



Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

A branch-and-cut framework for the consistent traveling salesman problem



Anirudh Subramanyam, Chrysanthos E. Gounaris*

Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA

ARTICLE INFO

Article history:

Received 28 January 2015

Accepted 10 July 2015

Available online 17 July 2015

Keywords:

Traveling salesman problem

Multi-period routing

Service consistency

Branch-and-cut

ABSTRACT

We develop an exact solution framework for the Consistent Traveling Salesman Problem. This problem calls for identifying the minimum-cost set of routes that a single vehicle should follow during the multiple time periods of a planning horizon, in order to provide consistent service to a given set of customers. Each customer may require service in one or multiple time periods and the requirement for consistent service applies at each customer location that requires service in more than one time period. This requirement corresponds to restricting the difference between the earliest and latest vehicle arrival-times, across the multiple periods, to not exceed some given allowable limit. We present three mixed-integer linear programming formulations for this problem and introduce a new class of valid inequalities to strengthen these formulations. The new inequalities are used in conjunction with traditional traveling salesman inequalities in a branch-and-cut framework. We test our framework on a comprehensive set of benchmark instances, which we compiled by extending traveling salesman instances from the well-known TSPLIB library into multiple periods, and show that instances with up to 50 customers, requiring service over a 5-period horizon, can be solved to guaranteed optimality. Our computational experience suggests that enforcing arrival-time consistency in a multi-period setting can be achieved with merely a small increase in total routing costs.

© 2015 Elsevier B.V. and Association of European Operational Research Societies (EURO) within the International Federation of Operational Research Societies (IFORS). All rights reserved.

1. Introduction

Vehicle routing problems with explicit focus on service quality and customer satisfaction have seen a recent surge in popularity and applicability. A key characteristic of high-quality customer service is its consistency over time, and this is particularly important when companies are faced with routing decisions spanning over multiple time periods. In such cases, it is generally desirable to provide consistent service to regular customers, i.e., customers who require frequent service across the different periods. Kovacs, Golden, Hartl, and Parragh (2014a) provide a review of routing problems in which explicit considerations of service consistency have been addressed. They define and classify service consistency as any of the requirements for arrival-time consistency, for which a customer should be visited at roughly the same time during the day, person-oriented consistency, for which a customer should always be visited by the same driver, and delivery consistency, for which a customer should be receiving roughly the same quantity of goods each time service is provided. Applications in which such consistent service considerations

are particularly relevant include small package shipping services (Groër, Golden, & Wasil, 2009), transportation of people with disabilities (Feillet, Garaix, Lehuédé, Péton, & Quadri, 2014) and Vendor-Managed Inventory distribution (Coelho, Cordeau, & Laporte, 2012).

This paper focuses on the aspect of arrival-time consistency. In this context, the supplier aims to reduce the variability in the actual times during the routing horizon at which a customer is served, since doing so generally increases the value of service for the customer. For example, in the context of VMI distribution, arrival-time consistency reduces the need for the customer to commit loading-dock resources throughout the day. In the home-care industry, the elderly and disabled are sensitive to changes in their daily routines and developing consistent schedules is of particular importance. Moreover, from the service provider's point of view, reducing the variability across repetitive deliveries over multiple time periods can expose efficiencies that add up to significant cost savings.

The Consistent Traveling Salesman Problem (ConTSP) is a variant of the well-known Traveling Salesman Problem (TSP) that attempts to address the issue of arrival-time consistency in multi-period routing applications. In the ConTSP, we aim to design minimum-cost (or, minimum-makespan) routes over a finite, multi-period horizon so as to serve a set of customers with known demands and service durations using a single (uncapacitated) vehicle. In general, a customer

* Corresponding author. Tel.: +1 412 268 6974.

E-mail addresses: asubramanyam@cmu.edu (A. Subramanyam), gounaris@cmu.edu (C.E. Gounaris).

may or may not require service in a given time period and, thus, only a subset of customers need to be visited in each period. The consistency requirement applies to every customer who requires service in more than one time period, meaning that every such customer must be visited at roughly the same time in each period for which service is required. The exact time of service remains a decision variable, but the arrival times across the multiple periods at each customer site must not differ by more than some prespecified, constant bound, which we call the maximum allowable arrival-time differential. In practice, this bound may be set by either the customer or by the service provider. Note that, if the maximum allowable arrival-time differential is chosen to be equal to infinity, then the ConTSP reduces to a set of separable TSPs (one for each time period); hence, the ConTSP is \mathcal{NP} -hard, just like the TSP (Karp, 1972).

Kovacs et al. (2014a) remark that the problem of visiting customers at consistent times over the different periods in a planning horizon is similar to single-period, multi-vehicle routing problems with temporal synchronization of vehicles (Drexler, 2012). In particular, consistent routes in a multi-period setting can be viewed as equivalent to several synchronized vehicles that must arrive to customer sites almost simultaneously during a single-period; the solution of the latter problem can be recovered as the union of all single-vehicle routes in the former problem. Another related problem that addresses periodicity in routing is the Periodic Vehicle Routing Problem (Campbell & Wilson, 2014), wherein each customer requires one or more visits over a planning horizon and the decision-maker must select in which time periods to provide these visits, while minimizing the total distance traveled over the planning horizon.

We highlight that the ConTSP constitutes a special, single-vehicle case of the Consistent Vehicle Routing Problem (ConVRP), originally introduced by Groër et al. (2009). The ConVRP utilizes multiple capacitated vehicles in order to provide consistent service to a set of customers over multiple periods while minimizing total transportation cost. In addition to considering arrival-time consistency, the ConVRP also requires driver consistency, for which each customer must be visited by the same driver in all periods of the planning horizon. A number of metaheuristic approaches to solve the ConVRP have been proposed in the literature. Groër et al. (2009) present an algorithm that is based on generating a “template” route by following a precedence principle: if customer a is visited before customer b in a time period, then a should be visited before b in every time period. The template route is built by considering only customers that require service in more than one time period. The individual routes in each time period are then generated by deleting those customers in the template who do not require service in that period, while those customers that require service in this period but are not part of the template route are inserted into the solution at the best position. Kovacs, Parragh, and Hartl (2014c) build upon this algorithm by allowing deviations from the precedence template using an adaptive large neighborhood search procedure; Tarantilis, Stavropoulou, and Repoussis (2012) modify both the template route and the actual single period routes using tabu search. Note that, in all of the above approaches, waiting is not allowed; that is, the vehicle is not allowed to wait at a customer location before providing service. Moreover, in the papers by Groër et al. (2009) and Tarantilis et al. (2012), the departure times of the vehicles from the depot are fixed: vehicles must depart from the depot at time zero in each period. Kovacs et al. (2014c) relax the latter requirement by allowing the vehicle to delay its departure from the depot and demonstrate that service quality can be improved without increasing driver working time. In a more recent paper, Kovacs, Golden, Hartl, and Parragh (2014b) consider a generalization of the ConVRP in which a limited number of different drivers may visit a customer, the maximum difference in the earliest and latest arrival times is penalized in the objective (i.e., arrival-time consistency is treated as a soft requirement), and each customer is associated with either of two available time windows (e.g., AM or

PM). Finally, Luo, Qin, Che, and Lim (2015) study a multi-period variant of the Vehicle Routing Problem with Time Windows, wherein each customer may be visited by only a limited number of drivers over the routing horizon, in addition to being serviced within a given time window.

In this work, we present a new, branch-and-cut-based algorithm to address the ConTSP, which to the best of our knowledge is the first attempt in the open literature to address a routing problem with consistency requirements in an exact framework. Our motivation to address the special, single-vehicle case of the ConVRP is three-fold: (i) an exact approach, even if only applicable to single-vehicle instances, has the potential to provide higher quality solutions with provable guarantee of optimality for those instances, (ii) an efficient solution scheme for the ConTSP could serve as a component of a hybrid metaheuristic approach, or even a decomposition-based¹ exact approach, for the ConVRP, and (iii) our contribution can influence the development of new exact algorithms to address more complex problem settings for consistent routing. We note that branch-and-cut approaches have, in fact, been employed successfully in the past to address the “basic” TSP (Applegate, Bixby, Chvatal, & Cook, 2007; Fischetti, Lodi, & Toth, 2007) and many of its variants and extensions, including among others the TSP with Time Windows (Ascheuer, Fischetti, & Grötschel, 2001), the TSP with Pickups and Deliveries (Dumitrescu, Ropke, Cordeau, & Laporte, 2010), the Prize-collecting TSP (Bérubé, Gendreau, & Potvin, 2009) and the Capacitated Vehicle Routing Problem (Lysgaard, Letchford, & Eglese, 2004). The development of these approaches has largely contributed towards extending the results of polyhedral combinatorics developed for the TSP by Grötschel and Padberg (1979).

We present three alternative mixed-integer linear programming formulations for the ConTSP and compare their performance when serving as the basis of our branch-and-cut scheme. In order to strengthen these formulations, we separate valid single-period TSP cutting planes. More specifically, in our implementation we use Subtour Elimination Constraints and 2-matching Constraints. We further derive a new class of valid inequalities for the ConTSP, which we refer to as Inconsistent Path Elimination Constraints, and present an efficient heuristic to separate them. These new inequalities are cross-period cutting planes which forbid the simultaneous occurrence of two paths that, if allowed to co-exist in the solution, would violate the arrival-time consistency requirement for one or more customers (here, a “path” is a sequence of customers appearing consecutively in a tour). Our algorithm is able to address instances containing up to 50 customers that require service over a 5-period horizon. In the instances we considered, we observed that increasing the quality of provided service by visiting each customer at consistent times does not have to come at a high price. In fact, the average routing cost increase was a modest 1.3 percent of the costs that would have been incurred if consistency considerations were not taken into account.

We can synopsise the contributions of this paper as follows:

- We develop a branch-and-cut-based solution framework that can solve instances of the ConTSP to guaranteed optimality.
- We propose the novel class of Inconsistent Path Elimination Constraints, which are valid for the ConTSP, and we present a polynomial-time algorithm to separate these cuts in the context of the above solution framework.
- We show that, unlike the case of the TSP, the Subtour Elimination Constraints and the 2-matching Constraints are not, in general, facet-defining for the ConTSP polytope.
- We compile a comprehensive database of benchmark instances for the ConTSP by extending single-period TSP instances from the TSPLIB library.

¹ For example, if the assignment of customers to vehicles/drivers is fixed, then the ConVRP decomposes into several ConTSPs, i.e., one for each vehicle/driver.

- We calculate, for the above database, the additional cost that a distributor must incur, on average, in order to implement consistent delivery schedules; that is, we quantify the “price of consistency.”

The remainder of the paper is structured as follows. In Section 2, we formally define the problem and introduce necessary notation. In Section 3, we present three formulations of the ConTSP and discuss some empirical dominance relationships. In Section 4, we present a number of valid inequalities for the above formulations and we discuss their lack of the facet-inducing property. In Section 5, we discuss the implementation of our branch-and-cut algorithm, including separation routines for each of the valid inequalities. Finally, in Section 6, we present computational results on a set of ConTSP benchmark problems, elucidating the cost of providing consistent service.

2. Problem definition and notation

Let $G = (V, A)$ be the complete directed graph on $n + 1$ nodes, where $V := \{0, 1, \dots, n\}$ is the node set and $A := \{(i, j) \in V \times V : i \neq j\}$ is the arc set. Node 0 represents the depot, while the node subset $V_c = V \setminus \{0\}$ represents the set of customers. Associated with every arc $(i, j) \in A$ is a travel cost $c_{ij} \geq 0$ and a travel duration $t_{ij} \geq 0$. Service (a.k.a., processing) times can be easily incorporated into the travel durations via the operation $t_{ij} \leftarrow t_{ij} + s_i$, where $s_i \geq 0$ is the service time of each customer $i \in V_c$ and $s_0 = 0$. We remark that we do not require symmetric costs or travel times; that is, we allow $c_{ij} \neq c_{ji}$ or $t_{ij} \neq t_{ji}$ for any $(i, j) \in A$. In either case, since in general service times differ among nodes, the ConTSP constitutes an asymmetric problem even if the costs and travel times are themselves symmetric.

Let also $\mathcal{P} = \{1, \dots, h\}$ denote the set of time periods such that $h \geq 2$. For each period $p \in \mathcal{P}$, we define the node subset $V_p \subseteq V_c$ to be the set of customers requiring service in this period, and we define the associated subset of arcs to be $A_p := \{(i, j) \in (V_p \cup \{0\}) \times (V_p \cup \{0\}) : i \neq j\}$. W.l.o.g, we assume for each period $p \in \mathcal{P}$ that $V_p \cap [\cup_{q \in \mathcal{P} : q \neq p} V_q] \neq \emptyset$; that is, each set V_p is non-empty² and it in fact includes at least one customer requiring service in some additional period.³

A tour in period p is a Hamiltonian cycle in $G_p := (V_p \cup \{0\}, A_p)$. We refer to such a tour via notation $T_p = \langle 0, v_1, v_2, \dots, v_{|V_p|}, 0 \rangle^p$, where it is implied that each customer v_k is unique. Given a tour T_p , we define its cost as $c(T_p) := \sum_{k=1}^{|V_p|} c_{v_{k-1}v_k} + c_{v_{|V_p|}0}$. We also define the arrival time at the i th node, $v_i \in V_p$, as $a_{v_i}^p = \sum_{i'=1}^i t_{v_{i'-1}v_{i'}}$, where $v_0 = 0$ is introduced for notational convenience. We assume that no waiting is allowed; that is, the vehicle departs in each period at time 0 and does not wait at customer locations before starting service. The above definition of arrival time reflects this assumption.

A ConTSP solution is a collection of tours $\{T_1, T_2, \dots, T_h\}$, i.e., one tour for each period $p \in \mathcal{P}$. Given such a collection, we define $\Delta a_i^{\max} := \max_{p \in \mathcal{P} : i \in V_p} a_i^p - \min_{p \in \mathcal{P} : i \in V_p} a_i^p$ to be the arrival-time differential for a customer i . In the ConTSP, we want to enforce that this arrival-time differential does not exceed a maximum allowable value $L > 0$ for all customers $i \in V_c$; that is, a ConTSP solution is feasible if and only if $\max_{i \in V_c} \Delta a_i^{\max} \leq L$. A collection of tours is said to be consistent if they induce a feasible ConTSP solution. The objective of the ConTSP is then to determine the collection of consistent tours with the minimum sum of costs, $\sum_{p \in \mathcal{P}} c(T_p)$.

² If the input data provides for a period in which no customers require service, then that period can be removed from consideration.

³ If the input data provides for a time period p in which every customer that requires service in that period does not require service in any additional period $q \neq p$, then we can independently address period p by solving its corresponding TSP instance and appending its solution to the solution of the ConTSP instance induced by the remaining periods.

Throughout the paper, we use the following notation. For each customer $i \in V_p \cup \{0\}$, let $N_p^+(i)$ denote the set of nodes j for which there is an arc from i to j in the graph G_p , i.e., $N_p^+(i) := \{j \in V : (i, j) \in A_p\}$. Similarly, let $N_p^-(i) := \{j \in V : (j, i) \in A_p\}$. Finally, given a subset of nodes $S \subseteq V_p \cup \{0\}$, let $A(S)$ be the set of arcs with both end points in S , i.e., $A(S) := \{(i, j) \in S \times S : i \neq j\}$, and let $\delta(S)$ be the set of arcs with exactly one end point in S , i.e., $\delta(S) := \{(i, j) \in S \times S^c\} \cup \{(i, j) \in S^c \times S\}$, where $S^c = (V_p \cup \{0\}) \setminus S$.

3. Formulations

We now present three alternative mixed-integer linear programming formulations that can serve as the basis for a branch-and-cut approach. These formulations differ in how they encode the various useful quantities (e.g., the arrival times at customer locations) and how they enforce the ConTSP’s requirements, namely the requirement that each period’s tour corresponds to a Hamiltonian tour (i.e., each node is incident to one outgoing and one incoming arc and there are no subtours) as well as that the tours across all periods are consistent (in the sense described in Section 2). Sometimes, a requirement is described via a superpolynomial set of constraints, which will have to be added dynamically in the context of a branch-and-cut framework. In the following, we will discuss the various strategies that can be followed in each case with regards to which constraints are to be utilized as cutting planes.

Let binary variables x_{ijp} be defined as follows:

$$x_{ijp} = \begin{cases} 1, & \text{if arc } (i, j) \in A_p \text{ is used in the tour of period } p \\ 0, & \text{otherwise} \end{cases}$$

Using these variables, the ConTSP can be cast with the following conceptual formulation:

$$\min \sum_{p \in \mathcal{P}} \sum_{(i,j) \in A_p} c_{ij} x_{ijp} \tag{1}$$

$$\text{s.t. } x_{ijp} \in \{0, 1\} \quad \forall (i, j) \in A_p, \forall p \in \mathcal{P} \tag{2}$$

$$\sum_{j \in N_p^+(i)} x_{ijp} = 1 \quad \forall i \in V_p \cup \{0\}, \forall p \in \mathcal{P} \tag{3}$$

$$\sum_{j \in N_p^-(i)} x_{jip} = 1 \quad \forall i \in V_p \cup \{0\}, \forall p \in \mathcal{P} \tag{4}$$

$$\{(i, j) \in A_p : x_{ijp} = 1\} = T_p \quad \forall p \in \mathcal{P} \tag{5}$$

$$\{T_p, T_q\} \text{ is consistent} \quad \forall (p, q) \in \mathcal{P} \times \mathcal{P} : p < q \tag{6}$$

In this formulation, the assignment (degree) constraints (3) and (4) ensure that each node is incident to one outgoing and one incoming arc in each period where it appears. Constraints (5) eliminate subtours by imposing that only Hamiltonian circuits be considered in each time period. Finally, constraints (6) ensure that the tours across all periods are consistent (in the sense described in Section 2).

The formulations we are about to present below differ in the way they achieve the requirements of constraints (5) and (6). The first formulation involves only the binary arc variables x_{ijp} defined above and utilizes a superpolynomial number of constraints to eliminate subtours and enforce consistency. In addition to binary arc variables x_{ijp} , the second formulation utilizes continuous node variables α_{ip} to represent the arrival time at customer i in period p , and employs big-M constraints to appropriately encode them. The use of big-M constraints is a common way to encode arrival times in formulations for the Vehicle Routing Problem with Time Windows (see, Desrochers & Laporte, 1991). The third formulation uses continuous arc variables y_{ijp} in addition to binary variables x_{ijp} and is based on a single-commodity flow representation of travel time.

3.1. Formulation 1

This model does not utilize any additional variables beyond the binary variables x_{ijp} defined above. Constraints (5) are modeled via Eq. (7), which constitute the standard Subtour Elimination Constraints (SEC) originally proposed by Dantzig, Fulkerson, and Johnson (1954).

$$\sum_{(i,j) \in A(S)} x_{ijp} \leq |S| - 1 \quad \forall S \subseteq V_p \cup \{0\} : 2 \leq |S| \leq |V_p| - 1, \quad (7)$$

$$\forall p \in \mathcal{P}$$

The consistency constraints (6) are modeled using a new class of infeasible path elimination constraints (see, Ascheuer, Fischetti, & Grötschel, 2000), which we introduce in this work and refer to as Inconsistent Path Elimination Constraints (IPEC). Note that two paths in two different periods are said to be “inconsistent” if their simultaneous occurrence as sub-paths in the tours of their respective time periods renders the ConTSP solution infeasible, and each IPEC will forbid such an occurrence for a pair of inconsistent paths.

Let $P = (v_1, \dots, v_k)^p$ denote a non-empty path in period p that is formed by the arcs in the set $\{(v_i, v_{i+1}) : i = 1, \dots, k - 1\}$, where $(v_i, v_{i+1}) \subseteq A_p$ for all $i = 1, \dots, k - 1$. We assume this path to be open and simple, i.e., $k > 1$ and $v_i \neq v_j$ for $i \neq j$, and we use $|P| = k - 1$ to denote the path length, which is equal to the cardinality of the above arc set. Let also $\tau(P) = \sum_{i=1}^{k-1} t_{v_i v_{i+1}}$ denote the time spent traveling on this path and processing all its nodes except the last one, v_k . Given these definition, sufficient conditions for inconsistency of a pair of paths are given in Lemma 1.

Lemma 1. A pair of paths $\{P, Q\}$, where $P = (v_1, \dots, v_s)^p$, $Q = (w_1, \dots, w_t)^q$ and $p \neq q$, is inconsistent, if any of the following conditions holds:

- (i) $w_1 = v_1 = 0$, $w_t = v_s$ and $|\tau(P) - \tau(Q)| > L$
- (ii) $w_1 = v_1 \neq 0$, $w_t = v_s \neq 0$ and $|\tau(P) - \tau(Q)| > 2L$
- (iii) $w_1 = v_s \neq 0$, $w_t = v_1 \neq 0$ and $\tau(P) + \tau(Q) > 2L$

Proof. Consider a ConTSP solution in which P and Q appear as sub-paths in time periods p and q respectively. Denote by a_c^p the arrival time at customer c in period p . It holds from their definitions that $\tau(P) = a_{v_s}^p - a_{v_1}^p$ and $\tau(Q) = a_{w_t}^q - a_{w_1}^q$. Let us take into account these relationships in the context described by each of the three conditions:

- (i) Since $w_t = v_s$, we have $a_{w_t}^q = a_{v_s}^q$. Furthermore, since the vehicle always departs from the depot at time 0, we have $a_{v_1}^p = a_{w_1}^q = 0$. Therefore, the condition $|\tau(P) - \tau(Q)| > L$ implies that $|a_{v_s}^p - a_{v_s}^q| > L$; that is, customer v_s violates the maximum allowable arrival-time differential, rendering the pair of paths $\{P, Q\}$ to be inconsistent.
- (ii) Since $w_1 = v_1$ and $w_t = v_s$, we have $a_{w_1}^q = a_{v_1}^q$ and $a_{w_t}^q = a_{v_s}^q$. Therefore, the condition $|\tau(P) - \tau(Q)| > 2L$ implies that $|a_{v_s}^p - a_{v_1}^p - a_{v_s}^q + a_{v_1}^q| > 2L$, which in turn implies that either $|a_{v_s}^p - a_{v_1}^p| > L$ or $|a_{v_s}^q - a_{v_1}^q| > L$ (or both); that is, at least one of the two customers v_1 and v_s violates the maximum allowable arrival-time differential, rendering the pair of paths $\{P, Q\}$ to be inconsistent.
- (iii) Since $w_1 = v_s$ and $w_t = v_1$, we have $a_{w_1}^q = a_{v_s}^q$ and $a_{w_t}^q = a_{v_1}^q$. Therefore, the condition $\tau(P) + \tau(Q) > 2L$ implies that $(a_{v_s}^p - a_{v_1}^p + a_{v_1}^q - a_{v_s}^q) > 2L$, which in turn implies that $|a_{v_s}^p - a_{v_1}^p + a_{v_1}^q - a_{v_s}^q| > 2L$; that is, the same conclusion as in case (ii) above can be reached. \square

Lemma 1 states that if P and Q have common end nodes and if one of them, say v_1 , is to be visited at consistent times in periods p and q , then the other common end node, v_s , cannot be visited at consistent times in those time periods, if the travel times to v_s along paths P and Q are sufficiently different.

The basic form of the inequality that forbids the simultaneous occurrence of P and Q in a solution is presented in Eq. (8).

$$\sum_{(i,j) \in P} x_{ijp} + \sum_{(i,j) \in Q} x_{ijq} \leq |P| + |Q| - 1 \quad (8)$$

Every feasible ConTSP solution must satisfy constraints (8) for every inconsistent pair of paths $\{P, Q\}$. Moreover, every solution to the set of constraints (2)–(4) and (7) (i.e., a set of tours) that violates the arrival-time consistency requirement for at least one customer must satisfy condition (i) of Lemma 1 for at least one pair of paths. Therefore, every solution that satisfies the set of constraints (2)–(4), (7) and (8), where the latter is imposed for all possible pairs of paths that meet the conditions of Lemma 1, constitutes a feasible ConTSP solution. As a result, the model consisting of Eqs. (1)–(4), (7) and (8) is a complete and valid ConTSP formulation. Note that this formulation is similar to the TSP with Time Windows (TSPTW) formulation of Ascheuer et al. (2001), which also consisted of only binary arc variables.

We remark that the number of distinct SEC is $\mathcal{O}(2^n h)$. Similarly, the number of distinct IPEC in the worst case, when every possible pair of paths is inconsistent, is $\mathcal{O}(n!^2 h^2)$. Therefore, since the number of these constraints grows very fast (exponentially and factorially, respectively) with the size of the instance, we treat these inequalities as cutting planes and add them dynamically in a branch-and-cut solution framework. In practice, the number of such inequalities added is relatively small. Our separation procedures and associated separation protocols are discussed in detail in Section 5. Note that, although it may be more challenging in the case of a fractional solution, one can immediately and exactly (i.e., with guarantees to identify a violation, if one exists) separate these inequalities in the context of an integral solution (see Section 5 for details). To that end, the branch-and-cut framework is guaranteed to locate the optimal solution, as long as it is afforded enough computational resources. We discuss our computational experience in Section 6.

3.2. Formulation 2

This formulation explicitly encodes the arrival times at customers. To that purpose, we introduce continuous variables $\alpha_{ip} \geq 0$ to capture the arrival time at each customer $i \in V_p$ in each period $p \in \mathcal{P}$. These variables attain appropriate values via their participation in Miller–Tucker–Zemlin (MTZ) expressions (Miller, Tucker, & Zemlin, 1960), which are cast here in terms of travel time.⁴ For each $p \in \mathcal{P}$, let us define parameters $f_{ip} := \min_{k \in N_p^+(i)} t_{ik}$, $r_{ip} := \min_{k \in N_p^-(i)} t_{ki}$, for all $i \in V_p$ and $f_{0p} := 0$, $r_{0p} := 0$; also define $\xi_{ip} := \max_{j \in N_p^+(i)} \{t_{ij} + f_{jp}\}$, for all $i \in V_p$. Constraints (9) apply.

$$\alpha_{ip} - \alpha_{jp} + t_{ij}x_{ijp} - t_{ji}x_{jip} \leq (M_p - f_{ip} - r_{jp})(1 - x_{ijp} - x_{jip}) \quad (9a)$$

$$\forall (i, j) \in A(V_p), \forall p \in \mathcal{P}$$

$$\alpha_{ip} \geq \sum_{j \in N_p^-(i)} (r_{jp} + t_{ji})x_{jip} \quad \forall i \in V_p, \forall p \in \mathcal{P} \quad (9b)$$

$$\alpha_{ip} + \sum_{j \in N_p^+(i)} (t_{ij} + f_{jp})x_{ijp} \leq M_p(1 - x_{0ip}) + (t_{0i} + \xi_{ip})x_{0ip} \quad (9c)$$

$$\forall i \in V_p, \forall p \in \mathcal{P}$$

$$\alpha_{ip} \leq (M_p - f_{ip})(1 - x_{0ip}) + t_{0i}x_{0ip} \quad \forall i \in V_p, \forall p \in \mathcal{P} \quad (9d)$$

The parameters M_p are big-M coefficients that must be large enough so as not to exclude the optimal solution. If z_{UB} is the objective value of a known feasible solution, obtained via a heuristic

⁴ In the typical setting, MTZ expressions are cast in terms of order of customer visits or cumulative demand served (see, e.g., Desrochers & Laporte, 1991).

or otherwise, and if ζ_p^* is the optimal objective of the TSP in period p , then the value $M_p = z_{UB} - \sum_{q \in \mathcal{P}: q \neq p} \zeta_q^*$ suffices.⁵ This also implies that the value of the parameter M_p can be dynamically tightened during the search as new incumbent solutions are identified.

We remark that the MTZ expressions as presented in Eq. (9) incorporate several applicable liftings, which we have proposed here for the first time. We further remark that, if the triangle inequality on the travel time vector t is satisfied, namely if

$$t_{ij} + t_{jk} \geq t_{ik} \quad \forall i, j, k \in V \quad (10)$$

then one may further lift the formulation by updating the definitions of its parameters as follows: $f_{ip} := t_{i0}$, $r_{ip} := t_{0i}$, and $\xi_{ip} := \max_{j \in N_p^+(i)} \{t_{ij} + t_{j0}\}$.

Constraints (9) suffice to exclude subtours.⁶ Furthermore, the introduction of variables α_{ip} enables the explicit enforcement of arrival-time consistency via constraints (11), achieving the arrival-time consistency requirement (6).

$$\alpha_{ip} - \alpha_{iq} \leq L \quad \forall i \in V_p \cap V_q, \forall (p, q) \in \mathcal{P} \times \mathcal{P} : p \neq q \quad (11)$$

Therefore, the model consisting of Eqs. (1)–(4), (9) and (11) is a complete and valid ConTSP formulation.

3.3. Formulation 3

This is a single-commodity flow formulation, where the cumulative travel time is represented as a commodity, originating at the depot and flowing through the arcs. The single-commodity flow formulation for the TSP was originally proposed by Gavish and Graves (1978), where one unit of commodity was picked up along each traveled arc. A similar model was proposed by Maffioli and Sciomachen (1997) for the sequential ordering problem with time windows; the version presented here can be obtained from the latter by ignoring the precedence constraints. After introducing a set of continuous variables $y_{ijp} \geq 0$ to capture the commodity flow on each arc $(i, j) \in A_p$ in each period $p \in \mathcal{P}$, the following constraints apply.

$$\sum_{k \in N_p^+(j)} y_{jkp} = \sum_{i \in N_p^-(j)} (y_{ijp} + t_{ij}x_{ijp}) \quad \forall j \in V_p, \forall p \in \mathcal{P} \quad (12a)$$

$$\sum_{j \in N_p^+(0)} y_{0jp} = 0 \quad \forall p \in \mathcal{P} \quad (12b)$$

$$0 \leq y_{ijp} \leq (M_p - t_{ij} - f_{jp})x_{ijp} \quad \forall (i, j) \in A_p, \forall p \in \mathcal{P} \quad (12c)$$

The parameters M_p and f_{jp} are as described in Section 3.2. The implications of the triangle inequality (10) on the definitions of f_{jp} also carry over from the discussion there.

Constraints (12a) represent commodity-flow balances and enforce that the arrival time at node j equals the arrival time at its predecessor node plus the time it takes to travel from that predecessor to node j . Constraints (12b) simply require the vehicle to start from the depot at time 0 in each period. These constraints lead the commodity variables y_{ijp} to attain the value of the arrival time at node i , whenever $(i, j) \in A_p$ is part of the tour in period p , i.e., whenever $x_{ijp} = 1$. At the same time, constraints (12c) will ensure that $y_{ijp} = 0$, whenever $x_{ijp} = 0$.

The commodity-flow constraints (12) suffice to eliminate subtours.⁷ Furthermore, the presence of commodity variables y_{ijp} enables us to explicitly enforce arrival-time consistency via constraints

⁵ If no feasible solution to the problem is known, one may replace z_{UB} with the sum of the maximal traveling salesman tours across all periods.

⁶ On a rather technical remark, note that MTZ constraints suffice to exclude a potential subtour only when the total travel time along this subtour is strictly positive (which is the typical case). For data sets that involve arcs of zero travel time, which may give rise to subtours of zero total travel time, one should pay attention to consider the relevant SEC explicitly (e.g., by adding them as a lazy cut).

⁷ The remark of the previous footnote applies also for the case of commodity-flow constraints.

Table 1
Sizes of ConTSP formulations.

Formulation	# of Variables		# of Constraints
	Binary	Continuous	
1	$\mathcal{O}(n^2h)$	–	$\mathcal{O}(n^2h^2)$
2	$\mathcal{O}(n^2h)$	$\mathcal{O}(nh)$	$\mathcal{O}(n^2h + nh^2)$
3	$\mathcal{O}(n^2h)$	$\mathcal{O}(n^2h)$	$\mathcal{O}(n^2h + nh^2)$

(13), achieving the arrival-time consistency requirement (6).

$$\sum_{j \in N_p^+(i)} y_{ijp} - \sum_{j \in N_q^+(i)} y_{ijq} \leq L \quad \forall i \in V_p \cap V_q, \forall (p, q) \in \mathcal{P} \times \mathcal{P} : p \neq q \quad (13)$$

Therefore, the model consisting of Eqs. (1)–(4), (12) and (13) is a complete and valid ConTSP formulation.

Finally, we remark that an alternative formulation results by utilizing commodity-flow constraints merely to eliminate subtours while using the IPEC (8), instead of constraints (13), to enforce arrival-time consistency. In such a case, one could apply projection techniques as in (Gouveia, 1995) to obtain a formulation in the space of the binary arc variables x_{ijp} only. Conversely, we also note that it is possible to model the ConTSP without introducing binary arc variables at all. This technique, which was used by Langevin, Desrochers, Desrosiers, Gélinas, and Soumis (1993) to model the TSPTW, utilizes a two-commodity flow representation of the cumulative travel time that is subsequently exploited in a branch-and-bound solution framework. The lower bound given by the linear programming relaxation of the two-commodity flow formulation would be identical to that of the single-commodity flow formulation (Desrochers & Laporte, 1991).

3.4. Sizes and strength of proposed formulations

Formulation 1 utilizes only the binary variables x_{ijp} , but features a factorially-large set of constraints. In contrast, Formulations 2 and 3 feature a polynomial number of constraints, but utilize additional variables. Table 1 provides a synopsis of the sizes of the three formulations.

Formulation 1 can be viewed as an example of a ConTSP formulation where an asymmetric TSP formulation (the one introduced by Dantzig et al. (1954), in this case) is utilized to achieve requirement (5), while the arrival-time consistency requirement (6) is enforced via Eq. (8). In principle, any valid asymmetric TSP formulation can be used to model (5), and we refer the interested reader to Öncan, Altinel, and Laporte (2009), Godinho, Gouveia, and Pesneau (2011) and Roberti and Toth (2012) for recent surveys of such formulations. As long as binary arc variables are encoded explicitly, one can in conjunction use Eq. (8) to enforce requirement (6) for a comprehensive ConTSP formulation. Consequently, known tightness results and relationships between the utilized asymmetric TSP formulations will persist in their ConTSP counterparts. On that note, Roberti and Toth (2012) conducted a comparison of exact algorithms for the asymmetric TSP and found the branch-and-cut algorithm of Fischetti et al. (2007), which is based on the TSP formulation by Dantzig et al. (1954), to be the most efficient computationally; hence, our Formulation 1 is expected to also be the most efficient among the formulations in its class. Formulations 2 and 3, however, follow a different modeling paradigm. These formulations encode travel times explicitly, and they use this information to enforce requirement (6) without having to introduce the IPEC (8). While this precludes a straightforward analysis of tightness relationships among our three ConTSP formulations, we present in Section 6.1 empirical evidence indicating that the dual bound obtained using the LP relaxation of Formulation 1 is always stronger than the bounds obtained using the LP relaxations of

Formulations 2 and 3, while no empirical dominance relationship can be inferred between the latter two formulations.

4. Valid inequalities

Let us define the ConTSP polytope, $P_{\text{CON TSP}}$, to be the convex hull of all integer feasible solutions of Formulation 1,

$$P_{\text{CON TSP}} = \text{conv}\{x \in \mathbb{R}^{\sum_{p \in \mathcal{P}} |A_p|} : x \text{ satisfies (2), (3), (4), (7), (8)}\}.$$

Several families of inequalities are valid for $P_{\text{CON TSP}}$. These include all inequalities that are valid for the asymmetric TSP. Note that an inequality that is valid for any of the asymmetric TSP instance associated with a period $p \in \mathcal{P}$ can be directly applied on the ConTSP instance. In our study, we consider Subtour Elimination and 2-matching Constraints. In addition, we consider the cross-period Inconsistent Path Elimination Constraints introduced earlier in Section 3.1. As discussed, the inclusion of SEC and IPEC is necessary for Formulation 1, which relies on these inequalities to eliminate subtours and enforce arrival-time consistency, respectively. The inequalities are redundant for Formulations 2 and 3, as long as the integrality restrictions (2) on the variables x_{ijp} are retained; however, they are still capable of strengthening the linear relaxation and should, thus, be used in conjunction with these formulations as well. In fact, the use of these inequalities as cutting planes is of fundamental importance from a practical point of view. For the instances we considered in our computational study (see Section 6), adding the cutting planes was very helpful in expediting the proof of optimality.

4.1. Subtour elimination constraints

These inequalities, which were introduced as constraints (7), forbid the occurrence of subtours and enforce the overall connectivity of the tour. Note however that, because of the degree constraints (3) and (4), the SEC defined by a vertex set $S \subseteq V_p \cup \{0\}$ and its complement $S^c = (V_p \cup \{0\}) \setminus S$ (see Eq. 14 below) are equivalent. To that end, one may use form (14) as an alternate to form (7). Numerical criteria, such as constraint sparsity (which depends on the size of set S), can be used to decide which of the two forms to utilize as a cutting plane in each case.

$$\sum_{(i,j) \in A(S^c)} x_{ijp} \leq |S^c| - 1 \quad \forall S \subseteq V_p \cup \{0\} : 2 \leq |S| \leq |V_p| - 1, \quad \forall p \in \mathcal{P} \tag{14}$$

4.2. 2-matching constraints

The 2-matching Constraints (also known as Blossom Inequalities) are particular cases of a more general class of inequalities that is referred to as Comb Inequalities (Fischetti, 1991). For a given period $p \in \mathcal{P}$ and given vertex sets $H, T_1, T_2, \dots, T_k \subseteq V_p$, where $k \geq 3$ and odd, satisfying (i) $|H \cap T_i| = 1$ for $i = 1, \dots, k$, (ii) $|T_i \setminus H| = 1$ for $i = 1, \dots, k$, (iii) $T_i \cap T_j = \emptyset$ for $1 \leq i < j \leq k$, the corresponding 2-matching Constraint (2MC) is presented in Eq. (15).

$$\sum_{(i,j) \in A(H)} x_{ijp} + \sum_{i=1}^k \sum_{(i,j) \in A(T_i)} x_{ijp} \leq |H| + \frac{k-1}{2} \tag{15}$$

Such constraints are obtained by adding the degree constraints (3) and (4) for all $i \in H$, adding the subtour elimination constraints (7) defined by the vertex sets $S = T_i$ for all $i \in \{1, \dots, k\}$, dividing by 2, and rounding down the right hand side to the nearest integer. More general Comb Inequalities may be obtained by relaxing conditions (i) and (ii) above as follows: $|H \cap T_i| \geq 1$ and $|T_i \setminus H| \geq 1$ for $i = 1, \dots, k$. Constraint (15) may be equivalently cast in the form (16). As before,

constraint sparsity can be used in each case to decide whether a cutting plane should be represented in form (15) or form (16).

$$\sum_{(i,j) \in \delta(H) \setminus (\bigcup_{i=1}^k A(T_i))} x_{ijp} - \sum_{i=1}^k \sum_{(i,j) \in A(T_i)} x_{ijp} \geq 1 - k. \tag{16}$$

4.3. Inconsistent path elimination constraints

These inequalities, which were introduced in Section 3.1, are cross-period constraints that forbid pairs of paths belonging to different periods that are inconsistent to simultaneously appear in the solution. Given a pair of paths $\{P, Q\}$, where $P = (v_1, \dots, v_s)^p$ and $Q = (w_1, \dots, w_t)^q$, the basic form of the inequality that forbids their simultaneous occurrence is presented in Eq. (17).

$$\sum_{i=1}^{s-1} x_{v_i v_{i+1} p} + \sum_{i=1}^{t-1} x_{w_i w_{i+1} q} \leq s + t - 3 \tag{17}$$

However, it is possible to strengthen this basic form (see also Ascheuer et al. (2000) for how to strengthen the infeasible path elimination constraints they proposed in the context of the TSPTW). In particular, it can be strengthened into the so-called Tournament Constraint (18).

$$\sum_{i=1}^{s-1} \sum_{j=i+1}^s x_{v_i v_j p} + \sum_{i=1}^{t-1} \sum_{j=i+1}^t x_{w_i w_j q} \leq s + t - 3 \tag{18}$$

Furthermore, for a given inconsistent pair of paths $\{P, Q\}$ as described above, let the paths obtained by reversing paths P and Q be denoted as $P' = (v_s, \dots, v_1)^p$ and $Q' = (w_t, \dots, w_1)^q$, respectively. If all three pairs of paths $\{P, Q\}$, $\{P', Q\}$ and $\{P', Q'\}$ are also inconsistent, then the symmetric inequality (19), which corresponds to a lifting of inequality (17), is also valid and can be used instead of the latter.

$$\sum_{i=1}^{s-1} (x_{v_i v_{i+1} p} + x_{v_{i+1} v_i p}) + \sum_{i=1}^{t-1} (x_{w_i w_{i+1} q} + x_{w_{i+1} w_i q}) \leq s + t - 3 \tag{19}$$

Finally, if $\{P_1, Q_1\}$ and $\{P_2, Q_2\}$ are two inconsistent pairs of paths in periods p and q , such that P_1 is contained in P_2 and Q_1 is contained in Q_2 , then the IPEC defined by $\{P_2, Q_2\}$ is dominated by the one defined by $\{P_1, Q_1\}$.

4.4. Polyhedral analysis

A polyhedral analysis to determine whether the inequalities considered above are facet-defining for $P_{\text{CON TSP}}$ is typically a difficult task. For a fixed graph size, $|V_p|$, and for fixed L , small changes in the travel times t_{ij} can change the dimension of $P_{\text{CON TSP}}$ or even make the entire instance infeasible. Although there may exist specific instances in which individual inequalities coincide with facets of $P_{\text{CON TSP}}$, in general none of the three families of inequalities induces facets of the polytope, even if the instance is feasible. This observation is interesting inasmuch it implies that known polyhedral results for asymmetric TSP do not carry over to the case of ConTSP. For example, all SEC are known to induce facets of the asymmetric TSP polytope for the case of $n \geq 4$ (Grötschel & Padberg, 1985), while all 2MC are known to be facet-defining for the asymmetric TSP polytope for the case of $n \geq 6$ (Fischetti, 1991). However, as Proposition 1 shows, this is not true for the ConTSP.

Proposition 1. *The Subtour Elimination, 2-matching and Inconsistent Path Elimination Constraints do not induce facets of $P_{\text{CON TSP}}$, in general.*

Proof. Consider a ConTSP instance with $n = 7$, $h = 2$, $V_1 = \{1, 2, 3, 4, 5, 6\}$ and $V_2 = \{1, 7\}$. The arrival-time consistency requirement applies only to node 1, since node 1 is the only customer node common to both periods 1 and 2. Assume that the

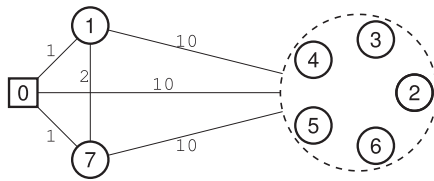


Fig. 1. Travel times for the ConTSP instance considered in the proof of Proposition 1.

travel times are symmetric and are as depicted in Fig. 1. For a maximum allowable arrival-time differential of $L = 1$, it is straightforward to see that all feasible ConTSP solutions are of the form $\{ \langle 0, 1, i_2, \dots, i_6 \rangle^1, \langle 0, 1, 7 \rangle^2 \}$, where i_2, \dots, i_6 is some permutation of nodes 2 through 6. In this case, one may verify that the rank of the set of feasible solutions is 20 and that 0 does not participate in the affine hull of this set.⁸ Therefore, the affine rank of P_{CONTSP} is 20 and $\dim(P_{\text{CONTSP}}) = 19$.

By a similar reasoning as above, it can be verified that the face induced by the SEC of the form (7), where $p = 1$ and $S = \{0, 2, 3\}$, is of dimension equal to 8 and, therefore, this inequality does not induce a facet of P_{CONTSP} . The face induced by the 2MC of the form (15), where $p = 1, k = 3, H = \{0, 1, 2\}, T_1 = \{0, 3\}, T_2 = \{1, 4\}$ and $T_3 = \{2, 5\}$, is of dimension equal to 7 and, therefore, this inequality does not induce a facet of P_{CONTSP} . In fact, it can be verified that none of the Comb Inequalities that are possible in period 1 (which include, as a special case, all 2-matching Constraints), induce facets of P_{CONTSP} . Finally, consider the IPEC of the form (8), where $P = (0, 2, 1)^1$ and $Q = (0, 7, 1)^2$. It can be verified that this inequality does not induce a face of P_{CONTSP} , i.e., there is no integral point in P_{CONTSP} that satisfies this inequality as an equality; therefore, it is not facet-defining for P_{CONTSP} . □

5. Branch-and-cut framework

We implemented three separate branch-and-cut algorithms, one based on each of the formulations presented in Sections 3.1, 3.2 and 3.3. At the interest of working with sufficiently tractable linear programming (LP) relaxations, a subset of applicable constraints are initially ignored and added later as cutting planes, if necessary. More specifically, in the case of Formulation 1, the initial LP relaxation consisted only of the degree constraints (3) and (4), along with continuous bounds on the x variables (relaxation of the integrality restrictions (2)). In the case of Formulation 2, constraints (9)–(11) were considered, along with the degree constraints and variable bounds, while in the case of Formulation 3, the initial LP relaxation consisted of constraints (12) and (13), in addition to the degree constraints and variable bounds.

The SEC, 2MC and IPEC were considered in all cases as cutting planes and dynamically added back to the applicable model, when found to be violated, at each node of the search tree. At the root node, if we are unable to generate any additional violated inequalities, we permanently fix nonbasic x variables to their current values using reduced cost information. Let UB be the current (incumbent) upper bound (obtained through a heuristic or otherwise) and let LB be the global (root-node) lower bound obtained from the applicable LP relaxation. Let \bar{c}_{ijp} be the reduced cost of each nonbasic variable x_{ijp} in the root-node LP solution. Then, if $x_{ijp} = 0$ and $LB + \bar{c}_{ijp} > UB$, we can fix this variable to zero. Conversely, if $x_{ijp} = 1$ and $LB - \bar{c}_{ijp} > UB$, then we can fix this variable to one. In a similar manner, using a local lower bound, one may set non-basic x variables to their current values in the sub-tree associated with each node of the search

process. These fixings ensure that the variables are never branched upon in the corresponding sub-trees. The remainder of this section elaborates on our separation routines and associated protocols.

5.1. Separation routines

5.1.1. Subtour elimination constraints

In our implementation, we utilize both heuristic procedures and an exact scheme to separate violated SEC. In particular, we first attempt to separate these cuts using the separation heuristics described in Applegate et al. (2007, chap. 6). The advantage of these routines is that they often lead to the identification of several violated inequalities, as opposed to the one most violated inequality, and we have found it beneficial to be adding all such identified inequalities in a single cutting-plane iteration. When the heuristic routines do not identify any violations, we employ the exact separation scheme that was introduced by Padberg and Rinaldi (1990). This is a polynomial-time routine that is based on a minimum-cut algorithm. The overall running time of our separation routine is dominated by the maximum-flow computations in the exact scheme, for which we use the Boost graph library (Siek, Lee, & Lumsdaine, 2000).

5.1.2. 2-matching constraints

In our implementation, we utilize both heuristic procedures and an exact scheme to separate violated 2MC. More specifically, we first attempt to separate these cuts using the so-called odd-component, Grötschel-Holland and block heuristic routines described in Applegate et al. (2007, chap. 7). Note that the latter heuristic also identifies violated general Comb Inequalities. As before, each of these heuristics may identify several violated inequalities, and we have found it beneficial to be adding all of them in a single cutting-plane iteration. If none of the heuristic routines succeeds in generating a violated cut, we employ the polynomial-time exact routine proposed by Letchford, Reinelt, and Theis (2004). The running time of this routine is dominated by the constructor of the Gomory-Hu cut-tree and, in our implementation, we utilize a code written by Skorobohatjy (2004) for this purpose. Given its computational burden, we only employ the exact routine during cutting-plane iterations at the root node. Alternatively, one may use the polynomial-time exact algorithm proposed by Padberg and Rao (1982) to separate violated 2MC.

5.1.3. Inconsistent path elimination constraints

Our separation routines to identify violated IPEC is guided by Lemma 1. We use the following polynomial-time enumerative procedure to identify inconsistent pairs of paths. Given a (fractional) solution x^* , consider the support graph in each time period p . This support graph has the same node set as G_p and involves those arcs $(i, j) \in A_p$ for which $x_{ijp}^* > 0$. By considering every node in each period's support graph as a start node, elementary paths are grown in a depth-first fashion by moving along the incident outgoing arcs of strictly positive flow. Each path $P = (v_1, \dots, v_s)^p$ is extended as long as the following two conditions hold: (i) the total flow on the path, $\sum_{i=1}^{s-1} x_{v_i v_{i+1} p}^*$, is strictly greater than $|P| - \frac{1}{2} = (s - 1) - \frac{1}{2}$, and (ii) $v_s \neq 0$, i.e., the path has not reached the depot. Because of the first condition, an arc $(i, j) \in A_p$ is added to the path only if $x_{ijp}^* > 0.5$, while the path-growing procedure stops as soon as all incident outgoing arcs have weight strictly less than 0.5. The degree constraint $\sum_{j \in N_p^+(i)} x_{ijp}^* = 1$ ensures that, for each node i , there is at most one incident outgoing arc with $x_{ijp}^* > 0.5$. Hence, there is at most one (unique) path extending out of each node and the path-growing procedure terminates in polynomial time.

After all such paths have been identified in all time periods, violated inequalities (17) are identified as follows: for each pair of nodes $v_1, v_s \in (V_p \cup \{0\}) \cap (V_q \cup \{0\})$, we consider all pairs of paths in periods p and q such that each path contains both v_1 and v_s . If any of the conditions (i)–(iii) in Lemma 1 are satisfied for the pair of sub-paths

⁸ This may be verified numerically by showing the following linear system (in variables $\lambda_i \in \mathbb{R}$) to be infeasible: $\sum_{i \in F} x_i \lambda_i = 0, \sum_{i \in F} \lambda_i = 1$. Here, $\{x_i\}_{i \in F}$ is the set of feasible ConTSP solutions.

corresponding to v_1 and v_s as terminal nodes, then we have identified a violated inequality.

For every pair of paths that satisfy either condition (ii) or (iii) in Lemma 1, we check whether the corresponding IPEC of the form (17) can be lifted to the symmetric inequality (19) (see Section 4.3). If such a lifting is not possible, we add the corresponding Tournament Constraint (Eq. 18), because any pair of paths which violates inequality (17) also violates (18), and we have found it computationally beneficial to enforce the IPEC via the latter, stronger form. We remark that, if the solution x^* is integral and does not contain any subtours, then condition (i) of Lemma 1 is sufficient to guarantee the exactness of the separation procedure.

Our computational experiments indicated that the number of violated IPEC identified during a given cutting-plane iteration at a node of the branch-and-cut tree may be very large, especially for Formulation 1. For this reason, we attempt to reduce the number of constraints added to the constraint matrix in the following two ways: (i) we remove any dominated inequalities (see Section 4.3), and (ii) whenever the number of violated inequalities exceeds $2|V|$, we calculate for each cut (considered here as being of form 18) the ratio between its absolute violation at the current solution and the number of non-zero coefficients in the cut, namely

$$\frac{\sum_{i=1}^{s-1} \sum_{j=i+1}^s x_{v_i v_j p}^* + \sum_{i=1}^{t-1} \sum_{j=i+1}^t x_{w_i w_j q}^* - (s+t-3)}{\frac{1}{2}s(s-1) + \frac{1}{2}t(t-1)},$$

and we only add to the constraint matrix the $2|V|$ cuts with the highest such ratio.

5.2. Separation protocol

We attempt to separate violated inequalities at each node of the branch-and-cut tree in the following sequence: (i) SEC for each time period $p \in \mathcal{P}$, (ii) 2MC for each time period $p \in \mathcal{P}$, and (iii) IPEC for each pair of time periods $p, q \in \mathcal{P} : p < q$. When we identify violated members of any of the above families of cuts in a given time period, then no attempt is made to separate members of other families of cuts that appear further in the separation sequence and that involve variables from that time period. At any given node other than the root node, whenever new cuts are added but the objective function value did not improve by at least 0.1 percent in the last 20 cutting-plane iterations, we exit the separation sequence and resort to branching. At the root node, we have found it beneficial to keep separating as many cuts as possible before branching, even if the lower bound does not show significant improvement at that moment. Finally, we remark that, since all inequalities considered are valid globally (i.e., throughout the branch-and-cut tree), any violated inequality identified is added as a global cut.

6. Computational results

Since no benchmark data sets are currently available for the ConTSP in the open literature, we compiled a set of benchmark problems by extending symmetric and asymmetric TSP instances from the well-known TSPLIB library (Reinelt, 1991) into 3- and 5-period ConTSP instances. In particular, we considered all 23 instances that involve up to 50 customers (i.e., 51 nodes). For each TSPLIB instance, we constructed 3- and 5-period ConTSP instances in a manner similar to that proposed by Groër et al. (2009) for the ConVRP. More specifically, we considered each of the n customers to have a probability f of requiring service in each period, where $f \in \{0.5, 0.7, 0.9\}$. Note that, in general, this results in instances in which a different number of customers require service in each time period; however, the average number of customers (across periods) is approximately

fn . The distances acquired from the TSPLIB data⁹ were interpreted as being both travel costs (c_{ij}) and travel times (t_{ij}), while we assumed all service times (s_i) to be zero. The first node in each of these data sets was arbitrarily regarded to be the depot. For every resulting multi-period instance, we selected the maximum allowable arrival-time differential, L , to be 10, 15 or 20 percent of the maximum (across periods) travel time of the corresponding optimal (single-period) traveling salesman tours,¹⁰ which correspond to practically reasonable selections for a required level of consistency.¹¹ The overall process resulted in nine 3-period and nine 5-period instances for each of the original TSPLIB problems, for a total of 414 ConTSP benchmark instances. These instances are available for download at <http://gounaris.cheme.cmu.edu/datasets/contsp/>.

In order to fairly compare among the different formulations we consider in this study, we explicitly supplied in each case the best available solution as an initial incumbent and we disabled any heuristic effort by the solver to obtain a better incumbent. By doing so, we eliminate the effect of the solver's (heuristic) primal bounding effort, and we can better appreciate the performance of our (exact) branch-and-cut framework in terms of its ability to generate dual bounds and prove optimality. All implementations were developed in the C++ programming language using the C API of ILOG CPLEX 12.6 and compiled using the GCC 4.8.2 compiler with optimization level -O2. We used the default CPLEX strong branching and best-bound node selection strategies. Moreover, unless otherwise mentioned, all CPLEX-generated cuts were disabled because we observed that using these general-purpose cuts increased overall computation times. All computations were conducted on a single-core (non hyper-threaded) of an Intel Xeon 2.8 GHz processor with a software-imposed memory limit of 3 GB RAM. Computational times reported correspond to actual "wall-clock" times.

6.1. Tightness of alternative formulations and effect of valid inequalities

In order to gain some insight about the tightness of the three formulations proposed in Section 3, we compare in Table 2 the corresponding root-node gaps.¹² The reported gap values constitute an average across all nine 3-period and nine 5-period ConTSP instances constructed from the TSPLIB data indicated in the first column. Furthermore, we report in each case, four characteristic root-node gap quantities, as follows: (C0) the gap obtained using the initial LP relaxation (as described in the preamble of Section 5), before any cut separation; (C1) the gap obtained after separation of SEC; (C2) the gap obtained after separation of SEC and 2MC; (C3) the gap obtained after separation of SEC, 2MC and IPEC. This analysis helps us appreciate the effect that each of the three families of cuts we consider in this

⁹ As per the TSPLIB standard, all distances were rounded down to the nearest integer. Note, however, that this assumption is not restrictive and does not invalidate any of the theoretical results presented in this paper.

¹⁰ We computed the optimal tours of these TSP instances with an in-house exact TSP solver implementation. We found this to be a rather trivial task, as each single period of a ConTSP instance involved no more than 50 customers.

¹¹ Assuming the optimal traveling salesman tours are to be traversed along an 8-hour shift of driving, a value of $L = 10\%$ corresponds to a maximum arrival time-difference of 48 minutes (i.e., a guaranteed arrival time within 24 minutes of some nominal time), while $L = 20\%$ corresponds to double that amount. To further put this into perspective, it should be remarked that, for the benchmark problems we considered, the maximum arrival-time difference (across all customers) of the multi-period solution that consists of the optimal traveling salesman tours in each period (whereby no consistency is enforced) is approximately 68% of the maximum vehicle travel-time across periods, i.e., a 5 hours and 26 minutes window within the 8-hour shift, on average. Therefore, arrival-time consistency at reasonable levels would not be enforced using a purely cost-based routing approach, further motivating the need for considering the ConTSP.

¹² Optimality gaps in this study are defined as $\frac{UB-LB}{LB} \times 100\%$ percent, where UB is the global upper bound and LB is the global lower bound of the branch-and-cut tree. Root-node gaps correspond to optimality gaps after processing the root of the search tree.

Table 2
Root-node gaps (%) obtained using each of the proposed formulations after various levels of separation of valid inequalities.

Instance	Formulation 1				Formulation 2				Formulation 3			
	C0	C1	C2	C3	C0	C1	C2	C3	C0	C1	C2	C3
burma14	19.96	0.84	0.84	0.47	12.05	0.84	0.84	0.84	9.21	0.84	0.84	0.84
ulysses16	17.24	0.91	0.91	0.42	11.91	0.91	0.91	0.91	8.45	0.91	0.91	0.86
br17	59.85	0.00	0.00	0.00	32.83	0.00	0.00	0.00	49.20	0.00	0.00	0.00
gr17	22.13	0.73	0.73	0.60	10.98	0.73	0.73	0.73	15.50	0.73	0.73	0.73
gr21	13.24	0.31	0.28	0.21	1.47	0.31	0.28	0.28	7.41	0.31	0.28	0.28
ulysses22	22.41	0.12	0.12	0.04	13.45	0.11	0.11	0.11	12.93	0.12	0.12	0.12
gr24	16.62	0.96	0.91	0.84	5.24	0.96	0.91	0.90	12.37	0.96	0.91	0.90
fri26	20.29	1.58	1.39	1.32	7.71	1.58	1.39	1.39	16.67	1.58	1.39	1.38
bayg29	11.23	1.15	1.12	0.89	5.21	1.15	1.12	1.12	8.06	1.15	1.12	1.12
bays29	14.42	0.56	0.49	0.40	4.64	0.56	0.49	0.49	11.71	0.56	0.49	0.48
ftv33	13.38	4.02	3.81	3.66	8.37	4.00	3.79	3.70	12.10	3.78	3.60	3.60
ftv35	10.94	3.81	3.56	3.50	6.91	3.79	3.55	3.52	9.32	3.74	3.56	3.54
ftv38	8.80	3.84	3.74	3.66	6.10	3.83	3.73	3.66	7.83	3.74	3.63	3.59
dantzig42	24.70	1.35	1.22	1.06	12.11	1.35	1.22	1.22	16.43	1.35	1.22	1.22
swiss42	21.57	1.58	1.33	1.13	6.98	1.58	1.33	1.33	18.80	1.58	1.33	1.32
p43	76.90	0.11	0.11	0.10	52.87	0.11	0.10	0.10	72.70	0.10	0.10	0.10
ftv44	10.69	4.24	3.79	3.79	6.79	4.24	3.80	3.80	9.78	4.23	3.78	3.78
ftv47	8.87	3.19	3.14	3.14	5.03	3.19	3.14	3.14	7.99	3.15	3.11	3.11
att48	22.20	1.42	1.37	1.25	10.79	1.42	1.37	1.37	18.19	1.42	1.37	1.37
gr48	16.58	1.29	0.94	0.88	6.12	1.29	0.95	0.95	13.56	1.29	0.94	0.94
hk48	17.59	1.51	1.39	1.33	5.75	1.51	1.39	1.39	14.37	1.51	1.39	1.39
ry48p	14.86	1.76	1.63	1.63	8.78	1.76	1.63	1.63	12.55	1.67	1.55	1.55
eil51	15.36	2.50	1.66	1.59	4.78	2.50	1.66	1.66	13.55	2.50	1.65	1.65
Avg. (sym.)	18.37	1.12	0.98	0.83	7.95	1.12	0.98	0.98	13.15	1.12	0.98	0.97
Avg. (asym.)	25.54	2.62	2.47	2.43	15.96	2.61	2.47	2.45	22.69	2.55	2.42	2.41
Avg. (all)	20.86	1.64	1.50	1.39	10.73	1.64	1.50	1.49	16.47	1.62	1.48	1.47

study has on each formulation's tightness. Note that, in order to ensure a fair comparison, we had disabled all CPLEX options pertaining to performing any possible MIP-based bound strengthening.

We observe that the initial LP bounds (gaps C0) are generally weak, in the order of 10 to 20 percent, but these bounds get significantly reduced, down to roughly 1.6 percent, when violated SEC are added as cutting planes (gaps C1). The additional separation of 2MC further reduces the gaps by 0.14 percent, on average. The gaps C2 are indicative of the strength of the lower bounds possible after separating over all the structural, TSP-related cuts. Finally, the separation of IPEC further improves the bounds, on average, by 0.11 percent for Formulation 1 and by 0.01 percent for Formulations 2 and 3 (gaps C3). Note that the gaps C3 constitute the effective "root-node gaps" of our branch-and-cut framework.

Comparing across formulations, Formulation 1 has the weakest initial LP bound, while the strongest initial LP bound is featured by Formulation 2. However, Formulation 1 appears to be more amenable for bound improvement after separation of valid inequalities. More specifically, after the separation of all three families of cuts considered in this study (gaps C3), the lower bounds of Formulation 1 become stronger and are higher by 0.10 and 0.08 percent, on average, compared to the respective bounds of Formulations 2 and 3. In 81 out of the 414 instances considered, the final root-node gaps (C3) were equal in all three formulations. In the remaining 333 instances, the root node bound of Formulation 1 was strongest in 215 instances, while those of Formulations 2 and 3 were strongest in 9 and 93 instances, respectively. Table 2 also shows the averages across the symmetric and asymmetric instances separately. It is interesting to note that Formulation 3 provided the strongest bound in most of the asymmetric instances, while Formulation 1 gave the strongest bound in most of the symmetric instances. This can be attributed to the fact that, while Formulation 3 is inherently asymmetric, Formulation 1 takes into account any asymmetry only through the IPEC. It remains to be investigated whether the addition of any cuts specific to the asymmetric TSP polytope (see, e.g., Fischetti & Toth, 1997) can significantly improve our lower bounds.

Table 3 presents the time spent and the number of cuts added at the root node during the runs that resulted in gaps C3; that is, after separation of all three families of cuts (SEC, 2MC and IPEC). Each reported value is averaged across 18 instances (similarly to Table 2). The average time spent at the root node is lowest in Formulation 1 and highest in Formulation 3. Thus, Formulation 1 is able to provide a stronger bound in a smaller amount of time, on average. We further observe that, at the root node, the number of IPEC added in the cases of Formulations 2 and 3 is negligible. This is probably because the explicit arrival-time consistency constraints (11) and (13) in these formulations prevent the frequent occurrence of (fractional) inconsistent paths for us to separate.

6.2. Performance of branch-and-cut framework

We attempted to solve all of our 414 benchmark problems using each of the branch-and-cut algorithms under a time limit of 2 hours. The goal of this study was to determine if there is a clear "winner" amongst the formulations. In Table 4, we report for each formulation the number of instances (out of 18) for which optimality was proved as well as the total number of branch-and-cut nodes explored and time required (averaged across all instances for which optimality was proved). For those instances which could not be solved in 2 hours, the residual gap (i.e., optimality gap at the time limit) is reported as an average. The three formulations are also compared in Fig. 2, in which we plot performance profiles (Dolan & Moré, 2002) across the totality of our data set.

Within the imposed time limit of 2 hours, Formulations 1, 2 and 3 were able to prove the optimality of 308, 291 and 297 instances respectively. Out of the 414 instances, the optimality of 96 instances could not be proved by any of the 3 formulations within the time limit. In the remaining 318 instances, the branch-and-cut framework based on Formulation 1 was fastest in 90 percent (285) of the instances, the one based on Formulation 2 in 2 percent (7), while the one based on Formulation 3 was fastest in 8 percent (26) of the instances. For the 96 instances in which none of the algorithms could

Table 3
Time spent and number of cuts added at the root node.

Instance	Formulation 1				Formulation 2				Formulation 3			
	t (sec)	SEC	2MC	IPEC	t (sec)	SEC	2MC	IPEC	t (sec)	SEC	2MC	IPEC
burma14	0.01	32.6	0.7	39.9	0.03	14.0	0.0	0.8	0.06	29.9	0.0	0.8
ulysses16	0.01	45.1	2.6	46.3	0.04	21.0	0.3	0.3	0.10	39.3	0.3	0.5
br17	0.00	40.6	0.2	1.3	0.02	30.2	0.1	0.1	0.04	47.4	0.0	0.0
gr17	0.01	51.0	0.0	38.3	0.05	23.6	0.0	0.4	0.15	49.0	0.0	1.3
gr21	0.01	37.7	0.3	48.3	0.06	6.6	0.3	0.3	0.29	37.5	0.3	0.2
ulysses22	0.02	67.2	1.3	42.6	0.09	35.2	0.0	0.1	0.37	63.7	0.0	0.1
gr24	0.03	47.9	3.6	52.4	0.14	15.4	2.2	1.5	0.51	45.9	2.2	2.8
fri26	0.03	61.9	4.4	43.4	0.13	31.4	4.0	0.3	0.67	69.7	4.4	0.6
bayg29	0.05	72.1	4.5	72.4	0.19	25.7	2.3	0.2	0.95	65.6	2.3	0.3
bays29	0.04	68.0	8.3	57.9	0.30	27.9	4.7	0.2	1.21	66.3	4.5	1.1
ftv33	0.05	67.9	27.7	22.3	0.27	50.0	22.8	3.6	1.63	76.8	33.6	0.0
ftv35	0.04	69.4	22.5	14.1	0.20	51.0	22.8	2.1	1.77	76.4	22.8	0.2
ftv38	0.04	62.7	22.7	1.2	0.26	51.4	23.7	0.3	2.94	76.0	39.9	0.6
dantzig42	0.10	113.4	5.2	56.1	0.72	49.2	3.9	0.0	5.61	105.8	3.8	0.0
swiss42	0.17	101.7	17.4	116.6	1.33	43.8	10.8	0.1	6.46	113.8	11.8	0.2
p43	0.51	134.7	46.3	41.7	2.08	89.7	49.4	0.3	80.49	189.9	63.6	0.1
ftv44	0.06	83.7	60.3	0.0	0.43	59.7	63.7	0.0	8.90	87.1	63.0	0.0
att48	0.17	164.1	7.3	73.2	1.45	79.4	4.3	0.0	11.47	161.1	4.6	0.0
ftv47	0.09	70.9	33.4	7.7	0.54	55.1	35.7	3.2	12.71	78.9	35.7	0.0
gr48	0.16	125.9	27.1	54.8	2.52	53.3	24.1	0.1	23.19	137.4	24.3	0.2
hk48	0.11	113.9	17.6	44.7	1.49	49.7	9.4	0.2	13.77	137.6	13.7	0.4
ry48p	0.08	161.9	29.8	0.3	0.60	110.2	28.6	0.1	11.83	174.7	32.9	0.1
eil51	0.40	131.4	72.9	168.5	3.21	53.1	46.9	0.0	29.07	137.9	51.6	0.1
Avg. (sym.)	0.09	82.2	11.5	63.7	0.78	35.3	7.6	0.3	6.26	84.0	8.3	0.6
Avg. (asym.)	0.11	86.5	30.4	11.1	0.55	62.2	30.9	1.2	15.04	100.9	36.4	0.1
Avg. (all)	0.10	83.7	18.1	45.4	0.70	44.6	15.7	0.6	9.31	89.9	18.1	0.4

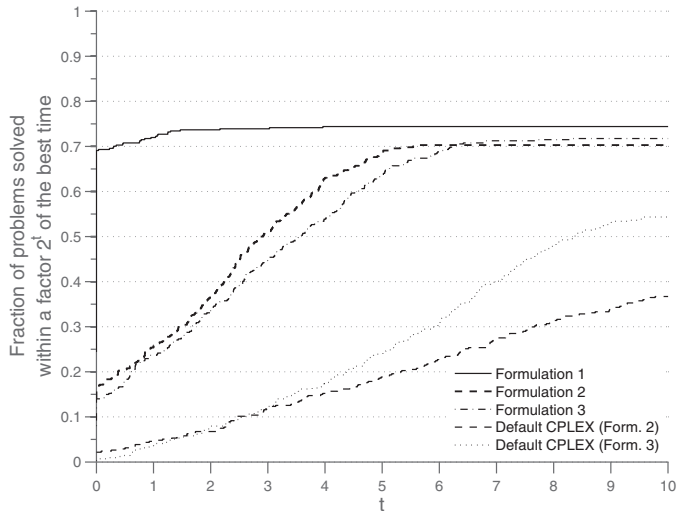
Table 4
Computational comparison of branch-and-cut algorithms.

Instance	Formulation 1					Formulation 2					Formulation 3				
	Proven optimal			Residual gap		Proven optimal			Residual gap		Proven optimal			Residual gap	
	#	Nodes	t (sec)	#	(%)	#	Nodes	t (sec)	#	(%)	#	Nodes	t (seconds)	#	(%)
burma14	18	30.2	0.1	0	–	18	15.2	0.4	0	–	18	19.2	1.3	0	–
ulysses16	18	23.5	0.1	0	–	18	15.7	0.5	0	–	18	19.9	2.8	0	–
br17	18	0.0	0.0	0	–	18	0.0	0.1	0	–	18	0.0	0.1	0	–
gr17	18	171.3	1.8	0	–	18	91.8	2.9	0	–	18	32.6	7.0	0	–
gr21	18	12.6	0.1	0	–	18	11.3	0.3	0	–	18	7.3	0.7	0	–
ulysses22	18	2.7	0.1	0	–	18	2.7	0.5	0	–	18	3.3	2.5	0	–
gr24	18	1,025.7	88.7	0	–	18	714.9	183.3	0	–	18	646.2	398.6	0	–
fri26	18	6,656.2	513.4	0	–	16	4,985.2	860.8	2	0.20	17	1,742.4	518.1	1	0.20
bayg29	17	3,489.2	416.9	1	1.77	17	1,243.9	190.3	1	1.25	17	239.9	95.2	1	1.25
bays29	18	240.9	5.1	0	–	18	170.4	36.0	0	–	18	105.8	47.5	0	–
ftv33	11	186.7	3.3	7	1.69	11	263.5	66.8	7	2.41	11	111.3	220.6	7	2.41
ftv35	10	10,955.2	1,267.8	8	1.05	9	5,341.9	1,374.0	9	1.34	8	3,058.3	1,986.7	10	1.34
ftv38	8	3,623.3	323.9	10	1.61	7	3,073.4	477.1	11	2.00	7	1,401.9	547.1	11	2.00
dantzig42	16	1,497.2	78.9	2	0.41	16	2,051.7	618.7	2	0.49	15	432.5	720.1	3	0.49
swiss42	16	3,584.1	518.8	2	0.49	16	2,316.6	928.3	2	0.68	15	1,753.3	1,285.6	3	0.68
p43	0	–	–	18	0.05	1	12,286.0	3,242.7	17	0.04	3	1,415.3	2,311.1	15	0.04
ftv44	6	21,588.7	2,496.7	12	1.81	4	14,962.5	2,607.4	14	2.01	2	8,205.5	4,755.0	16	2.01
att48	13	8,073.8	1,091.4	5	0.74	10	3,498.7	1,718.0	8	0.80	10	1,755.1	1,931.9	8	0.80
ftv47	6	6,104.0	891.8	12	1.25	5	913.8	243.8	13	1.72	5	370.0	346.6	13	1.72
gr48	13	3,406.9	353.5	5	0.39	12	1,949.4	1,221.2	6	0.74	13	1,026.4	1,426.4	5	0.74
hk48	11	6,580.9	743.3	7	0.90	8	3,631.5	853.1	10	0.52	11	1,173.5	1,299.8	7	0.52
ry48p	13	12,969.0	866.3	5	0.38	10	5,490.6	1,709.1	8	0.78	12	1,133.1	1,532.2	6	0.78
eil51	6	8,924.5	1,324.7	12	1.08	5	3,505.6	1,935.7	13	1.24	7	1,329.9	1,702.3	11	1.24
Low L	98	3,533.2	378.2	40	1.17	92	1,860.4	478.6	46	1.20	90	548.6	546.0	48	1.38
Med. L	104	2,958.3	267.0	34	0.94	99	1,537.6	456.8	39	1.01	103	750.4	625.6	35	1.27
High L	106	3,624.2	420.7	32	0.80	100	1,748.8	533.0	38	0.83	104	764.9	557.8	34	0.99
3-period	179	2,949.4	332.1	28	0.79	172	1,638.6	427.5	35	0.83	177	767.2	586.8	30	1.07
5-period	129	3,954.6	387.5	78	1.05	119	1,818.7	580.0	88	1.10	120	586.9	564.4	87	1.29
Sym.	236	2,384.5	264.9	34	0.84	226	1,301.6	416.5	44	0.80	231	582.7	488.6	39	0.83
Asym.	72	6,602.0	651.4	72	1.05	65	3,140.0	745.0	79	1.15	66	1,085.0	889.6	78	1.44
All	308	3,370.4	355.3	106	0.98	291	1,712.2	489.9	123	1.03	297	694.3	577.7	117	1.24

Table 5

Time spent in separation routines and number of cuts added (averaged across all 414 ConTSP instances).

Formulation	Sep. time (%)	Cuts	SEC (%)	2MC (%)	IPEC (%)
1	8.42	3,689	27.6	7.8	64.6
2	1.43	1,378	41.0	15.8	43.2
3	0.73	563	61.2	17.6	21.3

**Fig. 2.** Log-scaled performance profiles across all 414 ConTSP benchmark instances.

prove optimality after 2 hours, the final optimality gap featured by Formulation 1 was tightest in 71 instances and was second best in 13 instances; Formulation 2 was best in 9 instances and second best in 72 instances, while Formulation 3 was best in 16 instances and second best in 11 instances. Although there does not appear to be any one formulation that performs best across the whole range of instances, Formulation 1 clearly dominates in terms of number of instances solved, quality of lower bounds obtained and solution times. Table 4 also summarizes separately the computational performance on instances with low, medium and high levels of the maximum allowable arrival-time differential (L), on 3-period and 5-period instances, as well as on symmetric and asymmetric instances. We observe that, on average, the instances with lower values of L (higher service quality), higher number of periods and asymmetric travel costs are harder to solve than their counterparts with higher values of L (lower service quality), lower number of periods and symmetric costs, respectively.

To better demonstrate the usefulness of the proposed branch-and-cut framework, we also attempted to solve all of the 414 instances with CPLEX in its default setting; that is, with a branch-and-cut implementation that uses general-purpose cuts only, and without separation of the valid inequalities described in Section 4. We utilized Formulations 2 and 3, which are our two formulations that are polynomial in size (as opposed to Formulation 1, which is not), and can thus serve as the basis for this experiment. As before, the best available solution was provided from the beginning as the upper bound in each case, and a time limit of 2 CPU hours was imposed. The performance of default CPLEX is also compared in Fig. 2. In summary, using Formulations 2 and 3, default CPLEX was able to prove the optimality of 182 and 246 instances, respectively. The average times were 845.9 and 664.8 seconds, the average numbers of nodes were 284,085 and 35,819, while the average optimality gaps for the unsolved instances were 6.46 and 5.13 percent, respectively. We observe that the computational performance of these formulations with default CPLEX branch-and-cut is significantly worse than the performance of the same formulations in the context of our proposed framework,

Table 6

Cost of providing consistent service (averaged across all instances).

	Low L	Med. L	High L
$f = 50\%$	1.85	1.32	1.09
$f = 70\%$	2.10	1.64	1.37
$f = 90\%$	0.99	0.75	0.67
$h = 3$	1.52	1.18	1.03
$h = 5$	1.77	1.30	1.06
	1.65	1.24	1.05

both in terms of number of instances solved and in terms of residual optimality gaps. This observation eludes to the fact that a custom-built algorithm, such as the one developed in this study, is necessary to efficiently address the ConTSP instances we considered.

Table 5 reports, for each of the formulations, the time spent in separation routines as a percentage of the total computation time (subject to a time limit of 2 CPU hours, as in Table 4) as well as the number of different families of violated cuts identified during the course of the search process. It is evident that separation time is not a major part of the total computing time, especially in the case of Formulations 2 and 3. The majority of the time is instead spent towards the solution of the LP relaxations. On average, more cuts are added in the case of Formulation 1 (predominantly IPEC), while fewer cuts are added in the case of Formulation 3. Comparing with the data reported in Table 3, it is interesting to note that, although SEC are the majority of cuts violated early in the solution process, IPEC violations become more common, raising the relative percentage of separated IPEC. This can be attributed to the fact that, at deeper levels of the search-tree, the tours are refined enough to not admit any more subtour or 2-matching violations, leading to more cut separation effort towards IPEC.

To conclude, we highlight that the size of an instance does not necessarily correlate with solution tractability. Sometimes, instances of larger size may solve significantly faster than some of their smaller counterparts. For example, the 3-period 34 node instance $ftv33$, at the service frequency $f = 0.7$ and medium consistency level, is very difficult to solve with any of the 3 formulations. After 2 hours of computation time, the branch-and-cut algorithm based on Formulation 1 was only able to close the gap down to 2.17 percent. To that end, 19,408 nodes were opened and 10,034 cuts were added. On the other hand, the 3-period 48 node instance $gr48$, at the same service frequency and consistency levels, can be solved to optimality in under a second, requiring only 66 nodes and generating 286 cuts in the process. The largest instance solved to guaranteed optimality features 51 nodes and 5 time periods. However, a number of smaller instances could not be solved by the framework. The smallest unsolved instance features 34 nodes and 3 time periods.

6.3. The price of consistency

In this section, we aim to estimate the additional cost that one must incur, on average, in order to provide consistent service. To that purpose, we report in Table 6 the difference between the optimal ConTSP cost and the sum across all time periods of the costs of the optimal traveling salesman tours (without consistency), as a percentage of the latter. This quantity is reported separately for each different

level of service frequency (f), length of the planning horizon (h) and maximum allowable arrival-time differential (L) considered in this study, in order to also investigate the effect of these parameters on the overall cost increase.

Our study shows that an arguably small cost increase, 1.31 percent, must be incurred, on average, in order to provide consistent service. Across all instances we considered, the required additional cost varies between 0 and 8.25 percent, with four out of five instances commanding less than 2 percent of a cost increase. For low values of the maximum allowable arrival-time differential, i.e., high levels of service consistency, the additional cost that must be incurred reaches 1.65 percent, on average, and this number decreases to 1.05 percent when consistency levels are reduced (high L). Moreover, it is interesting to observe that a relatively higher cost must be incurred when the service frequency is 70 percent, as opposed to when it is 50 or 90 percent, and when the number of time periods is 5, as opposed to when it is 3. We also remark that the price of consistency is, on average, higher for asymmetric instances.

7. Conclusions

Multi-period routing problems with consistency requirements represent a practically relevant class of problems, as distributors can gain significant competitive advantages by providing consistent service to their customers. Arrival-time consistency, i.e., the requirement to visit customers at approximately the same time during the routing horizon, has been identified as one plausible avenue to add such value. In this paper, we introduced three mixed-integer linear programming formulations for the Consistent Traveling Salesman Problem and analyzed their potential worthiness in a branch-and-cut framework. This constitutes the first exact approach in the open literature that addresses a routing problem with consistency constraints. The effectiveness of the branch-and-cut algorithm was tested on a set of benchmark problems that was compiled by extending the well-known TSPLIB database. Instances with up to 50 customers that require service over a 5-period planning horizon were solved to guaranteed optimality. A formulation that uses only binary variables and that relies on cutting planes to enforce all consistency requirements was shown to be the most attractive from a computational viewpoint. Finally, our study suggests that a modest routing cost increase of the order of 1–2 percent would typically suffice so as to provide consistent service. Expected benefits for the distributor, however, may well make up for this small cost increase. Evidently, consistency of service constitutes a value proposition that distributors should consider further.

Acknowledgments

The authors gratefully acknowledge support from the U.S. National Science Foundation (award number CMMI-1434682).

References

- Applegate, D. L., Bixby, R. E., Chvatal, V., & Cook, W. J. (2007). *The traveling salesman problem: A computational study (Princeton series in applied mathematics)*. NJ, USA: Princeton University Press.
- Ascheuer, N., Fischetti, M., & Grötschel, M. (2000). A polyhedral study of the asymmetric traveling salesman problem with time windows. *Networks*, 36(2), 69–79.
- Ascheuer, N., Fischetti, M., & Grötschel, M. (2001). Solving the asymmetric travelling salesman problem with time windows by branch-and-cut. *Mathematical Programming*, 90(3), 475–506.
- Bérubé, J.-F., Gendreau, M., & Potvin, J.-Y. (2009). A branch-and-cut algorithm for the undirected prize collecting traveling salesman problem. *Networks*, 54(1), 56–67.
- Campbell, A. M., & Wilson, J. H. (2014). Forty years of periodic vehicle routing. *Networks*, 63(1), 2–15.
- Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2012). Consistency in multi-vehicle inventory-routing. *Transportation Research Part C: Emerging Technologies*, 24(0), 270–287.
- Dantzig, G., Fulkerson, R., & Johnson, S. (1954). Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2(4), 393–410.
- Desrochers, M., & Laporte, G. (1991). Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research Letters*, 1(February), 27–36.
- Dolan, E. D., & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2), 201–213.
- Drexl, M. (2012). Synchronization in vehicle routing – A survey of vrps with multiple synchronization constraints. *Transportation Science*, 46(3), 297–316.
- Dumitrescu, I., Ropke, S., Cordeau, J.-F., & Laporte, G. (2010). The traveling salesman problem with pickup and delivery: Polyhedral results and a branch-and-cut algorithm. *Mathematical Programming*, 121(2), 269–305.
- Feillet, D., Garaix, T., Lehuédé, F., Péton, O., & Quadri, D. (2014). A new consistent vehicle routing problem for the transportation of people with disabilities. *Networks*, 63(3), 211–224.
- Fischetti, M. (1991). Facets of the asymmetric traveling salesman polytope. *Mathematics of Operations Research*, 16(1), 42–56.
- Fischetti, M., Lodi, A., & Toth, P. (2007). Exact methods for the asymmetric traveling salesman problem. In G. Gutin, & A. P. Punnen (Eds.), *The traveling salesman problem and its variations. Combinatorial optimization: vol. 12* (pp. 169–205). US: Springer.
- Fischetti, M., & Toth, P. (1997). A Polyhedral Approach to the Asymmetric Traveling Salesman Problem. *Management Science*, 43(11), 1520–1536.
- Gavish, B., Graves, S. C. (1978). The travelling salesman problem and related problems. Working Paper OR 078–78, Operations Research Center, Massachusetts Institute of Technology.
- Godinho, M. T., Gouveia, L., & Pesneau, P. (2011). On a time-dependent formulation and an updated classification of atsp formulations. In A. R. Mahjoub (Ed.), *Progress in combinatorial optimization* (pp. 251–305). Wiley.
- Gouveia, L. (1995). A result on projection for the vehicle routing problem. *European Journal of Operational Research*, 85(3), 610–624.
- Groër, C., Golden, B., & Wasil, E. (2009). The consistent vehicle routing problem. *Manufacturing & Service Operations Management*, 11(4), 630–643.
- Grötschel, M., & Padberg, M. W. (1979). On the symmetric travelling salesman problem I: Inequalities. *Mathematical Programming*, 16(1), 265–280.
- Grötschel, M., & Padberg, M. W. (1985). Polyhedral theory. In E. Lawler, J. Lenstra, A. R. Kan, & D. Shmoys (Eds.), *The Travelling Salesman Problem* (pp. 251–305). New York: John Wiley and Sons.
- Karp, R. M. (1972). Reducibility among combinatorial problems. In R. E. Miller, J. W. Thatcher, & J. D. Bohlinger (Eds.), *Complexity of Computer Computations. In The IBM Research Symposia Series* (pp. 85–103). US: Springer.
- Kovacs, A. A., Golden, B. L., Hartl, R. F., & Parragh, S. N. (2014a). Vehicle routing problems in which consistency considerations are important: A survey. *Networks*, 64(3), 192–213.
- Kovacs, A. A., Golden, B. L., Hartl, R. F., & Parragh, S. N. (2014b). The generalized consistent vehicle routing problem. *Transportation Science*. Available at: <http://dx.doi.org/10.1287/trsc.2014.0529>.
- Kovacs, A. A., Parragh, S. N., & Hartl, R. F. (2014c). A template-based adaptive large neighborhood search for the consistent vehicle routing problem. *Networks*, 63(1), 60–81.
- Langevin, A., Desrochers, M., Desrosiers, J., Gélinas, S., & Soumis, F. (1993). A two-commodity flow formulation for the traveling salesman and the makespan problems with time windows. *Networks*, 23(7), 631–640.
- Letchford, A. N., Reinelt, G., & Theis, D. O. (2004). A faster exact separation algorithm for blossom inequalities. In D. Bienstock, & G. Nemhauser (Eds.), *Integer Programming and Combinatorial Optimization. In Lecture Notes in Computer Science: vol. 3064* (pp. 196–205). Berlin Heidelberg: Springer.
- Luo, Z., Qin, H., Che, C., & Lim, A. (2015). On service consistency in multi-period vehicle routing. *European Journal of Operational Research*, 243(3), 731–744.
- Lysgaard, J., Letchford, A. N., & Eglese, R. W. (2004). A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2), 423–445.
- Maffioli, F., & Sciomachen, A. (1997). A mixed-integer model for solving ordering problems with side constraints. *Annals of Operations Research*, 69, 277–297.
- 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.
- Öncan, T., Altinel, I. K., & Laporte, G. (2009). A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers & Operations Research*, 36(3), 637–654.
- Padberg, M., & Rinaldi, G. (1990). An efficient algorithm for the minimum capacity cut problem. *Mathematical Programming*, 47(1–3), 19–36.
- Padberg, M. W., & Rao, M. (1982). Odd minimum cut-sets and b -matchings. *Mathematics of Operations Research*, 7(1), 67–80.
- Reinelt, G. (1991). Tsplib – A traveling salesman problem library. *ORSA Journal on Computing*, 3(4), 376–384.
- Roberti, R., & Toth, P. (2012). Models and algorithms for the asymmetric traveling salesman problem: An experimental comparison. *EURO Journal on Transportation and Logistics*, 1(1–2), 113–133.
- Siek, J., Lee, L.-Q., Lumsdaine, A. (2000). Boost Graph Library. Available at: <http://www.boost.org/libs/graph/> [Accessed 12th December, 2014].
- Skorobohatyy, G. (2004). Finding a minimum cut between all pairs of nodes in an undirected graph. Available at: <http://elib.zib.de/pub/Packages/mathprog/mincut/index.html> [Accessed 12th December, 2014].
- Tarantilis, C., Stavropoulou, F., & Repoussis, P. (2012). A template-based Tabu Search algorithm for the Consistent Vehicle Routing Problem. *Expert Systems with Applications*, 39(4), 4233–4239.