The travelling salesman problem with positional consistency constraints: An Application to Healthcare Services

Luís Gouveia[1,2,3], **Ana Paias**[1,2,4], **Mafalda Ponte**[1,2,5,*]

[1] DEIO, Faculdade de Ciências, Universidade de Lisboa, C6, Piso 4, 1749-016 Lisboa, Portugal.

[2] Centro de Matemática, Aplicações Fundamentais e Investigação Operacional, Faculdade de Ciências, Universidade de Lisboa.

[3] E-mail: legouveia@fc.ul.pt [4] E-mail: ampaias@fc.ul.pt [5] E-mail: mafaldaponte@gmail.com

[*] Corresponding author

Abstract

In this paper we study the *Consistent Traveling Salesman Problem* with positional consistency constraints (CTSP), where we seek to generate a set of routes with minimum cost, in which all the clients that are visited in several routes require *total positional consistency*, that is, they need to appear in the same relative position in all the routes they are visited in. This problem was motivated by a scheduling application in healthcare. We present several compact formulations for the CTSP, which have been adapted from models known from the Time-dependent TSP (TDTSP) literature, and propose a new model, which is an aggregated version of a model adapted from the TDTSP. A preliminary computational experience allows us to identify the three most competitive models. These models were then evaluated in more detail, first through a set of instances with 2, 3 or 5 routes and characteristics that derive from an healthcare application; and second through a set of tests with 5 routes and seven different and more general consistency configurations. The computational results show that for consistency configurations in which the consistent nodes appear in all, or most, of the routes, the new aggregated model can outperform the best model adapted from the literature. For the cases where the consistent nodes appear in fewer routes or less frequently, the original time-dependent model is more efficient.

***K*eywords** Combinatorial optimisation · Travelling salesman problem · Positional consistency · Healthcare services

1 Introduction

Vehicle Routing problems have become very popular as mathematical models for a wide variety of applications, ranging from the most obvious cases, namely pickup and delivery problems such as newspapers distribution and children transportation, to less obvious cases, such as security patrols and the order planning of machinery tasks. Quite often, instead of planning one single route, the planning of several routes is desirable. This might be due to the existence of more than one entity, or vehicle, whose activity we need to optimise, or only one entity, to optimise its activity for a time horizon composed by multiple time periods. In such situations, synchronisation and/or consistency requirements become relevant in many applications.

[Kovacs et al., 2014b] present several examples of applications where synchronisation and consistency in route planning is relevant, and define different types of consistency that can be provided. In particular, we can have *person-oriented consistency*, when the same client is served by the same driver in several periods; *delivery consistency*, when the same client receives similar quantities of a given good each time that there is a delivery; and *service consistency*, when the same client is served roughly at the same time in different time periods. In many situations, these three, or some of these three, consistency types might be desirable. The first type results in routes that are more predictable, that may be used in a more efficient way, and is beneficial for both the driver and the client. The second type makes service more predictable, increasing customer satisfaction. The third type results in a more personalised and predictable service, also improving customer satisfaction. Additionally, besides being considered desirable, *service consistency* can be a prerequisite for the service. Also, service consistency may be interpreted in two different ways. On one hand, it might mean that a client must be served (or a task must be performed) roughly at the same time in two different time periods (days, for instance). On the other hand, this consistency type may appear in the planning of a single time period, for several vehicles/workers. In this context, service consistence represents the need for several vehicles/workers to be in the same place at the same time, which may be, for instance, due to the fact that some tasks require more than one worker to be performed, or for transshipment purposes. Finally, the same authors also mention the possibility of aiming for *inconsistency*, instead of aiming for consistency (see [Kovacs et al., 2014a]). This is relevant in applications related to security, or transportation of expensive items, where the unpredictability introduced by inconsistency may improve safety.

In the context of service consistency, the *Consistent Travelling Salesman Problem* (CTSP), a variant of the *Travelling Salesman Problem* (TSP), is an adequate model. In the CTSP one determines a minimum-cost set of routes to be performed by one vehicle, ensuring that each client is visited in all the periods service is required, and that the consistency constraints are verified. Each client may require service in one or several periods, and the clients requiring more than one service, may or may not require consistency. *A priori* it is known which clients must be visited in each period of the time horizon, as well as which clients require consistency.

Two different modelling views can be used for service consistency. The first, *temporal consistency*, consists of incorporating travelling (and eventually service) times as parameters, and using them to write constraints that impose an upper bound to the difference between arrival/service times for the same client in different routes. This approach is suitable for scheduling applications where travel times are relevant, and where one service can start at any moment in the planning horizon, for instance, in applications where healthcare professionals visit patients in their homes, and some patients must be visited by several healthcare professionals. This is usually the case in home healthcare applications, of which we can find examples in [Braekers et al., 2016], [Decerle et al., 2017] and [En-nahli et al., 2016]. In fact, all of these examples use this approach, although they enforce consistency in different ways. In the first case, each patient has an ideal service time and a preferred nurse. Consistency between the real service time and the ideal service time is not enforced as a constraint, but the lack of consistency is penalised in an objective function that represents customer inconvenience, in a bi-objective framework. In the second case, the aim is to minimise the total travelling time associated with all the routes, the difference between the real service time and ideal service time as well as between the arrival times of the healthcare professionals, when several professionals must perform the service, and the difference between the travelling times of the longest and the shortest routes. Different objectives receive different weights. Finally, in the third case, consistency is enforced through constraints, guaranteeing that each task must be performed within a time window, and tasks that must be performed by several workers are subject to synchronisation constraints. The second modelling view, *positional consistency*, consists of dividing the plan horizon in fixed intervals (time slots), with each interval being associated to a single task/client. The relative position that a client occupies in a given route represents the allocation of the client to a time slot. As a consequence, we are able to write consistency constraints in terms of positions. As we will see in this study, this approach is adequate to plan schedules for healthcare professionals in a health centre, where the work shift can be divided in tasks that take nearly the same time, and the travel time from one task to the other, although it still can affect the efficiency of the shift, is mostly negligible.

Also, the definition of consistency between two or more routes needs to be more precise. We can define two tasks/clients as being consistent if they have to be performed roughly at the same time in different time periods or routes. This definition of consistency can be easily modelled using temporal consistency with suitable time intervals modelling the difference between arrival times. One can also aim for *total consistency* describing the situation where the tasks are performed exactly at the same time and which is modelled by setting the time interval to zero. We observe, that in many situations, total consistency is far from being practical. On the other hand, total consistency is adequate in positional consistency where the tasks need to be performed exactly at the same relative position.

Subramanyam and Gounaris ([Subramanyam and Gounaris, 2016] and [Subramanyam and Gounaris, 2017]) studied the CTSP with temporal consistency constraints. For this type of consistency, an upper limit is imposed for the difference between the latest and the earliest vehicle arrival-time for each client that requires consistency. The authors proposed exact models for the CTSP with temporal consistency for the vehicle arrival-times in [Subramanyam and Gounaris, 2016], namely two compact models, a MTZ based formulation and a Single Commodity Flow based model, as well as a non-compact formulation, containing subtour elimination constraints and consistency constraints (defined by pairs of inconsistent paths). The authors described a branch-and-cut method based on the non-compact model, in addition to strengthened versions of the consistency constraints and two other families of valid inequalities. Besides comparing it with the performance of the two compact models, this method allowed to solve, to optimality, instances with up to 50 customers and 5 time periods. For the instances tested by the authors, the upper limit was set to 10, 15 or 20 percent of the maximum travel time among all the routes, when optimised individually and without taking consistency into consideration. The work [Subramanyam and Gounaris, 2017] describes a decomposition method for a variant of the problem where each vehicle is allowed to be idle between two clients and where each route is optimised independently of the others, and consistency is introduced while branching, using time windows. This method allowed to solve, to optimality, instances with up to 100 customers and 5 time periods. This study also showed that allowing the vehicle to be idle between different services can lead to solutions which have a lower total cost.

In this paper, we will focus on compact formulations, i.e., those that are polynomial in the number of variables and constraints, for the problem with *total positional consistency* constraints (to which we will refer as *total consistency* in the remainder of the text). In the scheduling of healthcare services, which motivates the current study, it is common to divide the working day in blocks, namely 30 minutes each, and to assign a task to each block. We will consider each route to represent a healthcare professional in the same working day. The tasks that must be performed by several healthcare professionals must be assigned to the same time block, which is attained with total consistency. Since in many cases we have the same number of doctors and nurses in the healthcare facility, and the same doctor is usually paired with the same nurse, within certain boundaries each pair doctor/nurse can plan their own schedules fairly independently from the other pairs' schedules. Our main motivation for focusing our study on compact formulations is that they can be used in combination with off-the-shelf optimisation software and more easily used by a practitioner, e.g, a healthcare centre practitioner or someone acquainted with a healthcare centre practitioner, not acquainted with specialised optimisation techniques. Formulations that may involve the use of exponentially sized sets of constraints, or variables, usually require the use of specialised methods, such as constraint separation, which may not always be easy to understand, implement and use for a practitioner not acquainted with specialised optimisation techniques.

The fact that we are studying a positional version of the CTSP allows us to refer to another related problem, namely the Time-Dependent Travelling Salesman Problem (TDTSP). In the TDTSP we assume that the travelling costs depend not only from the origin and destination of one trip, but also from the position that such trip occupies in the route. The connection to the CTSP is that the position index in the variables of the TDTSP models can be interpreted as the position of the tasks in the routes. Thus, formulations that have been proposed to model the TDTSP can be expanded for the CTSP by adding adequate consistency constraints that are easily written due to the extra index in the variables. However, the advantage of writing more straightforward consistent constraints has the drawback of the extra number of variables/constraints (at least when compared to typical models of the TSP). Thus, we will also focus our study on the derivation of valid aggregated formulations that do not use variables indexed by vehicle/worker. It is worth pointing out that such aggregated models for routing problems where nodes/clients are visited by more than one route are not easy to derive. In Subsection 3.2 we make a brief reference to such problems. However, and as we shall show later on, the characteristics of the problem under study allow us to obtain an aggregated version of a particular time-dependent model.

In Section 2 we present a formal definition of the problem and give more details about a healthcare application that motivated this study. In Section 3, we present formulations for the CTSP that are adapted from known models from the TDTSP and augmented with different sets of consistency constraints We also show some theoretical results that allow us to derive dominance relations between the linear programming relaxation of the different models. In particular, we present a discussion concerning aggregated formulations, that is, formulations with variables that are not indexed by route. Additionally, we propose an enhancement for the two most competitive models. Section 4 describes the test instances as well as the results of the computational experiment. Finally, in Section 5 we present the main conclusions of this work.

2 Problem definition and the healthcare application

In the problem we are studying, we assume that the routes can have different sizes, and thus we need to clarify what we mean by the consistency of a node that is included in routes of different sizes. We could assume that all the routes start at position 1 and thus smaller routes would finish earlier. However, we may be able to obtain better solutions if we shift the smaller route(s) forward and allow it (them) to start later than the largest route. To do this, we distinguish two position concepts: the *local position* from the *global position*, where the former is the position a client occupies in a route, when that route is taken independently from the others, while the latter is the position the client occupies in a route, when compared with the largest route. That is, for a route that starts later, the node in the first *local* position may be in the kth ($k > 1$) *global* position. Of course, for the largest route there is no need to distinguish between local and global position. In this work the consistency constraints will be written in terms of global positions and hereinafter whenever we refer to a client position we are referring to a client global position. We will get back to this later on, when we describe the application in healthcare. Also, in order to limit the spread of the work shifts, and also to avoid the permanency of idle professionals in the health center, we do not allow any route to have unoccupied positions between two occupied positions. In other words, for each route all the occupied positions are consecutive.

To define the CTSP with total consistency we consider a graph $G = (V, A)$, where V is the set of nodes, $n = |V|$, and A is the set of arcs. For each arc (i, j), the cost of travelling from node i to node j, is given by C_{ij}. Node 1 is the depot and the subset $V \setminus \{1\}$ defines the set of clients. We also consider m subsets of clients, V_l, $l = 1, ..., m$, which are not disjoint. The aim of the problem is to define m routes, each route l, covering the subset of clients $V_l \cup \{1\}$. The clients that must be served in more than one route require total positional global consistency. That is, these clients must be served in the same global position in all the routes they are included in. The total cost, given by the sum of the costs of the m routes, is to be minimised. Observe that if we remove the consistency constraints, the CTSP reduces to m TSPs, one for each period. Therefore, since the TSP is a NP-hard problem, we can conclude that the CTSP is also NP-hard.

This work is motivated by an application arising in the scheduling of healthcare services and thus we will also describe the application in terms of the generic problem described above. This particular application consists of determining minimum cost schedules for a set of healthcare professionals that work in a center of the National Health Service in Portugal. For this purpose, the working day is divided in time slots (all with the same duration), and each task must be assigned to a time slot. Some tasks must be performed by several healthcare professionals at the same time, and therefore all the healthcare professionals that participate at one such task must assign it to the same time slot. This problem can be solved as a CTSP with total consistency. Node 1 represents that the healthcare professionals are inactive, and the remaining nodes represent the tasks. Each route represents the schedule for a healthcare professional. The relative position that a task occupies in a route represents the allocation of that task to a time slot in the corresponding schedule. Consequently, we can formulate the need for all the agents that perform one task to assign it to the same time slot using consistency constraints.

For this application, we will make a few more considerations about the parameters that define the problem. The arc costs represent the cost of performing one task right after another, and which therefore include aspects like storing back material that will no longer be needed, going from one room to another, and preparing material for the next task. Therefore, these costs will take smaller, more subjective values than they would if they were measured by travel times, distances or monetary costs. For instance, if i and j are similar tasks, we assign to arc (i, j) the lowest cost, 0. If the arc represents the need of getting new equipment, or of switching computer systems, we assign costs 1 or 2, respectively. If one machine needs to be turned on or off, or if the healthcare professional needs to switch rooms, we assign cost 3 or 4, respectively. As a result, an arc with cost 1 will be preferred to one with cost 2. However, this does not mean that the latter is two times worse than the former, therefore, to avoid such an interpretation, we sum a constant value, 10, to all the costs, thus obtaining costs that vary between 10 and 14. Additionally, although there is some flexibility regarding the starting/finishing time of the shifts, their maximum spread is fixed. In particular, in this study we will assume that the work shifts cannot have unoccupied time slots between occupied slots, that is, all the tasks must be consecutive avoiding the permanency of idle professionals at the center. Finally, the fact that routes may not have the same number of nodes, corresponds to the cases where the healthcare professionals do not work the same number of hours in the same day. In this case, for simplicity, we will assume that the work day has $(T-1)$ time slots, with T being equal to the number of tasks in the largest work shift plus a dummy task that represents inactivity. In other words, the work day has as many time slots as the route with the largest number of nodes. The healthcare professionals with a smaller number of tasks can start work later than the professional with the largest number of tasks, leave work earlier, or both. To illustrate the importance of this last consideration, we consider a small example where the starting time of the smallest route is relevant for feasibility purposes. Consider two healthcare professionals, $H1$ and $H2$. $H1$ has a 6-hour shift, whereas $H2$ has a 4-hour shift. Both shifts are divided in tasks that take 1 hour each. $H1$ has to perform tasks $\{t1, t2, t3, t4, t5, t6\}$ and $H2$ needs to perform tasks $\{t5, t6, t7, t8\}$. Since tasks $t5$ and $t6$ belong to both professionals, they require total consistency, that is, they must appear in the same position/time slot.

As noted before, we create an extra node to use as a depot, which represents that the healthcare professionals are idle; assign a node to each task; and therefore can use routes to represent the sequence of tasks, as shown in Figure 1.

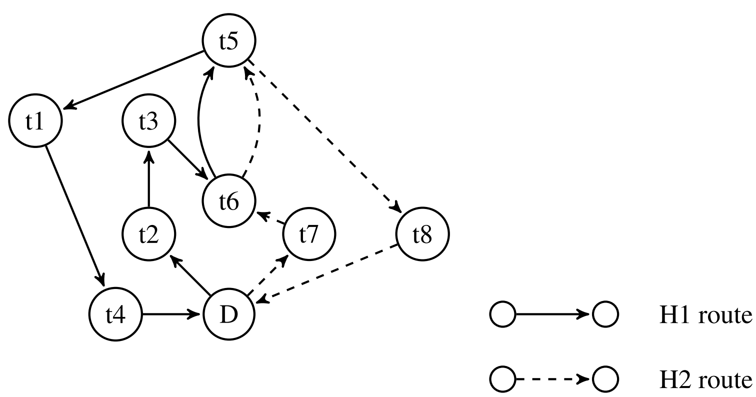

Figure 1: Using routes to represent sequences of tasks

In Figure 1 both healthcare professionals start their shift at the depot (D), which represents inactivity, and follow a route that goes through all the nodes representing tasks they have to perform and finish again at the depot, which means that the shift has ended. The relative position that each node occupies in the route (or routes) it appears in gives information about the time slot it is assigned to. One way to interpret the routes given in Figure 1 in terms of scheduling is represented in Table 1.

In Table 1 we assign each task directly to the relative position it occupies in the routes it appears in. Because professional $H2$ does not have to perform as many tasks as professional $H1$, the last time slots are empty, which means $H2$ finishes

Table 1: Scheduling example - both professionals start shift at slot 1

	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
H1	t2	t3	t6	t5	t1	t4
H2	t7	t6	t5	t8	-	-

the shift earlier than $H1$. This solution is not feasible, since tasks $t5$ and $t6$, which are common to both routes and therefore require total consistency, are not performed in the same time slot by both professionals. However, by shifting 1 slot forward the sequence of tasks of professional $H2$ we obtain a feasible solution, as shown in Table 2.

Table 2: Scheduling example - the professional with less tasks is allowed to start at a later time slot

	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
H1	t2	t3	t6	t5	t1	t4
H2	-	t7	t6	t5	t8	-

This example shows the relevance of introducing the concepts of local position and global position. For the route associated with $H1$ there is no need to distinguish these concepts, but we need to do so for $H2$. For instance, locally $t6$ occupies position 2 and $t5$ occupies position 3. However, by allowing these tasks to occupy, respectively, global positions 3 and 4 we are able to obtain a feasible solution that is represented by the two routes in Figure 1.

Finally, it is important to remark that the fact that smaller routes are allowed to start later than larger routes, despite allowing for cheaper optimal solutions, also adds the complexity of having several feasible solutions, which differ in terms of the positions that the nodes occupy, but which are not distinguishable in terms of the arcs that are used, or the order of the nodes. One consequence of this is that it may not be possible to adapt, in a straightforward way, formulations that do not explicitly include information about the position of nodes/arcs, namely some formulations that are valid for the TSP with temporal consistency.

3 Mixed Integer Linear Programming Formulations

In this section we present and discuss two classes of compact formulations that will be considered in this study. The first, and larger, class, described in Section 3.1, includes *disaggregated* models, with the variables having an index identifying the route or worker. The second class, described in Section 3.2, discusses *aggregated* models, which result from aggregating some of the models discussed in the previous subsection, and with the variables of the model not having explicit information on the route/worker. In Section 3.3 we propose two sets of inequalities that enhance the models discussed in sections 3.1 and 3.2. Before presenting the formulations, we define the following parameters:

- Parameter S_i^l, that takes value 1 if node i must be in route l, and 0 otherwise.
- The number of nodes in each route, N^l. Note that $N^l = \sum_{i=1}^{n} S_i^l;\ \forall l = 1, ..., m$. This number includes the depot.
- The set of clients that must be visited in route l, $V_l = \{i : S_i^l = 1,\ 2 \leq i \leq n\}$. Note that $|V_l| = N^l - 1$.
- The longest route has T arcs with $T = \max\{N^l : l = 1, ..., m\}$, or T positions.

3.1 Route/Worker Indexed Models

In this section we introduce three classes of compact formulations, the *Single Commodity Flow Model* and two time-dependent models, the *FGG3* model and the *Picard and Queyranne Model*, the last two containing two formulations each, depending on the choice of the consistency constraints.

3.1.1 The Single Commodity Flow Model

Since the work by Gavish and Graves (1978), single commodity flow based models have been providing a basic framework for modelling many routing problems. The model described in this section follows this framework and it is an adapted and slightly enhanced version of one of the models presented in [Subramanyam and Gounaris, 2016] for the CTSP with temporal consistency (the other is a Miller-Tucker-Zemlin ([Miller et al., 1960]) based model, which is discussed in the Appendix). The model allows the flow on the first and the last traversed arc to have a value different from 1 and T, respectively, in order to model routes with less than T arcs (as explained in Section 1). Also, and as explained below, it differs from the model described in [Subramanyam and Gounaris, 2016], since it uses an enhanced version of the linking constraints.

The *Single Commodity Flow* model (SCF) uses two sets of variables:

- Arc binary variables, x_{ij}^l, that take value 1 whenever arc (i, j) is in route l, and 0 otherwise.
- Flow variables y_{ij}^l, which indicate the amount of flow traversing arc (i, j) in route l.

The SCF model is as follows:

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{m} C_{ij}x_{ij}^l \tag{3.1}$$

$$\text{s.t. } \sum_{j=1}^{n} x_{ij}^l = S_i^l;\ \forall i = 1, ..., n;\ \forall l = 1, ..., m \tag{3.2}$$

$$\sum_{i=1}^{n} x_{ij}^l = S_j^l;\ \forall j = 1, ..., n;\ \forall l = 1, ..., m \tag{3.3}$$

$$\sum_{j=1}^{n} y_{ij}^l - \sum_{j=1}^{n} y_{ji}^l = S_i^l;\ \forall i = 2, ..., n;\ \forall l = 1, ..., m \tag{3.4}$$

$$x_{1j}^l \le y_{1j}^l;\ \forall j = 1, ..., n;\ \forall l = 1, ..., m \tag{3.5}$$

$$2 * x_{ij}^l \le y_{ij}^l;\ \forall i = 2, ..., n;\ \forall j = 1, ..., n;\ \forall l = 1, ..., m \tag{3.6}$$

$$y_{i1}^l \le T x_{i1}^l;\ \forall i = 1, ..., n;\ \forall l = 1, ..., m \tag{3.7}$$

$$y_{ij}^l \le (T-1)x_{ij}^l;\ \forall i = 1, ..., n;\ \forall j = 2, ..., n;\ \forall l = 1, ..., m \tag{3.8}$$

$$\sum_{i=1}^{n} y_{ij}^s = \sum_{i=1}^{n} y_{ij}^t;\ \forall s, t = 1, ..., m;\ \forall j \in V_s \cap V_t;\ s \ne t \tag{3.9}$$

$$y_{ij}^l \ge 0 \text{ and integer};\ \forall i, j = 1, ..., n;\ \forall l = 1, ..., m \tag{3.10}$$

$$x_{ij}^l \in \{0, 1\};\ \forall i, j = 1, ..., n;\ \forall l = 1, ..., m \tag{3.11}$$

The objective function (3.1) states that the total cost of the set of routes is minimised. For each route, if a client node must be visited in that route, then there must be an arc entering it and another one leaving it, as stated by constraints (3.2) and (3.3), respectively. Constraints (3.4) are the flow conservation constraints and indicate that in each route one unit of flow is added when a client node is visited. Constraints (3.5)-(3.8) are the constraints linking the two sets of variables. Essentially, they guarantee that if an arc is in a route, then there must be flow going through it and if there is flow going through an arc in one route, then that arc must be in the route. Constraints (3.5) guarantee that in the arc leaving the depot, the flow is at least equal to 1 and constraints (3.6) guarantee that on the remaining arcs of the route, the flow will be always equal to 2 units or more. Constraints (3.7) guarantee that the arc returning to the depot has at most a flow equal to T units, (which corresponds to the number of nodes in the route with the maximum number of clients). On the remaining arcs, constraints (3.8) state that the flow value never exceeds $T - 1$. The flow variables are defined as non-negative and integer (constraints (3.10)). Constraints (3.11) state that the arc variables are binary.

Constraints (3.5) and (3.6) are redundant and need not be included in the model, as in [Subramanyam and Gounaris, 2016]. However, these constraints are not redundant in the LP relaxation of the model leading to slightly tighter linear programming bounds. Furthermore, their inclusion allows us to state an equivalence result with a time-dependent model discussed in the next subsection.

In the model proposed above, we do not have the standard set of constraints arising, for instance, in TSP models, that state that one unit of flow leaves the depot in each route. In our model, the flow variables represent, in a certain sense, the position occupied by a node in a route. In fact, for the route with a larger number of nodes (or all the routes, if they have the same number of nodes) constraints (3.4), (3.7) and (3.8) guarantee that the nodes in the route will occupy consecutive positions 1 to $T - 1$. If one of the routes, l, has fewer nodes than the route with the larger number of nodes, then we do not know which positions will be occupied. The nodes in this route l will occupy N^l positions between 1 and $T - 1$, but it is not known *a priori* which positions will be occupied. Constraints (3.4) guarantee that those N^l positions are consecutive in the route.

As noted before, the flow variables indicate the position of an arc in the longest route, since one unit of flow leaves the depot, in that route, and the flow increases by one unit for each arc it passes through in the same route. Also, as explained before, the flow system allows smaller routes to "start later" or "end earlier". Constraints (3.9) use this information, by stating that for any node j requiring consistency, the value of the flow entering that node is equal in the routes the node is in. These constraints result from the consistency constraints given in [Subramanyam and Gounaris, 2016] by setting the limit equal to 0.

3.1.2 The FGG3 Model

The formulation presented in this section is an adaptation of the model proposed by Fox, Gavish and Graves [Fox et al., 1980] (see also [Gavish and Graves, 1978]) and uses time-dependent binary variables, z_{ij}^{kl}, which take value 1 if arc (i, j) is in position k in route l, and 0 otherwise. Some of the time-dependent variables are not considered in the model. In practice, these variables are not generated, however, in order to make the reading of the formulations easier, we generate them but set their value to 0, as stated in the expressions (3.12) and (3.13):

$$z_{ij}^{1l} = 0;\ \forall i = 2, ..., n;\ \forall j = 1, ..., n;\ \forall l = 1, ..., m \tag{3.12}$$

$$z_{ij}^{Tl} = 0;\ \forall i = 1, ..., n;\ \forall j = 2, ..., n;\ \forall l = 1, ..., m \tag{3.13}$$

Constraints (3.12) indicate that an arc in position 1 cannot be leaving from a node other than the depot, while equalities (3.13) state that an arc in position T cannot be entering a node other than the depot.

Note, however, and in contrast to what happens in the TSP, we cannot eliminate the variables associated to arcs in position 2 to $T - 1$ that are adjacent to the depot. To allow for consistency in routes with less than T arcs, arcs leaving the depot may be in positions greater than 1 and arcs entering the depot may be in positions less than T.

Before introducing the model we observe that the time-dependent variables are related to the decision variables from the SCF model in the following way:

$$x_{ij}^{l} = \sum_{k=2}^{T-1} z_{ij}^{kl};\ \forall i = 1, ..., n;\ \forall j = 1, ..., n;\ \forall l = 1, ..., m \tag{3.14}$$

$$x_{1j}^{l} = \sum_{k=1}^{T-1} z_{1j}^{kl};\ \forall j = 2, ..., n;\ \forall l = 1, ..., m \tag{3.15}$$

$$x_{i1}^{l} = \sum_{k=2}^{T} z_{i1}^{kl};\ \forall i = 2, ..., n;\ \forall l = 1, ..., m \tag{3.16}$$

and

$$y_{ij}^{l} = \sum_{k=2}^{T-1} k z_{ij}^{kl};\ \forall i = 1, ..., n;\ \forall j = 1, ..., n;\ \forall l = 1, ..., m \tag{3.17}$$

$$y_{1j}^{l} = \sum_{k=1}^{T-1} k z_{1j}^{kl};\ \forall j = 2, ..., n;\ \forall l = 1, ..., m \tag{3.18}$$

$$y_{j1}^{l} = \sum_{k=2}^{T} k z_{i1}^{kl};\ \forall i = 2, ..., n;\ \forall l = 1, ..., m \tag{3.19}$$

One particular case is obtained when $T = N^l$. In this case we know that an arc leaving the depot is in position 1, and an arc entering the depot is in position N^l. If, on the other hand, $T > N^l$, then we do not know *a priori* in which positions are the first and the last arc in the route. In these relations, we emphasise again the difference to the standard TSP. The model we describe below is denoted by $wFGG3$, "w" from "weak" (this designation will be explained later on):

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{T} \sum_{l=1}^{m} C_{ij} z_{ij}^{kl} \tag{3.20}$$

$$\text{s.t.} \sum_{i=1}^{n} \sum_{k=1}^{T} z_{ji}^{kl} = S_j^l;\ \forall j = 1, ..., n;\ \forall l = 1, ..., m \tag{3.21}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{T} z_{ij}^{kl} = S_j^l;\ \forall j = 1, ..., n;\ \forall l = 1, ..., m \tag{3.22}$$

$$\sum_{j=1}^{n}\sum_{k=1}^{T} kz_{ij}^{kl} - \sum_{j=1}^{n}\sum_{k=1}^{T} kz_{ji}^{kl} = S_i^l;\ \forall i = 2, ..., n;\ \forall l = 1, ..., m \tag{3.23}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{T} kz_{ij}^{ks} = \sum_{i=1}^{n}\sum_{k=1}^{T} kz_{ij}^{kt};\ \forall s, t = 1, ..., m;\ \forall j \in V_s \cap V_t;\ s \neq t \tag{3.24}$$

$$(3.12) - (3.13)$$

$$z_{ij}^{kl} \in \{0; 1\};\ \forall i = 1, ..., n;\ \forall j = 1, ..., n;\ \forall k = 1, ..., T;\ \forall l = 1, ..., m \tag{3.25}$$

Due to relations (3.14)-(3.19) it is easy to conclude that the $wFGG3$ model contains most of the constraints of the SCF model, rewritten with the time-dependent variables. In fact:

- Due to (3.14)-(3.16), the objective functions (3.1) and (3.20) are equivalent.
- For the same reason, the assignment constraints (3.21) and (3.22) are equivalent to the assignment constraints (3.2) and (3.3) of the SCF model.
- Due to (3.17)-(3.19), constraints (3.23) are the flow conservation constraints (3.4) of the SCF model, rewritten with the time-dependent variables.
- For the same reason, constraints (3.24) are the consistency constraints (3.9) of the SCF model, rewritten with the time-dependent variables.

Similarly to what has been said in the context of the SCF model, if we have two routes, s and h, and $N^s < N^h$, then in route h all positions $1, 2, ..., T$, with $T = N^h$ are occupied, while the arcs in route s will occupy N^s positions between 1 and T, although we do not know *a priori* which positions will be occupied. However, and as before, the flow conservation constraints (3.23) guarantee that those positions are consecutive.

The time-dependent variables are binary, as stated by constraints (3.25). As known from the literature (see, [Gouveia, 1995] and subsequent papers on this reformulation technique, e.g, [Gouveia and Voß, 1995]), one of the main ideas of the $wFGG3$ model is that the linking constraints (3.5)-(3.8) of the SCF model, rewritten with the time-dependent variables by using (3.14)-(3.19), are redundant and can be, and are, omitted from the model.

In fact, the relations between the sets of variables of the two models, allow us to adapt the following result, stated without proof since the modifications to the original proof of [Gouveia, 1995] are minor (also, a similar proof is added in the Appendix for aggregated versions of these two models). Let $LP(F)$ denote the optimal value of the linear programming relaxation of model F:

Theorem 3.1. $LP(wFGG3) = LP(SCF)$.

This result indicates that, in terms of the corresponding LP relaxations, we have not gained by rewriting the SCF model as the $wFGG3$ model. However, the time dependent variables also allow us to write the following new set of consistency constraints whose validity is straightforward:

$$\sum_{i=1}^{n} z_{ij}^{ks} = \sum_{i=1}^{n} z_{ij}^{kt};\ \forall j \in V_s \cap V_t;\ \forall s,t = 1,...,m;\ s \neq t;\ \forall k = 1,...,T-1 \tag{3.26}$$

We denote by $FGG3$, the $wFGG3$ model with constraints (3.24) replaced by constraints (3.26). Observe that if for a fixed j, k and l, we multiply the constraints (3.26) by k, then add for all k, we obtain constraints (3.24). This shows that constraints (3.26) imply constraints (3.24) (and in fact, for some instances, they provide better LP bounds). This dominance argument leads to the following result and explains the letter "w", from "weak", in the designation of the model $wFGG3$.

Theorem 3.2. $LP(FGG3) \geq LP(wFGG3)$.

Although providing a better LP bound, the computational results reported in Section 4, show that the $FGG3$ is not competitive with the SCF model when solving instances of the CTSP. The advantage of using the time-dependent variables will appear in the context of the model discussed in Section 3.1.2.

We conclude this section by pointing out that, for each route l, both the $wFGG3$ and the $FGG3$ models contain $3n$ constraints, explaining the $"3"$ in the designation $FGG3$. Fox, Gavish and Graves (see [Fox et al., 1980]) also proposed a model with a stronger linear programming relaxation, obtained from $wFGG3$, or $FGG3$, by adding a set of constraints stating that, for each position t, one and only one arc must be in that position. In the problem studied here, and since some routes may have less than T arcs, we could use a relaxed version of the constraints stating that for each position k, in each route, at most one arc could be in that position. Preliminary computational experiments have shown that the performance of such an augmented model is similar to the performance of the $wFGG3$, or $FGG3$ model, respectively, and thus this augmented model, besides a short reference later on, will not be considered any further.

3.1.3 The Picard and Queyranne Model

The model discussed in this section is an adaptation of the model proposed in [Picard and Queyranne, 1978], and uses the same decision variables as the models of the previous subsection. The model includes the objective function (3.20), constraints (3.22), (3.25), (3.12), (3.13) and the following constraints:

$$\sum_{i=1}^{n} z_{ij}^{kl} = \sum_{i=1}^{n} z_{ji}^{k+1,l};\ \forall j = 2,...,n;\ \forall k = 1,...,T-1;\ \forall l = 1,...,m \tag{3.27}$$

Constraints (3.27) state that one arc is entering node j in position k in route l if and only if in the same route one arc is leaving that node in position $k+1$. Due to the new constraints (3.27), one of the sets of the assignment constraints (3.22) and (3.21) is redundant, and therefore we did not include constraints (3.21) in the model.

We define by wPQ, the time-dependent model just described with the consistency constraints (3.24). We denote by PQ, the wPQ model with constraints (3.24) replaced by constraints (3.26).

The next result is similar to Theorem 3.2

Theorem 3.3. $LP(PQ) \geq LP(wPQ)$.

We conclude this subsection by relating the two families of time-dependent models. The result is stated without proof since the modifications to the original proof of [Gouveia and Voß, 1995] are minor.

Theorem 3.4. *$LP(wPQ) \geq LP(wFGG3)$, and $LP(PQ) \geq LP(FGG3)$.*

The computational results show that the constraints (3.27), that characterise the PQ and wPQ models, lead to much more efficient models than the constraints (3.24) that characterise de FGG3 and wFGG3 models. Also, with exception to the aggregated model discussed in the next subsection, the PQ model is, by far, more adequate to solve the instances we have tested for the CTSP.

3.1.4 The Time-Dependent Models with Node Position Variables

In this section we rewrite the two models, PQ and $FGG3$, with node position variables. We omit from this discussion the wPQ and $wFGG3$ models since they have a weaker linear programming relaxation than their stronger versions and, as our results will show, they are not competitive when compared with the corresponding stronger versions. We will see that the position variables can be eliminated from the new models leading to the original model in case of the PQ model and to a model with an additional set of non-redundant constraints in the case of the $FGG3$ model. The reason for introducing the new variables and a corresponding set of constraints is that they will be relevant for the discussion of Section 3.2, where we discuss aggregated versions of these two models. Consider the binary variables, p_i^k, which take value 1 if node i is in position k in any route l, and 0 otherwise. Observe that these variables are consistently defined, since if a node is included in several routes, then it must be included in the same position in these routes. Using these variables, we can write the following system (denoted in the remainder of the text as $node - pos$ system) of consistency constraints:

$$\sum_{k=1}^{T-1} p_j^k = 1; \; \forall j = 2, ..., n \tag{3.28}$$

$$\sum_{j=2}^{n} S_j^l p_j^k \leq 1; \; \forall l = 1, ..., m; \; \forall k = 1, ..., T-1 \tag{3.29}$$

$$p_i^k \in \{0; \; 1\}; \; \forall i = 2, ..., n; \; \forall k = 1, ..., T-1 \tag{3.30}$$

Constraints (3.28) state that every node j is associated with one and only one position k. Several nodes may occupy the same position, as long as they do not belong to the same route, as stated by constraints (3.29). Finally, the position variables are binary, as stated by constraints (3.30). These node variables have been used before in Gouveia and Voss ([Gouveia and Voß, 1995]) to define a quadratic model for the TDTSP.

We can combine the system described above with the time-dependent arc variables z_{ij}^{kl} with

$$S_j^l p_j^k = \sum_{i=1}^{n} z_{ij}^{kl}; \; \forall j = 2, ..., n; \; \forall k = 1, ..., T-1; \; \forall l = 1, ..., m \tag{3.31}$$

and we redefine the PQ model with constraints (3.28)-(3.31) instead of (3.26). The linking constraints (3.31) can also be written in a symmetric way as follows:

$$S_j^l p_j^{k-1} = \sum_{i=1}^{n} z_{ji}^{kl};\ \forall j = 2, ..., n;\ \forall k = 2, ..., T;\ \forall l = 1, ..., m \tag{3.32}$$

However, these constraints are implied by the PQ constraints (3.27) and the previous set of linking constraints (3.31). Thus, they do not need to be included in the model, however, they will be used in the proof of the validity of an aggregated version of the PQ model discussed in the next section. As observed before, with the linking constraints described above the node position variables p_i^k can be eliminated leading to the original PQ model.

We redefine the $FGG3$ model adding the same set of constraints (3.28)-(3.31) and removing constraints (3.26). In contrast with the PQ model, the model obtained in this way has a LP relaxation slightly stronger than the LP relaxation of the original $FGG3$ model. To see this, we observe that as with the PQ model, the node position variables p_i^k can also be eliminated from the modified $FGG3$ model. However, after eliminating the position variables, we obtain a model that is stronger than $FGG3$, namely with additional constraints stating that for each position k, in each route, at most one arc could be in that position. This set of constraints was already mentioned at the end of Section 3.1.2, in the context of a brief reference to augmented $FGG3$ and $wFGG3$ models.

3.2 Aggregated formulations

Many variants of the vehicle routing problem can be modelled using an arc or node-based integer linear programming formulation in which the arc or node variables are not indexed by route. This results in a fewer number of variables, which, in turn, allows the development of state-of-the-art solution methods. Although earlier formulations proposed for some routing problems used route-indexed variables, new insights on the problem later on led to the development of formulations that did not require the use of a route index. The Capacitated Vehicle Routing Problem (CVRP, see, e.g, [Toth and Vigo, 2014]) is one example, where the capacity constraints are easily modelled using variables indexed by route, but the so-called capacity-cut inequalities that were developed later on use aggregated variables that do not require a route index. The reader is referred to Magnanti ([Magnanti, 1981]) for a survey of earlier formulations of the CVRP.

It is worth pointing out that there still exist routing problem variants for which formulations that do not use route-index variables have not yet been described, and, we believe that it is unlikely that such models can be developed. Two such variants are the Split Delivery Vehicle Routing Problem (see, e.g., [Archetti et al., 2014]) and the Arc Routing Problem, (see ,e g., [Gouveia et al., 2010]). Routing problems involving several routes and synchronisation constraints are, in general, also difficult to model without route-indexed variables (see, e.g., [Drexl, 2012]). In fact, one example of this situation is illustrated by a straightforward and naive attempt to model the CTSP with an aggregated version of the PQ model that results from adding, for each route, the variables and constraints of the original model. It is far from clear as to whether it is possible to obtain an aggregated version of the constraints (3.26) without loosing the information given by the disaggregated variables used in the latter.

In this section, we show that after introducing the position variables p_i^k and the $node - pos$ system and the subsequent modifications described in Section 3.1.4, it is possible to obtain a valid aggregated version of the PQ model, although with a weaker linear programming relaxation. This aggregated model will be the most efficient one to solve several of the CTSP instances in our computational experiment. Besides the variables p_j^k, the aggregated formulation contains the aggregated variables u_{ij}^k that indicate the number of times arc (i, j) is traversed in position k. The aggregated variables are related with the original variables z_{ij}^{kl} as follows:

$$u_{ij}^k = \sum_{l=1}^{m} z_{ij}^{kl};\ \forall i, j = 1, ..., n;\ \forall k = 1, ..., T \tag{3.33}$$

An upper bound on the number of times an arc (i, j) is traversed in position k is given by the number of routes including both nodes i and j, that is, the number of routes l where $S_i^l = S_j^l = 1$. We denote this value by SR_{ij}. Observe that $SR_{ij} = \sum_{l=1}^{m} S_i^l S_j^l$. If either i or j is equal to 1 (the depot), we can simplify this expression to $SR_{1i} = SR_{i1} = \sum_{l=1}^{m} S_i^l$, since $S_1^l = 1;\ \forall l = 1, ..., m$.

Similar to what is done for the route indexed time-dependent formulations, some of the arc variables should not be generated, or be set to value 0 as indicated by the following constraints:

$$u_{ij}^1 = 0;\ \forall i = 2, ..., n;\ \forall j = 1, ..., n \tag{3.34}$$

$$u_{ij}^T = 0;\ \forall i = 1, ..., n;\ \forall j = 2, ..., n \tag{3.35}$$

Constraints (3.34) indicate that an arc in position 1 cannot be leaving from a node other than the depot, while constraints (3.35) state that an arc in position T cannot be entering a node other than the depot.

The previous relations allow us to obtain the aggregated PQ model to be presented in the next subsection. Later on, we will also show that a similar aggregated version of the FGG3 and the SCF models are not valid. This emphasises the modelling strength of the constraints defining the PQ model, namely constraints (3.27), and that they are needed in order to be able to obtain a valid aggregated model even after modifying it with the $node - pos$ system.

3.2.1 The Aggregated PQ Model

The aggregated PQ model, denoted by APQ, is defined as follows:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{T} C_{ij} u_{ij}^k \tag{3.36}$$

$$\text{s.t.} \sum_{i=1}^{n} \sum_{k=1}^{T} u_{ij}^k = \sum_{l=1}^{m} S_j^l;\ \forall j = 1, ..., n \tag{3.37}$$

$$\sum_{i=1}^{n} u_{ij}^k = \sum_{i=1}^{n} u_{ji}^{k+1};\ \forall j = 2, ..., n;\ \forall k = 1, ..., T-1 \tag{3.38}$$

$$\sum_{i=1}^{n} u_{ij}^k = (\sum_{l=1}^{m} S_j^l) p_j^k;\ \forall j = 2, ..., n;\ \forall k = 1, ..., T-1 \tag{3.39}$$

$$\sum_{k=1}^{T-1} p_j^k = 1;\ \forall j = 2, ..., n \tag{3.40}$$

$$\sum_{j=2}^{n} S_j^l p_j^k \leq 1;\ \forall l = 1, ..., m;\ \forall k = 1, ..., T-1 \tag{3.41}$$

$$(3.34) - (3.35)$$

$$u_{ij}^k \leq SR_{ij};\ \forall i, j = 1, ..., n;\ \forall k = 1, ..., T \tag{3.42}$$

$$u_{ij}^k \geq 0 \text{ and integer};\ \forall i, j = 1, ..., n;\ \forall k = 1, ..., T \tag{3.43}$$

$$p_j^k \in \{0; 1\};\ \forall j = 2, ..., n;\ \forall k = 1, ..., T-1 \tag{3.44}$$

In the objective function (3.36) we minimise the total cost of the set of routes. Constraints (3.37) are the assignment constraints and state that we have as many arcs entering a node j as routes in which the node must be visited. Constraints (3.38) are the aggregated version of the constraints that typically define the PQ model, namely constraints (3.27), and state that there are as many arcs leaving a node j in position $k+1$ as arcs entering j in position k. Constraints (3.39) link the two sets of variables and model the consistency constraints. They state that all arcs that enter node j, enter in the same position. Constraints (3.40) and (3.41) are taken from the $node - pos$ system. Constraints (3.40) state that each node occupies one and only one position, and constraints (3.41) state that each position cannot be occupied by more than one node appearing in the same route. To understand the meaning of constraints (3.42) we refer the reader to the definition of the parameter SR_{ij}. These constraints provide an upper bound on the value of the variables u_{ij}^k given by the number of routes where both i and j must appear in. We will say more about these constraints in Lemma 3.4. Observe also that these constraints prevent arcs between nodes that do not appear in the same route since $SR_{ij} = 0$. Finally, variables u_{ij}^k are integer and non-negative, while variables p_j^k are binary.

The symmetric behaviour of the PQ model is also seen in its aggregated version. Observe that combining constraints (3.37) together with constraints (3.38) leads to the redundant constraints

$$\sum_{i=1}^{n} \sum_{k=1}^{T} u_{ji}^k = \sum_{l=1}^{m} S_j^l;\ \forall j = 1, ..., n \tag{3.45}$$

stating that we have as many arcs leaving a node j as routes in which the node must be visited (which is a symmetric interpretation of the one given for constraints (3.37)). In a similar way, constraints (3.39) together with constraints (3.38) imply the constraints

$$\sum_{i=1}^{n} u_{ji}^{k+1} = (\sum_{l=1}^{m} Sj^l) p_j^k;\ \forall j = 2, ..., n;\ \forall k = 1, ..., T-1 \tag{3.46}$$

These constraints have the symmetric interpretation of constraints (3.39), namely that all arcs that leave node j, leave in the same position. These two sets of constraints, namely constraints (3.39) and (3.46), are relevant for the proof of the Lemma 3.4 which states that in any feasible solution for the APQ model, there are three sets of variables u_{ij}^k that either take its minimum or maximum feasible value. We start with four lemmas on which we state and prove conditions of the feasible solutions of the APQ model that will be used on the proofs of results to be presented later on.

Lemma 3.1. *The following condition is valid for any feasible solution $\{u_{ij}^k, p_j^k\}$ for the APQ model: For all i, j, k such that $i, j = 2, ..., n$, $k = 2, ..., T-1$, if $u_{ij}^k > 0$ then $p_i^{k-1} = p_j^k = 1$.*

Proof. The fact that $p_i^{k-1} = p_j^k = 1$ if $u_{ij}^k > 0$ follows from constraints (3.38) and (3.39). □

Lemma 3.2. *The following constraints are valid for any feasible solution $\{u_{ij}^k, p_j^k\}$ for the APQ model:*

$$u_{ij}^k \le SR_{ij} p_i^{k-1} p_j^k, \forall i, j = 2, ..., n, k = 2, ..., T-1 \tag{3.47}$$

$$\sum_{i=2}^{n} S_i^l S_j^l p_i^{k-1} p_j^k \le S_j^l p_j^k, \forall j = 2, ..., n, k = 2, ..., T-1, l = 1, ..., m \tag{3.48}$$

$$\sum_{j=2}^{n} S_i^l S_j^l p_i^{k-1} p_j^k \le S_i^l p_i^{k-1}, \forall i = 2, ..., n, k = 2, ..., T-1, l = 1, ..., m \tag{3.49}$$

Proof. Constraints (3.47) are implied by constraints (3.42), Lemma 3.1 and the fact that the variables u_{ij}^k are integer. To obtain inequalities (3.48), we note that for $k = 2, ..., T-1, l = 1, ..., m$ we can rewrite constraints (3.41) as $\sum_{i=2}^{n} S_i^l p_i^{k-1} \le 1$. Because, for $j = 2, ..., n$, the term $S_j^l p_j^k$ can only take values 0 or 1, we can multiply both sides of the previous inequality by this term, leading to (3.48). Inequalities (3.49) are obtained in a similar way, multiplying both sides of inequality $\sum_{j=2}^{n} S_j^l p_j^k \le 1$ by $S_i^l p_i^{k-1}$. □

The first set of constraints presented presented in Lemma 3.3 states that, for any route l, either there is a node in position 1 (that is, the first node visited by the route is in the first position) or there is a node j in a position k ($k \ge 2$), and such that there is no node in the route in position $k-1$ (later on, we shall show that this implies that node j is the first node in the route). The second constraint is an aggregated version of the first inequalities.

Lemma 3.3. *The following constraints are valid for any feasible solution $\{u_{ij}^k, p_j^k\}$ of the APQ model:*

$$\sum_{j=2}^{n} S_j^l p_j^1 + \sum_{k=2}^{T-1} \sum_{j=2}^{n} S_j^l p_j^k - \sum_{k=2}^{T-1} \sum_{i=2}^{n} \sum_{j=2}^{n} S_i^l S_j^l p_i^{k-1} p_j^k = 1, \forall l = 1, ..., m \tag{3.50}$$

$$\sum_{j=2}^{n} (\sum_{l=1}^{m} S_j^l) p_j^1 + \sum_{k=2}^{T-1} \sum_{j=2}^{n} (\sum_{l=1}^{m} S_j^l) p_j^k - \sum_{k=2}^{T-1} \sum_{i=2}^{n} \sum_{j=2}^{n} SR_{ij} p_i^{k-1} p_j^k = m \tag{3.51}$$

Proof. Since this proof is quite long we give an overview of the sequence of inequalities that shall be proved as valid: i) we start by proving that constraints (3.50) are verified as $\ge$ inequalities; ii) then as a consequence, we conclude that (3.51) is also satisfied as a $\ge$ inequality; iii) in a third step we prove, by a padding argument, that (3.51) is valid and finally, iv) we "backtrack" and prove that (3.50) is valid.

We start by proving that constraints (3.50) are verified as $\ge$ inequalities, that is, the following inequalities are valid

$$\sum_{j=2}^{n} S_j^l p_j^1 + \sum_{k=2}^{T-1} \sum_{j=2}^{n} S_j^l p_j^k - \sum_{k=2}^{T-1} \sum_{i=2}^{n} \sum_{j=2}^{n} S_i^l S_j^l p_i^{k-1} p_j^k \ge 1, \forall l = 1, ..., m \tag{3.52}$$

Consider an inequality (3.52) for a fixed l. First, note that because of constraints (3.41) the term$\sum_{i=2}^{n} S_i^l p_i^1$ only takes value 0 or 1 and, because of constraints (3.48), each term $S_j^l p_j^k - \sum_{i=2}^{n} S_i^l S_j^l p_i^{k-1} p_j^k$, for $j = 2, ..., n, k = 2, ..., T-1$ also only takes values 0 or 1. Therefore, for inequalities (3.52) to be false, we need to have

$$\sum_{j=2}^{n} S_j^l p_j^1 = 0 \tag{3.53}$$

and

$$\sum_{j=2}^{n} \sum_{k=2}^{T-1} S_j^l p_j^k - \sum_{i=2}^{n} \sum_{j=2}^{n} \sum_{k=2}^{T-1} S_i^l S_j^l p_i^{k-1} p_j^k = 0 \tag{3.54}$$

Observe, also, that because of (3.48), that is, $S_j^l p_j^k - \sum_{i=2}^{n} S_i^l S_j^l p_i^{k-1} p_j^k \geq 0$, for $j = 2, ..., n, k = 2, ..., T-1$, for (3.54) to be true we must have

$$\sum_{j=2}^{n} S_j^l p_j^k - \sum_{i=2}^{n} \sum_{j=2}^{n} S_i^l S_j^l p_i^{k-1} p_j^k = 0, \forall k = 2, ..., T-1 \tag{3.55}$$

We can prove that (3.53) and (3.55) imply that

$$\sum_{j=2}^{n} S_j^l p_j^k = 0, \forall k = 2, ..., T-1 \tag{3.56}$$

Before proving this implication, we show that (3.56) leads to a contradiction. To see this, observe that (3.56) implies

$$\sum_{j=2}^{n} \sum_{k=1}^{T-1} S_j^l p_j^k = 0 \tag{3.57}$$

However, equation (3.57) is not valid since each route has at least one client node and for that client, constraint (3.40) states that it has a position assigned to it. This leads to a contradiction showing that (3.53) and (3.55) cannot both be true. To prove that equations (3.53) and (3.55) imply (3.56) we first observe that by combining (3.55) with (3.49), added up for all $i = 2, ..., n$, we obtain $\sum_{j=2}^{n} S_j^l p_j^k - \sum_{i=2}^{n} S_i^l p_i^{k-1} \leq 0, \forall k = 2, ..., T-1$. By transitivity, we obtain $\sum_{j=2}^{n} S_j^l p_j^k - \sum_{i=2}^{n} S_i^l p_i^1 \leq 0, \forall k = 2, ..., T-1$. Combining these last inequalities with (3.53), and by using the fact that the parameters $S_j^l, j = 1, ..., n, l = 1, ..., m$ and the variables $p_j^k, j = 2, ..., n, k = 1, ..., T-1$ only take values 0 or 1, we obtain equations (3.56).

Having shown that constraints (3.52) are valid, we can sum them for $l = 1, ..., m$ to conclude that constraints (3.51) are also valid as $\geq$ inequalities, that is

$$\sum_{j=2}^{n} (\sum_{l=1}^{m} S_j^l) p_j^1 + \sum_{k=2}^{T-1} \sum_{j=2}^{n} (\sum_{l=1}^{m} S_j^l) p_j^k - \sum_{k=2}^{T-1} \sum_{i=2}^{n} \sum_{j=2}^{n} SR_{ij} p_i^{k-1} p_j^k \geq m \tag{3.58}$$

Because of inequalities (3.47), we can also conclude that

$$\sum_{j=2}^{n}(\sum_{l=1}^{m} S_j^l)p_j^1 + \sum_{k=2}^{T-1}\sum_{j=2}^{n}(\sum_{l=1}^{m} S_j^l)p_j^k - \sum_{k=2}^{T-1}\sum_{i=2}^{n}\sum_{j=2}^{n} u_{ij}^k \geq$$
$$\sum_{j=2}^{n}(\sum_{l=1}^{m} S_j^l)p_j^1 + \sum_{k=2}^{T-1}\sum_{j=2}^{n}(\sum_{l=1}^{m} S_j^l)p_j^k - \sum_{k=2}^{T-1}\sum_{i=2}^{n}\sum_{j=2}^{n} SR_{ij}p_i^{k-1}p_j^k \geq m \quad (3.59)$$

From constraints (3.39), for $j = 2,...,n, k = 2,...,T-1$, we know that $\sum_{k=2}^{T-1}\sum_{j=2}^{n}(\sum_{l=1}^{m} S_j^l)p_j^k - \sum_{k=2}^{T-1}\sum_{i=2}^{n}\sum_{j=2}^{n} u_{ij}^k = \sum_{j=2}^{n}\sum_{k=2}^{T-1} u_{1j}^k$, and therefore (3.59) can be rewritten as

$$\sum_{j=2}^{n}(\sum_{l=1}^{m} S_j^l)p_j^1 + \sum_{k=2}^{T-1}\sum_{j=2}^{n} u_{1j}^k \geq \sum_{j=2}^{n}(\sum_{l=1}^{m} S_j^l)p_j^1 + \sum_{k=2}^{T-1}\sum_{j=2}^{n}(\sum_{l=1}^{m} S_j^l)p_j^k - \sum_{k=2}^{T-1}\sum_{i=2}^{n}\sum_{j=2}^{n} SR_{ij}p_i^{k-1}p_j^k \geq m \quad (3.60)$$

Also, from constraints (3.39) for $j = 2,...,n$ and $k = 1$, combined with (3.34), we know that $\sum_{j=2}^{n}(\sum_{l=1}^{m} S_j^l)p_j^1 = \sum_{j=2}^{n} u_{1j}^1$. Therefore, (3.60) can be rewritten as

$$\sum_{j=2}^{n}\sum_{k=1}^{T-1} u_{1j}^k \geq \sum_{j=2}^{n}(\sum_{l=1}^{m} S_j^l)p_j^1 + \sum_{k=2}^{T-1}\sum_{j=2}^{n}(\sum_{l=1}^{m} S_j^l)p_j^k - \sum_{k=2}^{T-1}\sum_{i=2}^{n}\sum_{j=2}^{n} SR_{ij}p_i^{k-1}p_j^k \geq m \quad (3.61)$$

Since constraints (3.45), combined with (3.35), state that $\sum_{j=2}^{n}\sum_{k=1}^{T-1} u_{1j}^k = m$, we can see that constraint (3.61) is written in the form $A \geq B \geq C$, with $A = C$, so constraint (3.58) must be verified as an equality (which is equation (3.51)). Finally, the validity of equality (3.51) and inequalities (3.52) leads to the validity of the equalities (3.50). □

Lemma 3.4. *The following constraints are verified in any feasible solution for the APQ model:*

$$u_{1j}^1 = SR_{1j}p_j^1;\ \forall j = 2,...,n \quad (3.62)$$
$$u_{j1}^T = SR_{j1}p_j^{T-1};\ \forall j = 2,...,n \quad (3.63)$$
$$u_{ij}^k = SR_{ij}p_i^{k-1}p_j^k;\ \forall i,j = 2,...,n;\ \forall k = 2,...,T-1 \quad (3.64)$$
$$u_{1j}^1 \in \{0; SR_{1j}\};\ \forall j = 2,...,n \quad (3.65)$$
$$u_{j1}^T \in \{0; SR_{j1}\};\ \forall j = 2,...,n \quad (3.66)$$
$$u_{ij}^k \in \{0; SR_{ij}\};\ \forall i,j = 2,...,n;\ \forall k = 2,...,T-1 \quad (3.67)$$

Proof. First, observe that constraints (3.65)-(3.67) are directly implied, respectively, by constraints (3.62)-(3.64), because variables p_i^k are binary, for all $i = 2,...,n, k = 1,...,T-1$. Therefore, if we are able to show that (3.62)-(3.64) are valid, the validity of (3.65)-(3.67) is also proved.

Constraints (3.62) are obtained by combining (3.34) and (3.39), for all $j = 2,...,n$ and $k = 1$. Similarly, constraints (3.63) result from combining constraints (3.35) and (3.46) for all $j = 2,...,n$, with $k = T-1$.

The validity of constraints (3.64) is not as straightforward. If we can show that

$$\sum_{i=2}^{n}\sum_{j=2}^{n}\sum_{k=2}^{T-1} u_{ij}^k = \sum_{i=2}^{n}\sum_{j=2}^{n}\sum_{k=2}^{T-1} SR_{ij}p_i^{k-1}p_j^k \tag{3.68}$$

then, by using (3.47), that is, $u_{ij}^k \leq SR_{ij}p_i^{k-1}p_j^k, \forall i,j = 2,...,n, k = 2,...,T-1$, we can conclude the validity of (3.64).

To see that the equality (3.68) is valid, we note that the constraint (3.45) for $j = 1$, combined with (3.35) leads to $\sum_{j=2}^{n}\sum_{k=1}^{T-1} u_{1j}^k = m$. Combining this equality with (3.51) we obtain

$$\sum_{j=2}^{n}\sum_{k=1}^{T-1} u_{1j}^k = \sum_{j=2}^{n}(\sum_{l=1}^{m} S_j^l)p_j^1 + \sum_{k=2}^{T-1}\sum_{j=2}^{n}(\sum_{l=1}^{m} S_j^l)p_j^k - \sum_{k=2}^{T-1}\sum_{i=2}^{n}\sum_{j=2}^{n} SR_{ij}p_i^{k-1}p_j^k \tag{3.69}$$

Considering constraints (3.39) for $k = 1$, combining with (3.34) and summed for all $j = 2,...,n$ leads to $\sum_{j=2}^{n}(\sum_{l=1}^{m} S_j^l)p_j^1 = \sum_{j=2}^{n} u_{1j}^1$. Combining this equality with (3.69) leads to

$$\sum_{j=2}^{n}\sum_{k=2}^{T-1} u_{1j}^k = \sum_{k=2}^{T-1}\sum_{j=2}^{n}(\sum_{l=1}^{m} S_j^l)p_j^k - \sum_{k=2}^{T-1}\sum_{i=2}^{n}\sum_{j=2}^{n} SR_{ij}p_i^{k-1}p_j^k \tag{3.70}$$

Constraints (3.39), summed for $j = 2,...,n, k = 2,...,T-1$, can be rewritten as $\sum_{k=2}^{T-1}\sum_{j=2}^{n}(\sum_{l=1}^{m} S_j^l)p_j^k - \sum_{k=2}^{T-1}\sum_{i=2}^{n}\sum_{j=2}^{n} u_{ij}^k = \sum_{j=2}^{n}\sum_{k=2}^{T-1} u_{1j}^k$. Substituting in (3.70) leads to

$$\sum_{k=2}^{T-1}\sum_{j=2}^{n}(\sum_{l=1}^{m} S_j^l)p_j^k - \sum_{k=2}^{T-1}\sum_{i=2}^{n}\sum_{j=2}^{n} u_{ij}^k = \sum_{k=2}^{T-1}\sum_{j=2}^{n}(\sum_{l=1}^{m} S_j^l)p_j^k - \sum_{k=2}^{T-1}\sum_{i=2}^{n}\sum_{j=2}^{n} SR_{ij}p_i^{k-1}p_j^k \tag{3.71}$$

Cancelling the term $\sum_{k=2}^{T-1}\sum_{j=2}^{n}(\sum_{l=1}^{m} S_j^l)p_j^k$ which is in both sides of the equality leads to the equation (3.68). □

Constraints (3.65)-(3.67) and (3.50) will be used to guarantee that from any solution $\{u_{ij}^k, p_j^k\}$, that is feasible for the APQ model, we can decode a solution defined in the variables $\{z_{ij}^{kl}, p_j^k\}$ that is feasible for the PQ model augmented with the $node - pos$ system. In particular, with these constraints, we can show that each integer variable u_{ij}^k can be decomposed into binary variables z_{ij}^{kl}.

Theorem 3.5. *The APQ model is a valid formulation for the $CTSP$.*

Proof. To prove the validity of the APQ model we show how to transform an aggregated solution $\{u_{ij}^k, p_j^k\}$ into a feasible solution $\{z_{ij}^{kl}, p_j^k\}$ for the PQ model, and with the same objective function value. To transform one solution into the other we use the following relations:

$$z_{ij}^{kl} = 0;\ \forall i,j = 1,...,n;\ SR_{ij} = 0;\ \forall k = 1,...,T;\ \forall l = 1,...,m \tag{3.72}$$

$$z_{ij}^{1l} = 0;\ \forall i = 2,...,n;\ \forall j = 1,...,n;\ \forall l = 1,...,m \tag{3.73}$$

$$z_{ij}^{Tl} = 0;\ \forall i = 1, ..., n;\ \forall j = 2, ..., n;\ \forall l = 1, ..., m \quad (3.74)$$

$$z_{1j}^{1l} = S_j^l p_j^1 = S_j^l \frac{u_{1j}^1}{SR_{1j}};\ \forall j = 2, ..., n;\ \forall l = 1, ..., m \quad (3.75)$$

$$z_{j1}^{Tl} = S_j^l p_j^{T-1} = S_j^l \frac{u_{j1}^T}{SR_{j1}};\ \forall j = 2, ..., n;\ \forall l = 1, ..., m \quad (3.76)$$

$$z_{ij}^{kl} = S_i^l S_j^l p_i^{k-1} p_j^k = S_i^l S_j^l \frac{u_{ij}^k}{SR_{ij}};\ \forall i, j = 2, ..., n;\ \forall k = 2, ..., T-1;\ \forall l = 1, ..., m \quad (3.77)$$

$$z_{1j}^{kl} = S_j^l p_j^k - \sum_{i=2}^{n} z_{ij}^{kl};\ \forall j = 2, ..., n;\ \forall k = 2, ..., T-1;\ \forall l = 1, ..., m \quad (3.78)$$

$$z_{j1}^{kl} = S_j^l p_j^{k-1} - \sum_{i=2}^{n} z_{ji}^{kl};\ \forall j = 2, ..., n;\ \forall k = 2, ..., T-1;\ \forall l = 1, ..., m \quad (3.79)$$

The value of the variables z_{1j}^{kl} and z_{i1}^{kl} (for $i, j = 2, ..., n, k = 2, ..., T-1, l = 1, ..., m$) given in relations (3.78) and (3.79) are obtained from the value, previously obtained, of other variables (namely, variables, $z_{ij}^{kl}, i, j = 2, ..., n$). The reason why we use the relations (3.78) and (3.79) is that the variables associated to the arcs $(1, j)$ and $(i, 1)$, $i, j = 2, ..., n$, in positions $k = 2, ..., T-1$ are not included in Lemma 3.4. These variables are used in situations where we have routes starting later or finishing earlier than the longest route. Constraints (3.78) state that the variable z_{1j}^{kl} takes value 1 if and only if node j occupies position k in route l, and there are not other arcs entering node j in position k in route l. The relation defining variables z_{i1}^{kl} can be interpreted in a similar, symmetrical, way.

Observe that after proving that the variables defined in this way are binary, the proof that constraints (3.31) for $j = 2, ..., n, k = 2, ..., T-1, l = 1, ..., m$ is immediate because the relations (3.78) are the constraints (3.31) rewritten in a different equivalent way. A similar observation holds for constraints (3.79) for $j = 2, ..., n, k = 2, ..., T-1, l = 1, ..., m$ since these relations are constraints (3.32).

Finally, we observe that the variables p_j^k are the same in the two solutions.

The two solutions have the same objective function value:

This follows immediately from relation (3.33) between the variables of the two models, that is, $u_{ij}^k = \sum_{l=1}^m z_{ij}^{kl}, \forall i, j = 1, ..., n, \forall k = 1, ..., T$. To prove that (3.33) holds, consider the following five cases:

- **Arcs** $(i, j), i, j = 2, ..., n; k = 2, ..., T-1$**:** summing constraints (3.77) for $l = 1$ to m, we obtain $\sum_{l=1}^m z_{ij}^{kl} = u_{ij}^k$.

- **Arcs** $(1, j)$**,** $j = 2, ..., n$ **in position** 1**:** summing constraints (3.75) for $l = 1$ to m, we obtain $\sum_{l=1}^m z_{1j}^{1l} = (\sum_{l=1}^m S_j^l)\frac{u_{1j}^1}{SR_{1j}} = SR_{1j}\frac{u_{1j}^1}{SR_{1j}} = u_{1j}^1$;

- **Arcs** $(1, j)$**,** $j = 2, ..., n$ **in position** $k, k = 2, ..., T-1$**:** summing constraints (3.78) for $l = 1$ to m and combining with constraints (3.39) for the same j and the same k and (3.77) we obtain $\sum_{l=1}^m z_{1j}^{kl} = \sum_{l=1}^m S_j^l p_j^k - \sum_{l=1}^m \sum_{i=2}^n z_{ij}^{kl} = \sum_{i=1}^n u_{ij}^k - \sum_{i=2}^n u_{ij}^k = u_{1j}^k$;.

- **Arcs** $(j, 1)$**,** $j = 2, ..., n$ **in position** T**:** summing constraints (3.76) for $l = 1$ to m, we obtain $\sum_{l=1}^m z_{j1}^{Tl} = u_{j1}^T$.

- **Arcs** $(j,1), j = 2,...,n$ **in position** $k, k = 1,...,T-1$**:** summing constraints (3.79) for $l = 1$ to m and combining with (3.46) and (3.77) we conclude that $\sum_{l=1}^{m} z_{j1}^{kl} = u_{j1}^{k}$.

By combining the five equalities obtained above we conclude that $\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{T}\sum_{l=1}^{m} C_{ij} z_{ij}^{kl} = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{T} C_{ij} u_{ij}^{k}$ and thus, the two solutions have the same cost.

Feasibility of the solution$\{z_{ij}^{kl}, p_j^k\}$**:**

First, we show that the variables z_{ij}^{kl} defined by (3.72)-(3.79), are binary. This follows from the fact that the variables from the APQ model are integer and satisfy the condition of the Lemma 3.4. To see this, observe that:

- Variables z_{1j}^{1l} and $z_{i1}^{Tl}, i, j = 2,...,n, l = 1,...,m$ are binary since they result from (3.75) and (3.76), respectively and the variables p_j^k are binary.

- Variables $z_{ij}^{kl};\ i, j = 2,...,n;\ k = 2,...,T-1;\ l = 1,...,m$ are binary since they result from (3.77) and because the variables u_{ij}^{k} either takes value 0 or SR_{ij}, as stated in Lemma 3.4.

- Variables $z_{1j}^{kl},\ k = 2,...,T-1, l = 1,...,m$ are defined as $z_{1j}^{kl} = S_j^l p_j^k - \sum_{i=2}^{n} z_{ij}^{kl}$ by (3.78). Using (3.77) for the variables z_{ij}^{kl}, with $i, j = 2,...,n$, we obtain $z_{1j}^{kl} = S_j^l p_j^k - \sum_{i=2}^{n} S_i^l S_j^l p_i^{k-1} p_j^k$, which only takes values 0 or 1 due to constraints (3.48).

- Similarly to the previous case, by using (3.77), (3.79) and (3.49), we conclude that variables $z_{i1}^{kl},\ k = 2,...,T-1, l = 1,...,m$ are binary.

We show next that the solution in variables $\{z_{ij}^{kl}\}$ also satisfies the remaining constraints of the PQ model:

- **PQ constraints (3.27):** We obtain constraints (3.27) for $j = 2,...,n$, $k = 1$ and $l = 1,...,m$ by summing expressions (3.75) and (3.73), for the same j and the same l, with $k = 1$ and $i = 2,...,n$, and then combining with constraints (3.32), for the same values of j, k and l. We obtain constraints (3.27) for $j = 2,...,n$, $k = 2,...,T-1$ and $l = 1,...,m$ by summing $\sum_{i=2}^{n} z_{ij}^{kl}$ on both sides of equation (3.78) and then combining with constraints (3.32), for the same values of j, k and l.

- **Assignment constraints (3.22):** We obtain constraints (3.22) for $j = 2,...,n$ and $l = 1,...,m$ by multiplying equations (3.40) by the corresponding S_j^l and then combining with (3.31) and (3.74) for the same j and the same l. For $j = 1$ and a given $l = 1,...,m$, the corresponding assignment constraint results from constraints (3.50). To see this, note that the latter can be rewritten as

  $\sum_{j=2}^{n} S_j^l p_j^{T-1} + \sum_{i=2}^{n}\sum_{k=2}^{T-1} S_i^l p_i^{k-1} - \sum_{k=2}^{T-1}\sum_{i=2}^{n}\sum_{j=2}^{n} S_i^l S_j^l p_i^{k-1} p_j^k = 1, l = 1,...,m$

  Combining the previous equality with (3.76) to replace the first summation term and (3.77) to replace the third summation term leads to

  $\sum_{j=2}^{n} z_{j1}^{Tl} + \sum_{i=2}^{n}\sum_{k=2}^{T-1} S_i^l p_i^{k-1} - \sum_{k=2}^{T-1}\sum_{i=2}^{n}\sum_{j=2}^{n} z_{ij}^{kl} = 1, l = 1,...,m$

  Finally, combining this equality with (3.79) to replace the second summation terms and rearranging, we obtain (3.22) for $j = 1$ and $l = 1,...,m$.

- $node - pos$ **system:** Constraints (3.40) and (3.41) are the same as constraints (3.28) and (3.29).

□

The proof above raises the point of checking why a similar proof does not also hold for fractional solutions (and as a consequence, the LP values of the PQ and the APQ models would be equal). To illustrate why not, consider a small example with 5 nodes. Suppose that node 1 is the depot, nodes $\{2,3,4\}$ belong to route 1 and $\{2,3,5\}$ belong to route 2. Nodes 2 and 3 require total consistency. Figure 2 shows a feasible solution for the linear relaxation of the APQ

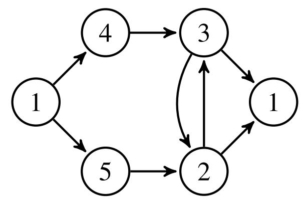

Figure 2: A feasible solution for the LP relaxation for the APQ model.

model for this example. For an easier interpretation of this feasible solution, we have duplicated the depot. For this solution, the values of the relevant variables are as follows: $u_{14}^1 = u_{15}^1 = u_{43}^2 = u_{52}^2 = u_{32}^3 = u_{23}^3 = u_{21}^4 = u_{31}^4 = 1$, $p_4^1 = p_5^1 = 1$ and $p_3^2 = p_2^2 = p_2^3 = p_3^3 = 0.5$.
Since $S_4^1 = S_3^1 = S_5^2 = S_2^2 = 1$, the equalities (3.77) lead to $z_{43}^{21} = z_{52}^{22} = 1$ violating the linking constraints (3.31) because the corresponding position variables, p_2^2 and p_3^2, are both equal to 0.5.
This example illustrates the relevance of Lemma 3.4 for the proof of Theorem 3.5. In the above solution, although the arc variables are integer, they do not satisfy the constraints given by Lemma 3.4. (e.g, $u_{23}^3 = 1$, and $SR_{23} = 2$; Lemma 3.4 states that either $u_{23}^3 = 0$ or $u_{23}^3 = 2$) because some of the positional variables are not integer.
We conclude this section by presenting the following inequalities, whose validity is easily proved, and that strengthen the LP relaxation of the APQ model.

$$\sum_{k=1}^{T} u_{ij}^k \leq SR_{ij};\ \forall i,j = 1,...,n; \tag{3.80}$$

Two observations can be made about about inequalities (3.80). First, the LP relaxation of the APQ model satisfies a weaker version of these inequalities, namely $\sum_{k=1}^{T} u_{ij}^k \leq \sum_{l=1}^{m} S_j^l;\ \forall i,j = 1,...,n;$. This indicates that the new inequalities (3.80) are expected to be more effective for instances with more routes or more complicated consistency configurations. Second, the new inequalities (3.80) are implied by the linear programming relaxation of the PQ model. To see this, we first claim that the inequalities

$$\sum_{k=1}^{T} z_{ij}^{kl} \leq S_i^l S_j^l;\ \forall i,j = 1,...,n; l = 1,...,m \tag{3.81}$$

are implied by the LP relaxation of the PQ model. Then, by adding these inequalities for all $l = 1,...,m$, and using (3.33) we obtain (3.80). To see that the inequalities (3.81) are implied by the linear programming relaxation of the PQ

model we observe that i) the inequalities $\sum_{k=1}^{T} z_{ij}^{kl} \leq S_j^l;\ \forall i,j=1,...,n; l=1,...,m$ are implied by (3.22) and ii) the $\sum_{k=1}^{T} z_{ij}^{kl} \leq S_i^l;\ \forall i,j=1,...,n; l=1,...,m$ are implied by inequalities (3.21), which in turn are redundant in the PQ model, due to constraints (3.22) and (3.27). Finally, the result follows since the minimum of S_i^l and S_j^l equals $S_i^l S_j^l$.

Preliminary computational tests show that substituting constraints (3.42) for (3.80) in the APQ model leads to a model with a slightly tighter LP bound and, in general, smaller CPU times to obtain the optimal integer solution. However, in order not to have too many models in this study, we maintain the same designation, APQ, to the model APQ with constraints (3.80), since in the remainder of the text we will focus only on this augmented model. Also, it is easy to see that the observation made before concerning the LP relaxation of the models PQ and APQ, still holds with this stronger version.

The following trivial result relates the linear programming relaxations of the models PQ, and APQ.

Theorem 3.6. $LP(PQ) \geq LP(APQ)$.

It is easy to find instances for which the inequality above is strict. Also, in the discussion of the next section, concerning an aggregated $FGG3$ model, we consider the model including the stronger set of inequalities.

3.2.2 The Aggregated $FGG3$ Models

We can try to derive an aggregated version of the model $FGG3$, denoted in the following by $AFGG3$. To obtain this model, we only need to substitute constraints (3.38) in APQ, for the following constraints:

$$\sum_{i=1}^{n} \sum_{k=1}^{T} u_{ji}^k = \sum_{l=1}^{m} S_j^l;\ \forall j=1,...,n \tag{3.82}$$

$$\sum_{i=1}^{n} \sum_{k=1}^{T} k u_{ji}^k - \sum_{i=1}^{n} \sum_{k=1}^{T} k u_{ij}^k = \sum_{l=1}^{m} S_j^l;\ \forall j=2,...,n \tag{3.83}$$

Constraints (3.82) are obtained from the sum of constraints (3.21), whereas constraints (3.83) are obtained from the sum of constraints (3.23). We show next that these models are in general not valid. The reason why is that the relations (3.65), (3.66) and (3.67) are not necessarily valid and as a consequence, an integer solution of the AFFG3 model may not be decodable into a feasible integer solution of the problem.

We start by giving some intuition on why these relations may not be valid by focusing on the main difference between the two classes of models, the PQ class and the FGG3 class, namely between constraints (3.38) of the first class, and (3.83) of the second one. The relevant part for this discussion is given by the summation terms involved in the left-hand side of constraints (3.83).

To illustrate, consider the solution, described in Figure 3, that is optimal for the $AFGG3$ model, for a given instance. Node 1 is the depot, nodes $\{2,3,4\}$ must be visited in route 1, and nodes $\{2,3,4,5,6,7\}$ must be visited in route 2. Nodes 2, 3 and 4 are visited in both routes and require total consistency. Consider the arc $(3,4)$ for which we have $SR_{34} = 2$. A decodable solution satisfying equations (3.67) would have $u_{34}^k = 2$ for a given value of k and $u_{34}^k = 0$ for the remaining values of k. However, there are other solutions satisfying the flow conservation constraints (3.83) one of them being the depicted solution, in which $u_{34}^6 = 1$. Observe, also, that to guarantee the validity of the indegree constraint of node 4, the solution also has $u_{64}^{6} = 1$. Finally, observe, also, that $u_{32}^{2} = 1$, that is, there is a return arc to node 2 to guarantee the validity of the outdegree constraint for node 3. This situation is quite common in many of the undecodable solutions provided by the AFGG3 model.

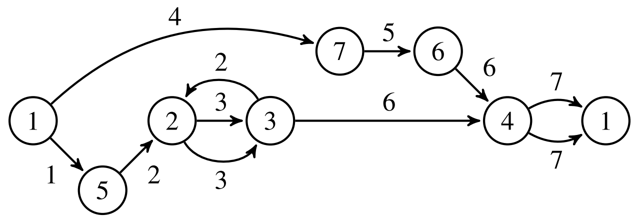

Figure 3: A feasible solution for the $AFGG3$ model

The counterexample given to show that the $AFGG3$ model is not valid is based on an example with two routes not containing the same number of nodes. For all the instances we have tried (even with more than 2 routes) and routes of the same size the $AFGG3$ has produced the same integer optimal solution as the APQ model. That is, it remains to be proved, or not, whether the $AFGG3$ is a valid model in this case. However, since the $AFGG3$ model is far from being competitive, we leave this as an open question.

One consequence of the discussion on the $AFGG3$ model is that defining a valid aggregated version of the SCF model is very unlikely. To see this, we observe that if such a valid model would exist, then, by using the same relation between the SCF model and the $wFGG3$ model, we would obtain a weaker version of the $AFGG3$ model. However, this version would be weaker than the $AFGG3$ model and would not be valid since the $AFGG3$ model is not. To formalise this reasoning, we start by defining an aggregated version of the $wFGG3$ model, the $AwFGG3$ model, which is obtained from the $AFGG3$ model by using a weaker version of constraints (3.39).

$$\sum_{i=1}^{n} \sum_{k=1}^{T-1} k u_{ij}^{k} = (\sum_{l=1}^{m} S_j^l) \sum_{k=1}^{T-1} k p_j^k;\ \forall j = 2, ..., n \tag{3.84}$$

Using a transformation similar to the one to prove the equivalence between the LP relaxations of the SCF and $wFGG3$ model (see section 3.1.2), we can prove the equivalence between the LP relaxation of the $AwFGG3$ model and the LP relaxation of an aggregated version, denoted by $ASCF$, of the SCF model. This equivalence also proves that the integer version of the two models are either both valid or both not valid. This aggregated version of the SCF model is described in the Appendix together with the equivalence with the $AwFGG3$ model. We also provide an optimal solution of the $ASCF$ model, showing that it is not valid, also for the case where all routes have the same size.

3.3 Enhanced formulations

In this section we describe an enhancement of the PQ and APQ models, which explores the fact that, depending on the length of a given route, the positions of the first and last nodes can be bounded. In order to accomplish this idea we add to the PQ model the following variable fixing constraints:

$$z_{1j}^{kl} = 0, \forall j = 2, ..., n, l = 1, ..., m, k = T - N^l + 2, ..., T \tag{3.85}$$

$$z_{i1}^{kl} = 0, \forall i = 2, ..., n, l = 1, ..., m, k = 1, ..., N^l - 1 \tag{3.86}$$

As an example consider two routes, one with 4 nodes (including the depot) and another with 3 nodes. Constraints (3.85) state that, for the longer route, arcs leaving the depot in position greater than 1 cannot be included in the solution. For the second route, arcs leaving the depot in position greater than 2 cannot be included in the solution. Constraints (3.86) have a similar interpretation, for arcs converging into the depot. We denote by PQ_0 the model PQ with these extra constraints. This enhancement can be used in combination with any other time-dependent model described before. However, the computational results indicate that, of the disaggregated models presented until now, the only model where a positive impact on its efficiency was obtained is the PQ model and thus, we have focused on enhancing this model.

For the aggregated model, APQ, we have considered the following constraints:

$$u_{1j}^{k} = 0, \forall j = 2, ..., n, k = T - a + 2, ..., T \tag{3.87}$$

$$u_{i1}^{k} = 0, \forall i = 2, ..., n, k = 1, ..., a - 1 \tag{3.88}$$

where $a = \min\{N^l : l = 1, ..., m\}$.

We denote by APQ_0 the model APQ with these extra constraints. By using the linking constraints (3.33) we can see that constraints (3.87) and (3.88) are implied by (3.85) and (3.86) leading to

Theorem 3.7. $LP(PQ_0) \geq LP(APQ_0)$.

Observe, however, that constraints (3.87) and (3.88) are weaker than (3.85) and (3.86). In fact, the former only forbid arcs entering or leaving the depot in positions where none of the routes can have such arcs. In the example previously mentioned, with one route with 4 nodes and the other with 3, arcs leaving the depot would be forbidden only in positions greater than 2. To further strengthen the LP relaxation of the APQ model, we could write, for each position, constraints defining an upper bound for the arcs entering or leaving the depot. Revisiting the same example, we could write the inequality $\sum_{j=2}^{n} u_{1j}^2 \leq 1$, which would be valid because the largest route must start in position 1, and only the smaller route can start either in position 1 or 2. The results of some preliminary tests showed that the addition of such constraints did not improve the linear programming relaxation bounds and for this reason we have omitted them from the text.

Finally, we present in Figure 4 the relations that have been established between the LP relaxations of all the valid models in this study. In the figure, an arrow going from model A to model B means that the optimal value of the linear programming relaxation for model B is greater or equal to the optimal value of the linear programming relaxation for model A. A double arrow between two models, A and B, indicates that the two models are equivalent in terms of the linear programming relaxation. Note that valid relations between models that result from transitivity are not represented in this figure. For instance, because we have determined that model PQ has a stronger linear programming relaxation

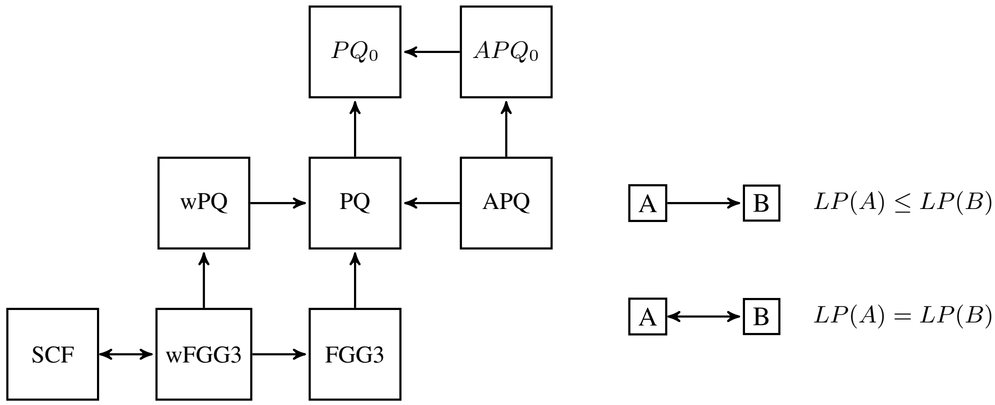

Figure 4: Relations between formulations for the *Consistent Travelling Salesman Problem*

than model $FGG3$, and model $FGG3$ has a stronger linear programming relaxation than model $wFGG3$, then we can conclude than the linear programming relaxation for model PQ is stronger than the linear programming relaxation for model $wFGG3$, due to transitivity. However, because this relation results from transitivity, an arrow from model $wFGG3$ to model PQ is not shown.

4 Computational Experiment

A computational experiment has been carried out to compare the performance of the models presented in section 3. For this purpose we have implemented the models, using the Concert Technology from CPLEX, version 12.9.0.0., in a computer with an Intel® Core™ i7-4790 CPU @ 3.60GHz processor, with 8GB of RAM. We considered the default values for the CPLEX parameters except for the time limit, which we set to 10800 seconds.

For this comparison we use several performance indicators: the *linear programming gap* (LPg), the computational time needed to solve the linear programming relaxation (LPt), measured in seconds; the computational time needed to solve the model ($Intt$), also measured in seconds and the number of test instances that were solved to optimality within the time limit ($\#$). The *linear programming gap* (LPg) is computed as follows:

$$LPg = \frac{best\ value - LP\ value}{best\ value} * 100 \tag{4.1}$$

where *best value* is the value of the best feasible solution, optimal or not, found among all the models, whereas the *LP value* is the optimal value of the linear programming relaxation of the model under consideration. These indicators will be used in a set of preliminary tests to analyse all the models proposed in Section 3, in order to evaluate which are the most efficient for a more in-depth analysis with an additional set of tests.

For this detailed analysis, we also assess the number of processed branch-and-bound nodes (BB) and the *value gap* (Vg), which compares the best feasible solution that each model has found in the time limit (*model value*) with the *best value* (the value of the best feasible solution found among all the models), as given by:

$$Vg = \frac{model\ value - best\ value}{best\ value} * 100 \tag{4.2}$$

In order to keep the results in the tables clearer, we present the measures $Intt$ and BB only for the instances that a given model has been able to solve to optimality, within the time limit. All the unsolved cases are marked with "$-$". Regarding the Vg values, three cases need to be distinguished. One case is when the model is not able to solve the instance to optimality, but at least one feasible solution is found within the time limit. The other two cases, refer to when no feasible solution is found within the time limit (thus making impossible the computation of this gap), also marked with a "$-$", and to when the model is able to solve the instance to optimality, marked with a "$*$". When computing the averages for each of the selected measures, cases marked with "$-$" or "$*$" are ignored.

We observe that comparing directly the averages, especially for the integer times ($Intt$), might be a bit misleading since the number of instances solved differs from model to model, and this might penalise models that are able to solve more complicated instances (which tend to require more time). Thus, in the more in-depth analysis, and to provide a more fair comparison between the models, we will also compute the averages, denoted by *all-solved*, which consider only the instances that were solved by all the models that are being compared.

Several instances were generated considering the application in healthcare services, introduced in Section 2. Thus, the arc costs of these instances were randomly generated, following a Uniform distribution, with values ranging between 10 and 14. If the matrix is asymmetric, all the costs C_{ij}, with $i \neq j$, are randomly generated. If the matrix is symmetric, we generate the costs C_{ij} for $i < j$, and set $C_{ji} = C_{ij}$. Following this procedure, we have generated cost matrices $medC$, where C is the number of client nodes in the instance (that is, an instance using matrix $medC$ has a total of $C + 1$ nodes, C clients and the depot). Matrices $med32$ and $med40$ are asymmetrical, whereas matrices $med36$ and $med44$ are symmetrical. The other instances were adapted from instances known from the literature, namely from the asymmetrical matrices $ftv33$, $ftv35$, $ftv38$ and $ftv44$, taken from the TSPLIB (http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html).

Concerning the parameters defining the required consistency, besides considering configurations fitting the healthcare application we also consider and analyse different configurations that may arise in other type of applications. Therefore, the computational experiment is divided in three subsections.

In Section 4.1, we compare all the models presented in Section 3, using a small set of test instances. The models with the best performances in this preliminary experiment are selected and further compared in the next two subsections. Section 4.2 refers to the healthcare application and the models are evaluated with a set of test instances that are generated with characteristics from the application. In Section 4.3 we analyse the effect of considering different and more general consistency configurations in the performance of the models. This experiment is also used to assess the two enhanced models proposed in Section 3.3. All the test instances that were generated for this study can be accessed online (https://sites.google.com/view/constsp/home).

4.1 Preliminary results

In this section we compare the models proposed and studied in Section 3. For this experiment we consider 2 routes, that the number of consistent nodes can be either 12.5% or 25% of the total number of client nodes, and that the free nodes are distributed as equally as possible between the two routes. For this experiment, we use cost matrices $med32$, $med36$,

$med40$, $med44$, $ftv33$ and $ftv35$, leading to a total of 12 test instances. For the med instances, the two routes can be interpreted as the schedules done by a doctor and a nurse.

Table 3: Preliminary results for all the models

	SCF			wFGG3			FGG3			wPQ			PQ			APQ		
	LPg	LPt	Intt	LPg	LPt	Intt	LPg	LPt	Intt	LPg	LPt	Intt	LPg	LPt	Intt	LPg	LPt	Intt
med32-12.5	0.00	0	0	0.00	0	601	0.00	0	17	0.00	1	2	0.00	1	0	0.00	0	0
med32-25	0.00	0	100	0.00	0	96	0.00	0	9941	0.00	1	54	0.00	2	18	0.00	1	11
med36-12.5	0.88	0	2223	0.88	0	-	0.88	0	-	0.88	1	7769	0.88	1	2793	0.92	1	1655
med36-25	0.61	0	-	0.61	0	-	0.61	0	-	0.61	1	-	0.61	2	8486	0.63	1	-
med40-12.5	0.00	0	162	0.00	0	-	0.00	0	-	0.00	2	28	0.00	2	3	0.21	1	44
med40-25	0.00	0	-	0.00	0	-	0.00	0	-	0.00	2	3002	0.00	5	80	0.00	2	225
med44-12.5	0.18	0	206	0.18	0	9094	0.18	0	-	0.18	2	36	0.18	5	270	0.18	2	90
med44-25	0.16	0	-	0.16	0	-	0.16	1	-	0.16	5	-	0.16	8	-	0.16	4	7998
ftv33-12.5	12.96	0	16	12.96	0	-	12.96	0	-	12.46	0	1036	12.25	0	482	14.54	0	481
ftv33-25	14.73	0	318	14.73	0	5802	14.65	0	8758	14.22	0	3548	13.76	1	1695	14.71	1	1229
ftv35-12.5	10.46	0	77	10.46	0	7458	10.36	0	4224	10.06	0	4137	9.89	0	488	10.68	0	1244
ftv35-25	13.86	0	2197	13.86	0	-	13.62	0	-	13.28	1	-	12.66	1	1960	14.65	1	7580
Average	4.49	0	589	4.49	0	4610	4.45	0	5735	4.32	1	2179	4.20	2	1480	4.72	1	1869

The results in Table 3 empirically confirm the theoretical relations between the models presented in Section 3. In terms of the computational time needed to obtain the integer solution, the worst performance is by the models $wFGG3$ and $FGG3$, with these models only being able to solve, respectively, 5 and 4 instances, despite the corresponding very small values for LPt. With respect to the computational times to obtain the optimal integer solutions ($Intt$), the model with the smallest average time is clearly the SCF model, followed by PQ and APQ model. However, the PQ and APQ models are able to solve 11 instances to optimality, whereas the SCF model only solves 9 instances. In fact, within the given time limit, the APQ model was the only one able to solve the med44-25 instance and the PQ is the only model that solves the med36-25 instance. A comparison between the wPQ model and the PQ model indicates that using the less compact set of consistency constraints (3.26) of the PQ model compensates over using the more compact set (3.24) of the wPQ model, since the wPQ model only solves 9 of the 12 instances, and needs, in average, more time to solve them.

It is also interesting to compare the results for the two sets of instances, the med instances and the ftv instances. The linear programming gaps, LPg, are much smaller for the med instances when compared with the ones of the ftv instances. However, the med instances are much more difficult to solve. This might be explained by the fact that the arc costs in these instances are very similar. This also might explain getting in all these cases the same Lpg value for all models.

Models SCF, PQ and APQ were selected to be evaluated in more detail in the next subsections, with an additional set of test instances.

4.2 The healthcare application experiment

For this experiment we compare models SCF, PQ and APQ using a set of instances that were generated with characteristics based on the healthcare application. In this study we use the same matrices $med32$, $med36$, $med40$ and $med44$ as used in the preliminary experiment, and besides instances with 2 routes (one doctor and one nurse), we will consider instances with 3 routes (two doctors and one nurse) and 5 routes (three doctors and two nurses). The number of consistent nodes can be 12.5% or 25% of the total number of client nodes and the free nodes are distributed as equally as possible between the routes. The distribution of consistent nodes per route depends on the number of routes as well as the percentage of consistent nodes. For $m = 3$ and $m = 5$ we also consider a consistent node to represent a meeting for all the healthcare professionals. Additionally, whenever the number of consistent nodes is 25% of the client nodes, we consider a consistent node to represent a meeting of the doctors and whenever $m = 5$ another one representing a meeting of the two nurses. The consistent nodes that do not represent meetings are distributed between the pairs of doctor and nurse routes according to Table 4, which presents the instances' description.

Table 4: Instance description

	Route correspondence	Consistent nodes (%)	Distribution of consistent nodes per route
$m = 2$	doctor, nurse	12.5%	all in all
		25%	all in all
$m = 3$	doctor1, doctor2, nurse	12.5%	1 in all the routes and the remaining equally distributed between doctor1/nurse and doctor2/nurse
		25%	1 in all the routes, 1 in the two doctor routes and the remaining equally distributed between doctor1/nurse and doctor2/nurse
$m = 5$	doctor1, doctor2, doctor3 nurse1, nurse2	12.5%	1 in all the routes and the remaining: 33% to doctor1/nurse1, 33% to doctor2/nurse2, 17% to doctor3/nurse1, 17% to doctor3/nurse2
		25%	1 in all the routes, 1 in the three doctor routes and the remaining: 33% to doctor2/nurse2, 17% to doctor3/nurse1, 17% to doctor3/nurse2

We have not considered the case with $m = 4$ since most of the consistent nodes would be distributed among the pairs $doctor1/nurse1$ and $doctor2/nurse2$, which, in general, would be working independently from each other. Due to the independence between the two pairs of workers, this situation corresponds to two independent problems with $m = 2$. In this experiment we will consider a total of 24 test instances.

Table 5: Application experiment results

		SCF						PQ						APQ					
	Instance	Vg	LPg	LPt	Intt	BB	#	Vg	LPg	LPt	Intt	BB	#	Vg	LPg	LPt	Intt	BB	#
m=2	med32-12.5	*	0.00	0	0	0		*	0.00	1	0	0		*	0.00	0	0	0	
	med32-25	*	0.00	0	99	19778		*	0.00	2	18	139		*	0.00	1	11	58	
	med36-12.5	*	0.88	0	2204	608973		*	0.88	1	2793	21347		*	0.92	1	1655	7863	
	med36-25	-	0.61	0	-	-		*	0.61	2	8486	32755		0.21	0.63	1	-	-	
	med40-12.5	*	0.00	0	163	90070		*	0.00	2	3	0		*	0.21	1	44	328	
	med40-25	0.19	0.00	0	-	-		*	0.00	5	80	191		*	0.00	2	225	2431	
	med44-12.5	*	0.18	0	204	31402		*	0.18	5	270	570		*	0.18	2	90	580	
	med44-25	-	0.16	0	-	-		0.18	0.16	8	-	-		*	0.16	4	7998	25237	
	Average	0.19	0.23	0	534	150045	5(8)	0.18	0.23	3	1664	7857	7(8)	0.21	0.26	2	1432	5214	7(8)
	All-solved				534	150045	5(8)				617	4411	5(8)				360	1766	5(8)
m=3	med32-12.5	*	0.25	0	1	381		*	0.24	0	2	0		*	0.46	0	2	19	
	med32-25	*	0.67	0	108	38438		*	0.64	1	124	2542		*	1.11	0	779	19186	
	med36-12.5	*	0.86	0	352	101068		*	0.85	1	130	2541		*	1.84	0	1075	10052	
	med36-25	-	1.48	0	-	-		*	1.47	1	1231	9517		0.00	1.56	1	-	-	
	med40-12.5	*	0.00	0	125	54197		*	0.00	1	11	295		*	0.00	1	18	232	
	med40-25	-	1.07	0	-	-		*	0.98	4	2569	18052		0.18	1.28	1	-	-	
	med44-12.5	*	0.69	0	6943	1480296		*	0.69	2	1809	15412		*	0.87	1	7290	42469	
	med44-25	-	1.16	0	-	-		0.50	1.14	4	-	-		0.33	1.30	1	-	-	
	Average	-	0.77	0	1506	334876	5(8)	0.5	0.75	2	839	6908	7(8)	0.17	1.05	1	1833	14392	5(8)
	All-solved				1506	334876	5(8)				415	4158	5(8)				1833	14392	5(8)
m=5	med32-12.5	*	0.88	0	1	1073		*	0.58	0	1	4		*	0.94	0	1	15	
	med32-25	*	1.87	0	310	80226		*	1.37	0	1	475		*	2.71	0	3	140	
	med36-12.5	*	1.00	0	32	12925		*	0.75	0	1	113		*	2.06	0	3	100	
	med36-25	0.00	2.19	0	-	-		*	2.00	0	355	8999		*	3.70	0	93	3790	
	med40-12.5	*	1.05	0	11	3291		*	0.76	0	2	77		*	1.61	0	3	39	
	med40-25	-	1.90	1	-	-		*	1.39	1	937	24964		*	3.18	0	543	14701	
	med44-12.5	*	1.57	0	94	30071		*	1.32	1	21	696		*	2.10	0	13	651	
	med44-25	1.21	1.74	1	-	-		*	1.62	1	1055	11971		*	2.70	1	1417	21571	
	Average	0.61	1.53	0	90	25517	5(8)	*	1.22	0	297	5912	8(8)	*	2.38	0	260	5126	8(8)
	All-solved				90	25517	5(8)				5	273	5(8)				5	189	5(8)
Average		0.47	0.88	0	710	170146	15(24)	0.34	0.73	2	905	6848	22(24)	0.18	1.23	1	1115	7685	20(24)
All-solved					710	170146	15(24)				346	2947	15(24)				732	5449	15(24)

Table 5 presents the results for this study. We can see that, as expected, the linear programming gap (LPg) is smaller for the PQ model. The difference to the other models is not very significant because we are considering instances with small and similar costs. Also, and as expected, the computational times needed to solve the LP relaxation of the SCF and APQ model are also smaller than the ones needed to solve the LP relaxation of the PQ model. This difference becomes more significant for the larger instances. The results from this experiment allow us to illustrate the importance of using the *all-solved* averages to compare the models. For $m = 5$ we get an average $Intt$ value of 90, 297 and 260 for models SCF, PQ and APQ, respectively. Even if we take into consideration the number of instances solved by each method, that is, 5, 8, 8, instances solved, respectively, we could get the wrong idea that SCF is faster despite solving less instances. However, if for each model we compute the average $Intt$ value for the 5 instances the SCF model was able to solve, we obtain the averages of 90, 5 and 5. This means that SCF was not faster, it just solved the easier instances. In general, for $m = 2$ the APQ model needs, in average, less computational time to solve the problem to optimality. For $m = 3$ the PQ model is, in average, the fastest model, as well as the one that solves more instances, but there are still a couple of instances where the APQ model slightly outperforms the PQ model. For $m = 5$, models PQ and APQ have the same average $Intt$ value (which is lower than the one needed by the SCF model). These results seem to contradict the motivation for deriving the aggregated model, namely that the greater the number of routes, the greater the improvement given by the aggregated model. However, for the problem under study, the gain is also measured by the percentage of nodes that are consistent as well the number of routes they are in. For an application where most of the consistencies are defined by pairs of nodes, the proportion in the reduction of the size of the model is greater for the case with $m = 2$. In fact, this observation has motivated the study presented in the next subsection.

Table 6: Application experiment results - averages by proportion of consistent nodes

		SCF						PQ						APQ					
	Case	Vg	LPg	LPt	Intt	BB	#	Vg	LPg	LPt	Intt	BB	#	Vg	LPg	LPt	Intt	BB	#
m=2	12.5%	-	0.27	0	643	182611	4(4)	*	0.27	2	1183	5479	4(4)	*	0.33	1	447	2193	4(4)
	25%	0.19	0.19	0	99	19778	1(4)	0.18	0.19	4	2861	11028	3(4)	0.21	0.20	2	2745	9242	3(4)
	Average	0.19	0.23	0	534	150045	5(8)	0.18	0.23	3	1664	7857	7(8)	0.21	0.26	2	1432	5124	7(8)
	All-solved				534	150045	5(8)				617	4411	5(8)				360	1766	5(8)
m=3	12.5%	-	0.45	0	1855	408986	4(4)	*	0.45	1	488	4562	4(4)	*	0.79	1	2096	13193	4(4)
	25%	-	1.10	0	108	38438	1(4)	0.50	1.06	3	1308	10037	3(4)	0.17	1.31	1	779	19186	1(4)
	Average	-	0.77	0	1506	334876	5(8)	0.50	0.75	2	839	6908	7(8)	0.17	1.05	1	1833	14392	5(8)
	All-solved				1506	334876	5(8)				415	4158	5(8)				1833	14392	5(8)
m=5	12.5%	-	1.13	0	35	11840	4(4)	*	0.85	0	6	1701	4(4)	*	1.68	0	5	201	4(4)
	25%	0.61	1.93	1	310	80226	1(4)	*	1.60	1	587	11602	4(4)	*	3.07	0	514	10051	4(4)
	Average	0.61	1.53	0	90	25517	5(8)	*	1.32	0	297	5912	8(8)	*	2.38	0	260	5126	8(8)
	All-solved				90	25517	5(8)				5	273	5(8)				5	189	5(8)

Observe also that, in general, the performance of the models in terms of Vg, LPt and $Intt$ gets worse when the size of the instances increase. For a given number of nodes, when we increase the number of routes the performances of the models improves, since the same number of nodes must be now distributed by a larger number of routes, resulting in smaller routes. Also, in general the performance of the models is worse for instances with symmetrical cost matrices.

Table 6 represents the average results, grouped by percentage of the number of consistent nodes. The increase in the percentage on the number of consistent nodes worsens the performance of the models, especially in what concerns the computational time needed ($Intt$).

4.3 The consistency configuration experiment

In this section, we compare the performance of the models SCF, PQ and APQ in terms of the percentage of nodes that are consistent as well as the number of routes they are in. Thus, we will consider more general consistency configurations.

In this experiment we set $m = 5$ and use, for completeness, matrices $med32$, $med36$, $med40$, $med44$, $ftv33$, $ftv35$, $ftv38$ and $ftv44$. Again, the number of consistency nodes are 12.5% or 25% of the total number of client nodes, and the free nodes are equally distributed among the routes. For the distribution of the consistent nodes we consider seven different variants (to facilitate the description of the variants, we assume an ordering of the routes). Table 7 details the characteristics of each variant. The first six correspond to new distributions and the last one corresponds to the distribution considered in the previous section for $m = 5$ and 25% consistent nodes.

m	Variant	Distribution of consistent nodes					
		{1, 2, 3, 4, 5}	{1, 2, 4, 5}	{2, 3, 4, 5}	{1, 2, 5}	{3, 4, 5}	{4, 5}
5	v1	all	-	-	-	-	-
	v2	50%	-	-	-	50%	-
	v3	50%	-	25%	-	25%	-
	v4	25%	-	25%	-	25%	25%
	v5	-	-	-	50%	50%	-
	v6	-	25%	25%	25%	25%	-
	v7	See Application Experiment, for m=5 and 25% consistent nodes					

Table 7: Variant description

Observe that in these variants consistency is maximum on variant v1 and it decreases when we move forward to the other variants. In this computational experiment we will consider a total of 112 test instances.

Tables 8 and 9 show the results for this experiment, for the instances with 12.5% of consistent nodes, and for the instances with 25% of consistent nodes, respectively. Again, we provide the averages *all-solved* when these are relevant (that is, whenever it was not possible to solve all the instances with all the models). Additionally, due to the fact that there are many instances that the SCF model is not able to solve (in fact, for some cost matrices in Table 9 this model is not able to solve any of the variants), we also provide similar averages, denoted by ***all-PQ-APQ***, that only consider instances solved by both the PQ and the APQ models.

We can see that, in addition to not being able to solve many of of the tested instances, the SCF model has significantly larger computational times ($Intt$), when compared with the other two models. Comparing the APQ with the PQ model, we observe that there are many instances for which the APQ model outperforms (sometimes quite significantly) the PQ model. Additionally, there are some instances which are solved relatively quickly by the APQ model but not solved by the PQ model within the time limit.

In general, we should note that instances with more consistent nodes are more difficult to solve, as we can see from the larger gap values and computational times. The number of nodes is another factor that contributes to instances being harder to solve. In Table 9 we can see that when the instances increase from size 38 (or 40) to size 44 they become much more difficult to solve. In fact, the average $Intt$ is far greater in instances $ftv44$, when compared with instances $ftv38$. Similarly, both model PQ and APQ are able to solve only one instance $med44$ (when both were able to solve

Table 8: Consistency configurations experiment (12.5% consistent nodes)

	SCF						PQ						APQ					
Instance	Vg	LPg	LPt	Intt	BB	#	Vg	LPg	LPt	Intt	BB	#	Vg	LPg	LPt	Intt	BB	#
med32-v1	*	1.12	0	57	16335		*	0.45	1	1	5		*	1.28	0	3	26	
med32-v2	*	0.67	0	4	1808		*	0.18	0	1	37		*	1.18	0	1	31	
med32-v3	*	1.33	0	34	11927		*	0.82	0	2	124		*	1.84	0	3	44	
med32-v4	*	0.94	0	2	1218		*	0.31	0	0	0		*	1.35	0	2	15	
med32-v5	*	0.82	0	2	1412		*	0.44	0	0	0		*	1.35	0	1	68	
med32-v6	*	1.16	0	21	10437		*	0.56	0	1	42		*	1.82	0	1	58	
med32-v7	*	0.30	0	0	0		*	0.00	0	0	0		*	0.98	0	2	3	
Average	*	0.91	0	17	6162	7(7)	*	0.39	0	1	30	7(7)	*	1.21	0	2	35	7(7)
med36-v1	0.16	2.20	0	-	-		*	1.90	1	275	7382		*	2.82	0	24	913	
med36-v2	0.17	2.49	0	-	-		*	2.25	0	90	2023		*	3.22	0	12	402	
med36-v3	*	2.17	0	1994	531688		*	1.92	0	105	3215		*	3.21	0	10	385	
med36-v4	*	2.18	0	2786	764512		*	1.97	0	94	3068		*	3.32	0	10	405	
med36-v5	*	1.45	0	54	23361		*	1.35	0	79	2793		*	2.84	0	25	1627	
med36-v6	*	1.97	0	3485	925564		*	1.92	0	78	3687		*	3.60	0	16	895	
med36-v7	*	1.60	0	97	44880		*	1.28	0	7	539		*	2.48	0	4	283	
Average	0.17	2.01	0	1683	458001	5(7)	*	1.80	0	104	3244	7(7)	*	3.07	0	14	701	7(7)
All-solved				1683	458001	5(7)				73	2660	5(7)				13	737	5(7)
med40-v1	-	1.79	0	-	-		*	1.14	2	77	1180		*	1.94	0	18	439	
med40-v2	*	1.12	0	2701	628775		*	0.80	1	27	622		*	1.45	0	8	126	
med40-v3	*	1.28	0	3169	497928		*	0.88	1	23	579		*	1.59	0	8	172	
med40-v4	*	1.08	0	492	143572		*	0.77	1	9	247		*	1.66	0	7	187	
med40-v5	*	1.18	0	138	56146		*	0.94	1	13	1145		*	1.57	0	10	501	
med40-v6	*	1.46	0	1078	260431		*	1.09	1	45	442		*	1.98	0	8	279	
med40-v7	*	1.49	0	583	189817		*	1.06	0	12	986		*	1.86	0	4	78	
Average	-	1.34	0	1360	296112	6(7)	*	0.95	1	29	743	7(7)	*	1.72	0	9	255	7(7)
All-solved				1360	296112	6(7)				22	670	6(7)				8	224	6(7)
med44-v1	-	1.86	1	-	-		*	1.67	4	1546	13350		*	2.12	1	2602	19355	
med44-v2	-	1.96	0	-	-		*	1.86	4	2830	34549		*	2.58	1	2839	36355	
med44-v3	-	2.24	1	-	-		*	2.15	3	3934	47360		*	2.77	1	2302	21974	
med44-v4	*	1.97	1	9484	1233820		*	1.78	2	671	8503		*	2.63	1	44	858	
med44-v5	*	2.04	1	5440	1350131		*	1.95	1	1154	23383		*	2.91	0	1073	15165	
med44-v6	0.00	2.42	1	-	-		*	2.33	1	1065	14474		*	3.13	1	1357	17358	
med44-v7	*	1.70	0	649	203920		*	1.58	1	233	4262		*	2.02	1	16	604	
Average	0.00	2.03	1	5191	929290	3(7)	*	1.90	2	1633	20840	7(7)	*	2.59	1	1462	15953	7(7)
All-solved				5191	929290	3(7)				686	12049	3(7)				378	5542	3(7)
ftv33-v1	*	11.35	0	15	2944		*	8.21	0	20	1080		*	16.62	0	3	126	
ftv33-v2	*	11.85	0	6	1396		*	9.34	0	9	486		*	14.52	0	5	460	
ftv33-v3	*	11.81	0	6	2568		*	9.40	0	6	590		*	13.84	0	7	788	
ftv33-v4	*	10.09	0	1	81		*	7.85	0	5	435		*	13.49	0	2	39	
ftv33-v5	*	9.30	0	2	1323		*	6.90	0	1	160		*	13.97	0	2	127	
ftv33-v6	*	12.72	0	8	2372		*	10.11	0	7	644		*	15.94	0	3	180	
ftv33-v7	*	7.51	0	1	323		*	4.28	0	1	43		*	8.49	0	2	145	
Average	*	10.66	0	6	1572	7(7)	*	8.01	0	7	491	7(7)	*	13.84	0	3	266	7(7)
ftv35-v1	*	12.06	0	29	3005		*	10.81	0	67	3036		*	12.20	0	5	201	
ftv35-v2	*	11.70	0	10	3616		*	9.54	0	4	659		*	12.75	0	3	735	
ftv35-v3	*	14.51	0	28	7370		*	11.87	0	21	2831		*	15.02	0	3	644	
ftv35-v4	*	13.25	0	13	5510		*	10.57	0	27	2048		*	14.62	0	4	767	
ftv35-v5	*	10.11	0	7	2034		*	8.76	0	9	1334		*	14.19	0	6	685	
ftv35-v6	*	14.15	0	90	32545		*	11.94	0	22	1671		*	18.06	0	11	1804	
ftv35-v7	*	9.94	0	6	1066		*	7.86	0	10	1113		*	11.97	0	6	1858	
Average	*	12.25	0	26	7878	7(7)	*	10.19	0	23	1813	7(7)	*	12.54	0	5	956	7(7)
ftv38-v1	*	9.29	0	11	755		*	7.04	1	22	819		*	11.95	0	10	362	
ftv38-v2	*	10.65	0	17	1599		*	9.25	1	12	768		*	11.51	0	9	461	
ftv38-v3	*	13.79	0	93	10669		*	11.79	1	101	2880		*	15.54	0	8	353	
ftv38-v4	*	12.01	0	22	4252		*	8.91	0	12	633		*	12.85	0	4	74	
ftv38-v5	*	8.40	0	1	111		*	7.27	0	68	213		*	10.13	0	8	363	
ftv38-v6	*	13.82	0	187	37006		*	12.48	0	4	1587		*	15.39	0	8	286	
ftv38-v7	*	6.23	0	1	70		*	3.15	0	1	0		*	7.76	0	5	120	
Average	*	10.59	0	47	7780	7(7)	*	8.56	0	31	986	7(7)	*	12.16	0	7	288	7(7)
ftv44-v1	*	12.59	1	917	93594		*	10.01	2	297	3670		*	13.21	0	10	291	
ftv44-v2	*	12.80	1	1055	159905		*	10.45	1	270	5105		*	14.40	0	14	1041	
ftv44-v3	*	14.76	1	4227	431771		*	12.44	2	215	3574		*	16.37	0	28	1469	
ftv44-v4	*	13.66	1	237	34867		*	9.82	1	96	2127		*	13.95	0	13	770	
ftv44-v5	*	8.35	0	5	1511		*	6.21	1	17	612		*	12.07	0	46	1757	
ftv44-v6	*	15.97	1	1485	143978		*	13.00	1	516	11469		*	18.79	0	197	5827	
ftv44-v7	*	9.22	1	6	1159		*	7.21	0	29	1072		*	10.34	0	51	1709	
Average	*	12.48	1	1133	123826	7(7)	*	9.88	1	206	3947	7(7)	*	14.18	0	51	1838	7(7)
Average	0.11	6.53	0	832	160920	49(56)	*	5.21	1	254	4012	56(56)	*	7.83	0	194	2537	56(56)
All-solved				832	160920	49(56)				90	2129	49(56)				35	924	49(56)

Table 9: Consistency configurations experiment (25% consistent nodes)

	SCF						PQ						APQ					
Instance	Vg	LPg	LPt	Intt	BB	#	Vg	LPg	LPt	Intt	BB	#	Vg	LPg	LPt	Intt	BB	#
med32-v1	-	1.80	0	-	-		*	1.28	4	162	1850		*	2.05	0	41	1223	
med32-v2	0.00	1.56	0	-	-		*	1.13	2	53	886		*	2.33	0	17	816	
med32-v3	-	1.61	0	-	-		*	1.21	2	149	1611		*	2.55	0	23	883	
med32-v4	*	1.56	0	10251	1681884		*	1.18	1	57	1117		*	2.65	0	12	743	
med32-v5	*	1.23	0	595	221495		*	0.94	1	15	532		*	2.09	0	21	1292	
med32-v6	0.00	1.68	0	-	-		*	1.29	1	24	684		*	2.16	0	19	939	
med32-v7	*	1.87	0	311	80226		*	1.37	0	1	475		*	2.71	0	3	140	
Average	0.00	1.62	0	3719	661202	3(7)	*	1.20	2	66	1022	7(7)	*	2.36	0	19	862	7(7)
All-solved				3719	661202	3(7)				24	708	3(7)				12	725	3(7)
med36-v1	-	2.83	1	-	-		*	2.63	6	7562	62740		0.00	3.11	1	-	-	
med36-v2	-	2.49	0	-	-		*	2.40	4	5159	62891		*	2.92	0	2696	21886	
med36-v3	-	2.92	0	-	-		*	2.88	2	5869	63814		0.00	3.37	0	-	-	
med36-v4	-	3.33	0	-	-		0.00	3.23	3	-	-		*	4.15	0	7158	101511	
med36-v5	0.98	2.48	0	-	-		*	2.39	1	3746	67127		*	2.85	0	4736	49532	
med36-v6	-	2.78	1	-	-		*	2.69	3	5514	74620		*	3.12	0	6079	95081	
med36-v7	0.00	2.19	0	-	-		*	2.00	0	355	8999		*	3.70	0	86	3790	
Average	0.49	2.72	0	-	-	0(7)	0.00	2.60	3	5049	56699	6(7)	0.00	3.32	0	4151	54360	5(7)
All-PQ-APQ										3694	53409	4(7)				3399	42572	4(7)
med40-v1	-	2.00	1	-	-		*	1.48	12	6787	29165		*	2.18	2	6047	31686	
med40-v2	-	2.20	1	-	-		*	1.83	6	4421	33090		*	2.84	1	5729	54125	
med40-v3	-	2.12	1	-	-		0.25	1.74	7	-	-		*	2.61	2	6387	51498	
med40-v4	-	2.15	1	-	-		*	1.82	5	5379	52039		*	2.79	1	5137	57086	
med40-v5	-	2.29	1	-	-		*	2.08	4	7891	100032		0.15	3.11	0	-	-	
med40-v6	-	2.06	1	-	-		*	1.80	4	5235	47446		*	2.59	1	10644	83539	
med40-v7	-	1.90	1	-	-		*	1.39	1	937	24964		*	3.18	0	544	14701	
Average	-	2.10	1	-	-	0(7)	0.25	1.73	6	5108	47789	6(7)	0.15	2.76	1	5748	48773	6(7)
All-PQ-APQ										4552	37341	5(7)				5620	48227	5(7)
med44-v1	-	2.28	1	-	-		1.05	2.28	18	-	-		0.31	2.31	1	-	-	
med44-v2	-	2.46	1	-	-		0.47	2.38	32	-	-		0.35	2.79	1	-	-	
med44-v3	-	2.75	1	-	-		0.79	2.60	12	-	-		0.00	2.82	1	-	-	
med44-v4	-	2.48	1	-	-		0.62	2.43	19	-	-		0.00	2.96	1	-	-	
med44-v5	-	2.16	1	-	-		0.00	1.86	6	-	-		2.05	2.66	1	-	-	
med44-v6	-	2.43	1	-	-		0.00	2.16	9	-	-		0.38	2.53	1	-	-	
med44-v7	1.21	1.74	1	-	-		*	1.62	1	1055	16192		*	2.70	1	1419	21571	
Average	1.21	2.33	1	-	-	0(7)	0.49	2.19	14	1055	16192	1(7)	0.52	2.68	1	1419	21571	1(7)
ftv33-v1	*	16.95	0	19	1125		*	14.50	2	161	2184		*	16.09	0	21	454	
ftv33-v2	*	20.04	0	1828	257276		*	17.66	1	381	5904		*	23.32	0	14	755	
ftv33-v3	*	22.63	0	333	53044		*	20.53	1	429	8183		*	22.12	0	14	573	
ftv33-v4	*	21.27	0	814	107828		*	18.81	1	133	12006		*	24.48	0	16	922	
ftv33-v5	*	21.29	0	30	5313		*	19.33	0	300	2923		*	27.91	0	71	2207	
ftv33-v6	*	23.71	0	1169	237996		*	21.70	1	69	7835		*	27.11	0	24	1383	
ftv33-v7	*	19.70	0	91	31634		*	18.03	0	69	3108		*	21.81	0	93	3827	
Average	*	20.80	0	612	99174	7(7)	*	18.65	1	220	6020	7(7)	*	23.26	0	36	1446	7(7)
ftv35-v1	*	18.70	0	2413	171482		*	17-19	2	1213	9950		*	19.37	0	50	1630	
ftv35-v2	*	18.54	0	1313	148078		*	15.78	2	1089	13419		*	19.97	0	32	1862	
ftv35-v3	*	20.58	0	6768	478694		*	18.17	2	933	15016		*	20.05	0	42	2155	
ftv35-v4	*	18.99	0	3783	413610		*	16.61	2	795	10987		*	20.49	0	160	3953	
ftv35-v5	*	23.25	0	3036	558565		*	21.17	1	1115	18372		*	28.12	0	1133	21830	
ftv35-v6	-	23.05	0	-	-		*	20.62	1	883	17065		*	26.46	0	155	3033	
ftv35-v7	*	20.04	0	65	20883		*	17.70	0	128	3903		*	21.80	0	109	2824	
Average	-	20.45	0	2896	298552	6(7)	*	18.18	1	879	12673	7(7)	*	22.32	0	240	5327	7(7)
All-solved				2896	298552	6(7)				879	11941	6(7)				254	5709	6(7)
ftv38-v1	-	20.23	1	-	-		*	18.40	4	1737	11538		*	20.42	1	387	3979	
ftv38-v2	*	20.15	0	1321	136708		*	18.10	5	1906	14739		*	21.92	0	346	4592	
ftv38-v3	*	21.47	0	9887	829362		*	19.56	3	3991	33784		*	21.96	0	347	4772	
ftv38-v4	*	22.44	1	8844	963596		*	19.61	2	1265	13749		*	22.75	0	275	3476	
ftv38-v5	*	23.43	0	854	146327		*	21.50	1	1774	27497		*	27.49	0	2071	25410	
ftv38-v6	-	23.12	0	-	-		*	21.00	2	2067	25238		*	25.32	0	1863	18484	
ftv38-v7	*	22.52	0	236	42745		*	20.81	0	814	22916		*	27.70	0	352	7820	
Average	-	21.91	0	4228	423748	5(7)	*	19.85	2	1936	21352	7(7)	*	23.94	0	806	9790	7(7)
All-solved				4228	423748	5(7)				1950	22537	5(7)				678	9214	6(7)
ftv44-v1	-	20.46	1	-	-		0.00	17.80	11	-	-		*	21.29	1	1355	7746	
ftv44-v2	-	20.25	1	-	-		0.00	18.13	6	-	-		*	21.32	1	2138	14570	
ftv44-v3	-	21.25	1	-	-		0.07	19.15	7	-	-		*	22.30	1	8828	41783	
ftv44-v4	-	21.85	1	-	-		*	19.08	5	8558	56055		*	23.01	1	2914	39261	
ftv44-v5	*	17.87	1	2694	306230		*	16.60	2	8531	87002		*	21.75	0	6742	50569	
ftv44-v6	-	21.40	1	-	-		0.00	19.41	4	-	-		0.00	24.72	0	-	-	
ftv44-v7	*	18.97	1	276	34123		*	16.67	1	1205	28473		*	20.38	0	607	10173	
Average	-	20.29	1	1485	170177	2(7)	0.02	18.12	5	6098	57177	3(7)	0.00	22.11	1	3764	27350	6(7)
All-solved				1485	170177	2(7)				4868	57738	2(7)				3675	30371	2(7)
All-PQ-APQ										6098	57177	3(7)				3421	33334	3(7)
Average	0.44	11.53	0	2475	301227	23(56)	0.27	10.32	4	2271	25048	44(56)	0.33	12.84	0	1890	18958	46(56)
All-solved				2475	301227	23(56)				1147	14960	23(56)				546	6668	23(56)
All-PQ-APQ										1917	21354	41(56)				1490	15975	41(56)

six instances $med40$). Observe, also, that matrix $med44$ is symmetric, which is not the case for matrix $med40$, which also increases the difficulty to solve these instances.

Table 10: Consistency configuration experiment results - Averages per variant

	SCF						PQ						APQ					
Variant	Vg	LPg	LPt	Intt	BB	#	Vg	LPg	LPt	Intt	BB	#	Vg	LPg	LPt	Intt	BB	#
v1-12.5%	0.16	6.53	0	206	23327	5(8)	*	3.17	1	288	3815	8(8)	*	7.77	0	334	2714	8(8)
v1-25%	-	10.66	1	1216	86304	2(8)	0.53	9.45	7	2937	19571	6(8)	0.16	10.85	1	1317	7786	6(8)
v1-Average	0.16	8.59	0	494	41320	7(16)	0.53	7.31	4	1423	10568	14(16)	0.16	9.31	0	755	4888	14(16)
v1-All-solved				494	41320	7(16)				254	2963	7(16)				15	441	13(16)
v1-All-PQ-APQ										951	6555	13(16)				709	4668	13(16)
v2-12.5%	0.17	6.53	0	632	132850	6(8)	*	5.46	1	405	5531	8(8)	*	7.70	0	361	4951	8(8)
v2-25%	0.00	10.96	0	1487	180687	3(8)	0.24	9.68	7	2168	21822	6(8)	0.18	12.18	0	1567	14087	7(8)
v2-Average	0.09	8.81	0	917	148796	9(16)	0.24	7.57	4	1161	12513	14(16)	0.18	9.94	0	924	9214	15(16)
v2-All-solved				917	148796	9(16)				411	4638	9(16)				48	1118	9(16)
v2-All-PQ-APQ										1161	12513	14(16)				860	8832	14(16)
v3-12.5%	-	7.74	0	1364	213417	7(8)	*	6.41	1	551	7644	8(8)	*	8.77	0	296	296	8(8)
v3-25%	-	11.92	0	5663	453700	3(8)	0.37	10.73	5	2274	24482	5(8)	0.00	12.22	1	2607	16944	6(8)
v3-Average	-	9.83	0	2654	285502	10(16)	0.37	8.57	3	1214	14120	13(16)	0.00	10.50	0	1286	9107	14(16)
v3-All-solved				2654	285502	10(16)				583	7078	10(16)				47	1136	10(16)
v3-All-PQ-APQ										826	9979	12(16)				706	2851	12(16)
v4-12.5%	*	6.90	0	1630	273479	8(8)	*	5.25	1	114	2133	8(8)	*	7.98	0	11	389	8(8)
v4-25%	-	11.76	1	5923	791730	4(8)	0.31	10.35	5	2698	24326	6(8)	0.00	12.91	0	2239	29565	7(8)
v4-Average	-	9.33	0	3061	446229	12(16)	0.31	7.80	3	1222	11644	14(16)	0.00	10.45	0	1051	14004	15(16)
v4-All-solved				3061	446229	12(16)				264	4577	12(16)				46	1017	12(16)
v4-All-PQ-APQ										1222	11644	14(16)				614	7754	14(16)
v5-12.5%	*	5.21	0	706	179504	8(8)	*	4.23	0	168	3705	8(8)	*	7.38	0	146	2537	8(8)
v5-25%	0.98	11.75	0	1442	247586	5(8)	0.00	10.73	2	3339	43355	7(8)	1.01	14.5	0	2462	25140	6(8)
v5-Average	0.98	8.48	0	989	205689	13(16)	0.00	7.48	1	1648	22208	15(16)	1.01	10.94	0	1405	12224	14(16)
v5-All-solved				989	205689	13(16)				1006	12767	13(16)				862	9354	13(16)
v5-All-PQ-APQ										1202	16650	14(16)				1405	12224	14(16)
v6-12.5%	0.00	7.96	0	908	201762	7(8)	*	6.68	0	217	4252	8(8)	*	9.84	0	200	3336	8(8)
v6-25%	0.00	12.53	1	1169	237996	1(8)	0.00	11.33	3	2299	28815	6(8)	0.19	14.25	0	3131	33743	6(8)
v6-Average	0.00	10.24	0	940	206291	8(16)	0.00	9.01	2	1109	14779	14(16)	0.19	12.05	0	1456	16368	14(16)
v6-All-solved				940	206291	8(16)				93	3422	8(16)				34	1339	8(16)
v6-All-PQ-APQ										1109	14779	14(16)				1456	16368	14(16)
v7-12.5%	*	4.75	0	168	55154	8(8)	*	3.30	0	37	1002	8(8)	*	5.74	0	11	600	8(8)
v7-25%	0.61	11.12	0	196	41922	5(8)	*	9.95	0	571	13629	8(8)	*	13.00	0	402	8106	8(8)
v7-Average	0.61	7.93	0	179	50065	13(16)	*	6.63	0	304	7315	16(16)	*	9.37	0	206	4353	16(16)
v7-All-solved				179	50065	13(16)				193	5145	13(16)				96	2276	13(16)
v7-All-PQ-APQ										304	7315	16(16)				206	4353	16(16)
Average	0.315	9.03	0	1357	205740	72(112)	0.24	7.77	2	1142	13268	100(112)	0.29	10.37	0	959	9942	102(112)
All-solved				1357	205740	72(112)				428	6228	72(112)				198	2759	72(112)
All-PQ-APQ										957	11342	97(112)				742	8217	97(112)

To have a more concise understanding of these results, Table 10 shows the average results per variant. Here we can confirm that, although in average the APQ model outperforms the other models in this experiment, the PQ model still performs better in some variants, namely variants v5 and v6. This means that, as expected, the PQ model is more efficient for instances where the consistent nodes do not appear in many routes, but the aggregated model, also as expected, becomes better when we test instances where the consistent nodes appear in most, or even all the routes.

We also note that although variant v1 is the one that has the most (all) consistent nodes in the five routes, none of the models has its worst performance in this variant. This might be explained by the fact that in variant v1 all consistent nodes appear in all the routes and free nodes are equally distributed, resulting in routes that have the same length, or with lengths that differ by 1 at most. In other variants we have more considerable differences in route lengths, which results in more positions in which smaller routes may start, which in turn may make instances more difficult to solve. In fact, the worst performance of the SCF model is obtained for variant v4, the worst performance of the PQ model is for variant v5, while the worst performance of the APQ model is for v6. However, the three models have their best performance in variant v7.

Finally, Table 11 provides an overview on the impact of the use of constraints (3.85) and (3.86) in the PQ model and (3.87) and (3.88) in the APQ model. This experiment includes all the instances from the consistency configurations tests with 25% of consistent nodes, and in Table 11 the results are presented using averages per variant. We can see that, in general, imposing a limit in the positions occupied by arcs adjacent to the depot results in an improvement both to the LP gaps and to the performance of the models. In fact, models PQ_0 and APQ_0 are able to solve more instances to optimality than models PQ and APQ. In terms of CPU time, we can see a significant improvement in model PQ_0, in most of the variants. This is not the case for the APQ_0 model, which, despite being the model that solves the most instances, has, in general, worse computational times than model APQ. In average, the PQ_0 model is the quickest to solve instances, but in variants where consistent nodes appear in many or all of the routes, that is, variants v1-v3, it still compensates to use the APQ model.

The full results for this experiment have been included in the Appendix, as well as some results for the Miller-Tucker-Zemlin based model mentioned in Section 3.1.1. With respect to the latter, the most relevant fact is that both the SCF and MTZ models are not competitive, when compared with the PQ and APQ models.

5 Conclusions

We studied the *Consistent Travelling Salesman Problem* with total positional consistency constraints with emphasis on a particular case, namely an application of scheduling of healthcare services.

We presented several compact formulations, which have been evaluated using three computational experiments based on two sets of test instances. Several of the models proposed in this study are augmented versions of models known from the literature for the TDTSP. The reason for this choice is that these models are adequate for this problem due to the inclusion of the position index in the variables of the model, allowing the derivation of strong positional consistency constraints. A set of preliminary tests allowed us to evaluate which models were the most efficient, namely a slightly enhanced version of the Single Commodity Flow model presented by [Subramanyam and Gounaris, 2016], one model based on an augmented version of the model presented [Picard and Queyranne, 1978], the PQ model, and one model that is an aggregated version of the previous model, the APQ model. This aggregated model is new and does not use variables indexed by route/worker. These models were then tested using a set of instances with characteristics of the healthcare application, and, in general, the PQ model showed the best computational performance, with the SCF model and the APQ model still obtaining, in average, very similar computational times. A third experiment allowed us to compare the performances of the models for different and more general consistency configurations. The main conclusion is that, although the PQ model is the most efficient for instances where the consistent nodes appear in fewer routes or less frequently, in instances where many consistent nodes appear in many or even all the routes, the APQ model outperforms (strongly outperforms in many cases) the other models.

We also proposed an enhancement for the PQ and APQ models, based on bounding the positions that can be occupied by arcs adjacent to the depot, which resulted in a significant improvement in the CPU times for the enhanced PQ model. Although a similar improvement was not verified in the enhanced APQ model, there are still instance variants where APQ is the most efficient model. Therefore, one possible line of future research may be the study of general inequalities and further enhancements, both for the PQ and the APQ models, such as constraints based on layered graphs (see, for instance, [Gouveia et al., 2019] and [Godinho et al., 2011]).

Table 11: Consistency configuration experiment results (enhanced models) - Averages per variant

	PQ						APQ						PQ_0						APQ_0					
Variant	Vg	LPg	LPt	Intt	BB	#	Vg	LPg	LPt	Intt	BB	#	Vg	LPg	LPt	Intt	BB	#	Vg	LPg	LPt	Intt	BB	#
v1-Average	0.53	9.45	7	2937	19571	6(8)	0.16	10.85	1	1317	7786	6(8)	0.26	9,06	8	3099	19822	6(8)	0.73	10,48	1	2547	16078	7(8)
v1-All-solved				2012	10937	5(8)				1309	7794	5(8)				1685	10131	5(8)				1824	11538	5(8)
v2-Average	0.24	9.68	7	2168	21822	6(8)	0.18	12.18	0	1567	14087	7(8)	0.00	8.98	4	1643	21156	6(8)	0.00	11.63	0	2553	16899	7(8)
v2-All-solved				2168	21822	6(8)				1472	14006	6(8)				1643	21156	6(8)				2105	15123	6(8)
v3-Average	0.37	10.73	5	2274	24482	5(8)	0.00	12.22	1	2607	16944	6(8)	0.74	10.15	7	1835	18298	6(8)	0.90	11.92	0	3603	22283	7(8)
v3-All-solved				1376	14649	4(8)				107	2096	4(8)				944	10614	4(8)				244	2712	4(8)
v4-Average	0.31	10.35	5	2698	24326	6(8)	0.00	12.91	0	2239	29565	7(8)	0.03	9.58	3	2230	35101	6(8)	0.12	12.58	0	1502	17176	7(8)
v4-All-solved				1526	17980	5(8)				1120	13236	5(8)				809	10286	5(8)				1067	12137	5(8)
v5-Average	0.00	10.73	2	3339	43355	7(8)	1.01	14.5	0	2462	25140	6(8)	*	10.03	1	1652	25932	8(8)	0.68	13.94	0	2301	21704	7(8)
v5-All-solved				2580	33909	6(8)				2462	25140	6(8)				1098	21094	6(8)				1129	12754	6(8)
v6-Average	0.00	11.33	3	2299	28815	6(8)	0.19	14.25	0	3131	33743	6(8)	0.00	10.69	3	1909	20885	7(8)	0.38	13.77	0	3136	23324	7(8)
v6-All-solved				2299	28815	6(8)				3131	33743	6(8)				935	13293	6(8)				2457	20664	6(8)
v7-Average	*	9.95	0	571	13629	8(8)	*	13.00	0	402	8106	8(8)	*	9.32	0	262	6436	8(8)	*	12.61	0	512	8507	8(8)
Average	0.24	10.32	4	2271	25048	44(56)	0.29	12.85	0	1890	18958	46(56)	0.23	9.69	4	1734	20707	47(56)	0.47	12.42	0	2295	17806	50(56)
All-solved				1751	20487	40(56)				1454	10331	40(56)				1010	13232	40(56)				1342	12213	40(56)

It may also be of interest to study if the APQ model could be adapted to other variants of the problem. At first, situations with costs differing from route to route appear to not be modelled by the APQ model since the main idea of this model is to have undistinguished routes. However, in the case where all the routes have the same size, the APQ model still models the problem correctly, if we assign to each arc (i, j) a cost that equals the average of the costs of the routes that share nodes i and j. A different situation (also easily modelled using the disaggregated formulations) concerns clients requiring consistency, but not in all the routes they appear in. In this case we could either consider a partially aggregated formulation where we maintain disaggregated position variables for nodes that are in several routes but do not require consistency in all of them and maintain disaggregated arc variables for arcs that have at least a node in these conditions, and use aggregated variables as in the APQ model for the other nodes, or a completely aggregated formulation where a node is replicated for every route where it appears but does not require consistency. Clearly, this can increase the size of the model and thus it would be worth comparing the behaviour of this complete aggregated model with node replicas, with the partial aggregated model mentioned before.

Finally, although in the application to healthcare services that is considered in this study, idle times between tasks are not allowed (in order to limit the permanency of the professionals inside of the healthcare facility) a topic worth of investigation is to check whether the models under study could handle such situations.

For the disaggregated models, one idea would be to have variables representing arcs of length-position greater than 1, e.g, from a node i in position t to a node j in position $t + k$ (similar to the arcs leaving node 1 and entering node 1, but in this case, entering and leaving any node). Such an arc would represent the fact that a worker would do task i in position t and then "move" to task j in position $t + k$ (meaning that it was idle between these two tasks). This would, however, increase the size of the model because we would need variables $z_{ij}^{t,t+k}$. Observe that in the previous models, the variables z_{ij}^{t} unequivocally identify node i as being in position t-1 and node j in position t (with a slight different explanation for variables associated to arcs leaving and entering the depot). An alternative idea is using arcs of length 1 (as they usually appear in many time-dependent models). In fact, we could use loop variables which as far as we know have appeared first in a work by Gouveia(1998) ([Gouveia, 1998]) in the context of a time-dependent multi-commodity flow model. Such variables represent arcs of the form (i, i, t) indicating that there is a transition from node i in position t to node i in position $t + 1$ (and being interpreted as indicating that a worker "moved" from task i in period t to the same task in period $t + 1$). We observe that it might have also been possible to use this idea in the current paper for the arcs leaving and entering node 1.

Although these two ideas could lead to define valid disaggregated models, and one could use ideas similar to the one used for aggregating the node position version of the PQ model into the APQ model, to obtain an aggregated model, proving its validity is another matter. Observe that the proof of the validity of the APQ model makes use of equality (3.50) that states that in each route there is a single node j such that j occupies a position k and position $k - 1$ is vacant (either because $k = 1$, and therefore position $k - 1$ is not defined, or because $k > 1$ and position $k - 1$ is not occupied by any node that appears in route l). If we allow for there to be idle positions (that are not occupied by any node) in the middle of a route, then this condition no longer holds.

As a conclusion, we think that before attempting aggregated formulations, we would need to focus on disaggregated models and start by comparing models using 4-index variables and allowing length position greater than 1, with models using 3-index variables of length-position equal to 1 and also including loop variables (assuming valid models could be obtained in both settings). The reader is also referred to [Gouveia et al., 2011] where loop variables are used to model a very complicated objective function in a traffic engineering problem.

Acknowledgements

This research was supported by Portuguese National Funding from FCT - Fundação para a Ciência e a Tecnologia, under projects: PTDC/MAT-NAN/2196/2014 and UIDB/04561/2020 and grant SFRH/BD/146812/2019.

References

[Archetti et al., 2014] Archetti, C., Bianchessi, N., and Speranza, M. G. (2014). Branch-and-cut algorithms for the split delivery vehicle routing problem. *European Journal of Operational Research*, 238(3):685–698.

[Braekers et al., 2016] Braekers, K., Hartl, R. F., Parragh, S. N., and Tricoire, F. (2016). A bi-objective home care scheduling problem: Analyzing the trade-off between costs and client inconvenience. *European Journal of Operational Research*, 248(2):428–443.

[Decerle et al., 2017] Decerle, J., Grunder, O., El Hassani, A. H., and Barakat, O. (2017). A general model for the home health care routing and scheduling problem with route balancing. *IFAC-PapersOnLine*, 50(1):14662–14667.

[Desrochers and Laporte, 1991] Desrochers, M. and Laporte, G. (1991). Improvements and extensions to the miller-tucker-zemlin subtour elimination constraints. *Operations Research Letters*, 10(1):27–36.

[Drexl, 2012] Drexl, M. (2012). Synchronization in vehicle routing—a survey of vrps with multiple synchronization constraints. *Transportation Science*, 46(3):297–316.

[En-nahli et al., 2016] En-nahli, L., Afifi, S., Allaoui, H., and Nouaouri, I. (2016). Local search analysis for a vehicle routing problem with synchronization and time windows constraints in home health care services. *IFAC-PapersOnLine*, 49(12):1210–1215.

[Fox et al., 1980] Fox, K. R., Gavish, B., and Graves, S. C. (1980). An n-constraint formulation of the (time-dependent) traveling salesman problem. *Operations Research*, 28(4):1018–1021.

[Gavish and Graves, 1978] Gavish, B. and Graves, S. C. (1978). The travelling salesman problem and related problems.

[Godinho et al., 2011] Godinho, M. T., Gouveia, L., and Pesneau, P. (2011). *On a Time-Dependent Flow-Based Formulation for the Asymmetric Travelling Salesman Problem*. Wiley.

[Gouveia, 1995] Gouveia, L. (1995). A 2n constraint formulation for the capacitated minimal spanning tree problem. *Operations research*, 43(1):130–141.

[Gouveia, 1998] Gouveia, L. (1998). Using variable redefinition for computing lower bounds for minimum spanning and steiner trees with hop constraints. *INFORMS Journal on Computing*, 10(2):180–188.

[Gouveia et al., 2019] Gouveia, L., Leitner, M., and Ruthmair, M. (2019). Layered graph approaches for combinatorial optimization problems. *Computers & Operations Research*, 102:22–38.

[Gouveia et al., 2010] Gouveia, L., Mourão, M. C., and Pinto, L. S. (2010). Lower bounds for the mixed capacitated arc routing problem. *Computers & Operations Research*, 37(4):692–699.

[Gouveia et al., 2011] Gouveia, L., Patrício, P., and de Sousa, A. (2011). Models for optimal survivable routing with a minimum number of hops: comparing disaggregated with aggregated models. *International Transactions in Operational Research*, 18(3):335–358.

[Gouveia and Voß, 1995] Gouveia, L. and Voß, S. (1995). A classification of formulations for the (time-dependent) traveling salesman problem. *European Journal of Operational Research*, 83(1):69–82.

[Kovacs et al., 2014a] Kovacs, A. A., Golden, B. L., Hartl, R. F., and Parragh, S. N. (2014a). The generalized consistent vehicle routing problem. *Transportation Science*, 49(4):796–816.

[Kovacs et al., 2014b] Kovacs, A. A., Golden, B. L., Hartl, R. F., and Parragh, S. N. (2014b). Vehicle routing problems in which consistency considerations are important: A survey. *Networks*, 64(3):192–213.

[Magnanti, 1981] Magnanti, T. L. (1981). Combinatorial optimization and vehicle fleet planning: Perspectives and prospects. *Networks*, 11(2):179–213.

[Miller et al., 1960] Miller, C. E., Tucker, A. W., and Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, 7(4):326–329.

[Öncan et al., 2009] Öncan, T., Altınel, İ. K., and Laporte, G. (2009). A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers & Operations Research*, 36(3):637–654.

[Picard and Queyranne, 1978] Picard, J.-C. and Queyranne, M. (1978). The time-dependent traveling salesman problem and its application to the tardiness problem in one-machine scheduling. *Operations Research*, 26(1):86–110.

[Roberti and Toth, 2012] Roberti, R. and Toth, P. (2012). Models and algorithms for the asymmetric traveling salesman problem: an experimental comparison. *EURO Journal on Transportation and Logistics*, 1(1):113–133.

[Subramanyam and Gounaris, 2016] Subramanyam, A. and Gounaris, C. E. (2016). A branch-and-cut framework for the consistent traveling salesman problem. *European Journal of Operational Research*, 248(2):384–395.

[Subramanyam and Gounaris, 2017] Subramanyam, A. and Gounaris, C. E. (2017). A decomposition algorithm for the consistent traveling salesman problem with vehicle idling. *Transportation Science*, 52(2):386–401.

[Toth and Vigo, 2014] Toth, P. and Vigo, D. (2014). *Vehicle routing: problems, methods, and applications*. Society for Industrial and Applied Mathematics.

1 The Aggregated Single Commodity Flow model

We present next the aggregated version of the SCF model. This models uses aggregated variables $h_{ij}, i, j = 1, ..., n$ representing the number of times arc (i, j) is traversed, and variables $f_{ij}, i, j = 1, ..., n$ representing the total flow that traverses arc (i, j). The aggregated variables are related with the original variables as follows:

$$h_{ij} = \sum_{l=1}^{m} x_{ij}^{l};\ \forall i, j = 1, ..., n \tag{1.1}$$

$$f_{ij} = \sum_{l=1}^{m} y_{ij}^{l};\ \forall i, j = 1, ..., n \tag{1.2}$$

The inequalities presented in the model below result from adding, for all $l = 1, ..., m$, the corresponding constraints in the SCF model.

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij}h_{ij} \tag{1.3}$$

$$\text{S.t.} \sum_{i=1}^{n} h_{ij} = \sum_{l=1}^{m} S_j^l;\ \forall j = 1, ..., n \tag{1.4}$$

$$\sum_{i=1}^{n} h_{ji} = \sum_{l=1}^{m} S_j^l;\ \forall j = 1, ..., n \tag{1.5}$$

$$\sum_{i=1}^{n} f_{ji} - \sum_{i=1}^{n} f_{ij} = \sum_{l=1}^{m} S_j^l;\ \forall j = 2, ..., n \tag{1.6}$$

$$h_{1j} \leq f_{1j};\ \forall j = 1, ..., n \tag{1.7}$$

$$2 * h_{ij} \leq f_{ij};\ \forall i = 2, ..., n;\ \forall j = 1, ..., n \tag{1.8}$$

$$f_{i1} \leq (T)h_{i1};\ \forall i = 1, ..., n \tag{1.9}$$

$$f_{ij} \leq (T-1)h_{ij};\ \forall i = 1, ..., n;\ \forall j = 2, ..., n \tag{1.10}$$

$$\sum_{i=1}^{n} f_{ij} = (\sum_{l=1}^{m} S_j^l)\sum_{k=1}^{T-1} kp_j^k;\ \forall j = 2, ..., n \tag{1.11}$$

$$\sum_{k=1}^{T-1} p_j^k = 1;\ \forall j = 2, ..., n \tag{1.12}$$

$$\sum_{j=2}^{n} S_j^l p_j^k \leq 1;\ \forall l = 1, ..., m;\ \forall k = 1, ..., T-1 \tag{1.13}$$

$$h_{ij} \leq SR_{ij};\ \forall i, j = 1, ..., n \tag{1.14}$$

$$h_{ij}, f_{ij} \geq 0 \text{ and integer};\ \forall i, j = 1, ..., n \tag{1.15}$$

$$p_j^k \in \{0; 1\};\ \forall j = 2, ..., n;\ \forall k = 1, ..., T-1 \tag{1.16}$$

2 An optimal solution for the Aggregated Single Commodity Flow model

We present below an optimal solution obtained using the aggregated SCF model. Node 1 is the depot. Nodes $\{2, 3, 4, 5\}$ must appear in both routes (and therefore require total consistency), nodes $\{6, 7\}$ appear only in route 1 and nodes $\{8, 9\}$ appear only in route 2.

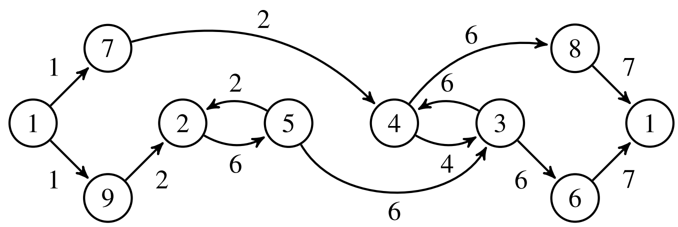

Figure 1: A feasible solution for the aggregated SCF model

The integer values associated to each arc, represent the corresponding aggregated flow value (or aggregated position value). In this solution, the arc $(2, 5)$ is used in two routes, but only one arc appears in the figure because, due to the fact that we are using an aggregated model, we know the total flow going from 2 to 5, but not the distribution of the flow between the two routes. Clearly, the solution is not valid for the problem. This is easily seen by the 2-cycles in the solution.

3 The aggregated weak $FGG3$ model

The aggregated weak $FGG3$ model, $AwFGG3$ model, is obtained from the $AFGG3$ by replacing constraints (3.39) by constraints (3.62). Since we provide, below, an equivalence between the linear programming relaxations of the two aggregated model, we describe now all the constraints of the $AwFGG3$:

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{T} C_{ij}u_{ij}^{k} \tag{3.1}$$

$$\text{s.t. } \sum_{i=1}^{n}\sum_{k=1}^{T} u_{ij}^{k} = \sum_{l=1}^{m} S_j^l;\ \forall j = 1, ..., n \tag{3.2}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{T} u_{ji}^{k} = \sum_{l=1}^{m} S_j^l;\ \forall j = 1, ..., n \tag{3.3}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{T} ku_{ji}^{k} - \sum_{i=1}^{n}\sum_{k=1}^{T} ku_{ij}^{k} = \sum_{l=1}^{m} S_j^l;\ \forall j = 2, ..., n \tag{3.4}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{T-1} ku_{ij}^{k} = (\sum_{l=1}^{m} S_j^l) \sum_{k=1}^{T-1} kp_j^k;\ \forall j = 2, ..., n \tag{3.5}$$

$$\sum_{k=1}^{T-1} p_j^k = 1;\ \forall j = 2, ..., n \tag{3.6}$$

$$\sum_{j=2}^{n} S_j^l p_j^k \leq 1;\ \forall l = 1, ..., m;\ \forall k = 1, ..., T-1 \tag{3.7}$$

$(3.34) - (3.35)$

$$\sum_{k=1}^{T} u_{ij}^k \leq SR_{ij};\ \forall i, j = 1, ..., n; \tag{3.8}$$

$$u_{ij}^k \geq 0 \text{ and integer};\ \forall i, j = 1, ..., n;\ \forall k = 1, ..., T \tag{3.9}$$

$$p_j^k \in \{0; 1\};\ \forall j = 2, ..., n;\ \forall k = 1, ..., T-1 \tag{3.10}$$

4 Relationship between the *ASCF* and the *AwFGG3* models

The following result is an adaptation of the proof taken in [Gouveia, 1995]. First, we note that variables $p_i^k, i = 2, ..., n, k = 1, ..., T-2$ are the same for both models, and that we can relate the other variables from the two models as follows:

$$h_{ij} = \sum_{k=1}^{T} u_{ij}^k;\ \forall i, j = 1, ..., n \tag{4.1}$$

$$f_{ij} = \sum_{k=1}^{T} k u_{ij}^k;\ \forall i, j = 1, ..., n \tag{4.2}$$

We start by stating the following straightforward lemmas:

Lemma 4.1. *Objective functions (1.3) and (3.1) are equivalent under constraints (4.1).*

Lemma 4.2. *The assignment constraints (1.4) and (3.2) are equivalent under constraints (4.1).*

Lemma 4.3. *The assignment constraints (1.5) and constraints (3.3) are equivalent under constraints (4.1).*

Lemma 4.4. *The flow conservation constraints (1.6) and constraints (3.4) are equivalent under constraints (4.2).*

Lemma 4.5. *The linking constraints (1.11) and (3.5) are equivalent under constraints (4.2).*

Lemma 4.6. *Constraints (1.13) and (3.8) are equivalent under constraints (4.1).*

Lemmas 4.1. to 4.6., along with the fact that constraints (1.12) and (1.13) are the same as, respectively, constraints (3.6) and (3.7), show us that all the constraints from model $AwFGG3$ are equivalent to constraints from the $ASCF$ model, with the objective functions of the two models being equivalent as well. These lemmas allow us to prove the following result:

Theorem 4.1. *The optimal value for the linear programming relaxation for the $AwFGG3$ model is equal to the optimal value of the linear programming relaxation for the $ASCF$ model.*

Proof. By using relations (4.1) and (4.2), we can construct, from a solution in the space of variables $\{u_{ij}^k, p_i^k\}$ and feasible for the $AwFGG3$ model, a solution in the space of variables $\{h_{ij}, f_{ij}, p_i^k\}$. By substituting expressions (4.1) and (4.2) in constraints (1.7)-(1.10), and considering constraints (3.34) - (3.35), we see that we obtain expressions that are redundant. Due to lemmas 4.1.-4.6. we conclude that this solution is feasible for the $ASCF$ model, and both solutions have the same cost. Consequently, we have that the optimal value for the linear programming relaxation for the $AwFGG3$ model is greater or equal to the optimal value of the linear programming relaxation for the $ASCF$ model.

To prove the converse relation we follow [Gouveia, 1995], and show that from any solution in the space of the variables $\{h_{ij}, f_{ij}, p_i^k\}$ and feasible for the $ASCF$ model, it is possible to construct a solution with the same cost in the space of the variables $\{u_{ij}^k, p_i^k\}$ and feasible for the $AwFGG3$ model. For each arc (i, j), we let $r = \frac{f_{ij}}{h_{ij}}$. If r is integer, then we set $u_{ij}^r = h_{ij}$, and $u_{ij}^k = 0;\ \forall k = 1, ..., T;\ k \neq r$. On the other hand, if r is non-integer, let $a = \lfloor r \rfloor$ and $b = \lceil r \rceil$. Then, we set $u_{ij}^k = 0;\ \forall k = 1, ..., T;\ k \neq a, b$, then, due to expressions (4.1) and (4.2), the values for variables u_{ij}^a and u_{ij}^b can be obtained by solving the system:

$u_{ij}^a + u_{ij}^b = h_{ij};\ au_{ij}^a + bu_{ij}^b = f_{ij}$.

Consequently, and observing that $b = a + 1$, we have:

$u_{ij}^a = bh_{ij} - f_{ij}; u_{ij}^b = f_{ij} - ah_{ij}$.

Note that if $j = 1$, and because we do not consider arcs in the form (i, i), we cannot have $r = 1$ or $a = 1$ without violating (1.8). Similarly, if $i = 1$ we cannot have $r = T$ or $b = T$ without violating (1.10). Therefore, we can conclude that by constructing the solution as shown, constraints (3.34) and (3.35) will always be verified.

Due to lemmas 4.1.-4.6., we can establish that, by using these expressions, from any solution that is feasible for the linear relaxation of the $ASCF$ model we can build a solution that is feasible for the $AwFGG3$ model, and both solutions have the same cost. Therefore, we have that the optimal value for the linear relaxation for the $ASCF$ model is greater or equal to the optimal value for the linear relaxation for the $AwFGG3$ model.

Combining this statement with the previous statement that the optimal value for the linear relaxation for the $AwFGG3$ model is greater or equal to the optimal value of the linear relaxation for the $ASCF$ model we can conclude then that the two optimal values are equal. □

5 Additional methods/models and some computational experiments

In this section of the Appendix we describe in more detail the computational results of some additional methods and models that were mentioned in the paper.

5.1 Experiments with enhanced models: PQ_0 and APQ_0

Table 1 presents the full results of the models PQ_0 and APQ_0. Recall that Table 11, in Section 4, provides the average results per variant, as well as a comparison of the performances of these two models with the performances of models PQ and APQ.

5.2 Experiments with a Miller-Tucker-Zemlin Model

Subramanyam and Gounaris ([Subramanyam and Gounaris, 2016]) have also presented a model based on the Miller-Tucker-Zemlin constraints ([Miller et al., 1960]), denoted by MTZ in the sequel, that can be adapted to the $CTSP$ in

Table 1: Consistency configurations experiment (25% consistent nodes)

	PQ_0						APQ_0					
Instance	Vg	LPg	LPt	Intt	BB	#	Vg	LPg	LPt	Intt	BB	#
med32-v1	*	1.24	2	58	486		*	1.97	0	666	6812	
med32-v2	*	1.08	1	87	1523		*	1.33	0	231	4413	
med32-v3	*	1.16	1	82	1165		*	2.55	0	32	1316	
med32-v4	*	1.09	1	22	546		*	2.61	0	18	1049	
med32-v5	*	0.80	0	6	386		*	1.96	0	94	2477	
med32-v6	*	1.14	1	12	386		*	2.14	0	139	2844	
med32-v7	*	1.31	0	2	262		*	2.63	0	10	1258	
Average	*	1.12	1	38	679	7(7)	*	2.31	0	170	2881	7(7)
med36-v1	*	2.33	5	10170	68277		*	2.77	1	3733	39132	
med36-v2	*	2.31	3	3353	54069		*	2.92	0	3885	24666	
med36-v3	*	2.69	3	4736	50721		*	3.37	0	7919	69209	
med36-v4	*	3.04	2	9331	159179		*	4.15	0	3397	44268	
med36-v5	*	2.27	1	2176	41176		*	2.80	0	3040	29900	
med36-v6	*	2.38	3	2515	40789		*	3.12	0	4027	46809	
med36-v7	*	1.98	0	94	3575		*	3.69	0	553	15870	
Average	*	2.43	2	4625	59684	7(7)	*	3.26	0	3793	38550	7(7)
med40-v1	*	1.48	7	4329	23904		*	2.18	2	6768	36245	
med40-v2	*	1.78	4	3875	33476		*	2.84	1	7939	51526	
med40-v3	*	1.67	5	2498	16613		*	2.61	1	7506	41408	
med40-v4	*	1.76	4	2552	23358		*	2.79	1	4763	49919	
med40-v5	*	2.00	2	2471	40346		*	3.11	0	9335	75404	
med40-v6	*	1.72	3	1234	10392		*	2.59	1	8052	45282	
med40-v7	*	1.25	1	331	9370		*	3.11	0	314	7795	
Average	*	1.67	4	2470	22494	7(7)	*	2.76	1	6382	43940	7(7)
med44-v1	0.52	2.17	30	-	-		0.73	2.31	3	-	-	
med44-v2	0.00	2.32	8	-	-		0.00	2.79	2	-	-	
med44-v3	1.47	2.46	31	-	-		0.90	2.82	1	-	-	
med44-v4	0.00	2.19	11	-	-		0.12	2.96	1	-	-	
med44-v5	*	1.75	3	4155	42398		0.68	2.53	1	-	-	
med44-v6	0.00	2.03	12	-	-		0.38	2.53	1	-	-	
med44-v7	*	1.57	1	1122	20360		*	2.64	1	2150	23679	
Average	0.40	2.07	14	2639	31379	2(7)	0.47	2.65	1	2150	23679	1(7)
ftv33-v1	*	14.50	1	183	2213		*	16.09	0	50	2464	
ftv33-v2	*	16.87	1	292	7954		*	22.37	0	67	2151	
ftv33-v3	*	19.86	1	238	6850		*	21.53	0	10	361	
ftv33-v4	*	17.83	1	304	7939		*	24.19	0	22	995	
ftv33-v5	*	18.48	0	195	9795		*	26.63	0	63	2015	
ftv33-v6	*	20.29	1	150	4984		*	25.99	0	79	1805	
ftv33-v7	*	16.96	0	16	2301		*	21.29	0	32	1689	
Average	*	17.83	1	197	6005	7(7)	*	22.58	0	46	1640	7(7)
ftv35-v1	*	16.51	2	1427	12228		*	18.74	0	662	6012	
ftv35-v2	*	14.43	1	700	11927		*	18.50	0	178	3322	
ftv35-v3	*	17.50	1	827	12913		*	19.53	0	193	3935	
ftv35-v4	*	14.79	1	511	9255		*	19.63	0	232	4538	
ftv35-v5	*	19.37	1	319	8638		*	26.98	0	582	11276	
ftv35-v6	*	19.74	1	620	9167		*	25.73	0	626	11193	
ftv35-v7	*	16.20	0	64	2910		*	20.37	0	137	3003	
Average	*	16.93	1	638	9577	7(7)	*	21.35	0	373	6183	7(7)
ftv38-v1	*	17.21	7	2427	11824		*	19.20	1	975	6156	
ftv38-v2	*	16.46	4	1549	17986		*	21.17	0	332	4662	
ftv38-v3	*	17.97	4	2630	21526		*	21.43	0	741	5237	
ftv38-v4	*	18.55	2	658	10331		*	22.52	0	301	4184	
ftv38-v5	*	20.13	1	1754	32104		*	26.77	0	1305	16686	
ftv38-v6	*	19.88	2	1076	14038		*	24.42	0	1819	16048	
ftv38-v7	*	19.58	0	110	3922		*	26.86	0	327	5525	
Average	*	18.06	3	1451	15817	7(7)	*	23.19	0	829	8357	7(7)
ftv44-v1	0.00	17.00	9	-	-		*	20.57	1	4973	15722	
ftv44-v2	0.00	16.62	7	-	-		*	20.13	1	6411	27553	
ftv44-v3	0.00	17.88	7	-	-		*	21.53	1	8818	34514	
ftv44-v4	0.05	17.41	4	-	-		*	21.83	1	1784	15281	
ftv44-v5	*	15.43	2	2137	34466		*	20.72	0	1690	14170	
ftv44-v6	*	18.35	3	7757	66437		*	23.64	1	7212	39285	
ftv44-v7	*	15.67	1	359	8786		*	20.29	0	574	9237	
Average	0.01	16.91	5	3418	36563	3(7)	*	21.24	1	4495	22252	7(7)
All-solved				3418	36563	3(7)				3159	20897	3(7)
Average	0.23	9.69	4	1734	20707	47(56)	0.47	12.42	0	2295	17806	50(56)
All-solved				1682	20236	46(56)				2017	17331	46(56)

a similar way as the SCF model was. This adapted model is given below:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{l=1}^{m} C_{ij} x_{ij}^{l} \tag{5.1}$$

$$\text{S.t.} \sum_{i=1}^{n} x_{ij}^{l} = S_{j}^{l};\ \forall j = 1, ..., n;\ \forall l = 1, ..., m \tag{5.2}$$

$$\sum_{i=1}^{n} x_{ji}^{l} = S_{j}^{l};\ \forall j = 1, ..., n;\ \forall l = 1, ..., m \tag{5.3}$$

$$w_{i}^{l} - w_{j}^{l} + (T-1)x_{ij}^{l} + (T-3)x_{ji}^{l} \leq T-2;\ \forall l = 1, ..., m;\ \forall i, j \in V^{l};\ i \neq j \tag{5.4}$$

$$w_{j}^{l} - \sum_{i \in NC^{l}} x_{ij}^{l} \geq 1;\ \forall l = 1, ..., m;\ \forall j \in V^{l} \tag{5.5}$$

$$w_{j}^{l} + \sum_{i \in NC^{l}} x_{ji}^{l} \leq T-1;\ \forall l = 1, ..., m;\ \forall j \in V^{l} \tag{5.6}$$

$$w_{j}^{s} = w_{j}^{t};\ \forall s, t = 1, ..., m;\ s \neq t;\ \forall j \in V^{s} \cap V^{t} \tag{5.7}$$

$$x_{ij}^{l} \in \{0, 1\};\ \forall i, j = 1, ..., n;\ \forall l = 1, ..., m \tag{5.8}$$

$$w_{i}^{l} \geq 0 \text{ and integer};\ \forall l = 1, ..., m;\ \forall j \in NC^{l} \tag{5.9}$$

This model uses binary variables x_{ij}^{l} as in the other models, as well as integer variables w_{i}^{l}, which indicate the position of node i in route l. Essentially, the main difference between the MTZ model and the SCF model is that the former associates the position information to a node while the latter associates the information to an arc. As before, we want to minimise total costs (5.1), subject to the standard assignment constraints, (5.2) and (5.3). Constraints (5.4) are an enhanced version of the standard MTZ-based constraints (observe that they guarantee that $w_{j}^{l} = w_{i}^{l} + 1$ if $x_{ij}^{l} = 1$) and were firstly proposed Desrochers and Laporte ([Desrochers and Laporte, 1991]). Constraints (5.5) and (5.6) are, respectively, lifted lower bound and upper bound constraints for the w_{i}^{l} variables. Consistency is modelled through constraints (5.7), whereas constraints (5.8) state that variables x_{ij}^{l} are binary and constraints (5.9) state that variables w_{i}^{l} are integer and non-negative.

We have tested this model and some results are given in Table 2, where we compare the LP relaxation gap, the time needed to solve the LP relaxation, and the time needed to solve the integer problem, for models SCF, MTZ, PQ and APQ. For this comparison we have used some of the test instances from the experiment involving consistency configurations. The results indicate that, in general, the MTZ model provides better LP bounds than the SCF model (the non-dominance between the linear programming relaxations of the SCF and of the MTZ model with the lifted constraints (5.4) is known from the literature for the ATSP and is easily seen to apply here). The results also indicate that with respect to the CPU times to obtain the integer solution, the SCF models is slightly preferable but the most relevant fact is that both the SCF and MTZ models are not competitive, when compared with the PQ and APQ models.

Instance	N consistent	SCF			MTZ			PQ			APQ		
		LPg	LPt	Intt	LPg	LPt	Intt	LPg	LPt	Intt	LPg	LPt	Intt
ftv33	v1	16.95	0	19	13.44	0	31	14.50	2	161	16.09	0	21
	v2	20.04	0	1828	14.93	0	1654	17.66	1	381	23.32	0	14
	v3	22.71	0	333	17.18	0	1162	20.53	1	429	22.12	0	14
	v4	21.27	0	814	18.80	0	511	18.81	1	133	24.48	0	16
	v5	21.29	0	30	11.30	0	40	19.33	0	300	27.91	0	71
	v6	23.71	0	1169	14.70	0	396	21.70	1	69	27.11	0	24
	v7	19.70	0	91	12.00	0	92	18.03	0	69	21.81	0	93
ftv35	v1	18.70	0	2413	12. 94	0	5942	17.19	2	1213	19.37	0	50
	v2	18.54	0	1313	15.39	0	-	15.78	2	1089	19.97	0	32
	v3	20.58	0	6768	17.53	0	-	18.17	2	933	20.05	0	42
	v4	18.99	0	3783	18.36	0	3077	16.61	2	795	20.49	0	160
	v5	23.25	0	3036	17.18	0	5894	21.17	1	1115	28.12	0	1133
	v6	23.05	0	-	18.19	0	-	20.62	1	883	26.46	0	155
	v7	20.04	0	65	15.15	0	253	17.70	0	128	21.80	0	109
med32	v1	1.80	0	-	1.97	0	-	1.28	4	162	2.05	0	41
	v2	1.56	0	-	1.62	0	4184	1.13	2	53	2.33	0	17
	v3	1.61	0	-	1.66	0	-	1.21	2	149	2.55	0	23
	v4	1.56	0	10251	1.69	0	1673	1.18	1	57	2.65	0	12
	v5	1.23	0	595	1.33	0	268	0.94	1	15	2.09	0	21
	v6	1.68	0	-	1.80	0	4640	1.29	1	24	2.16	0	19
	v7	1.78	0	311	1.96	0	282	1.37	0	1	2.71	0	3
med36	v1	2.83	1	-	2.39	0	-	2.63	6	7562	3.11	1	-
	v2	2.49	0	-	2.03	0	-	2.40	4	5159	2.92	0	2696
	v3	2.92	0	-	2.21	0	-	2.88	2	5869	3.37	0	-
	v4	3.33	0	-	2.57	0	-	3.23	3	-	4.15	0	7158
	v5	2.48	0	-	1.88	0	-	2.39	1	3746	2.85	0	4736
	v6	2.78	1	-	2.26	0	-	2.69	3	5514	3.12	0	6079
	v7	2.19	0	-	1.41	0	-	2.00	0	355	3.70	0	86

Table 2: Comparing the MTZ model with the other models

5.3 Testing and discussing exponentially sized sets of constraints

Additionally, and despite this paper being focused on compact formulations, we also implemented a branch and cut method that implicitly adds subtour elimination constraints (SEC)

$$\sum_{i \in (NC^l \cup \{1\}) \setminus G; j \in G} x_{ij}^l \geq 1;\ \forall l = 1, ..., m;\ \forall G \subseteq NC^l \tag{5.10}$$

or

$$\sum_{i \in (NC^l \cup \{1\}) \setminus G; j \in G} \sum_{k=1}^{T} z_{ij}^{kl} \geq 1;\ \forall l = 1, ..., m;\ \forall G \subseteq NC^l \tag{5.11}$$

The constraint separation is modelled in the usual way, by solving several maximum flow problems, each one for a route l and between the depot and node j such that $S_j^l = 1$, in a capacitated graph where the capacity of each arc (i, j) is equal to x_{ij}^{*l} in the fractional solution of the linear programming relaxation.

Instance	|RC|	SCF model			PQ model			APQ model			SCF model + SEC			PQ model + SEC		
		LPg	LPt	Intt	LPg	LPt	Intt	LPg	LPt	Intt	LPg	LPt	Intt	LPg	LPt	Intt
ftv33	0	6.06	0	0	4.81	0	0				0.28	0	0	0.28	2	0
	4	11.35	0	16	8.21	0	20	16.62	0	3	5.81	1	38	3.41	4	6
	8	16.95	0	19	14.50	2	161	16.09	0	21	2.18	1	14	0.47	12	9
ftv35	0	8.45	0	0	7.81	0	2				0.09	0	0	0.09	2	0
	4	12.06	0	33	10.81	0	67	12.20	0	5	4.11	1	13	2.94	3	15
	9	18.70	0	2413	17.19	2	1213	19.37	0	50	6.35	1	10800	5.14	20	848
ftv38	0	6.71	0	0	5.63	0	4				0.51	0	0	0.48	4	8
	5	9.20	0	11	7.04	1	23	11.95	0	10	3.42	1	1131	1.41	7	4
	10	20.23	1	10800	18.40	4	1738	20.42	1	387	8.25	2	10800	6.96	30	10800
ftv44	0	6.68	0	0	5.72	0	4				0.00	1	0	0.00	8	0
	6	12.59	1	917	10.01	2	297	13.21	0	10	6.55	2	10800	4.97	19	6288
	11	20.46	1	10800	20.03	11	10800	21.29	1	1355	10.65	3	10800	8.47	62	10800
med32	0	0.31	0	0	0.26	0	0				0.14	0	0	0.14	1	0
	4	1.12	0	61	0.45	1	1	1.28	0	3	0.99	1	345	0.40	2	5
	8	1.80	0	10800	1.28	4	162	2.05	0	41	1.79	1	10800	1.28	7	1294
med36	0	0.79	0	0	0.76	0	1				0.00	0	0	0.00	3	0
	5	2.20	0	10800	1.90	1	275	2.82	0	24	1.42	1	10800	1.14	8	1126
	9	2.83	1	10800	2.63	6	7562	3.11	1	10800	2.39	3	10800	2.11	41	10800
med40	0	0.19	0	0	0.17	0	0				0.08	0	0	0.08	4	17
	5	1.79	0	10800	1.14	2	78	1.94	0	18	1.69	1	10800	1.08	7	7536
	10	2.00	1	10800	1.48	12	6782	2.18	2	6047	2.00	3	10800	1.47	25	10800
med44	0	0.66	0	0	0.65	0	7				0.00	1	0	0.00	10	0
	6	1.86	1	10800	1.67	4	1547	2.12	1	2602	1.20	3	10800	0.92	31	10800
	11	2.28	1	10800	2.28	18	10800	2.31	1	10800	2.04	7	10800	1.95	172	10800

Table 3: Impact of using subtour elimination constraints in model performance

For this discussion, we have considered two variants, one with the SCF model as base method, the other with the PQ model without any enhancement.

We have included tests where we consider the same instances with no consistency constraints and with 12.5% and 25% of consistent nodes as considered in the paper. The results from this method can be found in Table 3.

As expected, with the inclusion of SECs, the LP bounds are better, much better in the case of the ftv instances. However, the reported CPU times for obtaining the optimal integer solution also indicate that using SECs may not compensate, if our main motivation is to solve instances of the problem under study. With respect to the med instances, the models with SECs produce much worse CPU times than the corresponding models without SECs. With respect to the ftv instances, there are clear improvements but, in almost all of the cases, these improvements are dominated by the results obtained by the APQ model.

We conclude this section of the Appendix by making a reference to the large set of inequalities associated to inconsistent pairs of paths (and enhanced versions) proposed by Subramanyam and Gounaris ([Subramanyam and Gounaris, 2016]). This time-consistent problem does not exactly become the position-consistent problem we are considering, after setting the interval duration to zero, since the "time-length" of the arcs diverging or converging from the depot node 1 are variable, and not fixed to 1. This difference explains why these inequalities cannot be adapted for the new problem for the CTSP with positional consistency (or, at least, their adaptation is far from clear). To clarify, consider an instance with two routes, where node 1 is the depot, nodes 2, 3 and 4 must be visited in both routes (and require consistency) and nodes 5 and 6 are visited only in route 1.

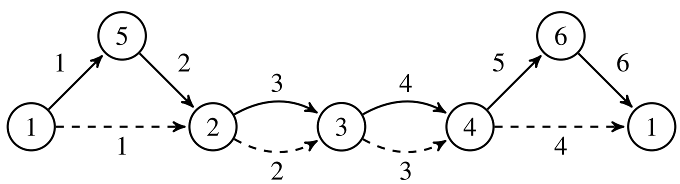

Figure 2: An infeasible solution for the CTSP

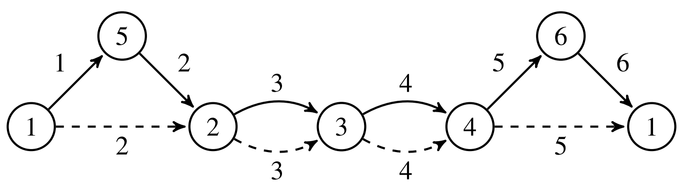

Figure 3: A feasible solution for the CTSP

Figures 2 and 3 show two solutions for this instance. The integer values associated to each arc indicate its position. Figure 2 shows an infeasible solution since none of the consistent nodes verifies the desired consistency. One constraint involving the pair of inconsistency paths $\{(1,5),(5,2)\}$ and $\{(1,2)\}$, and which would cut off this solution is given by $x_{15}+x_{52}+x_{12} \leq 2$. However, the solution shown in Figure 3 is feasible (as shown by the the position information attached to the arcs) and would be cut off by the same inequality. That is, the two solutions are different with respect to the position information associated to the arcs, but in terms of the x_{ij} variables alone these two solutions are not distinguishable.