Exact methods for the Selective Assessement Routing Problem

JOAN SALVÀ SOLER

The Selective Assessment Routing Problem (SARP) addresses the site selection and routing decisions of the rapid needs assessment teams which aim to evaluate the post-disaster conditions of different community groups, each carrying a distinct characteristic. SARP constructs an assessment plan that maximizes the covering of different characteristics in a balanced way. This project explores exact approaches that maximally exploit or extend the capabilities of numerical solvers. Different mathematical formulations are presented, and theoretical results regarding their strength are given. The models are tested on a set of test instances, and the best model is applied to a real-world scenario.

CCS Concepts: • **Mixed Integer Linear Programming**; • **Humanitarian Logistics**;

Additional Key Words and Phrases: Team Orienteering, Vehicle Routing Problem

Supervision:
Vera C. Hemmelmayr, PhD, Wirtschaftsuniversität Wien, main supervisor
Günther Raidl, Ao. Univ. Prof, Technische Universität Wien, co-supervisor

Area	Mathematical Optimization
Student	Joan Salvà Soler, 12223411
Semester	WS 2023/2024
Domain lecture	Humanitarian Logistics
Research group	Institute for Transport and Logistics - WU Wien
Lecture done or in parallel	In parallel

TU WIEN Informatics WU VIENNA

Author's address: Joan Salvà Soler, e12223411@student.tuwien.ac.at.

1 INTRODUCTION

Once a disaster occurs, humanitarian agencies first carry out rapid assessments of the affected region. The objective of the rapid assessment stage is to perform a broad evaluation of the disaster impact and population needs Maya [2013]. The rapid assessment process may start as early as a few hours after the disaster occurrence and is typically completed within three days Maya [2013]. Assessments are conducted by experts, who are familiar with the local context and have specialties such as nutrition, shelter, etc. ACAPS [2011]. A rapid needs assessment process involves "making observations to feel the situation through sounds, smells, and visual impressions" and "conducting interviews by key informants who have prior knowledge of the affected community" IFRC [2008]. The findings of the rapid needs assessment stage help agencies to determine the scale and the scope of the required response, prepare funding appeals, and prioritize needs IFRC [2008]. Therefore, developing effective rapid assessment plans is crucial for achieving successful disaster response. Starting to deliver assistance without a rapid assessment may lead to significant gaps or overlaps in relief efforts and wastes of precious resources Maya [2013].

Since rapid needs assessment stage must be completed quickly, assessment teams do not make observations at all of the affected sites; therefore, sites to be visited are sampled. The general aim of sampling in the rapid needs assessment stage is to choose a limited number of sites that will allow the assessment teams to observe and compare the post-disaster conditions of different community groups (e.g., farmers versus pastoralists). In practice, two sampling methods are mainly used to select sites IFRC [2008]. In random sampling, sites to be visited are selected randomly, which may be reasonable if all community groups in the region are affected by the disaster in a similar way. However, random sampling may not produce accurate findings if the post-disaster conditions change throughout the affected region. Therefore, relief organizations usually implement a purposive sampling strategy, in which only sites that carry certain characteristics are visited ACAPS [2011]. The steps of developing an assessment plan by using purposive sampling are as follows ACAPS [2011]:

- *Specify the critical characteristics.* The critical characteristics that define the target community groups are identified. These characteristics may be related to geographical (e.g., topography, altitude), demographical (e.g., age, gender), socio-economical (e.g., economic activity, literacy), and socio-cultural (e.g., ethnicity, language) aspects ACAPS [2011]. In general, it is important to avoid considering too many characteristics and to focus on the most critical ones ACAPS [2011].
- *Site selection.* A large number of sites may not be visited within the assessment period by using the available resources (such as staff and vehicles). While selecting sites, agencies mainly consider: (i) sample richness (i.e., observing each community group at least once is important), (ii) collecting adequate information (i.e., observing a community group at several sites is desirable to understand the situation better), and efficiency (e.g., visiting a site that involves multiple community groups may be beneficial).
- *Logistical planning.* Selected sites are assigned to teams and the order of visits to the sites for each team is determined. If feasible assessment routes cannot be constructed, the steps above are repeated to reduce the number of critical characteristics and/or the number of selected sites.

The above trial-and-error process may be time-consuming and may not guarantee obtaining an effective assessment plan. In this study, we focus on developing a systematic method that can be used by relief agencies to determine site selection and vehicle routing decisions simultaneously while implementing purposive sampling. We assume that the critical characteristics that define community groups are specified in advance. Each community group is identified by a single characteristic, and each site may involve different community groups. Given the set of critical characteristics carried by each site, Balcik [2017] defines the Selective Assessment Routing Problem (SARP) as the problem of determining: (i) the sites to visit and (ii) the vehicle routes (i.e., the assignment of the selected sites to the assessment teams and the order of visits to the sites by each team), while ensuring sufficient coverage of the critical characteristics within the assessment period.

Defining and measuring coverage is not straightforward in this context. In this study, we take a first step and define a coverage objective to address the site selection criteria considered in practice. Specifically, in SARP, we define a coverage objective, which maximizes the minimum coverage ratio across the critical characteristics; the coverage ratio of a characteristic is calculated by dividing the number of times that the characteristic is covered in the assessment plan by the total number of sites in the network carrying that characteristic. This objective aims to cover each critical characteristic at least once in order to ensure sample richness; and if the duration of the assessment period allows one to make further observations, additional sites are selected so that the critical characteristics are covered in a balanced way, which facilitates collecting adequate information while ensuring diversity. The SARP involves route duration constraints, which define the assessment deadline and make visiting sites that involve multiple characteristics desirable.

Execution time and solution quality a key elements to take into consideration in the context of rapid needs assessment after a humanitarian disaster, and it could be contended that it should be prioritized before the quality of the solution. However, we argue that it is still advantageous to define algorithms that can prove optimality despite the additional time investment. This approach yields lower bounds that can serve as benchmarks for subsequent research, enabling the evaluation of solutions provided by approximate methods. Moreover, working on exact methods for the SARP also offers potential inspiration for the development of exact methods in related problems like the Team Orienteering Problem (TOP). With this motivation, our work will study the SARP problem from an exact perspective.

In Section 2, we review the relevant literature. In Section 3, we present four different mathematical formulations for the SARP problem. Section 4 contains prepositions and their proofs related to the strength relationship of the proposed formulations. We perform some problem-specific analysis in Section 5 that gives some insightful findings. We discuss improvements to one of the formulations in Section 6. Section 7.2 contains the computational results, and Section 8 solves a case study. Finally, we conclude and discuss future work in Section 9.

2 LITERATURE REVIEW

The most important reference of our work is naturally Balcik [2017] since it first defines the SARP problem. It also provides a mathematical MTZ formulation that is solved using CPLEX, and defines a Tabu Search heuristic that provides much better objectives. Our work is an extension of theirs because we take the same problem and instances, and propose new algorithms for it. Another paper that considers the SARP problem with the same definition is Hakimifar et al. [2022], which presents a heuristic with the ability to distinguish the "best" solutions among those with the same objective.

It is helpful to study our problem getting inspiration from related problems. SARP is naturally a variant of the Vehicle Routing Problem (VRP), which finds the cost-minimizing routes that visit all nodes in a given network. However, in some applications, visiting all of the nodes in the network may not be possible due to logistical concerns, and the goal becomes to visit the most profitable nodes in the network. Depending on how the objectives of maximizing the profits collected from the visited nodes and minimizing travel costs are addressed, different routing problems are defined in the literature; Feillet et al. [2005] call these class of problems “routing problems with profits”. For instance, Allahviranloo et al. [2014] motivate using a selective routing problem to support humanitarian relief distribution; since demand usually exceeds available supply in relief operations, prioritizing critical nodes is essential in these settings. Allahviranloo et al. [2014] focus on a setting where the utility to be gained from each visited node is uncertain and develop solution methods to solve the formulated stochastic selective routing problem.

The closest problem to SARP is the Team Orienteering Problem (TOP), where the objective is to maximize the total profit collected from the visited locations while constraining the routes to a maximum duration. It was first introduced by Butt and Cavalier [1994], and Golden et al. [1987] proves it is an NP-hard problem. There is rich

literature focusing on exact and heuristic solution approaches for TOP (Archetti et al. [2014a] is an extensive review of them). Boussier et al. [2007] and Poggi et al. [2010a] are examples of work that focus on exact algorithms, and their models will be of inspiration for our formulations.

The SARP differs from the TOP by its objective function. While the TOP maximizes total collected profits, the SARP maximizes the minimum coverage ratio. Further, TOP and SARP differ in terms of the benefits that can be collected by visiting a node; in TOP, the benefit from visiting a site is defined by a single measure (e.g., profit, revenue, etc.), whereas in SARP, it is difficult to quantify the benefits of visiting a particular site without knowing the characteristics of the other sites that may be included in the assessment plan. Therefore, existing solution algorithms developed for the TOP cannot be directly applied to solve the SARP, and there does not seem to be a natural function that maps instances of SARP to instances of TOP. However, the inverse direction is very straightforward. Defining the set of characteristics to be just one enables the use of any SARP algorithm to solve the TOP.

3 MATHEMATICAL FORMULATIONS

In this section we present four mathematical formulations that model the SARP problem. They all lead to Mixed-Integer Linear Programs. One formulation already was considered in the literature (MTZ-3), and in this work we propose a Single Commodity Flow formulation (SCF), a Cutting Set exponential formulation (CutSet), and a 2-index adaptation of MTZ-3 (that refer to as MTZ-2).

Notation. The following notation is common for all formulations. N is the set of affected sites. K is the set of vehicles (or teams), and each team $k \in K$ departs from the origin node $\{0\}$ and returns back to the origin after completing the route. the travel time between nodes is represented by $t_{ij},\ \forall i, j \in N_0$. Each vehicle is allowed to travel at most $T_{\max}$ time units. Let C denote the set of critical characteristics of interest. A coverage parameter α_{ik} is defined, which takes value 1 if node $i \in N$ carries characteristic $c \in C$, and 0 otherwise. The total number of sites that carry characteristic $c \in C$ is represented by τ_c.

3.1 MTZ - 3 index

This formulation is introduced in Balcik [2017]. We call it MTZ because it uses the well-known Miller Tucker Zemlin constraints for cycle elimination Miller et al. [1960]. We will also refer to it as MTZ-3index or MTZ-3 because the edge variables have a triple index. The formulation uses the following variables:

- x_{ijk} takes value 1 if vehicle $k \in K$ visits site $j \in N_0$ right after $i \in N_0$, and 0 otherwise.
- y_{ik} takes value 1 if vehicle $k \in K$ visits site $i \in N_0$, and 0 otherwise.
- z is the continuous variable a maximize that models the coverage rate. It is bounded by 0 and 1.
- u_i is a continuous variable that defines a sequence on the visited nodes $i \in N$ that is valid for all routes.

$$\max z \tag{1}$$

$$\text{s.t. } z \cdot \tau_c \leq \sum_{i \in N} \sum_{k \in K} \alpha_{ic} \cdot y_{ik} \qquad \forall c \in C \tag{2}$$

$$\sum_{j \in N_0} x_{ijk} = y_{ik} \qquad \forall i \in N_0, k \in K \tag{3}$$

$$\sum_{j \in N_0} x_{jik} = y_{ik} \qquad \forall i \in N_0, k \in K \tag{4}$$

$$\sum_{k \in K} y_{ik} \leq 1 \qquad \forall i \in N \tag{5}$$

$$\sum_{k \in K} y_{0,k} = |K| \tag{6}$$

$$\sum_{i \in N_0} \sum_{j \in N_0} t_{ij} x_{ijk} \leq T_{\max} \qquad \forall k \in K \tag{7}$$

$$u_i - u_j + |N| \cdot x_{ijk} \leq |N| - 1, \qquad \forall k \in K, i \neq j \in N \tag{8}$$

$$x_{iik} = 0 \qquad \forall i \in N, k \in K \tag{9}$$

$$x_{ijk} \in \{0, 1\} \qquad \forall i, j \in N_0, k \in K \tag{10}$$

$$y_{ik} \in \{0, 1\} \qquad \forall i \in N, k \in K \tag{11}$$

$$0 \leq z \leq 1 \tag{12}$$

$$0 \leq u_i \leq |N| \qquad \forall i \in N \tag{13}$$

The objective function (1) maximizes the minimum coverage ratio, which is defined through constraints (2). Constraints (4) and (3) ensure that an arc enters and leaves the depot and each selected site. Constraints (5) guarantee that each site is visited at most once. Constraint (6) limits the number of routes by the available number of teams. Constraints (7) ensure that each route is completed within the allowed duration. Constraints (8) are for eliminating subtours (adapted from Miller et al. [1960]); note that when $x_{ijk} = 0$, these constraints are not binding, while they force $u_j \geq u_i + 1$ when $x_{ijk} = 1$. In this way, nodes are sequenced by avoiding subtours but allowing total tours containing the depot node. Constraints (9) avoid vehicles staying at one node. Finally, constraints (10)-(13) define x, y as binary variables, and constraints z and u as continuous variables with the respective bounds.

3.2 SCF

We propose a two-index Single Commodity Flow formulation. It uses the following variables:

- f_{ij} is the flow circulating from $i \in N_0$ to $j \in N_0$.
- x_{ij} takes value 1 if some vehicle visits site $j \in N_0$ right after $i \in N_0$, and 0 otherwise.
- y_i takes value 1 if some vehicle visits site $i \in N$, and 0 otherwise.
- z is the continuous variable a maximize that models the coverage rate. It is bounded by 0 and 1.

$$\max z \tag{14}$$

$$\text{s.t. } z \cdot \tau_c \leq \sum_{i \in N} \alpha_{ic} \cdot y_i \qquad \forall c \in C \tag{15}$$

$$\sum_{i \in N_0} x_{0i} = |K| \tag{16}$$

$$\sum_{i \in N_0} x_{i0} = |K| \tag{17}$$

$$f_{ij} \leq T_{\max} \cdot x_{ij} \qquad \forall i \in N_0, j \in N_0 \tag{18}$$

$$f_{i0} = x_{i0} \cdot t_{i0} \qquad \forall i \in N \tag{19}$$

$$\sum_{j \in N_0} (f_{ji} - f_{ij}) = \sum_{j \in N_0} x_{ji} \cdot t_{ji} \qquad \forall i \in N \tag{20}$$

$$\sum_{j \in N_0} (x_{ji} + x_{ij}) = 2y_i \qquad \forall i \in N \tag{21}$$

$$\sum_{j \in N_0} x_{ji} = y_i \qquad \forall i \in N \tag{22}$$

$$\sum_{j \in N_0} x_{ij} = y_i \qquad \forall i \in N \tag{23}$$

$$x_{ii} = 0 \qquad \forall i \in N_0 \tag{24}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in N_0 \tag{25}$$

$$y_i \in \{0, 1\} \qquad \forall i \in N \tag{26}$$

$$0 \leq z \leq 1 \tag{27}$$

$$0 \leq f_{ij} \leq |T_{\max}| \qquad \forall i, j \in N \tag{28}$$

As in MTZ-3, the objective function (14) maximizes the minimum coverage ratio, which is defined through constraints (15). Constraints (16) and (17) ensure that $|K|$ vehicles leave and enter the depot. Constraints (18) limit the maximum flow at one edge, while forcing it to be zero when the edge is not selected. The SCF model considers the depot a source which can generate at most $|K|$ paths of at most $T_{\max}$ units. A flow path is forced to go back to the depot consuming all units (19). For every visited node j in a path, the node consumes t_{ij} units of flow, where i is the previous visited node in the path (20). Constraints (21)-(23) guarantee that each site is visited at most once. Constraints (24) avoid vehicles staying at one node. Finally, constraints (25)-(28) define x, y as binary variables, and z and f as continuous variables with the respective bounds.

3.3 Cutting Sets

We propose an exponential-sized formulation that replaces the MTZ constraints with Cutting Set constraints, a common modelling approach considered, for instance, in Archetti et al. [2014b], and also implemented for SARP by Balcik [2017]. The variables used in this formulation are x, y, and z as in MTZ-3, and the constraints are all the same except for (8), which is replaced by (31).

- x_{ijk} takes value 1 if vehicle $k \in K$ visits site $j \in N_0$ right after $i \in N_0$, and 0 otherwise.
- y_{ik} takes value 1 if vehicle $k \in K$ visits site $i \in N_0$, and 0 otherwise.
- z is the continuous variable a maximize that models the coverage rate. It is bounded by 0 and 1.

$$\max \ z \tag{29}$$

$$\text{s.t. (2) - (7), (9) - (13)} \tag{30}$$

$$\sum_{(i,j)\in\delta^+(S)} x_{ijk} \geq y_{hk}, \qquad \forall S \subset N, h \in S, k \in K \tag{31}$$

$$\tag{32}$$

The set of constraints (31) grows exponentially with the size of N. In order to progressively add constraints to the initial model, we propose the following separation procedure to be applied at each node of the Branch & Bound tree that finds an integer solution.

Separation procedure. For each vehicle $k \in K$, we consider a copy of the original network that has edges ij with capacity x_{ijk}, and call it G^*. For each node $i \in N$ in G^*, we calculate the Max-Flow f between the depot 0 and i. If $f < y_{ik}$, we find a Min-Cut partition of G^*, we take S as the side that contains i, and we finally add the CutSet constraint (31) for S. The pseudo-code is provided in Algorithm **??**.

3.4 MTZ - 2 index

This formulation adapts the MTZ-3 formulation to drop the vehicle subindex. We also refer to it as MTZ-2 or MTZOpt. The formulation uses the following variables:

- x_{ij} takes value 1 if some vehicle visits site $j \in N_0$ right after $i \in N_0$, and 0 otherwise.
- y_i takes value 1 if some vehicle visits site $i \in N$, and 0 otherwise.
- z is the continuous variable a maximize that models the coverage rate. It is bounded by 0 and 1.
- u_i is a continuous variable that models the time at which a vehicle visits site $i \in N_0$. In the origin, u_0 models the last time that a vehicle returns to the depot.

$$\max z \tag{33}$$

$$\text{s.t. } z \cdot \tau_c \leq \sum_{i \in N} \alpha_{ic} \cdot y_i \qquad \forall c \in C \tag{34}$$

$$\sum_{j \in N_0} x_{ij} = y_i \qquad \forall i \in N \tag{35}$$

$$\sum_{j \in N_0} x_{ji} = y_i \qquad \forall i \in N \tag{36}$$

$$\sum_{i \in N} x_{0i} = |K| \tag{37}$$

$$\sum_{i \in N} x_{i0} = |K| \tag{38}$$

$$u_j - u_i \geq x_{ij}(T_{\max} + t_{ij}) - T_{\max}, \qquad \forall i \in N, j \in N_0, i \neq j \tag{39}$$

$$x_{ii} = 0 \qquad \forall i \in N \tag{40}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in N_0 \tag{41}$$

$$y_i \in \{0, 1\} \qquad \forall i \in N \tag{42}$$

$$0 \leq z \leq 1 \tag{43}$$

$$x_{0i} \cdot t_{0i} \leq u_i \leq T_{\max} \qquad \forall i \in N_0 \tag{44}$$

The objective function (34) maximizes the minimum coverage ratio, which is defined through constraints (34). Constraints (35) and (36) ensure that an arc enters and leaves the depot and each selected site. Constraints (5) guarantee that each site is visited at most once. Constraints (37) and (38) limit the number of routes by the available number of teams. Constraints (39) are eliminate subtours; note that when $x_{ijk} = 0$, these constraints are not binding, while they force u to model a time sequence that accounts for travel times: $u_j \geq u_i + t_{ij}$ when $x_{ijk} = 1$. Constraints (40) avoid vehicles staying at one node. Finally, constraints (41)-(44) define x, y as binary variables, and constraints z and u as continuous variables with the respective bounds.

4 THEORETICAL STUDY

In this section we mathematically compare the strength of the four presented formulations: SCF, CutSet, MTZ-2, and MTZ-3. This kind of analysis had not beed addressed yet in the literature. The figure below summarizes our findings:

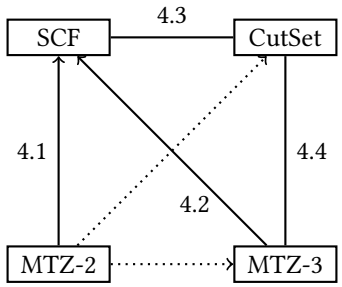

Fig. 1. B stronger than A is noted by $A \longrightarrow B$. A and B incomparable is noted by $A - B$. Dashed lines or arrows indicate that the we did not prove the relation, but experimental tests strongly suggest could be true. The edges refer to the proposition that proves the relationship.

Our study proves that our Single Commodity Flow formulation now becomes the theoretically strongest formulation for SARP existing in the literature, thanks to proposition 4.2. MTZ-2 is weaker than SCF and our computational experiments show that is probably also weaker than Cutset and MTZ, although we didn't focus on proving these relationships. Our experiments consisted in solving the relazations for >2000 randomly generated instances and comparing the optimal values. The CutSet formulation is not comparable to SCF nor MTZ-3, suggesting that the CutSet constraints could be make this two compact formulations even tighter.

Proposition 4.1 (SCF vs MTZ-2). *SCF is stronger than MTZ-2, i.e., $P_{SCF} \subset P_{MTZ-2}$ and they are not equal.*

Proof of 4.1: We project the two formulations to the common space $(\dots y_i \dots, \dots x_{ij} \dots)$. We start with an SCF solution and define

$$u_0 = T_{\max} \tag{45}$$

$$u_i = T_{\max} + \sum_{k \in N_0} x_{ki} t_{ki} - \sum_{k \in N_0} f_{ki}, \qquad i \in N, \tag{46}$$

which gives a solution for MTZ-2. The goal is to see that it is feasible for MTZ-2, i.e., that it satisfies (39) and (44). The remaining MTZ-2 constraints are already satisfied because x_{ij}, y_i define a feasible VRP solution.

$$u_j - u_i \geq x_{ij} t_{ij} - T_{\max}(1 - x_{ij}), \qquad \forall i \in N, j \in N_0$$

$$x_{0i} t_{0i} \leq u_i \leq T_{\max}, \qquad \forall i \in N_0.$$

Constraint (44) is easy to check:

$$u_i = T_{\max} + \sum_{k \in N_0} x_{ki} t_{ki} - \sum_{k \in N_0} f_{ki} \underset{(18)}{\geq} T_{\max} + \underbrace{\sum_{k \in N_0} x_{ki} t_{ki}}_{\geq x_{0i} t_{0i}} - T_{\max} \geq x_{0i} t_{0i}$$

$$u_i = T_{\max} + \sum_{k \in N_0} x_{ki} t_{ki} - \sum_{k \in N_0} f_{ki} \underset{(20)}{=} T_{\max} - \sum_{k \in N_0} f_{ik} \leq T_{\max}$$

For proving (39), we start with the case $j = 0$.

$$u_0 - u_i = -\sum_{k\in N_0} x_{ki}t_{ki} + \sum_{k\in N_0} f_{ki} \underset{(20)}{=} \sum_{k\in N_0} f_{ik} = f_{i0} + \sum_{k\in N} f_{ik} \underset{(19)}{=} x_{i0}t_{i0} + \sum_{k\in N} f_{ik}$$

Since $\sum_{k\in N} f_{ik} \geq 0$ and $-T_{\max}(1 - x_{ij}) \leq 0$, (39) holds for $j = 0$.

We now assume $j \neq 0$.

$$u_j - u_i = \underbrace{\sum_{k\in N_0} x_{kj}t_{kj}}_{\geq x_{ij}t_{ij}} - \sum_{k\in N_0} x_{ki}t_{ki} + \sum_{k\in N_0} f_{ki} - \sum_{k\in N_0} f_{kj} \underset{(20)}{\geq}$$

$$\geq x_{ij}t_{ij} + \underbrace{\sum_{k\in N_0} f_{ik}}_{\geq f_{ij}} - \sum_{k\in N_0} f_{kj} \geq x_{ij}t_{ij} - \sum_{k\in N_0:k\neq i} f_{kj} \underset{(18)}{\geq}$$

$$\geq x_{ij}t_{ij} - T_{\max} \sum_{k\in N_0:k\neq i} x_{kj} \geq x_{ij}t_{ij} - T_{\max}(1 - x_{ij}).$$

In the last inequality, we used (22):

$$\sum_{k\in N_0} x_{kj} = y_j \leq 1 \iff \sum_{k\in N_0:k\neq i} x_{kj} \leq 1 - x_{ij}.$$

This proves that $P_{SCF} \subset P_{MTZ-2}$. To see that they are not equal, consider the following example:

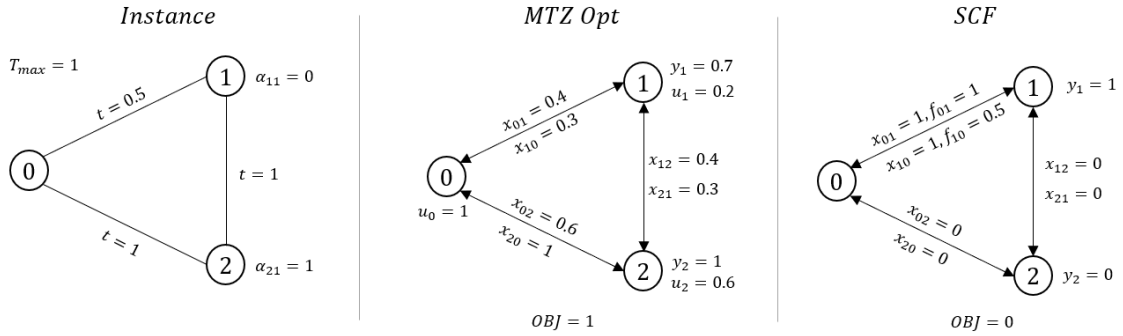

Fig. 2. On the left, the instance presented, which only considers one characteristic: $C = \{0\}$. In the middle, the MTZ Opt and SCF solutions to the respective Linear Relaxations

.

The optimal LR of the MTZ-2 formulation gives a worse bound than the LR of the SCF formulation, which proves $P_{SCF} \neq P_{MTZ-2}$. □

Proposition 4.2 (SCF vs MTZ-3). *SCF is stronger than MTZ-3, i.e., $P_{SCF} \subset P_{MTZ-3}$ and they are not equal.*

Proof of 4.2: We start with an SCF solution that we note as $(...x_{ij}..., ...y_i..., ...f_{ij}...)$, and we will build an MTZ-3 noted as $(...x_{ijk}..., ...y_{ik}..., ...u_i...)$. The x and y variables have different subindices in each formulation, so there

should be no confusion regarding the formulation they belong to.

We define the MTZ-3 solution as follows:

$$x_{ijk} = \frac{x_{ij}}{|K|}, \qquad i, j \in N_0, k \in K \tag{47}$$

$$y_{ik} = \sum_{j \in N_0} x_{ijk}, \qquad i \in N_0, k \in K \tag{48}$$

$$\tilde{u}_i = T_{\max} + \sum_{k \in N_0} x_{ki} t_{ki} - \sum_{k \in N_0} f_{ki}, \qquad i \in N, \tag{49}$$

$$u_1, \dots u_n = F(\tilde{u_1}, \dots \tilde{u_n}) \tag{50}$$

Note that (48) is defined based on (47). Definition (50) uses F, a map that enumerates its arguments based on its relative order, starting from 1. For example, $F(0.3, 0.1, 0.6) = (2, 1, 3)$.

We need to prove that the definition above corresponds to an MTZ-3 solution, i.e, we have to check constraints (2)-(13)

Before we start, a useful result is the following:

$$\sum_{k \in K} y_{ik} = \sum_{k \in K} \sum_{j \in N_0} x_{ijk} = \sum_{k \in K} \sum_{j \in N_0} \frac{x_{ij}}{|K|} \underset{(23)}{=} y_i \tag{51}$$

We start with (2):

$$z \cdot \tau_c \leq \sum_{i \in N} \alpha_{ic} \cdot y_i \underset{(23)}{=} \sum_{i \in N} \alpha_{ic} \cdot y_i \underset{(51)}{=} \sum_{i \in N} \sum_{k \in K} \alpha_{ic} \cdot y_{ik}.$$

(3) is true by definition. We again use the proved relationship (51) to prove (4):

$$\sum_{k \in K} \sum_{j \in N_0} x_{jik} = \sum_{k \in K} \sum_{j \in N_0} \frac{x_{ji}}{|K|} \underset{(22)}{=} y_i = \sum_{k \in K} y_{ik}.$$

(5) follows naturally from (51):

$$\sum_{k \in K} y_{ik} = y_i \leq 1$$

(7) is less trivial and uses the fact that $\sum_{i \in N_0} (f_{ji} - f_{ij}) \leq 0$, which states that nodes consume flow. This can be easily derived from (20).

$$\sum_{i \in N_0} \sum_{j \in N_0} x_{ijk} t_{ij} = \frac{1}{|K|} \sum_{i \in N_0} \sum_{j \in N_0} x_{ij} t_{ij} = \frac{1}{|K|} \sum_{i \in N_0} \sum_{j \in N_0} x_{ji} t_{ji} \underset{(20)}{=} \frac{1}{|K|} \left(\sum_{i \in N_0} x_{0i} t_{0i} + \sum_{i \in N_0} \sum_{j \in N} (f_{ji} - f_{ij}) \right) \leq$$

$$\underset{(19)}{\leq} \frac{1}{|K|} \left(T_{\max} \underbrace{\sum_{i \in N_0} x_{0i}}_{|K|} + \sum_{j \in N} \underbrace{\sum_{i \in N_0} (f_{ji} - f_{ij})}_{\leq 0} \right) \leq \frac{1}{|K|} \cdot T_{\max} |K| = T_{\max}$$

We finally need to prove the MTZ constraint (8). The proof of proposition 4.1 shows that given an SCF solution, the definition of visiting times like (49) satisfies the MTZ-2 constraint. Therefore, our $\tilde{u}_i$ values satisfy:

$$\tilde{u}_i - \tilde{u}_j + x_{ij} t_{ij} \leq T_{\max}(1 - x_{ij}), \forall i, j \in N$$

The choice of the function F implies that the mapping of the times $\tilde{u}_i$ to integer ordering u_i satisfies

$$u_i - u_j + 1 \leq |N|(1 - x_{ijk}).$$

This proves that $P_{SCF} \subset P_{MTZ-3}$. To see that they are not equal, the same example as in 4.1 works:

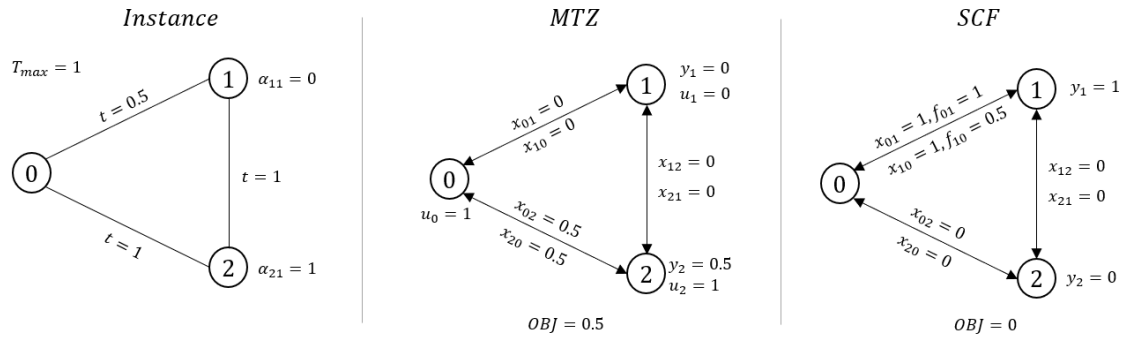

Fig. 3. On the left, the instance presented. In the middle, the MTZ-3 and SCF solutions to the respective Linear Relaxations

.

The optimal LR of the MTZ-3 formulation gives a worse bound than the LR of the SCF formulation, which proves $P_{SCF} \neq P_{MTZ-3}$. □

Proposition 4.3 (SCF vs CutSet). *SCF and CutSet are not comparable, i.e., $P_{SCF} \not\subset P_{CutSet}$ and $P_{Cutset} \not\subset P_{SCF}$.*

Proof of 4.3: Consider the following examples.

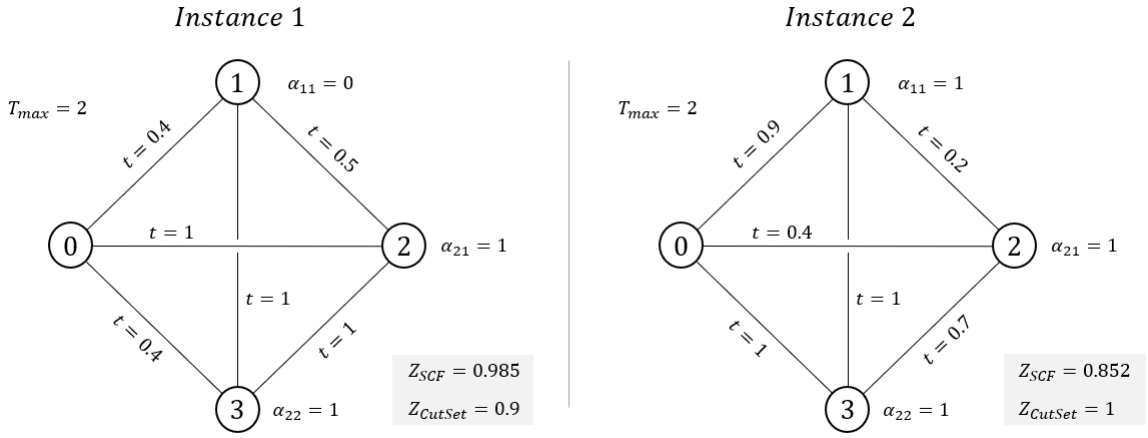

Fig. 4. Instance 1 shows that $P_{SCF} \not\subset P_{CutSet}$. Instance 2 shows $P_{Cutset} \not\subset P_{SCF}$.

Proposition 4.4 (MTZ-3 vs CutSet). *MTZ-3 and CutSet are not comparable, i.e., $P_{MTZ-3} \not\subset P_{CutSet}$ and $P_{Cutset} \not\subset P_{MTZ-3}$.*

Proof of 4.4: Consider the following examples.

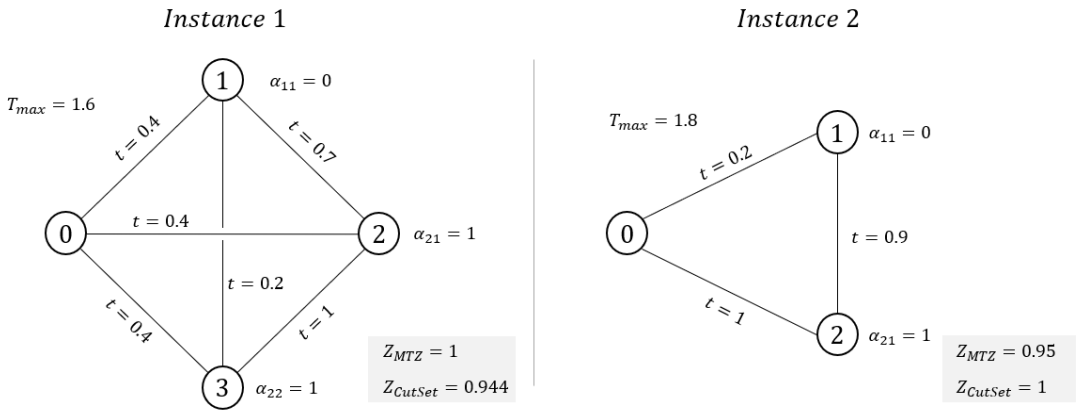

Fig. 5. Instance 1 shows that $P_{MTZ-3} \not\subset P_{CutSet}$. Instance 2 shows $P_{Cutset} \not\subset P_{MTZ-3}$.

5 PROBLEM-SPECIFIC ANALYSIS

In this section we want to investigate the general instance difficulty. Difficulty can be well measured by how fast can we reach low optimality gaps. When are we more likely to prove optimality? Are there instances intrinsically "harder" than others?

5.1 Effect of parameter magnitudes

As explained in 7.1, our 48 large networks define two classes of problems each. The first class has smaller travel times, with an average of 1.3 while the second one has an average of 31.8. We compare the performance of the solver using the SCF and MTZ-3 formulations, but all others behave in a similar way.

	% inst. prove opt.		**% inst. improve TS**		**Opt. Gap %**	
Class type	MTZ-3	SCF	MTZ-3	SCF	MTZ-3	SCF
1	16,67%	37,50%	0,00%	14,58%	77,10%	32,93%
2	25,00%	45,83%	14,58%	31,25%	38,45%	9,71%

Table 1. We show the amount of times that we prove optimality (% inst. prove opt.), and the average optimality gap (Opt. Gap %).

It is clear that better results are obtained on the class 2 instances. Additionally, a Wilcoxon signed-rank test on the paired Opt. Gap % results gives a p-value of 0.0064, which allows us to conclude that the distribution of the gap of the two instances is significantly different. Table 1 shows how the average optimality gap is reduced, and the number of instances that can be solved to optimality under the time limit increases. The results suggest a data processing technique that could speed up the solver for instances with small travel times (average 1): scaling the travel times and $T_{\max}$ to obtain an equivalent network with higher travel times. The optimization results would need to be translated to the original magnitudes accordingly.

5.2 Difficulty of instance

We propose the following experiment in order to study how the maximum route duration and fleet size parameters affect the difficulty of the instance. For this purpose we propose the following experiment: we consider the instance `large_RC50_K4T5` and solve the problem for multiple choices of $T_{\max}$ and $|K|$.

Distance to 0 - 1		$T_{\max}$						
		1	2	3	4	5	6	7
$\|K\|$ **Fleet size**	1	0,000	0,000	0,000	0,100	0,200	0,286	0,429
	2	0,000	0,000	0,214	0,400	0,400	0,357	0,091
	3	0,000	0,000	0,357	0,429	0,182	0,000	0,000
	4	0,000	0,000	0,357	0,286	0,000	0,000	0,000
	5	0,000	0,000	0,357	0,071	0,000	0,000	0,000
	6	0,000	0,000	0,429	0,071	0,000	0,000	0,000

Table 2. The distance from the best objective found to 0 or 1 is measured: $\min(z, 1 - z)$.

Solve time		$T_{\max}$						
		1	2	3	4	5	6	7
$\|K\|$ **Fleet size**	1	0,09	3,05	191,98	410,13	3417,65	2358,06	922,68
	2	0,13	35,05	3600,13	3600,25	3600,54	3600,25	3600,20
	3	0,14	1064,12	3600,22	3600,20	3600,11	1347,63	199,09
	4	0,15	1463,84	3600,16	3600,10	62,85	9,45	10,72
	5	0,13	1276,51	3600,22	3600,28	14,00	7,80	1,78
	6	0,13	3600,04	3600,17	3600,33	2,40	9,65	7,27

Table 3

Opt. Gap		$T_{\max}$						
		1	2	3	4	5	6	7
$\|K\|$ **Fleet size**	1	0,000	0,000	0,000	0,000	0,000	0,000	0,000
	2	0,000	0,000	0,185	0,350	0,071	0,226	0,100
	3	0,000	0,000	0,441	0,301	0,222	0,000	0,000
	4	0,000	0,000	0,800	0,400	0,000	0,000	0,000
	5	0,000	0,000	1,262	0,077	0,000	0,000	0,000
	6	0,000	0,000	1,115	0,077	0,000	0,000	0,000

Table 4

Tables 2 - 4 make very clear that there exists a correlation between the optimal objective and the difficulty. We conclude that the instance can be solve very fast to optimality when the optimal objective is either 0 or 1. The difficulty (Opt. Gap and Solve time) increase when the objective value is far from both 0 and 1.

Naturally, the optimal value being 0 or 1 is related to the parameters $T_{\max}$ and $|K|$. Few teams or a short route duration do not allow to visit all characteristics, thus implying $z = 0$. Many teams or long route durations allow the existance of routes that visit all sites, implying $z = 1$. In this two extreme cases, the solver has half of the job done. In the case $z = 0$, finding an integer solution is trivial because there is no need to leave the depot, and the only work remaining is lowering the upper bound until 0. On the other hand, $z = 1$ all relaxations satisfy $z_{LR} \leq 1$, so the best dual bound will be available during the whole optimization from the root node of the Branch and Bound tree. The only job remaining is finding the integer solution that visits all sites.

6 IMPROVEMENTS TO THE SCF MODEL

Among the 4 basic formulations proposed in Section 3, the best-performing one was SCF (this will be discussed in Section 7). A consequent research question was: can we make some improvements to this model in order to speed up the solver's times? This section discusses two different improvement approaches (6.1, 6.2), and Section 6.3 describes the experimental results.

6.1 Dual strength

We want have strengthen the formulation using families of valid inequalities with the promise that could provide better relaxation bounds and therefore speed up solvers.

In Poggi et al. [2010b], a family of cuts named Triangle Clique Cuts is considered for the Team Orienteering Problem. While promising, we decided to focus on the Cut-Set inequalities, which we have already explored as a single formulation and has been already studied in SARP-related literature Balcik [2017].

As we previously proved in 4.3, SCF and Cut-Set are not comparable, which suggests that a formulation that includes all SCF and Cut-Set constraints could be stronger than the two. However, the constraints of the Cut-Set formulation are defined in a 3-index way, which makes the comparison with SCF not straightforward and was not studied.

The translation of the Cut-Set constraints (31) to a 2-index formulation results in:

$$\sum_{(i,j)\in\delta^+(S)} x_{ij} \geq y_h, \qquad S \subset N, h \in S \tag{52}$$

The separation procedure is analog to the one for the 3-index case the (3.3), with the only change that we do not iterate over the vehicles. Instead, we consider a copy of the original network with capacities x_{ij} and perform the Max-Flow and Min-Cut routine in the same way. The pseudo-code is provided in Algorithm **??**.

There are two natural ways in which we could use this family of constraints:

- Add a small, polynomial-sized subset of Cut-Set constraints to the base model and solve it. We consider two extended formulations, one with all subsets S such that $|S| = 2$, and the other with $|S| = 3$.
- Add constraints sequentially using the defined separation procedure during the optimization process. We will do at each node where an integer or fractional solution has been found.

6.2 Greedy heuristic for initial solutions

Another classic approach for speeding up an exact MIP solver is providing initial feasible solutions that set the incumbent at the very first branching node. However, the improvement is not guaranteed, and the effect could vary among problems, instances, formulations...

We define a Greedy Heuristic (GH) for feasibility inspired by the one defined by Balcik [2017], but considering randomization on the heuristic rules. The procedure works as follows. For each critical characteristic, we first compute the average distance among the nodes carrying that characteristic. A small average distance implies that the critical characteristic can be observed in a specific area of the network, while a large average distance value indicates that the characteristic is spread throughout the network. Therefore, we sort the characteristics by average distance (Sort 1) and choose the lowest $|K|$. For each of the selected characteristics $C*$, we sort the nodes carrying it by the distance to the depot (Sort 2), and start the route one vehicle to the closest one, therefore starting $|K|$ routes. After this, the iterative improvement starts: we sort the characteristics according to the current coverage ratios (Sort 3), and take the lowest one; we try to improve it by making some route visit a non-visited node; this node is chosen as the one with the lowest insertion cost to some route (Sort 4); if the characteristic couldn't be improved, we try with the next characteristic in the sort; otherwise we recompute the coverage ratios and iterate again. The pseudo-algorithm is presented in Algorithm **??**.

The randomization of this Greedy Heuristic is done by modifying the order of the four sorting steps. This is achieved by adding random perturbations to the keys of the sort. Since the execution time of the heuristic is very fast (0.1 seconds for all instances), we introduce a time-limit parameter and compute many GH solutions with different randomization parameters. We finally return the best solutions.

GH performance. We tested our GH heuristic on the 96 large instances and compared it against the quality of the GH heuristic and Tabu Search algorithm reported in Balcik [2017]. We set a TL of 5 seconds, which allows the exploration of approximately up to 100 solutions for the 100-node instances, and about 5000 for the 25-node instances. A first observation is that randomization is extremely helpful. On average, the (randomized) GH improves the deterministic GH's objective by 35.7%. Moreover, our heuristic is 40.8% better than the GH in Balcik [2017] which is very similar to our deterministic heuristic. However, the GH is -16.8% worse than the TS heuristic, but it is not a disappointing result considering the simplicity of the heuristic. In fact, in 12 out of the 96 instances, we tie or improve the objective of the TS.

6.3 Experimental results

The names for the different improvements of the SCF formulation are:

- *scf_cuts_2* includes to the SCF model all constraints of type (52) with $|S| = 2$.
- *scf_cuts_3* includes to the SCF model all constraints of type (52) with $|S| \leq 3$.
- *scf_sep_cuts* considers the basic SCF model and performs the separation routine for all fractional and integer solutions.
- *scf_start* is the basic SCF model and provides the feasible GH solution to the solver
- *scf_cuts_2_start* is the *scf_cuts_2* model and also provides the feasible GH start solution. The introduction of this variant was motivated by the fact that both *scf_cuts_2* and *scf_start* showed a positive effect.

We tested the improvements on a total of 12 instances (3 for each size). For each size, we selected instances such that at least one could be solved to optimality. The empty results for the *scf_cuts_3* mean that the solver could not handle the model size and threw a memory error (see used hardware specifications in Section 7). *scf_sep_cuts* showed very bad results on the first computations and we didn't solve the instances 58, 75, and 49. Table 5 shows the difference between the optimality gap of the improved formulation minus the optimality gap of SCF. Table 6 shows the difference in solve times. There is a 1-hour time limit for all executions.

Results clearly suggest discarding *scf_sep_cuts* for its very bad performance (is only competitive with SCF in 25-node instances) and *scf_cuts_3* because the model can only solve up to 50 node instances (and not with good results).

scf_cuts_2 and *scf_start* show promising results. The optimality gaps improve in 3 instances and get worse in also 3 others. *scf_start* improves the gap 4 times and worsens it 4 times. The solve times behave in a similar way. There seems to be significant variability in the way these two improvements behave, but at least they seem competitive with SCF. This motivated the definition of *scf_cuts_2_star*, which provides similar results to the two improvements (noticeable variability), but with a significantly better gap increase for instances 14, 24, and 1. We conclude that *scf_cuts_2_star* is a promising candidate to outperform SCF on the complete set of large instances, and will be considered as one formulation for the computational results of Section 7.2.

Opt. Gap		mip_gap (column) - mip_gap (SCF)				
N	**id**	**scf_cuts_2**	**scf_cuts_3**	**scf_sep_cuts**	**scf_start**	**scf_cuts_2_start**
25	7	0,000	0,000	0,000	0,000	0,000
25	11	0,000	0,000	0,000	0,000	0,000
25	58	0,000			0,000	0,000
50	14	-0,020	0,053	0,047	0,016	-0,020
50	15	0,004	0,151	0,031	0,001	0,003
50	63	0,013			-0,007	-0,004
75	23	-0,032		0,080	-0,048	-0,045
75	24	0,000		0,000	0,000	0,000
75	69	-0,005			-0,005	0,010
100	1	0,000		0,012	-0,002	-0,018
100	27	0,042		0,100	0,050	0,064
100	49	0,001			0,003	0,001

Table 5. Optimality gap differences between basic SCF and different SCF improvements

Solve time		solve_time (column) - scf_solve_time (SCF)				
N	**id**	**scf_cuts_2**	**scf_cuts_3**	**scf_sep_cuts**	**scf_start**	**scf_cuts_2_start**
25	7	2	9	14	2	1
25	11	141	238	2130	63	34
25	58	4			50	21
50	14	-1829	0	17	0	-2335
50	15	0	0	51	0	0
50	63	-105			-105	-105
75	23	0		154	0	0
75	24	164		3626	-41	-30
75	69	0			0	0
100	1	0		117	0	0
100	27	2391		4024	2391	2391
100	49	0			0	0

Table 6. Solve time differences between basic SCF and different SCF improvements

7 COMPUTATIONAL RESULTS

We describe our test instances in Section 7.1, and present the results in Section 7.2.

7.1 Test instances

We test our model in the exact same set of test instances as Balcik [2017], which will be described below. Some of them adapt Solomon's 100-node instances widely used in the literature Solomon [1987]. We refer to Balcik [2017] paper for the details about their generation.

The test set include small instances and large instances. All the 64 small instances can could be solved to optimality in the work of Balcik [2017], so we expected our model to do the same. We take them in order to validate our model and observe optimal solutions. There are also 96 large instances more similar to realistic-size scenarios.

The small instances present 12-node networks, half of them being random (R), and the others being clustered (C). The number of characteristics ranges from two to seven.

The 96 large instances come from 48 networks with 25, 50, 75, and 100 nodes. We distinguish random (R) and random-clustered (RC) instances. For each of the 48 networks, two different instances originate by setting different travel times between the nodes. In the first group, the travel times are simply the euclidean distance between the node coordinates, while the second group divides the times by a factor of 30. For all large instances, the number of characteristics is set to $|C| = 12$. The travel times are symmetric in all of the hypothetical instances. The route duration $T_{\max}$ is set between two and eight in Group I instances, and between 75 and 250 in Group II instance. $|K|$ values range between two and six in both groups of instances.

We used Gurobi 10.0.1 to solve all problem instances by implementing the model in Python using the *gurobipy* API. The rest of the code (experiment design, Greedy Heuristic...) is also implemented in Python. All computational experiments were carried out on a 64-bit Windows machine with an Intel i5-12450H processor of 8 cores, 12 logical processors, and 16 GB of RAM. Therefore, the Gurobi Threads parameter was set to 12.

7.2 Results

We have considered 9 formulations or variants, and there are 64 + 96 test instances. Even though not all combinations have been considered, this leads to a results table with more than 1000 rows. This full table is available in the project repository in *csv* format. In this section, we try to provide an overview of the results. We analyze the performance of the different modeling approaches by comparing them with each other, but we additionally compare their performance with the reported results in Balcik [2017]. There, a Tabu Search heuristic and a CPLEX with 1h of time limit solved all large instances.

Small instances with basic formulations. The 64 12-node small instances can be solved to optimality with Gurobi fast (usually < 1 second). Therefore, we will just look at the time it takes the solver to solve the instances. We compute the average and standard deviation of the solve time for the four basic formulations: CutSet, MTZ-3, MTZ-2, and SCF.

Formulation	Avg. Solve time	Std. Solve time
CutSet	1,21	0,79
MTZ-3	4,07	6,12
MTZ-2	14,00	21,05
SCF	1,11	1,94

Table 7. Solving times on the 64 small instances, by the four basic formulations-

CutSet and SCF perform the best on the small instances. MTZ-3 is the original formulation in Balcik [2017], and seems to be significantly worse than the previous two. MTZ-2 shows a very bad performance, much worse than the 3-index equivalent MTZ formulation. The TS heuristic also solves all small instances to optimality (without proof) in 0.56 on average. That is faster than our best average time, but we would argue that solving times for such small instances are not really relevant.

Basic formulations on 25-node instances. We first show the aggregated results on the subset of 25-node instances. Table 8 shows how CutSet and SCF formulations completely dominate the two MTZ formulations. In terms of objective, MTZ-2 and MTZ-3 still do a good job with respect to the TS objective, they only fall 3.59% and 0.60%, respectively. However, their dual bounds are not of good quality and only allow the formulations to achieve a poor 88.24% and 65.58% optimality gap. SCF and CutSet provide extremely good results: the optimality gaps are very small 2.26% and 1.12%, and the real gaps are equal for the two formulations. The objectives are never worse than the ones from the TS heuristic. In fact, all found objectives between the two formulations were equal, thus giving the same real gap value (1.04%). CutSet seems a bit better since takes less time on average, has a lower gap, and proves optimality two times more.

Model	$Z/Z_{db} - 1$	$Z/Z^*_{db} - 1$	$Z/Z_{TS} - 1$	$\#Z > Z_{TS}$	$\#Z = Z_{TS}$	$\#Z < Z_{TS}$	#Gap = 0	Avg. time diff
MTZ-2	88,24%	5,53%	-3,59%	0	18	6	6	2799,4
MTZ-3	65,58%	1,74%	-0,60%	0	23	1	9	2307,4
SCF	2,26%	1,04%	0,00%	0	24	0	20	580,9
CutSet	1,13%	1,04%	0,00%	0	24	0	22	485

Table 8. Results on the 25-node large instances, aggregated over the four basic models.

Performance limitations of the Cutting Set formulation. While arguably being one of the best formulations for small instances, our Cutting Set formulation is unable to solve some large instances. For example, in the 100-node instances 1 or 27, the solver could improve the dual bound but was unable to find any solution with an objective higher than zero. Note that the trivial solution with no routes is feasible, and this is the actual incumbent returned at the end of the time limit for these two formulations. That made us discard this formulation and we just focused on MTZ-2, MTZ-3, and SCF for the basic formulation analysis and later improvements. As suggested in Vigo [2001], 2-index VRP formulations are usually more suitable for Branch and Bound algorithms. Since no routes are distinguished in the solution graph, one advantage is that the separation routines introduce cuts that are valid for all vehicles, as opposed to our current vehicle-iterative approach. This could be a new line of research to overcome the existing limitations of our proposed exponential formulation.

Basic formulations on the large dataset. Recall that the large dataset contains 96 instances. Table 9 aggregates the results of the MTZ-2, MTZ-3, SCF, and improved SCF (named SCF*) over the large instances. $Z/Z_{db} - 1$ is the optimality bound (Z_{db} is the best dual bound found during the optimization). $Z/{Z_{db}}^* - 1$ is the average of the relative gap considering the best-known dual bound known for the instance, Z^*_{db} (could come from a solution found in another formulation). $Z/Z_{TS} - 1$ is the average of the relative difference between the found objective and the TS objective. The columns with "#" count the number of times that our objective is better, is equal, or is worse than the TS objective, and also the number of times where optimality was proven under the time limit of 1 hour. Finally, the average of the solve time (max. is 3600 seconds) minus the reported execution time for the TS heuristic is given.

Model	$Z/Z_{db} - 1$	$Z/Z^*_{db} - 1$	$Z/Z_{TS} - 1$	$\#Z > Z_{TS}$	$\#Z = Z_{TS}$	$\#Z < Z_{TS}$	#Gap = 0	Avg. time diff
MTZ-2	120,15%	63,62%	-25,89%	0	22	74	7	2908,9
MTZ-3	57,77%	24,98%	-7,84%	0	41	55	20	2087,0
SCF	21,32%	20,51%	-2,18%	13	60	23	40	1192,1
SCF *	20,70%	20,04%	-1,79%	14	61	21	41	1131,4

Table 9. Results on the large instances, aggregated over different models.

Table 9 allows us to take the following conclusions:

- **The 2-index MTZ formulation performs much worse than all others**. The relative gaps that it provides are on average 6 times the gaps of the improved SCF formulation, and it proves optimality the least amount of times (7/96 instances). The real optimality gaps (60%) are also much lower than all other formulations, which are around 20%, suggesting that the primal feasibility is the struggle of this formulation. With no surprise, it also takes the most time on average. Therefore, we conclude that this modification of the original MTZ-3 formulation in Balcik [2017] gives much worse results. This result aligns with our experimental results of section 4, where we compared the root relaxations of the two MTZ formulations, and MTZ-3 always provided better or equal dual bounds.
- **The basic SCF formulation strongly dominates the two MTZ formulations**. Besides providing significantly better optimality gaps, this formulation proves optimality in 40/96 instances, while MTZ-3 could only do it for 20. Moreover, it can improve the TS objective 13 times, something that MTZ formulations do not achieve in any instance. When comparing the relative difference in objectives, SCF is just 2.18% worse than TS. This experimental result supports our theoretical findings: Propositions 4.2 and 4.1, prove that SCF was a stronger formulation than MTZ-2 and MTZ-3.
- **The SCF improvements traduce to a performance improvement**. The gap $Z/Z_{db} - 1$ is reduced by 0.62 points, and the real gap $Z/Z^*_{db} - 1$ is reduced by 0.47. The fact that there is an improvement in these two values implies that the improvements are reflected on both the primal and dual bounds. The relative difference to the TS solutions is also decreased in comparison to the basic SCF: from 2.18% to 1.79%. Additionally, the improved SCF can improve or tie TS on two more instances, and it proves optimality in under one hour one more time (for a total of 41/96 =42.7%). However, it is worth mentioning that the Wilcoxon test on the MIP gaps does not allow us to conclude that the improvements are statistically significant (p-value = 0.3562).
- **Our best model (improved SCF) is totally competitive with the Tabu Search heuristic**. The relative difference in the objective is only 1.79% on average, and the real optimality gaps are not significantly different according to a Wilcoxon test (p-value = 0.8581). However, the 20.04% of the improved SCF still comes short of the 12.56% of the TS. However, we argue that a comparison between a heuristic and an exact method will never be totally fair, because the solver spends a lot of work trying to prove optimality. If only feasibility would matter, the solver could have been set in a way that prioritizes feasibility heuristics and primal bounds.

In conclusion, the main contribution of this work is the definition of a Single Commodity Flow model that:

- Is theoretically and empirically stronger than the MTZ-3 model proposed in the literature.
- Is highly competitive with the current-best Tabu Search heuristic, with the benefit that is an exact method.

8 CASE STUDY

We will reproduce the case study that Balcik [2017] proposed, which focuses on the affected towns after the earthquake in Van, a city in eastern Turkey. The earthquake occurred on the 23d of October 2011 and had a magnitude of 7.2; it killed more than 600 people and left more than 200,000 people homeless and in need. Disaster response activities were coordinated by the Disaster and Emergency Management Authority (AFAD), which operates under the Prime Ministry. The Turkish Red Crescent (TRC), the main humanitarian organization in the country, was the main responder. The rapid needs assessment operations were conducted by the TRC teams that immediately arrived in the city from agency offices located in the neighboring cities of Van. According to Balcik [2017], communication with the TRC officers indicated that a systematic approach was not used while selecting sites and making assessment plans after the earthquake, and they could not obtain the implemented assessment plan from the organization. This case study illustrates an example application, which can help practitioners understand how purposive sampling could be used in practice by using the proposed approach. The case study also allows us to test our solution algorithm on a data set based on a real network, and compare it to the current-best Tabu Search algorithm of Balcik [2017] in this additional set of instances.

8.1 Data set

The data set used by Balcik [2017] is a network of 93 sites (towns and villages) affected by the Van earthquake. These sites, whose population range between 112 and 20,000, are dispersed through a rural region. In fact, it is based on the post-disaster network presented in Noyan et al. [2016]. The network is illustrated in Figure **??**. The road travel times for this network were determined by Google Maps and can be asymmetric as opposed to the in small and large instances.

Researchers in Balcik [2017] define the critical characteristics to cover for each site by studying and analyzing several aspects of the affected region. Those are: elevation (3 classes), coastal/inland, population (3 classes), children (3 classes), elderly (3 classes), disabled (3 classes), and impact (2 classes). This gives a total of 19 characteristics. Some case instances are also defined with just a subset of nine characteristics: low altitude, high altitude, coastal, small town, large town, high children population, high elderly population, high disabled population, and high impact zone. For the details about the choice of the characteristics, we refer to Balcik [2017].

We consider instances with two, three, and four teams, where each team has 8, 12, 16, and 24 hours to complete the assessments. We assume that each team is expected to spend one hour to complete the assessments at a site, which is incorporated into the travel time values. Therefore, if we assume that each team may work 8–12 hours per day, an assessment deadline of one to three days is considered in our case instances, which is consistent with real-world practices.

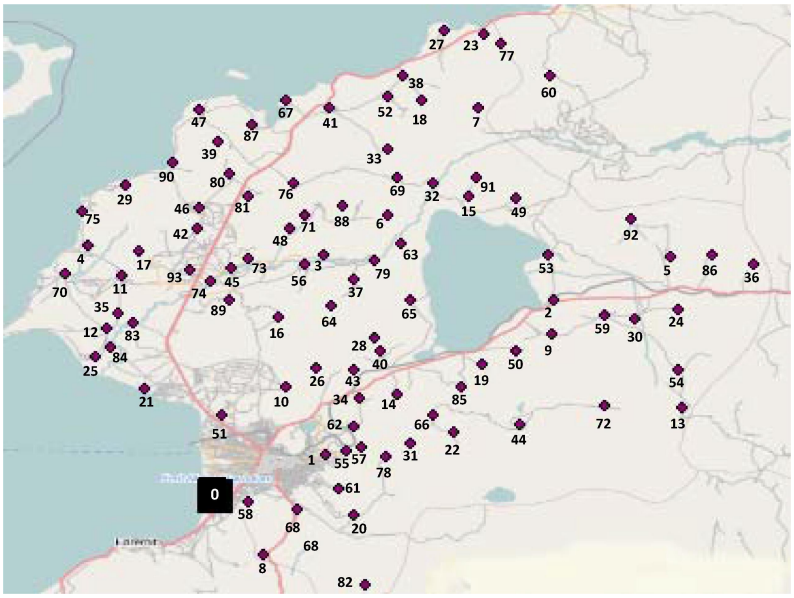

Fig. 6. The case study network. Figure from Balcik [2017].

8.2 Results

We use our best model (SCF with improvements) to solve the 24 case instances with a time limit of 1 hour, and we measure the same variables as in Section 7.2, with the addition of $Z_{TS}/Z_{db} - 1$, which is the optimality gap of the TS heuristic.

	$\|C\| = 9$	$\|C\| = 19$	Total
#Instances	12	12	24
$Z/Z_{db} - 1$	9,18%	13,96%	11,57%
$Z_{TS}/Z_{db} - 1$	15,36%	25,60%	20,48%
$Z/Z_{TS} - 1$	5,71%	10,20%	7,95%
$\#Z > Z_{TS}$	10	11	21
$\#Z = Z_{TS}$	2	1	3
$\#Z < Z_{TS}$	0	0	0
#gap = 0	0	0	0
Avg. time diff	2749,0	2107,1	2428,0

Table 10. Results on case instances, aggregated by the number of characteristics.

In this set of instances, our solution method seems to completely dominate the TS heuristic in terms of solution quality. Overall, SCF provides an 11,57% relative gap, while TS can only reach 20,48%. This is a relative improvement of 7,95%, significant according to a Wilcoxon test on the correspondent gaps (p-value = 1e-7). In 21/24 instances, we improved the TS objective, and we obtained the same objective in the rest of the instances. However, no solution could be proven optimal under the time limit. However, the solve time of 1 hour is still higher than the solve time used by the TS.

Another observation is that the number of characteristics $|C|$ seems to be influential in the instance difficulty. Note that this could not be studied on the large instances because they all considered $|C| = 12$. This is especially noticeable for the TS heuristic, whose optimality gaps drop from 15,35% on the $|C| = 9$ instances to 25,60%. Our model seems to be more robust to this parameter, and the gaps only change from 9,18% to 13,96%. In fact, the size of our model does not depend on the number of characteristics. On the other hand, the TS algorithm, as explained by Balcik [2017], contains steps that iterate over a subset of critical characteristics. A bigger set C both increases execution time and the probability of making erroneous decisions during the construction of solutions.

We conclude that the SCF is significantly stronger than TS in terms of solution quality in the network of the case study.

9 CONCLUSIONS

A ALGORITHMS

The pseudo-code for the algorithms considered in this work is presented.

Algorithm 1: Separation procedure for Cut-Set formulation

Input : Integer or fractional feasible solution (x, y), and the original network G
Output : If existing, a cutting set valid inequality that separates the current solution.

for $k \in K$ **do**
  $G^* \leftarrow$ copy of the original network with capacities x_{ijk} on the edges $(i, j) \in E(G)$;
  for $i \in V(G^*) : i \neq 0$ **do**
    $mf \leftarrow$ Max-Flow on G^* from 0 to i;
    if $mf < y_{ik}$ **then**
      $S_1, S_2 \leftarrow$ Min-Cut partition of G^*;
      return S_* the side of the partition that contains i ; // S_* becomes $S \subset N$ in (31)
    end
  end
end
return $\emptyset$

Algorithm 2: Separation procedure for Cut-Set constraints in SCF formulation

Input : Integer or fractional feasible solution x, y, original network G
Output : If existing, a cutting set valid inequality that separates the current solution.

$G^* \leftarrow$ copy of the original network with capacities x_{ij} on the edges $(i, j) \in E(G)$;
for $i \in V(G^*) : i \neq 0$ **do**
  $mf \leftarrow$ Max-Flow on G^* from 0 to i;
  if $mf < y_i$ **then**
    $S_1, S_2 \leftarrow$ Min-Cut partition of G^*;
    return S_* the side of the partition that contains i ; // S_* becomes $S \subset N$ in (52)
  end
end

Algorithm 3: Greddy Heuristic for feasible SARP solutions

Input : SARP instance
Output: SARP feasible solution

$d_c \leftarrow$ average distance between nodes carrying $c \in C$;
$C^* \leftarrow$ first $|K|$ elements of C sorted increasingly according to d_c ; // Sort 1
$R_k \leftarrow \{\}$ empty routes for $k \in K$ are initialized ;
for $c \in C^*$ **do**
    $N^* \leftarrow$ non-visited nodes carrying c $i \leftarrow$ closest node to the depot ; // Sort 2
    Append i to R_c
end
while *True* **do**
    $cr_c \leftarrow$ coverage ratios of $c \in C$;
    $C^* \leftarrow$ characteristics sorted increasingly according to cr_c ; // Sort 3
    for $c \in C^*$ **do**
        $N^* \leftarrow$ non-visited nodes carrying c sorted by insertion cost to some route ; // Sort 4
        remove from N^* the infeasible insertions (depot couldn't be reached within $T_{\max}$;
        if $N^* = \emptyset$ **then**
            continue
        end
        $i \leftarrow$ node with the minimum insertion cost;
        Append i to R_k
    end
    break
end
return $R_k, k \in K$

REFERENCES

ACAPS. 2011. Technical Brief: Purposive Sampling and Site Selection in Phase 2. (2011). https://www.acaps.org/sites/acaps/files/resources/files/purposive_sampling_and_site_selection_in_phase_2.pdf

Mahdieh Allahviranloo, Joseph YJ Chow, and Will W Recker. 2014. Selective vehicle routing problems under uncertainty without recourse. *Transportation Research Part E: Logistics and Transportation Review* 62 (2014), 68–88.

Claudia Archetti, M Grazia Speranza, and Daniele Vigo. 2014a. Chapter 10: Vehicle routing problems with profits. In *Vehicle routing: Problems, methods, and applications, second edition*. SIAM, 273–297.

Claudia Archetti, M. Grazia Speranza, and Daniele Vigo. 2014b. *Chapter 10: Vehicle Routing Problems with Profits*. Chapter 10, 273–297. https://doi.org/10.1137/1.9781611973594.ch10 arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611973594.ch10

Burcu Balcik. 2017. Site selection and vehicle routing for post-disaster rapid needs assessment. *Transportation Research Part E: Logistics and Transportation Review* 101 (2017), 30–58. https://doi.org/10.1016/j.tre.2017.01.002

Sylvain Boussier, Dominique Feillet, and Michel Gendreau. 2007. An exact algorithm for team orienteering problems. *4or* 5 (2007), 211–230.

Steven E Butt and Tom M Cavalier. 1994. A heuristic for the multiple tour maximum collection problem. *Computers & Operations Research* 21, 1 (1994), 101–111.

Dominique Feillet, Pierre Dejax, and Michel Gendreau. 2005. Traveling salesman problems with profits. *Transportation science* 39, 2 (2005), 188–205.

Bruce L Golden, Larry Levy, and Rakesh Vohra. 1987. The orienteering problem. *Naval Research Logistics (NRL)* 34, 3 (1987), 307–318.

Mohammadmehdi Hakimifar, Vera C. Hemmelmayr, and Fabien Tricoire. 2022. A Lexicographic Maximin Approach to the Selective Assessment Routing Problem. *OR Spectr.* 45, 1 (jul 2022), 205–249. https://doi.org/10.1007/s00291-022-00687-8

IFRC. 2008. Guidelines for Assessment in Emergencies. (2008).

ARII Maya. 2013. Rapid assessment in disasters. *JMAJ* 56 (2013), 19–24.

C. E. Miller, A. W. Tucker, and R. A. Zemlin. 1960. Integer Programming Formulation of Traveling Salesman Problems. *J. ACM* 7, 4 (oct 1960), 326–329. https://doi.org/10.1145/321043.321046

Nilay Noyan, Burcu Balcik, and Semih Atakan. 2016. A stochastic optimization model for designing last mile relief networks. *Transportation Science* 50, 3 (2016), 1092–1113.

Marcus Poggi, Henrique Viana, and Eduardo Uchoa. 2010a. The Team Orienteering Problem: Formulations and Branch-Cut and Price. In *10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'10) (OpenAccess Series in Informatics (OASIcs), Vol. 14)*, Thomas Erlebach and Marco Lübbecke (Eds.). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 142–155. https://doi.org/10.4230/OASIcs.ATMOS.2010.142

Marcus Poggi, Henrique Viana, and Eduardo Uchoa. 2010b. The Team Orienteering Problem: Formulations and Branch-Cut and Price. In *10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'10) (OpenAccess Series in Informatics (OASIcs), Vol. 14)*, Thomas Erlebach and Marco Lübbecke (Eds.). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 142–155. https://doi.org/10.4230/OASIcs.ATMOS.2010.142

Marius M. Solomon. 1987. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research* 35, 2 (1987), 254–265. https://doi.org/10.1287/opre.35.2.254

Paolo Toth; Daniele Vigo. 2001. *The Vehicle Routing Problem*. SIAM.

Received WS 2023; accepted WS 2023