is proposed. The algorithm, tested on 240 instances adapted from the literature using a *Intel(R) Core(TM)2* 6700 @ 2.66 GHz computer with 2 GB RAM, has been able to optimally solve instances with up to $|K| = 24$ and $|M| = 32$.

7.1.4. TPP with Total Quantity Discount (TPP-TQD)

Manerba and Mansini (2012a) introduce *Total Quantity Discount* (TQD) policies for the purchases in the TPP with restricted availabilities (R-TPP). According to TQD, the interval in which the total quantity purchased lies determines the discount applied by the supplier to the total purchase cost. More precisely, each supplier $i \in M$ defines a set $R_i = \{1, \dots, r_i\}$ of r_i consecutive and non-overlapping discount intervals $[l_i^r, u_i^r]$, where l_i^r and u_i^r are the minimum and maximum number of product units to be purchased from i to be in interval r. A discount rate δ_i^r is also associated with each interval $r \in R_i$ such that $\delta_i^{r+1} \ge \delta_i^r$, $r = 1, \dots, r_i - 1$. The authors generalize the classical R-TPP formulation to include the TQD policy modeling and propose a branch-and-cut approach exploiting known TPP valid inequalities as well as ad-hoc cuts and matheuristic strategies for the TQD subproblem (see Manerba & Mansini, 2012b; 2014). They solve instances with up to 100 suppliers, 500 products, and 5 discounts intervals per supplier using a *Intel Core Duo* 2 GHz computer, with 2 GB RAM.

7.2. Variants incorporating data uncertainty

In real problems, purchasing prices and product quantities might not be exactly known when the purchaser has to select suppliers and design the corresponding optimal tour. The presence of uncertain data forces to define when information becomes available and for which amount. This section describes the alternative approaches proposed to tackle this issue in the TPP literature.

7.2.1. Stochastic TPP

Kang and Ouyang (2011) analyze a stochastic variant of the U-TPP, where product prices are random variables following known independent (but not necessarily identical) probability distributions. In their setting, the purchaser will know the offered price (a realization from the distribution) after arriving at a supplier, and can decide whether to buy the product at the offered price, or reject it and visit another supplier (but it is not allowed to go back to any already visited supplier). The purchaser needs to determine the optimal routing and purchasing strategies that minimize the expected total costs. They propose an exact solution algorithm based on Dynamic Programming with a time and space complexity equivalent to the ones for the traditional TSP from which the method is derived. They also propose an approximate problem of lower complexity whose solution yields bounds for the minimum total expected cost, and a greedy heuristic for fast solutions to large-scale problems, quite similar to a nearest neighbor algorithm using expected prices. Instances are constructed considering origin, destination and 348 supplier locations randomly selected from nodes in the Chicago metropolitan transportation network. The numerical results show that the heuristic algorithm yields near-optimal strategies and the approximate problem provides very good estimates of the minimum total cost.

Beraldi et al. (2016) study a R-TPP in which both product quantities and prices are uncertain and propose a two-stage Stochastic Programming formulation where the first stage deals with the selection of suppliers and the minimal cost route to visit them (tactical decisions), whereas recourse decisions in the second stage are related to the products and the quantities to purchase at each supplier. To solve the deterministic equivalent problem, the authors develop a branch-and-cut method, incorporating the separation of some cuts, and three variants of the Beam Search described in Section 6.1.3. Extensive computational results show that the exact method is efficient finding the optimal solution for instances with up to 75 suppliers, 50 products and 200 scenarios in less than 2 hours.

7.2.2. Dynamic TPP

Angelelli, Mansini, and Vindigni (2009) introduce the first attempt to deal with a dynamic variant of the R-TPP, where relevant information is not completely known in advance, but revealed as time goes on. In real procurement problems, quantities available at the suppliers are time-dependent usually decreasing over time (stocks are folded each morning and reduce over the day). The authors assume to operate in a scenario where the decision maker exactly knows the current state of the product stocks. As soon as the quantity available for a given product reduces, the purchaser is informed on the amount of reduction and at which supplier it has occurred. However, he does not have any knowledge about future events. This gives rise to the need of algorithms able to take decisions rapidly on the basis of the dynamically changing information. The study aims at analyzing effectiveness of heuristics constructing solutions step by step through greedy criteria that determine the next supplier to visit and the quantities to buy there. The resulting heuristics are divided into 4 incremental levels. In any given level, the decisions take into account additional information from the previous levels, thus the last level is supposed to tackle the problem in a less myopic way. *First level* heuristics associate with each product a priority value and select, as next supplier to visit, the one with the highest priority (*product-driven criteria*). *Second level* heuristics choose the next supplier by looking more widely at all the products offered (*market-driven criteria*). In *third level*, some products are purchased in advance in sight of possible future scarcity (*consumption-driven criteria*). Finally, *fourth level* heuristics select the next supplier considering the trade-off between traveling and purchasing costs (*trade-off-driven criteria)*. Globally the authors propose 18 heuristics, 9 in the first, 5 in the second, 3 in the third, and 1 in the last level. Methods are tested on a set of problems generated by modifying a $|M| = 50$, $|K| = 50$ deterministic instance of *Class 4* (see Appendix A) and in which quantities reduce in all the suppliers according to a Poisson distribution (different values of consumption rate are considered). Since heuristics may fail in finding a feasible solution, they are also compared in terms of total product units not purchased at the end of the tour. Results show that myopic approaches are reasonable only under limited dynamism, while in more dynamic contexts the use of some future prediction may avoid to incur in highly infeasible solutions and also improve costs performance.

Afterwards, in Angelelli, Mansini, and Vindigni (2011), the same authors propose new methods considering some evaluation of future events (*look-ahead heuristics*) and compare them with the 4 best algorithms described in their previous work. The new heuristics build, execute, and eventually revise a long-term plan establishing where to go and what to buy in visited suppliers. Moreover, once obtained a feasible solution, a VNS-based improvement procedure is applied. Also look-ahead heuristics are divided into 3 incremental levels, differing for the way the solution is updated: (1) in *BasePlan*, the plan is built on the initial state of the world, and no information updates are taken into account; (2) in *Revise-Orders*, the suppliers to visit are decided once, but the list of products to buy is revised whenever necessary, in order to deal with scarceness; (3) in *DynamicPlan*, the whole plan is rebuilt when any change in the offer occurs. The authors use an experimental setting similar to Angelelli et al. (2009), generating several different sequences of consumption events. They show that look-ahead heuristics exhibit a more proactive behavior better contrasting product scarceness, although they are not always able to get feasibility.

Recently, Angelelli, Gendreau, Mansini, and Vindigni introduce a time-dependent R-TPP variant, where product quantities decrease over time at a constant rate, the replenishment of the product stocks for each supplier occurs early in the morning (before the purchasing tour starts), and product prices do not vary during the day. The problem is analyzed on a single-day horizon. The authors propose a natural MILP formulation for the problem and provide some simple valid inequalities to strengthen it. A new branching strategy, embedded in a branch-and-cut framework, is introduced to solve the problem at optimality. The authors test their method on 11 TSPLIB symmetric instances with up to 42 nodes and 10 products. Quantities decrease according to different consumption rates. Results show how the proposed approach outperforms plain CPLEX when directly used to solve the models. As expected, complexity is strongly correlated to dynamism, and for instances where feasible solutions exist, quick products depletion implies higher computational time to find optimal solutions.

7.2.3. Dynamic and stochastic TPP

Angelelli, Mansini, and Vindigni (2016) study a R-TPP including both dynamic and stochastic features. Due to the presence of other purchasers, product availabilities decrease over time according to Markov processes, independent with respect to products and suppliers. Purchasing can only be done on-site and the purchaser organizes his visit to a set of suppliers in order to maximize the probability to satisfy products demand and minimize expected total cost. Information about consumption events is made available at runtime, allowing for plan reorganization. The authors assume the presence of an executor (the driver) collecting information at visited suppliers and of a planner having the computing power to reformulate plans. Different operating scenarios occur depending on communication technology at hand that, in turn, influences the level of information available to the planner. In scenario *S1* no communication equipment is available, thus an a-priori plan is produced. On the contrary, if communication is active, in scenario *S2* a complete local information on supplier inventory is revealed at visit time, whereas scenario *S3* considers a complete global information, where the planner is continuously informed on stock levels in all markets. The problem models different application domains, from daily procurement of perishable foods to the hand out of vaccines in the spread of viral diseases.

The multi-objective nature of the problem is faced through a hierarchical evaluation of the objectives concerning unsatisfied demands and costs. Policies should first guarantee that the probability to miss some items cannot be larger than a fixed threshold, then the expected number of missing items and in sequence the expected overall costs should be minimized. The authors introduce 3 heuristic variants exploiting new information when it becomes available: the *stochastic planner* takes consumption processes into account, the *deterministic planner* cuts down the computation burden of the stochastic one by approximating the consumption processes with deterministic functions of time, and the *hybrid planner* combine the two previous ones by proposing a compromise between CPU time and quality of the results. In order to have a comparison, an *off-line* planner and a multi-scenario approach are also implemented (under scenario *S3*) and tested on randomly generated instances with up to 100 suppliers and 10 products (available at `http://or-brescia.unibs.it`) with different realizations of consumption processes. Results show that the hybrid planner does not work well when the decisions have to be taken in a long run and no re-optimization is available as in *S1*, where the stochastic planner is the winning approach. In richer information scenarios the approximation of hybrid planner becomes less critical and it highly improves its performance. In particular, in *S3* the hybrid planner comes out to be the best approach from all points of view (feasibility, cost errors, and missing items).

7.3. Multi-vehicle TPP variants

Different types of constraints (bounds on load capacity or on distance traveled, incompatibilities among products) may force, in real applications, the use of a fleet of vehicles instead of a single one. In the *multi-vehicle* TPP (MVTPP) a set F of homogeneous vehicles (with a limited capacity Q) is available at the depot for a set of purchasers collaborating to satisfy the products demand. The MVTPP aims at minimizing the overall purchasing and traveling costs deciding, for each vehicle, the purchasing plan and the corresponding visiting cycle.

Notwithstanding its relevance, the first MVTPP is proposed only quite recently by Choi and Lee (2010b), who test MTZ-based formulations for both the restricted and the unrestricted case. Later on, in Choi and Lee (2011), the same authors apply those formulations to solve the purchase of a complex system's components with the objective of maximizing the overall reliability (expressed as a non-linear function). No ad hoc solving procedure is proposed and a MIP solver is simply used to tackle a problem formulation in which the objective function is duly linearized. Due to the huge problem complexity, the largest instance solved in a reasonable

Table 5
MVTPP features in the surveyed works.

Reference	d_k	Multi-visit	$\|F\|$	Intra-route constraints	Side-constraints
Riera-Ledesma and Salazar-González (2012)	1	Denied	∞		
Riera-Ledesma and Salazar-González (2013)	1	Denied	∞	UB on route length, UB on route duration, UB on visited suppliers, LB on vehicle load	
Bianchessi et al. (2014)	$\mathbb{Z}^+$	Denied	$\mathbb{Z}^+$	UB on route length	
Manerba and Mansini (2015)	$\mathbb{Z}^+$	Allowed	$\mathbb{Z}^+$		Product incompatibilities
Shameli-Sendi et al. (2015)	$\mathbb{Z}^+$	Allowed	$\mathbb{Z}^+$		Product purchase order, independent purchasers
Gendreau et al. (2016)	1	Allowed	$\mathbb{Z}^+$		Product incompatibilities
Manerba and Mansini (2016)	$\mathbb{Z}^+$	Allowed	$\mathbb{Z}^+$	UB on route duration	Product incompatibilities

amount of time involves only 40 suppliers, 40 components and 4 vehicles. However, this paper has the merit to have gathered the attention of researchers on the MVTPP, creating a vivid stream of research. Table 5 summarizes all the papers on MVTPP surveyed in the following and published later than that initial work. The problems are classified depending on the type of product demand (unitary or not), on the possibility or not to visit each supplier more than once by different vehicles (multi-visit allowed or denied), and on the number of vehicles available in the fleet (limited or not). Moreover, intra-route constraints (besides the common upper bound on vehicle capacity) and other restrictions considered, if any, are shown.

7.3.1. School bus routing

Riera-Ledesma and Salazar-González (2012) use the MVTPP to model a school bus service where suppliers correspond to bus stops and products to students to pick-up. It is not allowed to visit the same stop with more than one bus and, since each student represents a singleton, the unrestricted MVTPP with unitary-demand is considered. The objective is to minimize the traveling costs represented by the distances traveled by buses to complete their tours and by students to reach the stops from their own homes. Note that, while constructing the best routes for the buses to carry the students to school, the problem simultaneously chooses the best stops for the students to reach. The authors present a branch-and-cut approach based on a two-index single-CF formulation (see Section 3.1.1) and the exact or heuristic separation of several families of cuts, e.g., the *Generalized Multistar*, the *Fractional Capacity*, the *zSEC* inequalities, the *lifted cycle* D_k^+ and D_k^- inequalities. An initial solution is generated by greedily finding a feasible assignment of the students to the stops and solving a VRP by using the classical Clarke and Wright (1964) method. A two-phase primal heuristic is also called after each branch-and-cut iteration in order to possibly obtain a feasible integer solution from a fractional one. The algorithm has been able to solve, in a reasonable amount of time, symmetric and asymmetric instances with up to 125 stops, 125 students, and 6 buses.

Riera-Ledesma and Salazar-González (2013) extend the school bus routing problem by including different restrictions significant for their application: (a) upper bounds on the length of each route; (b) upper bounds on the duration of each route; (c) upper bounds on the number of possible stops for each bus; (d) lower bounds on the number of students served by each bus (i.e., on the vehicle load). The resulting variants are suitable to be modeled by a set-partitioning formulation and to be efficiently solved by *column generation* (CG). In general, in fact, intra-route constraints allow the pricing problem to discard a large number of infeasible columns, speeding up its solution. The authors solve the pricing by a *q-route* (Christofides, Mingozzi, & Toth, 1981) pseudo-polynomial algorithm. Finally, they embed CG into a branch-and-bound framework to ensure integrality, and add some cuts to strengthen the LP relaxation at each node, yielding a *branch-and-price-and-cut* approach. A wide computational experience is conducted testing several combinations of the considered constraints on instances with up to 125 users and stops.

Although consolidated CG procedures exist in VRP literature, their adaptation to the MVTPP is not straightforward and deserves a brief explanation. Let R be the set of all feasible routes, θ^r a variable equal to 1 if route r belongs to the solution, and δ_{ik}^r a coefficient equal to 1 if product k is purchased from supplier i in the route r. Here, differently from common VRPs, a feasible route in R is not only a set of arcs representing a visiting tour but includes also the set of decisions guaranteeing the accomplishment of a feasible purchasing plan, i.e. ensuring that a product cannot be purchased where it is not available and that the vehicle capacity is not exceeded. Hence, a feasible MVTPP solution can be viewed as a collection of $O(|F|)$ routes in R such that the demand for each product is satisfied. A basic set partitioning formulation for the (unitary-demand) MVTPP is:

$$\min \sum_{r\in R}\left(\sum_{(i,j)\in r} c_{ij} + \sum_{k\in K}\sum_{i\in M_k} p_{ik}\delta_{ik}^r\right)\theta^r \tag{19}$$

$$\text{subject to} \quad \sum_{r\in R}\sum_{i\in M_k}\delta_{ik}^r\theta^r = 1 \quad k\in K \tag{20}$$

$$\theta^r \in \{0,1\} \quad r\in R. \tag{21}$$

This decomposition moves the complexity to the pricing problem, that results in a single-vehicle TPP for which the product demand is not defined. However, to take advantage of the well-known labeling-based solution algorithms existing in the literature, the pricing problem can be redefined as an *Elementary Shortest Path Problem with Resource Constraints* (Feillet et al., 2004) considering a graph in which nodes (k, i) correspond to all the potential assignments of a product $k \in K$ to a supplier $i \in M_k$ and the (reduced) cost of each arc depends on both the traveling and purchasing costs.

7.3.2. Distance-constrained MVTPP

Bianchessi, Mansini, and Speranza (2014) also propose a branch-and-price for a restricted MVTPP in which the length of each route is bounded. The authors decompose the problem in a more traditional way, by considering a route as a simple resource constrained tour through the suppliers and by dealing with the purchasing part in the master problem. Here, in fact, the product demand is non-unitary and the vehicles capacity is unlimited and this would have impacted dramatically on the size of a set R defined as in (19)–(21). Through the use of accelerating techniques and a restricted master heuristic, the authors are able to optimally solve instances with up to $|M| = 100$, $|K| = 200$, and $|F| = 8$. Moreover, for the first time, an empirical performance comparison of different (but equivalent) MVTPP compact formulations by using a MIP solver is presented.

7.3.3. MVTPP with pairwise incompatibility constraints (PICs)

. Manerba and Mansini (2015) introduce a MVTPP variant, named MVTPP-PIC, involving the presence of incompatibilities among product types to model real procurements where, e.g., foods should not be mixed with chemicals, or some dangerous substances may react if mixed together. This is the first MVTPP variant allowing multiple visit to the same supplier by different vehicles.

Note that, differently from common VRPs where split delivery allows savings in the traveling costs, here the multiple visit may be forced by the incompatibilities and thus may cause a traveling cost increase. The authors propose a branch-and-cut framework, based on the dynamic separation of several families of valid inequalities,and incorporating symmetry breaking constraints and a *four-step* heuristic able to find good integer solutions. The heuristic is similar to the Beam Search presented in Section 6.1.3, plus a fourth layer that represents the subproblem of assigning vehicles to products and suppliers. For the latter, the authors propose three greedy procedure (in which the assignment is done supplier by supplier, with or without privileging incompatible products, and product by product, respectively) and a MIP-based recovery procedure. The method is able to efficiently solve instances with up to 100 products, 20% of cross-incompatibilities among them, 50 suppliers, and 16 vehicles.

Gendreau, Manerba, and Mansini (2016) study the unitary-demand MVTPP-PIC applied to the daily scheduling of a set K of surgeries that can be accomplished by a set M of medical teams in a set F of multi-purpose operating rooms. Here, p_{ik} represents the time needed by team i to complete surgery k, whereas c_{ij} is the set-up time to prepare specific tools/equipment for team j if operating in the same room after team i. The objective is to minimize the overall time to accomplish the required surgeries. Incompatibilities among different surgeries arise when there exists a too high risk of reciprocal contamination for the patients or the time for correctly sterilizing the room would be prohibitive. The authors propose a branch-and-price method based on a problem decomposition similar to that proposed in Riera-Ledesma and Salazar-González (2013). The pricing problem is solved by a hybrid strategy combining two different exact methods, i.e. a labeling algorithm and a tailored branch-and-cut, that present complementary features and result effectively combinable in several ways. The algorithm strictly improve solutions for 8 open instances from the literature and optimally solve all the others within an average CPU time that is an order of magnitude lower with respect to the benchmark. The method remains effective over a new set of hard-to-solve instances with up to 70 products, 70% of cross-incompatibilities among them, and 50 suppliers.

Finally, Manerba and Mansini (2016) present a branch-and-price approach for a Nurse Routing Problem (NRP) modeled as a MVTPP. Here, a set F of nurses has to visit a set of patients M in order to perform a set K of services with different priority and importance, maximizing the overall profit associated with the care service. This routing/scheduling problem is further complicated by considering a daily working time limit for each nurse and the impossibility to perform two incompatible services to a certain patient in the same day. Note that, even if these incompatibilities can be modeled as PICs, the restriction is no longer an intra-route constraint.

7.3.4. Independent MVTPP with order

In the context of network security, where virtual appliances are chained in ordered sequences to perform different filters on the traffic, Shameli-Sendi et al. (2015) extend the basic MVTPP by considering: (a) many independent purchasers, each one having a source and a destination node, and a particular list of products to purchase (instead of multiple purchasers cooperating to satisfy a common products list); (b) a specific order for the product purchase. The resulting problem aims at minimizing the processing time needed by security functions, which is composed by the time required by the traffic to traverse the links and the time required by the traffic to be analyzed by the security appliances in the nodes. The experimental setup basically consists in integrating a MIP model solution in a cloud computing platform to show its suitability.

8. Conclusions and future research

The TPP is a procurement/routing problem that aims at selecting the purchasing plan of a set of products from a subset of suppliers, and the corresponding visiting tour, in order to satisfy a predefined products demand. The objective is to minimize the overall purchasing and traveling costs. Interesting for its wide applicability, the TPP is one of the most studied generalization of the TSP. Many papers have been written on this problem since its introduction in the optimization literature about 50 years ago, and it still attracts the attention of researchers and practitioners (more than 20 papers have appeared since the 2010). A main goal of this survey is to gather for the first time all the TPP works together[3] and to create a comprehensive starting point for anyone who will approach the problem, or some of its variants, in the future. However, we do not propose just a references sink. All the contributions have been critically reviewed, taxonomies and classifications have been standardized, and transversal comparisons have been conducted whenever possible.

It is clear that the basic TPP, in its symmetric/asymmetric or restricted/unrestricted versions, has been exhaustively analyzed by different authors and under different perspectives. A wide variety of solution methods, some of which very efficient, has been proposed to achieve optimal or near-optimal solutions. Although the computational classification of the problem predicts poor results when the purpose is to guarantee the optimality, a certain number of authors has proposed efficient exact algorithms (Dynamic Programming, branch-and-bound, branch-and-cut) to solve it. However, given the hardness of the problem and its complicate combinatorial structure, heuristic approaches represent not surprisingly the major part of existing solution methods. Several constructive and local search procedures have been presented, whereas contributions on metaheuristic are quite limited in number but, actually, explore the most part of the main known frameworks. The importance of having even a simple heuristic algorithm in place is confirmed by the fact that the quasi-totality of the existing exact methods for the TPP also include some heuristic components. In turn, mathematical/structural properties extracted to obtain exact methods have led, very often, to the basic ideas on which heuristic algorithms have been developed in later works. We believe that both exact and heuristic approaches have their own relevance, and complement one each other in the process of enriching the knowledge on the problem and its tractability.

It is clear as well that the attention of the researchers has moved, in the last years, to study interesting TPP variants able to model more realistic problems. In particular, the most recent papers focus on stochastic and multi-vehicle TPP, and on other variants involving more or less complicating side-constraints (budget restrictions, discount policies for the purchasing plan, pick-up and delivery requests, intra-route constraints, and so on). We strongly believe that future works will keep going in this direction. As a

[3] Only a bunch of papers have been excluded due to their scarce availability, as the technical report by Voß (1989), or because they are published on national journals not in English language (Choi & Lee, 2009; 2010a; 2010c).

suggestion, we remark that no existing TPP contributions consider uncertainty on the traveling costs, in contrast to the literature on other stochastic routing problems. Again, by interpreting the supplier selection part of the TPP as a tactical decision-making issue, an explicit *multi-period* variant of the problem may be of interest in organizing medium-term procurement logistics operations or in considering periodic visit to the suppliers.

Despite of the natural growing interest on more and more innovative and useful TPP variants, we believe that some work can still be done regarding the basic problem. Surely, the current literature lacks in some aspects:

- *benchmarking*: no systematic works exist on defining the generation process, the representing format, the dimension and the complexity of benchmark TPP instances. A contribution in this sense, following recent guidelines on the subject (Kendall et al., 2016), would be of great utility;
- *web-page*: in order to simplify the performance evaluation of upcoming solution methods, a maintained web-page should be of reference for the research community working on the problem, gathering and making available up-to-date best results for closed and non-closed instances;
- *libraries*: as for other well-known routing problems, open-source code libraries implementing the most used constructive heuristics as well as separation procedures for the most efficient cuts would be highly appreciated and foster the creation of more efficient TPP solution methods.

Appendix A. Benchmark instances

This appendix surveys the main classes of instances proposed in the literature for the basic TPP. We analyze the main characteristics of each class by putting the attention on maximum size of solved instances and indicating which ones (if any) have not been solved to optimality yet.

The first interesting benchmark set has been proposed in Singh and van Oudheusden (1997) for both U-ATPP and U-STPP. In the asymmetric instances, travel costs are integers randomly generated from a uniform distribution in [15, 30]. Product prices are also integer values fixed to a *big-M* constant in about the 50% of the cases, whereas the remaining ones range in $[a, a+10]$ (a is a non-specified integer value). A total of 65 instances (5 of each size) with $|M| = \{10, 15, 20, 25\}$ and $|K|$ ranging from 10 to 100 are generated and solved. In symmetric instances, traveling costs come from the 33-city TSP example described in Karg and Thompson (1964). Product prices are integers randomly generated in [0, 500]. A total of 40 instances with 10–20 suppliers, and 15–50 products are created. All instances have been solved to optimality, however symmetric ones require higher computing effort.

Almost in the same years, other authors propose benchmarks to test their methods. Ochi et al. (1997) generate a set of 80 instances divided in four classes with the following characteristics: (a) both $|M|$ and $|K|$ range in [20, 50]; (b) $|M|$ ranges in [20, 50], $|K|$ in [100, 500]; (c) $|M|$ ranges in [100, 500], $|K|$ in [20, 50]; (d) both $|M|$ and $|K|$ range in [100, 200]. Unfortunately, these instances are unusable since there are no indications on how distances and products prices have been generated and also no details on the solutions obtained on single instances. Pearn and Chien (1998) generate 30 random instances where $|M|$ ranges in [10, 50], $|K|$ in [5, 60], the traveling costs in $[1, x]$ with $15 \le x \le 140$, and the purchasing costs in $[0, y]$[4] with $5 \le y \le 75$. Some graphs are dense, others are sparse. Their implementation of the exact method by Ramesh (1981) finds the optimal solution in all the cases. Ochi et al. (2001) generate 36 U-TPP instances to test their sequential algorithms. Both $|M|$ and $|K|$ range in [50, 150], whereas the number of products per supplier is between 1 and 5. Traveling and purchasing costs are generated randomly in [10, 300]. Five additional instances with $|M|$ and $|K|$ randomly ranging in [100, 500], traveling and purchasing costs in [10, 500], and up to 100 products for supplier are also provided. No optimal solutions are reported for the instances.

Laporte et al. (2003) introduce 4 classes of test instances for the STPP, which have become the most used benchmarks (instances are available at http://webpages.ull.es/users/jriera/TPP.htm):

Class 1 contains the 33-supplier U-STPP instances defined in Singh and van Oudheusden (1997). For each instance 5 samples are generated for each $|K| = \{50, 100, 150, 200, 250\}$. Distances do not satisfy the triangle inequality. Product prices are generated in [1, 500] according to a discrete uniform distribution;

Class 2 contains 140 instances for the U-TPP randomly generated by using the process described in Pearn and Chien (1998), apart from the routing costs that are symmetric instead of asymmetric. For each combination of $|V| = \{50, 100, 150, 200, 250, 300, 350\}$ and $|K| = \{50, 100, 150, 200\}$, five samples are generated;

Class 3 contains U-TPP instances defined as for *Class 1*. Here, $|V|$ integer coordinate vertices are generated in the [0, 1000] $\times$ [0, 1000] square according to a uniform distribution and routing costs as truncated Euclidean distances. Moreover, each product k is associated with $|M_k|$ randomly selected suppliers, where $|M_k|$ is uniformly generated in $[1, |V|-1]$;

Class 4 contains R-TPP instances generated as in *Class 3*, adding a limit q_{ik} on offered quantities that is randomly taken in [1, 15] and an additional parameter λ to control the number of suppliers in a feasible solution through the product demand $d_k := \lceil \lambda \max_{i \in M_k} q_{ik} + (1-\lambda) \sum_{i \in M_k} q_{ik} \rceil$, with $0 < \lambda < 1$. Basically, the lower the λ, the higher the number of suppliers in a solution. The authors consider values of λ equal to 0.5, 0.7, 0.9, and 0.99.

All *Class 1* instances have been solved to optimality with the branch-and-cut approach proposed by Laporte et al. (2003). Similarly for *Class 2* but for two instances with ($|V|$, $|K|$) = (300, 50) and (300, 150), respectively. In *Class 3*, only 89 out of 140 instances have been solved to optimality. Euclidean travel costs seem to produce much harder instances to solve. The open instances are 51, the 40 largest ones with $|V| = \{300, 350\}$ plus other 11 instances, one instance with $(|V|, |K|) = (150, 200)$, one with (200, 200), three with (250, 100), two with (250, 150) and four with (250, 200), respectively. The best-known solution value has been provided for 3 instances (*EEuclideo.300.200.1*, *EEuclideo.300.200.5*, and *EEuclideo.350.200.4*) by the local search algorithm l-ConsecutiveExchange (Riera-Ledesma & Salazar-González, 2005a) and for all the remaining ones by the ACO approach (Bontoux & Feillet, 2008) or by the TA approach (Goldbarg et al., 2009). All *Class 4* open instances up to $(|V|, |K|) = (200, 100)$ are summarized in Table A.6. Larger instances have not been solved yet.

Finally, if we do not consider the fixed costs, the 192 instances introduced by Voß (1996) are U-TPP instances not solved to optimality yet. They are based on 3 graphs known from the literature for various routing problems with 10, 31, and 52 nodes. For each graph, $|K|$ is varied 8 times depending on $|M|$, with no more than 83 products for largest instances. The availability of each product at each supplier is decided according to a probability value (0.25,

[4] Note that allowing null product prices can generate some nonsensical solutions. We believe that, in these instances, a null price for a product should be interpreted as its non-availability in that market. However, this is not clearly explicated in the paper.

Table A.6
Class 4 instances not solved to optimality yet.

(|V|, |K|)	$\lambda = 0.5$	$\lambda = 0.7$	$\lambda = 0.8$	$\lambda = 0.9$	$\lambda = 0.95$	$\lambda = 0.99$
(100, 150)					4	
(100, 200)					1	
(150, 50)					1	1
(150, 100)				1	4	1
(150, 150)				1	5	1
(150, 200)					5	1
(200, 50)	1	1			4	1
(200, 100)		1		3	4	3
Total:	1	1	1	5	28	8

0.4, 0.6, and 0.75, respectively). The price of a non-available product is set to a prohibitively high value, otherwise is uniformly chosen in [0, 1000] in a first cost structure, or in [0, 100] in a second one.

References

Aissaoui, N., Haouari, M., & Hassini, E. (2007). Supplier selection and order lot sizing modeling: A review. *Computers and Operations Research, 34*(12), 3516–3540.
Almeida, C. P., Gonçalves, R. A., Goldbarg, E. F., Goldbarg, M. C., & Delgado, M. R. (2012). An experimental analysis of evolutionary heuristics for the biobjective traveling purchaser problem. *Annals of Operations Research, 199*(1), 305–341.
Angelelli, E., Gendreau, M., Mansini, R., & Vindigni, M. The traveling purchaser problem with time-dependent quantities, Manuscript submitted for publication.
Angelelli, E., Mansini, R., & Vindigni, M. (2009). Exploring greedy criteria for the dynamic traveling purchaser problem. *Central European Journal of Operations Research, 17*, 141–158.
Angelelli, E., Mansini, R., & Vindigni, M. (2011). Look-ahead heuristics for the dynamic traveling purchaser problem. *Computers and Operations Research, 38*(12), 1867–1876.
Angelelli, E., Mansini, R., & Vindigni, M. (2016). The stochastic and dynamic traveling purchaser problem. *Transportation Science, 50*(2), 642–658.
Balas, E., & Ng, S. M. (1989). On the set covering polytope: I. All the facets with coefficients in {0, 1, 2}. *Mathematical Programming, 43*, 57–69.
Balas, E., & Oosten, M. (2000). On the cycle polytope of a directed graph. *Networks, 36*, 34–46.
Batista-Galván, M., Riera-Ledesma, J., & Salazar-González, J.-J. (2013). The traveling purchaser problem with multiple stacks and deliveries: A branch-and-cut approach. *Computers and Operations Research, 40*(8), 2103–2115.
Bellmore, M., & Nemhauser, G. L. (1968). The traveling salesman problem: A survey. *Operations Research, 16*, 538–558.
Beraldi, P., Bruni, M. E., Manerba, D., & Mansini, R. (2016). A stochastic programming approach for the traveling purchaser problem. *IMA Journal of Management Mathematics*. http://imaman.oxfordjournals.org/content/early/2015/08/04/imaman.dpv022.
Bianchessi, N., Mansini, R., & Speranza, M. (2014). The distance constrained multiple vehicle traveling purchaser problem. *European Journal of Operations Research, 235*(1), 73–87.
Boctor, F. F., Laporte, G., & Renaud, J. (2003). Heuristics for the traveling purchaser problem. *Computers and Operations Research, 30*(4), 491–504.
Bontoux, B., & Feillet, D. (2008). Ant colony optimization for the traveling purchaser problem. *Computers and Operations Research, 35*(2), 628–637.
Burstall, R. M. (1966). A heuristic method for a job-scheduling problem. *Operational Research Quarterly, 17*(3), 291–304.
Buzacott, J. A., & Dutta, S. K. (1971). Sequencing many jobs on a multi-purpose facility. *Naval Research Logistics Quarterly, 18*(1), 75–82.
Cambazard, H., & Penz, B. (2012). A constraint programming approach for the traveling purchaser problem. In M. Milano (Ed.), *Principles and practice of constraint programming*. In *Lecture notes in computer science* (pp. 735–749). Berlin Heidelberg: Springer.
Cattrysse, D., Beullens, P., Collin, P., Duflou, J., & van Oudheusden, D. (2006). Automatic production planning of press brakes for sheet metal bending. *International Journal of Production Research, 44*(20), 4311–4327.
Choi, M. J., & Lee, S. H. (2009). Periodic heterogeneous multiple traveling purchaser problem for refuse logistics optimization. *Journal of the Korean Society of Supply Chain Management, 9*(2), 147–154.
Choi, M. J., & Lee, S. H. (2010a). Heterogenious multiple traveling purchaser problem with budget constraint. *Journal of the Korean Operations Research and Management Science Society, 35*(1), 111–124.
Choi, M.-J., & Lee, S.-H. (2010b). The multiple traveling purchaser problem. In *Proceedings of the fortieth international conference on computers and industrial engineering (CIE)* (pp. 1–5).
Choi, M. J., & Lee, S. H. (2010c). Uncapacitated multiple traveling purchaser problem. *Journal of the Korean Institute of Industrial Engineers, 36*(2), 78–86.
Choi, M.-J., & Lee, S.-H. (2011). The multiple traveling purchaser problem for maximizing system's reliability with budget constraints. *Expert Systems with Applications, 38*(8), 9848–9853.
Christofides, N., Mingozzi, A., & Toth, P. (1981). Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical Programming, 20*(1), 255–282.
Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a volume of delivery points. *Operations Research, 12*(4), 568–581.
Degraeve, Z., Labro, E., & Roodhooft, F. (2000). An evaluation of vendor selection models from a total cost of ownership prospective. *European Journal of Operational Research, 125*(1), 35–48.
Edmonds, J. (1965). Maximum matching and a polyhedron with 0,1 vertices. *Journal of the Research of the National Bureau of Standards, 69B*, 125–130.
El-Dean, R. A.-H. Z. (2008). A tabu search approach for solving the travelling purchase problem. In *Proceedings of the international conference on informatics and system, INFOS2008* (pp. 24–30). Faculty of Computers & Information, Cairo University.
Feillet, D., Dejax, P., & Gendreau, M. (2005). Traveling salesman problems with profits. *Transportation Science, 39*(2), 188–205.
Feillet, D., Dejax, P., Gendreau, M., & Gueguen, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks, 44*(3), 216–229.
Fischetti, M., Salazar-Gonzalez, J.-J., & Toth, P. (2007). The generalized traveling salesman and orienteering problems. In G. Gutin, & A. Punnen (Eds.), *The traveling salesman problem and its variations*. In *Combinatorial optimization: Vol. 12* (pp. 609–662). US: Springer.
Gendreau, M., Hertz, A., & Laporte, G. (1992). New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research, 40*(6), 1086–1094.
Gendreau, M., Manerba, D., & Mansini, R. (2016). The multi-vehicle traveling purchaser problem with pairwise incompatibility constraints and unitary demands: A branch-and-price approach. *European Journal of Operational Research, 248*(1), 50–71.
Goerler, A., Schulte, F., & Voß, S. (2013). An application of late acceptance hill-climbing to the traveling purchaser problem. In D. Pacino, S. Voß, & R. M. Jensen (Eds.), *Computational logistics*. In *Lecture notes in computer science: Vol. 8197* (pp. 173–183). Berlin Heidelberg: Springer.
Goldbarg, M. C., Bagi, L. B., & Goldbarg, E. F. G. (2009). Trans-genetic algorithm for the traveling purchaser problem. *European Journal of Operational Research, 199*(1), 36–45.
Goldberg, A., & Tarjan, R. (1988). A new approach to the maximum-flow problem. *Journal of the ACM, 35*(4), 921–940.
Goldberg, A. V., & Rao, S. (1998). Beyond the flow decomposition barrier. *Journal of the ACM, 45*(5), 783–797.
Golden, B., Levy, L., & Dahl, R. (1981). Two generalizations of the traveling salesman problem. *Omega, 9*(4), 439–441.
Gouveia, L., Paias, A., & Voß, S. (2011). Models for a traveling purchaser problem with additional side-constraints. *Computers and Operations Research, 38*(2), 550–558.
Grötschel, M., & Padberg, M. W. (1985). Polyhedral theory. In E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan, & D. B. Shmoys (Eds.), *The traveling salesman problem. A guided tour of combinatorial optimization* (pp. 251–305). Wiley. Ch. 8.
Gutin, G., & Punnen, A. P. (Eds.) (2002). The traveling salesman problem and its variations. Combinatorial optimization. Dordrecht, London: Kluwer Academic.
Helsgaun, K. (2000). An effective implementation of the Lin-Kernighan traveling salesman heuristic. *European Journal of Operational Research, 126*(1), 106–130.
Infante, D., Paletta, G., & Vocaturo, F. (2009). A ship-truck intermodal transportation problem. *Maritime Economics and Logistics, 11*(3), 247–259.
Kang, S., & Ouyang, Y. (2011). The traveling purchaser problem with stochastic prices: Exact and approximate algorithms. *European Journal of Operational Research, 209*(3), 265–272.
Karg, R. L., & Thompson, G. L. (1964). A heuristic approach to solving travelling salesman problems. *Management Science, 10*(2), 225–248.
Kendall, G., Bai, R., Błazewicz, J., De Causmaecker, P., Gendreau, M., John, R., et al. (2016). Good laboratory practice for optimization research. *Journal of Operational Research Society, 67*(4), 676–689.
Laporte, G., Riera-Ledesma, J., & Salazar-González, J.-J. (2003). A branch-and-cut algorithm for the undirected traveling purchaser problem. *Operations Research, 51*(6), 940–951.
Lin, S., & Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations Research, 21*(2), 498–516.
Lomnicki, Z. A. (1966). Job scheduling. *Operations Research, 17*(3), 314–317.
Manerba, D. (2015). Optimization models and algorithms for problems in procurement logistics. *4OR-A Quarterly Journal of Operations Research, 13*(3), 339–340.
Manerba, D., & Mansini, R. (2012a). The capacitated traveling purchaser problem with total quantity discount. In *Proceedings of the fifth international workshop on freight transportation and logistics, Odysseus 2012, Mykonos, Greece* (p. 42).
Manerba, D., & Mansini, R. (2012b). An exact algorithm for the capacitated total quantity discount problem. *European Journal of Operational Research, 222*(2), 287–300.
Manerba, D., & Mansini, R. (2014). An effective matheuristic for the capacitated total quantity discount problem. *Computers and Operations Research, 41*(1), 1–11.
Manerba, D., & Mansini, R. (2015). A branch-and-cut algorithm for the multi-vehicle traveling purchaser problem with pairwise incompatibility constraints. *Networks, 65*(2), 139–154.

Manerba, D., & Mansini, R. (2016). The nurse routing problem with workload constraints and incompatible services. *IFAC-PapersOnLine, 49*(12), 1192–1197. http://dx.doi.org/10.1016/j.ifacol.2016.07.670. 8th IFAC Conference on Manufacturing Modelling, Management and Control (MIM) Troyes, France, 28–30 June 2016

Mansini, R., Pelizzari, M., & Saccomandi, R. (2005). An effective tabu search algorithm for the capacitated traveling purchaser problem. *Technical report TR2005-10-49*. DEA, University of Brescia.

Mansini, R., & Tocchella, B. (2009a). Effective algorithms for a bounded version of the uncapacitated TPP. In J. A. Nunen, M. G. Speranza, & L. Bertazzi (Eds.), *Innovations in distribution logistics*. In *Lecture notes in economics and mathematical systems: Vol. 619* (pp. 267–281). Berlin: Springer.

Mansini, R., & Tocchella, B. (2009b). The traveling purchaser problem with budget constraint. *Computers and Operations Research, 36*(7), 2263–2274.

Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM, 7*(4), 326–329.

Ochi, L. S., Drummond, L., & Figueiredo, R. (1997). Design and implementation of a parallel genetic algorithm for the travelling purchaser problem. In *Proceedings of the 1997 ACM symposium on applied computing* (pp. 257–262). ACM.

Ochi, L. S., Santos, E. M., Montenegro, A. A., & Maculan, N. (1995). Genetic algorithm for the traveling purchaser problem. In *Proceedings of the metaheuristics international conference (MIC1 – 95)*.

Ochi, L. S., Silva, M. B., & Drummond, L. (2001). Metaheuristics based on grasp and VNS for solving traveling purchaser problem. In *Proceedings of the fourth metaheuristic international conference (MIC'2001)* (pp. 489–494).

Ong, H. L. (1982). Approximate algorithms for the travelling purchaser problem. *Operations Research Letters, 1*(5), 201–205.

Padberg, M., & Rinaldi, G. (1991). A branch and cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review, 33*, 60–100.

Pearn, W. L. (1991). On the traveling purchaser problem. *Technical report*. Department of Industrial Engineering and Management, National Chiao Tung University. Working Paper 91-01

Pearn, W. L., & Chien, R. C. (1998). Improved solutions for the traveling purchaser problem. *Computers and Operations Research, 25*(11), 879–885.

Petersen, H. L., & Madsen, O. B. (2009). The double travelling salesman problem with multiple stacks – Formulation and heuristic solution approaches. *European Journal of Operational Research, 198*, 139–147.

Ramesh, T. (1981). Travelling purchaser problem. *Opsearch, 18*(2), 78–91.

Ravi, R., & Salman, F. S. (1999). Approximation algorithms for the traveling purchaser problem and its variants in network design. In J. Nešetřil (Ed.), *Algorithms - ESA' 99*. In *Lecture notes in computer science: Vol. 1643* (pp. 29–40). Berlin Heidelberg: Springer.

Riera-Ledesma, J. (2002). *The traveling purchaser problem*. DEIOC, Univ. de La Laguna Ph.D. thesis.

Riera-Ledesma, J., & Salazar-González, J.-J. (2005a). A heuristic approach for the traveling purchaser problem. *European Journal of Operational Research, 162*(1), 142–152.

Riera-Ledesma, J., & Salazar-González, J.-J. (2005b). The biobjective travelling purchaser problem. *European Journal of Operational Research, 160*(3), 599–613.

Riera-Ledesma, J., & Salazar-González, J.-J. (2006). Solving the asymmetric traveling purchaser problem. *Annals of Operations Research, 144*(1), 83–97.

Riera-Ledesma, J., & Salazar-González, J.-J. (2012). Solving school bus routing using the multiple vehicle traveling purchaser problem: A branch-and-cut approach. *Computers and Operations Research, 39*(1), 391–404.

Riera-Ledesma, J., & Salazar-González, J.-J. (2013). A column generation approach for a school bus routing problem with resource constraints. *Computers and Operations Research, 40*(2), 566–583.

Schwarze, S., & Voß, S. (2015). Interaction of maritime shipping and hinterland traffic using a two-level hierarchical transport network. *International Journal of Logistics Research and Applications, 18*(3), 276–290.

Shameli-Sendi, A., Jarraya, Y., Fekih-Ahmed, M., Pourzandi, M., Talhi, C., & Cheriet, M. (2015). Optimal placement of sequentially ordered virtual security appliances in the cloud. In *Proceedings of the IFIP/IEEE international symposium on integrated network management (IM)* (pp. 818–821).

Singh, K. N., & van Oudheusden, D. L. (1997). A branch and bound algorithm for the traveling purchaser problem. *European Journal of Operational Research, 97*(3), 571–579.

Teeninga, A., & Volgenant, A. (2004). Improved heuristics for the traveling purchaser problem. *Computers and Operations Research, 31*(1), 139–150.

Toth, P., & Vigo, D. (2014). Vehicle routing: Problems, methods, and applications. MOS-SIAM series on optimization (2nd ed.). Philadelphia, PA: Society for Industrial and Applied Mathematics.

Voß, S. (1986). ADD- and DROP-procedures for the travelling purchaser problem. *Methods of Operations Research, 53*(3), 317–318.

Voß, S. (1989). The traveling purchaser problem with fixed costs. *Technical report*. TH Darmstadt.

Voß, S. (1990). Designing special communication networks with the traveling purchaser problem. In *Proceedings of the first ORSA telecommunications conference* (pp. 106–110).

Voß, S. (1996). Dynamic tabu search strategies for the traveling purchaser problem. *Annals of Operations Research, 63*(2), 253–275.

Wikström, P., & Eriksson, L. O. (2000). Solving the stand management problem under biodiversity-related considerations. *Forest Ecology and Management, 126*(3), 361–376.