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 discussed, and different exact optimization techniques are considered to improve the performance of numerical solvers.

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, TU 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

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

1 MOTIVATION

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].

Since the 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. Random sampling is always an option, but purposive sampling constructs an assessment plan as follows [ACAPS 2011]: (1) specify the critical characteristics that define the target community groups, (2) select sites to be visited according to sample richness, and (3) plan the logistics (assign teams and routes) according to routing and resources constraints. However, the before-mentioned process might become a trial and error procedure that oscillates between (2) and (3). The SARP addresses this problem by solving an optimization problem, the solution of which is the assessment plan.

To the best of our knowledge, exact methods for SARP remain largely unexplored in the existing literature. Though researchers in [Balcik 2017] propose a MILP formulation and solve some instances with an optimization solver, we do not have any record of another attempt to solve the SARP exactly.

Execution time is a key element 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 pursue 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) [Chao et al. 1996] or the so-called *routing problems with profits* [Feillet et al. 2005].

2 METHODOLOGY

2.1 Mathematical formulations and algorithm design

Alternative MILP formulations to the one in [Balcik 2017] will be investigated. Due to the closeness of the problems, we will explore how the well-studied formulations for the VRP translate into different formulations for the SARP: vehicle flow formulations (MTZ constraints), commodity flow formulations (SCF, MCF), or exponentially sized formulations that use, for instance, Subtour Elimination Constraints [Vigo 2001].

For exponential formulations, we will study different classes of strengthening inequalities and separation procedures. Different relaxations for the problem will be studied. In particular, we will pay attention to combinatorial and Lagrangian relaxations. Additionally, we will discuss possible primal heuristics to construct feasible solutions during the optimization process and for starting. Finally, if considered adequate and promising, we explore advanced techniques like branching schemes and search strategies that exploit the properties of SARP.

2.2 Implementation

The computational study will be done using Gurobi with its Python API (using a student license). Depending on the feasibility and how promising the different intended techniques are, further options are to use Gurobi or SCIP in conjunction with C++.

2.3 Solution to problem instances

We will work with the exact same instances considered in [Balcik 2017]. We hope to obtain them directly from the author of the paper. We will test our developed approaches on all instances agreeing on a time limit, will validate the solutions, and will analyze and visualize the results (solving times, optimality gap, nodes explored...).

We will finally apply our best developed algorithm to a case study that was proposed by [Balcik 2017]. It consists of assessing the rapid needs assessment to affected towns after the earthquake in Van, which is a city in eastern Turkey. The earthquake occurred on 23 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. We will discuss how the developed solution aligns with the goals of Humanitarian Logistics, and how the different key concepts of this discipline are utilized in our work.

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. 2014], 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.

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:

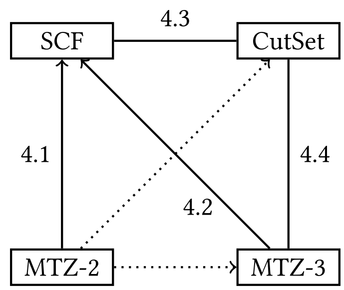

Fig. 1. B stronger than A is noted by $A \longrightarrow B$. A and B incomparable is noted by $A - B$. Dashed lines or arrows indicate that the we did not prove the relation, but experimental tests strongly suggest could be true. The edges refer to the proposition that proves the relationaship.

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:

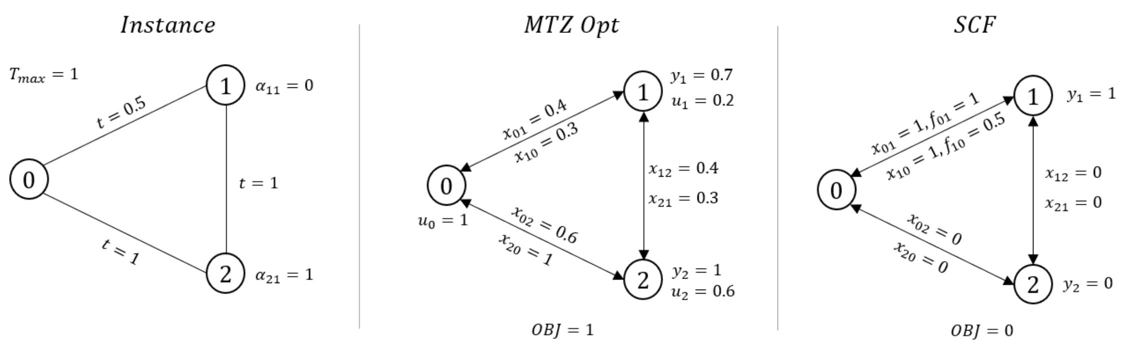

Fig. 2. On the left, the instance presented, which only considers one characteristic: $C = \{0\}$. In the middle, the MTZ Opt and SCF solutions to the respective Linear Relaxations

.

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).

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

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

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:

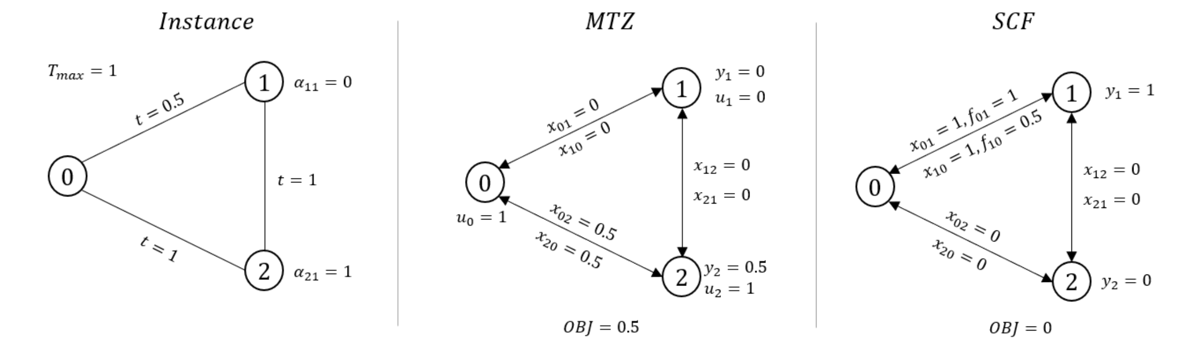

Fig. 3. On the left, the instance presented. In the middle, the MTZ-3 and SCF solutions to the respective Linear Relaxations
.

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.

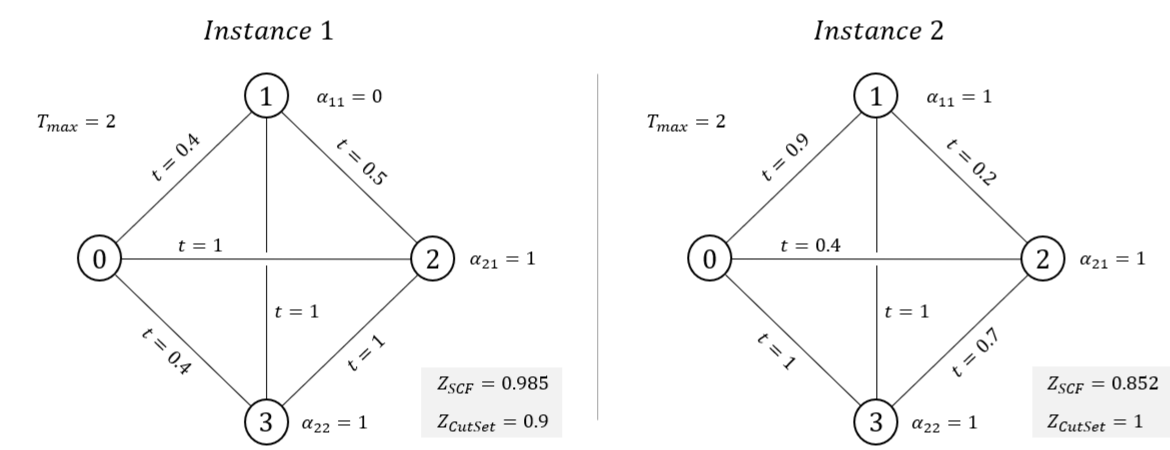

Fig. 4. Instance 1 shows that $P_{SCF} \not\subset P_{CutSet}$. Instance 2 shows $P_{Cutset} \not\subset P_{SCF}$.

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.

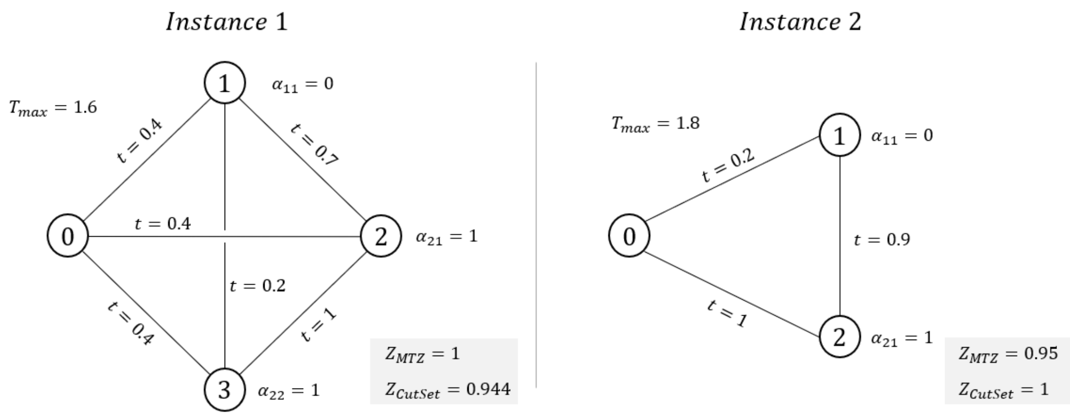

Fig. 5. Instance 1 shows that $P_{MTZ-3} \not\subset P_{CutSet}$. Instance 2 shows $P_{Cutset} \not\subset P_{MTZ-3}$.

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 **??**, our 48 large networks define two classes of problems each. The first class has smaller travel times (average of average travel time is 1.3), while the second one has an average of 31.8. We compare the performance of the solver using the SCF and MTZ formulations, but all other behave in a similar way.

	% inst. prove opt.		**% inst. improve TS**		**Opt. Gap %**	
Class type	mtz	scf	mtz	scf	mtz	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. The average optimality gap is reduced significantly, and the amount of instances that can be solved to optimality under the time limit increase. 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
$\lvert K\rvert$ **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
$\lvert K\rvert$ **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
$\lvert K\rvert$ **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 COMPUTATIONAL RESULTS

We describe our test instances in Section 6.1.

6.1 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.

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

Claudia Archetti, M. Grazia Speranza, and Daniele Vigo. 2014. *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

I-Ming Chao, Bruce L Golden, and Edward A Wasil. 1996. The team orienteering problem. *European journal of operational research* 88, 3 (1996), 464–474.

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

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

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