E-ARCP: User guide, software verification, and modification to other network connectivity problems

The "**E**xact Solution **A**pproach to the Collaborative Multi-vehicle **A**rc **R**outing for **C**onnectivity **P**roblem" (E-ARCP) is a Mixed Integer Linear Programming (MIP) model designed to solve the **Collaborative Multi-vehicle Prize Collecting Arc Routing for Connectivity Problem** (MPC-ARCP). The MPC-ARCP is introduced in Souza Almeida et al. (2023), where a full description of the MIP model can be found.

1) Program Requirements

- PuLP
- NetworkX
- Matplotlib
- Plotly
- A linear programming solver such as GUROBI or CMD

2) Usage

The structure of the E-ARCP is depicted in Figure 1. The program requires two input files: '*model parameters*' (.txt) and '*instance description*' (.py). The content of these files is covered in Subsection 2.1. After solving the MIP model, E-ARCP generates three output files: a '*mathematical model solution*' file (.txt), a '*log file*' (.log) and the '*solution visualization*' (.png), which graphically depicts the results for a network in an image file. The output files are explained in Subsection 2.2. A description of the steps required to run E-ARCP is presented in Subsection 2.3.

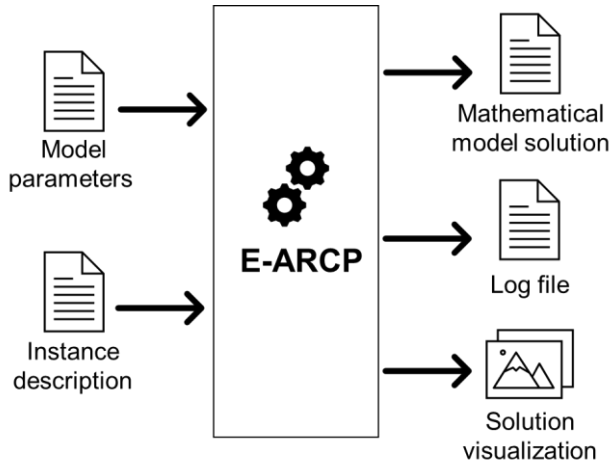


Figure 1 – Summary of the E-ARCP experimental setting

2.1) Input data

E-ARCP requires two input data files: a python file called '*input_data*' (.py) and the 'instance description' (.txt) with the network description. Subsection 2.1.1 explains how to set up the input files, and Subsection 2.1.2 covers the network's description.

2.1.1) input_data.py

The '*input_data*' (.py) is a Python file which contains the parameters used in the MIP model, such as the speed of the road clearing teams and the time horizon. An example is displayed in Figure 2. Notice that the user should **not** change the variable names, and the file must be kept in the same folder as E-ARCP.

The first variable is the "instance_name", which indicates the name of the text file describing the network's structure (further explained in Subsection 2.1.2). The string with the file name should **not** contain the file extension. For example, instead of "MPC-10.txt", the user should only type "MPC-10" as depicted in Figure 2. The "T_{max}" parameter of the MIP model is represented by the variable "t_max", and it indicates the time horizon in hours (i.e. the maximum duration of each team's route). For example, in Figure 2, $t_max = 72$, and therefore all the road clearing teams will be assigned a route that takes less than 72 hours to complete. The third parameter is "speed", the traversing speed of the road clearing teams. For instance, in Figure 1, teams traverse open roads at 50 km/h.

Network properties

```
#Name of the instance file without the extension (.txt)
instance_name = "MPC-10"
#Time horizon in hours
t_max = 72
#Traversing speed of the road clearing teams (km/h)
speed = 50
```

Solver properties

```
#Solver processing time limit in seconds
processing_time_limit = 3600
#Choice of a solver (GUROBI or Other)
solver = "GUROBI"
```

Visualization

```
#Display the results? (True if yes and False if no)
display_results = True
#Color of the intact roads
open_road_color = "green"
#Color of the closed roads
closed_road_color = "red"
#Color and size of the nodes
nodesize = 150
nodecolor = "white"
nodeboundary = "black"
#Color of the road clearing teams' routes
team_color = "blue"
```

Figure 2 – An example of the file '*input_data*' (.py)

Another parameter in the '*input_data*' (.py) is the processing time limit of the solver. This is the maximum time (in seconds) the solver will keep trying to find an optimal solution. The variable "solver" indicates

what type of solver should be used. This version of E-ARCP can only solve a model with GUROBI or the PuLP default solver, which as of January 2023, is CMD. The user needs a valid license to use GUROBI.

The last set of parameters is related to the solution visualization. First, the user should indicate if E-ARCP should generate the images with the solution. If the variable “display_results” is set as “True”, E-ARCP will plot the network; otherwise, the images will not be created. The color of the roads that were not affected by the disaster are given by variable “open_road_color”. The color of the blocked edges is given by the variable “closed_road_color”. The settings of the vertices are the size in pixels (variable “nodesize”), the fill color (“nodecolor”), and the outline color (“nodeboundary”). Finally, the color of the edges that the road clearing teams traverse is set using the variable “team_color”.

2.1.2) Instance file

The network’s structure is described in a text file (.txt). Consider the example of the “MPC-10”, an instance with ten vertices. The first part of the instance is the heading, displayed in Figure 3. The “NAME” is the network’s name, which is “MPC-10” in this example. The “TYPE” refers to the problem related to this network, which is MPC-ARCP. The name of the authors of the instance is available in the “COMMENT” section. The “DIMENSION” refers to the number of vertices, and the “EDGE_WEIGHT_TYPE” describes how the weight of the edge should be calculated. In this instance, the weights should be calculated using the 2D Euclidean distance. “DEPOT_IDS” is an array with the index of the depots. For example, vertices 0 and 6 are the teams’ start points. The “SPEED” is the road clearing teams’ speed (in km/h) while traversing an open edge. The “AVERAGE_DEGREE” is the average number of connections each vertex has in the network. “COMPONENTS” refers to the number of sub-networks in which the main network is divided after the disaster. The “NODE_COORD_SECTION” describes the coordinates of all vertices. For example, the first row (“0, 60, 100”) states that node 0 is located in the position [60,100].

```
NAME: MPC-10
TYPE: MPC-ARCP
COMMENT: Generated by Almeida, L., Ghasemi, A., Goerlandt, F.
DIMENSION: 10
EDGE_WEIGHT_TYPE: EUC_2D
DEPOT_IDS: [0, 6]
SPEED: 50
AVERAGE_DEGREE: 2.5
COMPONENTS: 4
NODE_COORD_SECTION
0       60      100
1       8       60
2       35      45
3       35      95
4       60      60
5       50      15
6       10      15
7       70      15
8       72      80
9       100     8
```

Figure 3 – The heading of the “MPC-10” instance description

A list with all the edges in the network is presented in the “LIST_OF_ALL_EDGES_DATA_SECTION”, as in Figure 4. Notice that it includes blocked and unblocked edges. The section “DAMAGED_EDGES_DATA_SECTION (EDGE, UNBLOCKING TIME)” consists of the edges that are blocked, showing also their respective unblocking times (in hours). For example, “1, 2, 20” means that edge [1,2] requires a team to clear for 20 hours before it becomes traversable. This list includes the edge’s symmetric counterpart (e.g. [1,2] and [2,1] have an unblocking time of 20).

```
LIST_OF_ALL_EDGES_DATA_SECTION
0       4
1       2
1       3
2       1
2       3
2       4
2       5
2       6
3       1
3       2
4       0
4       2
4       7
4       8
5       2
5       7
6       2
7       4
7       5
8       4
8       9
9       8
```

Figure 4 – The list of edges in the "MPC-10" instance description

```
DAMAGED_EDGES_DATA_SECTION (EDGE, UNBLOCKING TIME)
1       2       20
2       1       20
2       5       15
5       2       15
2       4       10
4       2       10
4       7       30
7       4       30
8       9       35
9       8       35
```

Figure 5 – The list of blocked edges and their respective unblocking times in the "MPC-10" instance description

The components and their respective indexes are listed in the "COMPONENTS" section (see Figure 6). For instance, in "MPC-10" the component with index 0 includes vertices 0, 4, and 8.

```
COMPONENTS
0       [0,4,8]
1       [1,2,3,6]
2       [5,7]
3       [9]
```

Figure 6 – The list of components in the "MPC-10" instance description

The components' relative prizes are described in the "RELATIVE PRIZES" section. As represented in the first row in Figure 7, the prize collected while reconnecting component 0 (vertices [0,4,8]) to component 1 (vertices [1,2,3,6]) is 10.

```
RELATIVE PRIZES
0       1       10
0       2       15
0       3       15
1       0       10
1       2       15
1       3       15
2       0       15
2       1       15
2       3       15
3       0       15
3       1       15
3       2       15
```

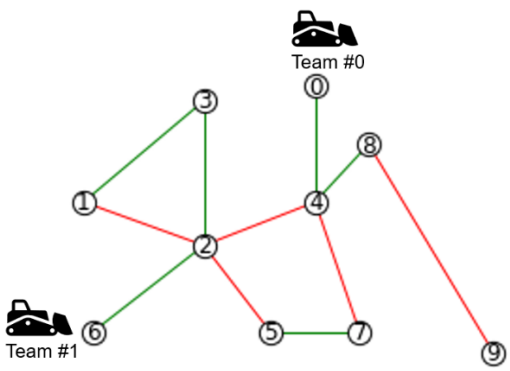
Figure 7 – The relative prizes to reconnect the components in the "MPC-10" instance description

The final part of the instance description is the number of available teams in each depot. As depicted in Figure 8, the section "TEAMS PER DEPOT (DEPOT_ID, TEAMS)" indicates the index of the depot followed by the number of teams. Thus, in the example, depot 0 has 1 team.

```
TEAMS PER DEPOT (DEPOT_ID, TEAMS)
0       1
6       1
```

Figure 8 – The number of available teams per depot in the "MPC-10" instance description

E-ARCP reads the instance description and converts it into a network. For instance, the network generated with the information in the "MPC-10" text file is shown in Figure 9.



Edges	Unblocking time
(1,2), (2,1)	20h
(2,5), (5,2)	15h
(2,4), (4,2)	10h
(4,7), (7,4)	30h
(8,9), (9,8)	35h

Legend:
Blocked road — Intact road — Community/road intersection/depots

Figure 9 – Network depiction and information associated with "MPC-10"

The user can find examples of academic instances to the MPC-ARCP in Souza Almeida et al. (2022) and in Souza Almeida et al. (2022).

2.2) Output data

E-ARCP generates three output files: a text file (.txt) with the MIP model's solution, a log file (.log) created by the solver, and the images of the road clearing teams' routes (.png). Subsection 2.2.1 explains how to read the MIP model solution, Subsection 2.2.2 explains the log file, and Subsection 2.2.3 covers the visualization.

2.2.1) Mathematical model solution

E-ARCP creates a text file with the results of the MIP (see an example in Figures 10 and 11). The first part of the solution file describes the "INPUT INFORMATION": the name of the instance description file ("Instance") and the total number of teams. For example, in MPC-10, there are 2 teams in total: one in depot zero and another in depot 6. The "RESULTS" Section in Figure 10 describes the processor's processing time (in seconds) to find a solution to the model. The "Objective Function" is the sum of the total prize collected while reconnecting the components. The output file also includes the routes of each of the road clearing teams. For instance, team 0 starts at vertex 0, then goes to vertices 4, 2, and 10 (sink vertex), in this order.

```
----------INPUT INFORMATION----------

Instance = MPC-10
Total # of teams = 2

----------RESULTS----------

Processing time (seconds) = 14.997645200000001
Objective function = 25.0
Team 0 = [0, 4, 2, 10]
Team 1 = [6, 2, 5, 7, 5, 10]
```

Figure 10 – General results of the run on MPC-10

The "VARIABLES" section contains a list of all positive variables, as exemplified in Figure 11. The authors refer to Souza Almeida et al. (2023) for a description of the decision variables.

```
----------VARIABLES----------

a_(2,_5,_1) = 15.0
a_(4,_2,_0) = 10.0
f_(0,_4,_0) = 3.0
f_(2,_10,_0) = 1.0
f_(2,_5,_1) = 4.0
f_(4,_2,_0) = 2.0
f_(5,_10,_1) = 1.0
f_(5,_7,_1) = 3.0
f_(6,_2,_1) = 5.0
f_(7,_5,_1) = 2.0
s_(2,_5,_1) = 3.5300000000000633
s_(4,_2,_0) = 0.7999999999999688
t_(0,_4,_2,_0) = 0.7999999999999972
t_(2,_5,_7,_1) = 19.200000000000063
t_(4,_2,_10,_0) = 11.379999999999969
t_(5,_7,_5,_1) = 19.600000000000062
t_(6,_2,_5,_1) = 0.7815631262524967
t_(7,_5,_10,_1) = 20.0
v_(10,_0) = 1.0
v_(10,_1) = 1.0
v_(2,_0) = 1.0
v_(2,_1) = 1.0
v_(4,_0) = 1.0
v_(5,_1) = 2.0
v_(7,_1) = 1.0
x_(0,_4,_2,_0) = 1.0
x_(2,_5,_7,_1) = 1.0
x_(4,_2,_10,_0) = 1.0
x_(5,_7,_5,_1) = 1.0
x_(6,_2,_5,_1) = 1.0
x_(7,_5,_10,_1) = 1.0
```

Figure 11 – Decision variables of the MPC-10 with a time horizon of 20 hours

2.2.2) Log file

The solver generates the log file. As this is a standard output from the applied solver, the authors refer to the solver's documentation for a detailed description of the file, as follows.

GUROBI: (https://www.gurobi.com/documentation/9.1/refman/logging.html)

CBC: (https://coin-or.github.io/Cbc/)

2.2.3) Solution visualization

E-ARCP creates an image for each one of the road-clearing teams (see an example in Figure 12). The image consists of a network in which the green lines represent roads not affected by the disaster (intact roads) and the red lines represent blocked roads. The bold blue arrows indicate the route of the road clearing

team. Since the dummy sink vertex is used in the MIP model only to ensure that all teams end their route, this vertex is not displayed in the solution.

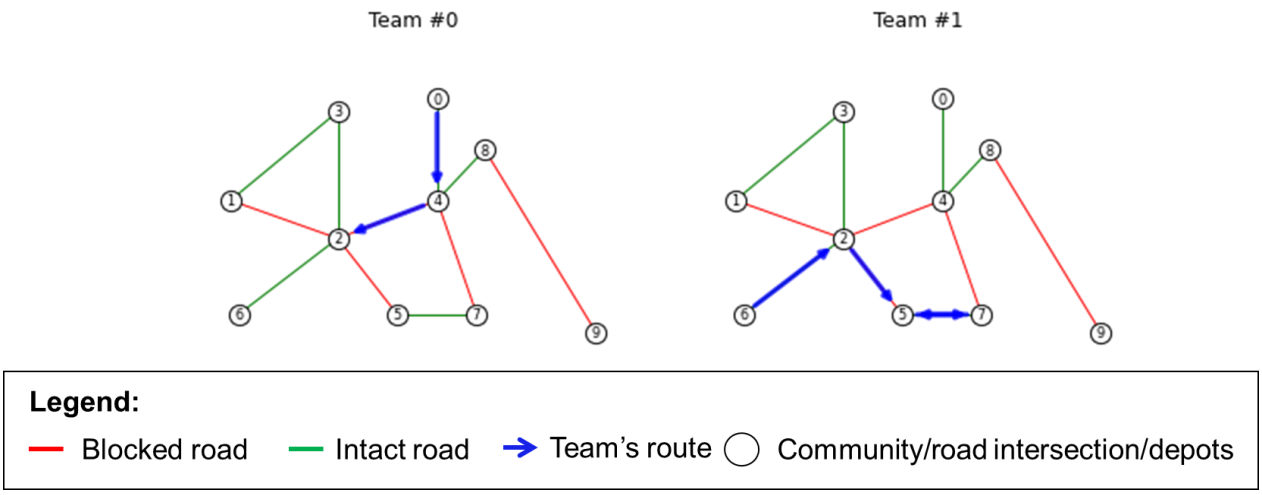


Figure 12 – Visualization of MPC-10

2.3) How to run the model

Step 1: Update the model parameters in the '*input_data*' (.py) file.

Step 2: Update the instance name in the '*input_data*' (.py) file.

Step 3: Ensure the Instance Description text file (e.g. "MPC-10.txt") is in the same folder as the '*input_data*' (.py) and the '*E-ARCP*' (.ipynb) files.

Step 4: Run the '*E-ARCP*' (.ipynb) on Jupyter Notebooks or Visual Studio Code.

3) Model Verification

The Mixed Integer Programming model to MPC-ARCP was tested on three instances (MPC-10, MPC-15, and MPC-20). All required input files for running the tests, as well as their results are provided in the "Model Verification" folder. The tests were performed on an AMD Ryzen 5 Microsoft Surface® Edition with six cores (base speed 2.2.GHz) and 8GB of RAM.

Two types of verification tests were performed. In the first type of test, a known solution was forced on instance MPC-10 to check if the model performs the calculations correctly (e.g. synchronization of road clearing teams), and if E-ARCP presents an inappropriate behaviour such as finding that a known solution is infeasible. Subsections 3.1 – 3.4 describe the first type of test and their respective results.

The second type of test consisted of running the model on three networks (MPC-10, MPC-15, and MPC-20) and checking if the model was being optimally solved and whether it behaved properly when the time horizon changed. The results of this second type of verification are presented in Subsections 3.5 - 3.7. A summary of the tests is available in Table 1.

Table 1 – Summary of the model verification tests

Test	Instance	Time Horizon	Test result
Null Test A	MPC-10	72 hours	Passed
Null Test B	MPC-10	72 hours	Passed

Single Team Test A	MPC-10	100 hours	Passed
Single Team Test B	MPC-10	72 hours	Passed
Single Team Test C	MPC-10	72 hours	Passed
Two Teams Test A	MPC-10	72 hours	Passed
Two Teams Test B	MPC-10	6.37 hours	Passed
Two Teams Test C	MPC-10	55 hours	Passed
Two Teams Test D	MPC-10	31 hours	Passed
Three Teams	MPC-10-2	7.75 hours	Passed
MPC-10 Test A	MPC-10	72h	Passed
MPC-10 Test B	MPC-10	20h	Passed
MPC-10 Test C	MPC-10	10h	Passed
MPC-15 Test A	MPC-15	72h	Passed
MPC-15 Test B	MPC-15	25h	Passed
MPC-20 Test A	MPC-20	72h	Passed
MPC-20 Test B	MPC-20	25h	Passed

3.1) Null Tests

The null tests checked if the model behaved properly with a null solution. For example, in this test, we checked (i) if the solver found the correct input solution, (ii) if no additional decision variables were added, and (iii) if the objective function and decision variables values were correct. Two null tests (A and B) were performed, as presented in Subsections 3.1.1 and 3.1.2. In Test A, two teams starting in different depots went directly to the sink vertex without unblocking any edge. In Test B, the two teams starting in different depots, traversed some intact roads but did not unblock any edge.

3.1.1) Null Test A

In this test, a known solution was forced into the model. The initial network is the MPC-10 (Figure 9), with a time horizon of 72 hours. Two additional constraints were added to the model (displayed in Figure 13), forcing the teams to go directly to the sink vertex. It is important to highlight that the routes are created with the decision variable x_{ijlk} meaning that team k goes from vertex i to j and then to vertex l. Thus, a team k would go directly from the depot (vertex 0) to the sink vertex (vertex 10) $x_{0,10,10,k}$ is set to 1. However, in MPC-ARCP, self-loops are not allowed (i.e. edge $(10,10)$). Therefore, for a team to go from the depot straight to the sink vertex, it must go to a neighbour vertex. The routes tested are as follows: team 0 starts at vertex 0, moves to 4, and ends its route at the sink vertex (10), and team 1 starts at vertex 6, moves to 2, and ends at vertex 10. The routes are displayed in Figure 14.

```
MPC_ARCP += x[0,4,10,0] == 1
MPC_ARCP += x[6,2,10,1] == 1
```

Figure 13 – Constraints included in '*Null Test A*'

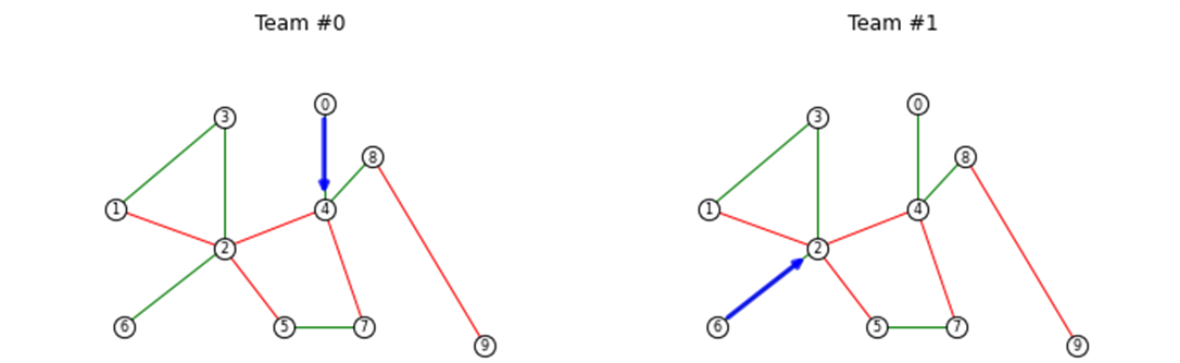


Figure 14 – Road clearing teams' routes in '*Null Test A*'

The results are available in Figure 15. As expected, the objective function is equivalent to zero, as no component was reconnected. In addition, the decision variables related to the unblocking of the routes (z_{ijk}, w_{ijk}, s_{ijk}, and $y_{q_x q_y}$) are null, as expected.

```
Processing time = 0.8935908000000001 seconds
--------
The problem is Optimal
Objective Function Value =  0.0
--------
f_(0,_4,_0) = 2.0
f_(2,_10,_1) = 1.0
f_(4,_10,_0) = 1.0
f_(6,_2,_1) = 2.0
t_(0,_4,_10,_0) = 72.0
t_(6,_2,_10,_1) = 72.0
v_(10,_0) = 1.0
v_(10,_1) = 1.0
v_(2,_1) = 1.0
v_(4,_0) = 1.0
x_(0,_4,_10,_0) = 1.0
x_(6,_2,_10,_1) = 1.0
```

Figure 15 – Results: '*Null Test A*'

3.1.2) Null Test B

This test considers that two teams (0 and 1) traverse the intact edges of a network and go to the sink vertex without clearing any edges. The initial network is MPC-10 (Figure 9), with a time horizon of 72 hours. Five constraints are added to the model to force the routes of teams 0 and 1. The constraints are displayed in Figure 16, and the teams' routes are in Figure 17.

```
MPC_ARCP += x[0,4,8,0] == 1
MPC_ARCP += x[4,8,10,0] == 1

MPC_ARCP += x[6,2,3,1] == 1
MPC_ARCP += x[2,3,1,1] == 1
MPC_ARCP += x[3,1,10,1] == 1
```

Figure 16 – Constraints included in Null Test B

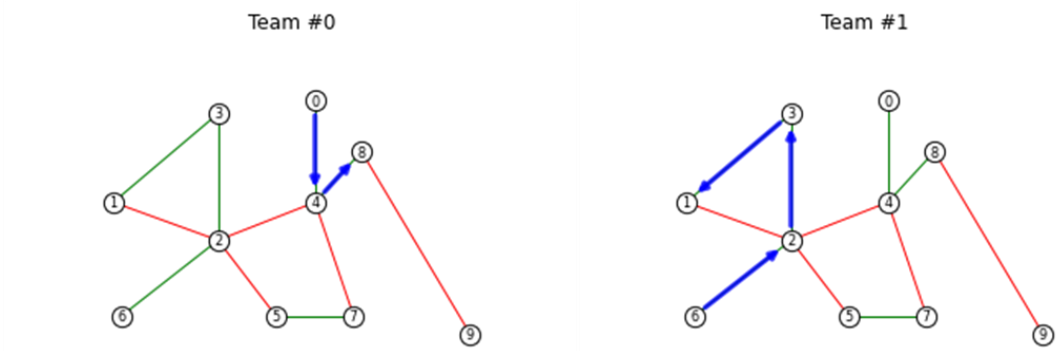


Figure 17 – Road clearing teams' routes in Null Test B

The result of this test is displayed in Figure 18. Since the teams do not reconnect any component, the objective function is zero. Additionally, this is a maximization problem, and since the time horizon is set at 72 hours, GUROBI finds the optimum value of all the decision variables, including the duration of the teams' routes (variable t_{ijlk}). Therefore, the duration of the time decision variables is also correct. For example, going from vertex 4 to 8 takes 0.47 hours, so 72 – 0.47 = 71.53, which is t_{0480}.

```
Processing time = 0.9568071000000007 seconds
--------
The problem is Optimal
Objective Function Value =  0.0
--------
f_(0,_4,_0) = 3.0
f_(1,_10,_1) = 1.0
f_(2,_3,_1) = 3.0
f_(3,_1,_1) = 2.0
f_(4,_8,_0) = 2.0
f_(6,_2,_1) = 4.0
f_(8,_10,_0) = 1.0
t_(0,_4,_8,_0) = 71.53000000000002
t_(2,_3,_1,_1) = 71.12000000000002
t_(3,_1,_10,_1) = 72.0
t_(4,_8,_10,_0) = 72.0
t_(6,_2,_3,_1) = 70.12000000000002
v_(1,_1) = 1.0
v_(10,_0) = 1.0
v_(10,_1) = 1.0
v_(2,_1) = 1.0
v_(3,_1) = 1.0
v_(4,_0) = 1.0
v_(8,_0) = 1.0
x_(0,_4,_8,_0) = 1.0
x_(2,_3,_1,_1) = 1.0
x_(3,_1,_10,_1) = 1.0
x_(4,_8,_10,_0) = 1.0
x_(6,_2,_3,_1) = 1.0
```

Figure 18 – Results: Null Test B

3.2) Single Team

The "Single Team" tests consider that one team unblocks all (or some) of the roads while the other team does not clear anything. In the test "Single Team Test A", a route is forced in the model so that team 0 reconnects **all** the components (described in Subsection 3.2.1). "Single Team Test B" considers that team 0 unblocks **some** of the roads (see Subsection 3.2.2), and "Single Team Test C" considers that team 0 unblocks **some** of the roads and goes back and forth on them (Subsection 3.2.3).

3.2.1) Single Team Test A

In this test, team 0 reconnects all the components in MPC-10. The components are $q_0 = \{0,4,8\}$, $q_1 = \{1,2,3,6\}$, $q_2 = \{5,7\}$, and $q_3 = \{9\}$. The constraints forcing the teams' routes are presented in Figure 19. The teams' routes are presented in Figure 20. Notice that team 1 goes straight to the sink vertex.

```
MPC_ARCP += x[0,4,2,0] == 1
MPC_ARCP += x[4,2,5,0] == 1
MPC_ARCP += x[2,5,7,0] == 1
MPC_ARCP += x[5,7,4,0] == 1
MPC_ARCP += x[7,4,8,0] == 1
MPC_ARCP += x[4,8,9,0] == 1
MPC_ARCP += x[8,9,10,0] == 1

MPC_ARCP += x[6,2,10,1] == 1
```

Figure 19 – Constraints included in Single Team Test A

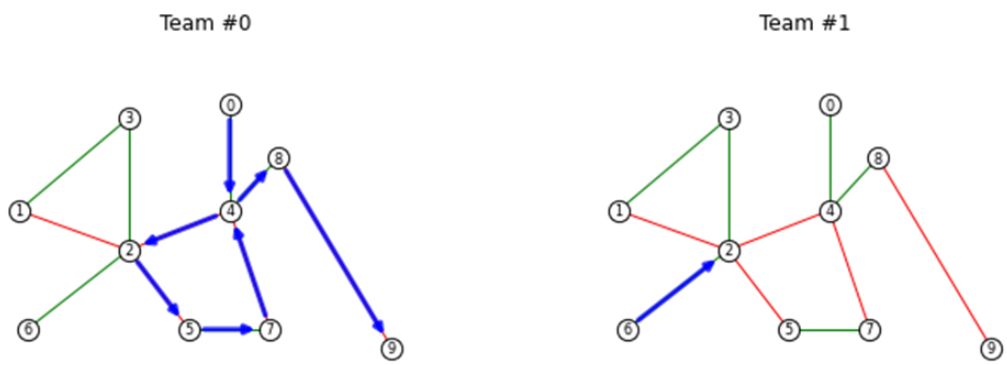


Figure 20 – Road clearing teams' routes in Single Team Test A

It is not possible for team 0 to reconnect all components in 72 hours. Edge (8,9) has an unblocking time of 35 hours, edge (7,4) of 30 hours, edge (2,5) of 15 hours, and edge (4,2) of 10 hours. Thus, the minimum time horizon to clear all the edges is 90 hours (plus the traversing times). As expected, while testing Single Team Test A with a time horizon of 72 hours, the solver could not solve the model (Figure 21).

```
Processing time = 1.2373347 seconds
--------
The problem is Not Solved
Objective Function Value =  None
--------
```

Figure 21 – Results: Single Team Test A – 72 hours

However, as anticipated, after updating the time horizon to 100 hours, the model did find the expected feasible solution. The decision variables values are presented in Figure 22. The schedule of road clearing team 0 is presented in Figure 23.

```
Processing time = 1.1267110000000002 seconds
--------
The problem is Optimal
Objective Function Value =  55.0
--------
a_(2,_5,_0) = 15.0
a_(4,_2,_0) = 10.0
a_(7,_4,_0) = 30.0
a_(8,_9,_0) = 35.0
f_(0,_4,_0) = 8.0
f_(2,_10,_1) = 1.0
f_(2,_5,_0) = 6.0
f_(4,_2,_0) = 7.0
f_(4,_8,_0) = 3.0
f_(5,_7,_0) = 5.0
f_(6,_2,_1) = 2.0
f_(7,_4,_0) = 4.0
f_(8,_9,_0) = 2.0
f_(9,_10,_0) = 1.0
s_(2,_5,_0) = 11.379999999999969
s_(4,_2,_0) = 0.7999999999999687
s_(7,_4,_0) = 27.449999999999967
s_(8,_9,_0) = 58.83999999999996
t_(0,_4,_2,_0) = 0.7999999999999687
t_(2,_5,_7,_0) = 27.04999999999997
t_(4,_2,_5,_0) = 11.379999999999969
t_(4,_8,_9,_0) = 58.83999999999997
t_(5,_7,_4,_0) = 27.449999999999967
t_(6,_2,_10,_1) = 100.0
t_(7,_4,_8,_0) = 58.36999999999997
t_(8,_9,_10,_0) = 95.38999999999996
v_(10,_0) = 1.0
v_(10,_1) = 1.0
v_(2,_0) = 1.0
v_(2,_1) = 1.0
v_(4,_0) = 2.0
v_(5,_0) = 1.0
v_(7,_0) = 1.0
v_(8,_0) = 1.0
v_(9,_0) = 1.0
x_(0,_4,_2,_0) = 1.0
x_(2,_5,_7,_0) = 1.0
x_(4,_2,_5,_0) = 1.0
x_(4,_8,_9,_0) = 1.0
x_(5,_7,_4,_0) = 1.0
x_(6,_2,_10,_1) = 1.0
x_(7,_4,_8,_0) = 1.0
x_(8,_9,_10,_0) = 1.0
y_(0,_1) = 1.0
y_(0,_2) = 1.0
y_(0,_3) = 1.0
y_(1,_0) = 1.0
y_(1,_2) = 1.0
y_(2,_0) = 1.0
y_(2,_1) = 1.0
y_(3,_0) = 1.0
z_(2,_5,_0) = 1.0
z_(4,_2,_0) = 1.0
z_(7,_4,_0) = 1.0
z_(8,_9,_0) = 1.0
```

Figure 22 – Results: Single Team Test A – 100 hours

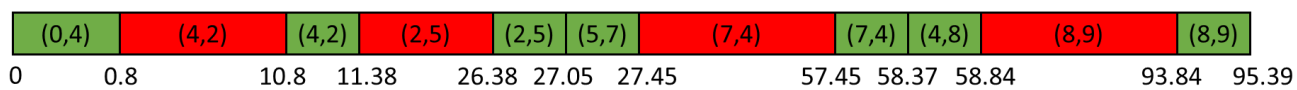


Figure 23 – Schedule of road clearing team 0 – Single Team Test A – 100 hours

3.2.2) Single Team Test B

In this test, team 0 reconnects some of the components in instance MPC-10, and team 1 does not unblock any edge. The components being reconnected are $q_0 = \{0,4,8\}$, $q_1 = \{1,2,3,6\}$ and $q_2 = \{5,7\}$. The constraints forcing the teams' routes are presented in Figure 24, with the routes presented in Figure 25. The time horizon is set at 72 hours.

```
MPC_ARCP += x[0,4,7,0] == 1
MPC_ARCP += x[4,7,5,0] == 1
MPC_ARCP += x[7,5,2,0] == 1
MPC_ARCP += x[5,2,4,0] == 1
MPC_ARCP += x[2,4,10,0] == 1

MPC_ARCP += x[6,2,3,1] == 1
MPC_ARCP += x[2,3,10,1] == 1
```

Figure 24 – Constraints included in Single Team Test B

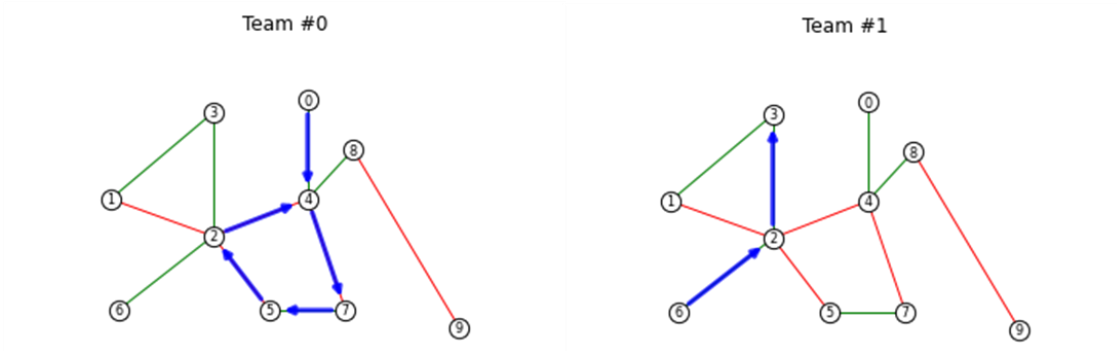


Figure 25 – Road clearing teams' routes in Single Team Test B

The results of the decision variables are presented in Figure 26. The model did not add any extra decision variable, and the time variables are correct. A schedule of team's 0 route is presented in Figure 27.

```
Processing time = 0.5571473999999998 seconds
--------
The problem is Optimal
Objective Function Value =  40.0
--------
a_(2,_4,_0) = 10.0
a_(4,_7,_0) = 30.0
a_(5,_2,_0) = 15.0
f_(0,_4,_0) = 6.0
f_(2,_3,_1) = 2.0
f_(2,_4,_0) = 2.0
f_(3,_10,_1) = 1.0
f_(4,_10,_0) = 1.0
f_(4,_7,_0) = 5.0
f_(5,_2,_0) = 3.0
f_(6,_2,_1) = 3.0
f_(7,_5,_0) = 4.0
s_(2,_4,_0) = 47.789999999999985
s_(4,_7,_0) = 0.799999999999983
s_(5,_2,_0) = 32.11999999999998
t_(0,_4,_7,_0) = 0.799999999999983
t_(2,_3,_10,_1) = 72.0
t_(2,_4,_10,_0) = 58.36999999999998
t_(4,_7,_5,_0) = 31.719999999999985
t_(5,_2,_4,_0) = 47.789999999999985
t_(6,_2,_3,_1) = 71.0
t_(7,_5,_2,_0) = 32.11999999999998
v_(10,_0) = 1.0
v_(10,_1) = 1.0
v_(2,_0) = 1.0
v_(2,_1) = 1.0
v_(3,_1) = 1.0
v_(4,_0) = 2.0
v_(5,_0) = 1.0
v_(7,_0) = 1.0
x_(0,_4,_7,_0) = 1.0
x_(2,_3,_10,_1) = 1.0
x_(2,_4,_10,_0) = 1.0
x_(4,_7,_5,_0) = 1.0
x_(5,_2,_4,_0) = 1.0
x_(6,_2,_3,_1) = 1.0
x_(7,_5,_2,_0) = 1.0
y_(0,_1) = 1.0
y_(0,_2) = 1.0
y_(1,_0) = 1.0
y_(1,_2) = 1.0
y_(2,_0) = 1.0
y_(2,_1) = 1.0
z_(2,_4,_0) = 1.0
z_(4,_7,_0) = 1.0
z_(5,_2,_0) = 1.0
```

Figure 26 – Results: Single Team Test B

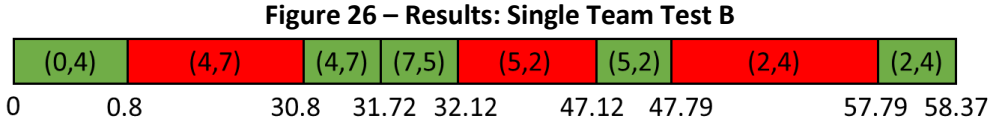


Figure 27 – Schedule of road clearing team 0 – Single Team Test B

3.2.3) Single Team Test C

In this test, team 0 reconnects some of the components in the instance MPC-10. Moreover, team 0 goes back and forth on one of the edges. Single Team Test C evaluates if the team clears an edge the first time the team traverses it. Team 1 does not clear any edge. The constraints included in this test for E-ARCP are presented in Figure 28, and the routes are displayed in Figure 29. The time horizon considered was 72 hours.

```
MPC_ARCP += x[0,4,2,0] == 1
MPC_ARCP += x[4,2,4,0] == 1
MPC_ARCP += x[2,4,7,0] == 1
MPC_ARCP += x[4,7,10,0] == 1

MPC_ARCP += x[6,2,10,1] == 1
```

Figure 28 – Constraints included in Single Team Test C

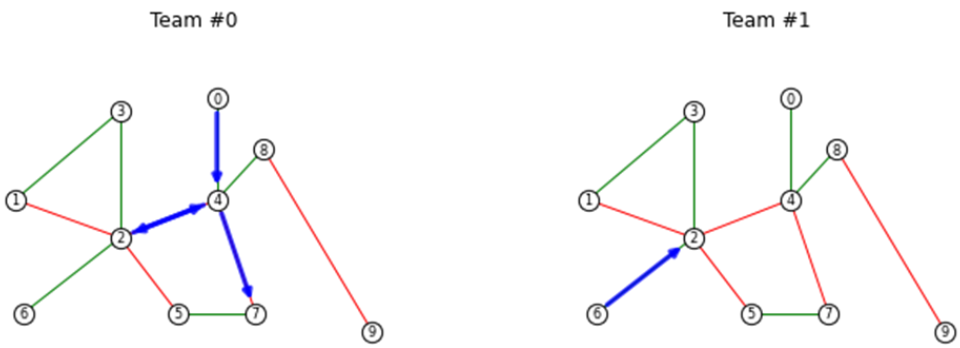


Figure 29 – Road clearing teams' routes in Single Team Test C

The results found by the model are presented in Figure 30. Since team 0 enters edge (4,2) and returns at (2,4), the correct answer is that team 0 clears edge (4,2) and traverses (2,4). As presented in Figure 30, z_{420} equals one, which means that team 0 is clearing in the direction (4,2). Additionally, the time at which team 0 can enter edge (2,4) is greater than the time it can enter edge (4,2). Therefore, the model passed this test. The schedule of road clearing team 0 is provided in Figure 31.

```
Processing time = 0.847353 seconds
--------
The problem is Optimal
Objective Function Value =  25.0
--------
a_(4,_2,_0) = 10.0
a_(4,_7,_0) = 30.0
f_(0,_4,_0) = 5.0
f_(2,_10,_1) = 1.0
f_(2,_4,_0) = 3.0
f_(4,_2,_0) = 4.0
f_(4,_7,_0) = 2.0
f_(6,_2,_1) = 2.0
f_(7,_10,_0) = 1.0
s_(2,_4,_0) = 11.379999999999983
s_(4,_2,_0) = 0.799999999999983
s_(4,_7,_0) = 11.95999999999998
t_(0,_4,_2,_0) = 0.799999999999983
t_(2,_4,_7,_0) = 11.959999999999983
t_(4,_2,_4,_0) = 11.379999999999983
t_(4,_7,_10,_0) = 42.87999999999998
t_(6,_2,_10,_1) = 72.0
v_(10,_0) = 1.0
v_(10,_1) = 1.0
v_(2,_0) = 1.0
v_(2,_1) = 1.0
v_(4,_0) = 2.0
v_(7,_0) = 1.0
x_(0,_4,_2,_0) = 1.0
x_(2,_4,_7,_0) = 1.0
x_(4,_2,_4,_0) = 1.0
x_(4,_7,_10,_0) = 1.0
x_(6,_2,_10,_1) = 1.0
y_(0,_1) = 1.0
y_(0,_2) = 1.0
y_(1,_0) = 1.0
y_(2,_0) = 1.0
z_(4,_2,_0) = 1.0
z_(4,_7,_0) = 1.0
```

Figure 30 – Results: Single Team Test C

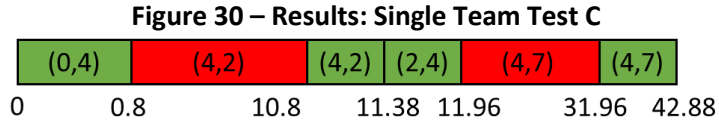


Figure 31 – Schedule of road clearing team 0 – Single Team Test C

3.3) Two Teams Tests

The '*Two Teams*' tests consider that two teams unblock the edges in a disconnected network, either separately or collaboratively. In the test '*Two Teams Test A*', the teams unblock some of the edges, but they do not collaborate (Subsection 3.3.1). The test '*Two Teams Test B*' considers that teams 0 and 1 collaborate to clear one edge and then go to the sink vertex (Subsection 3.3.2). In '*Two Teams Test C*', the teams collaborate to clear one edge and then clear other roads (Subsection 3.3.3). Test '*Two Teams Test D*' considers that the teams collaborate to clear two edges (Subsection 3.3.4).

3.3.1) Two Teams Test A

In this test, teams 0 and 1 unblock some of the edges in the instance MPC-10. However, they do not collaborate. Team 0 clears edges (8,9) and (4,7), whereas team 1 clears edges (2,4) and (2,5). The constraints forcing the teams' routes are presented in Figure 32. The teams' routes are presented in Figure 33.

```
MPC_ARCP += x[0,4,8,0] == 1
MPC_ARCP += x[4,8,9,0] == 1
MPC_ARCP += x[8,9,8,0] == 1
MPC_ARCP += x[9,8,4,0] == 1
MPC_ARCP += x[8,4,7,0] == 1
MPC_ARCP += x[4,7,10,0] == 1

MPC_ARCP += x[6,2,4,1] == 1
MPC_ARCP += x[2,4,2,1] == 1
MPC_ARCP += x[4,2,5,1] == 1
MPC_ARCP += x[2,5,10,1] == 1
```

Figure 32 – Constraints included in '*Two Teams Test A*'

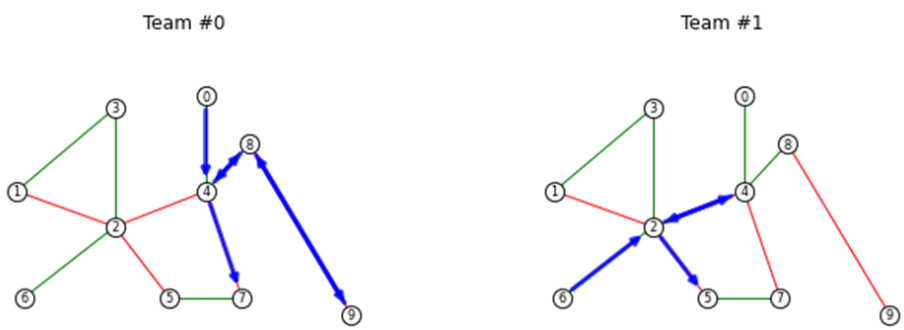


Figure 33 – Road clearing teams' routes in '*Two Teams Test A*'

The results of '*Two Teams Test A*' are available in Figure 34. As mentioned, team 0 clears edges (8,9) and (4,7), so variables $a_{470} = 30\ hours$ and $a_{890} = 35\ hours$ have the expected values. Similarly, team 1 clears edges (2,4) and (2,5), and variables $a_{241} = 10\ hours$ and $a_{251} = 15\ hours$.

```
Processing time = 0.8455250999999997 seconds
--------
The problem is Optimal
Objective Function Value =  55.0
--------
a_(2,_4,_1) = 10.0
a_(2,_5,_1) = 15.0
a_(4,_7,_0) = 30.0
a_(8,_9,_0) = 35.0
f_(0,_4,_0) = 7.0
f_(2,_4,_1) = 4.0
f_(2,_5,_1) = 2.0
f_(4,_2,_1) = 3.0
f_(4,_7,_0) = 2.0
f_(4,_8,_0) = 6.0
f_(5,_10,_1) = 1.0
f_(6,_2,_1) = 5.0
f_(7,_10,_0) = 1.0
f_(8,_4,_0) = 3.0
f_(8,_9,_0) = 5.0
f_(9,_8,_0) = 4.0
s_(2,_4,_1) = 0.7799999999999869
s_(2,_5,_1) = 11.939999999999985
s_(4,_2,_1) = 11.359999999999987
s_(4,_7,_0) = 39.839999999999904
s_(8,_9,_0) = 1.269999999999932
s_(9,_8,_0) = 37.81999999999991
t_(0,_4,_8,_0) = 0.7999999999999676
t_(2,_4,_2,_1) = 11.359999999999987
t_(2,_5,_10,_1) = 27.609999999999985
t_(4,_2,_5,_1) = 11.939999999999987
t_(4,_7,_10,_0) = 70.7599999999999
t_(4,_8,_9,_0) = 1.2699999999999676
t_(6,_2,_4,_1) = 0.7799999999999869
t_(8,_4,_7,_0) = 39.83999999999991
t_(8,_9,_8,_0) = 37.81999999999993
t_(9,_8,_4,_0) = 39.369999999999905
v_(10,_0) = 1.0
v_(10,_1) = 1.0
v_(2,_1) = 2.0
v_(4,_0) = 2.0
v_(4,_1) = 1.0
v_(5,_1) = 1.0
v_(7,_0) = 1.0
v_(8,_0) = 2.0
v_(9,_0) = 1.0
x_(0,_4,_8,_0) = 1.0
x_(2,_4,_2,_1) = 1.0
x_(2,_5,_10,_1) = 1.0
x_(4,_2,_5,_1) = 1.0
x_(4,_7,_10,_0) = 1.0
x_(4,_8,_9,_0) = 1.0
x_(6,_2,_4,_1) = 1.0
x_(8,_4,_7,_0) = 1.0
x_(8,_9,_8,_0) = 1.0
x_(9,_8,_4,_0) = 1.0
y_(0,_1) = 1.0
y_(0,_2) = 1.0
y_(0,_3) = 1.0
y_(1,_0) = 1.0
y_(1,_2) = 1.0
y_(2,_0) = 1.0
y_(2,_1) = 1.0
y_(3,_0) = 1.0
z_(2,_4,_1) = 1.0
z_(2,_5,_1) = 1.0
z_(4,_7,_0) = 1.0
z_(8,_9,_0) = 1.0
```

Figure 34 – Results: '*Two Teams Test A*'

3.3.2) Two Teams Test B

In this test, teams 0 and 1 collaborate to clear edge (2,4). Team 0 starts on the side (4,2) and team 1 starts at (2,4). Notice that team 1 is the first to arrive as it takes 0.78h to reach vertex 2, whereas team 0 takes 0.8h. In this problem, a time horizon of 6.37h was used to force the teams to collaborate. The constraints added to the model are given in Figure 35. The teams' routes are presented in Figure 36.

```
MPC_ARCP += x[0,4,2,0] == 1
MPC_ARCP += x[4,2,10,0] == 1

MPC_ARCP += x[6,2,4,1] == 1
MPC_ARCP += x[2,4,10,1] == 1
```

Figure 35 – Constraints included in '*Two Teams Test B*'

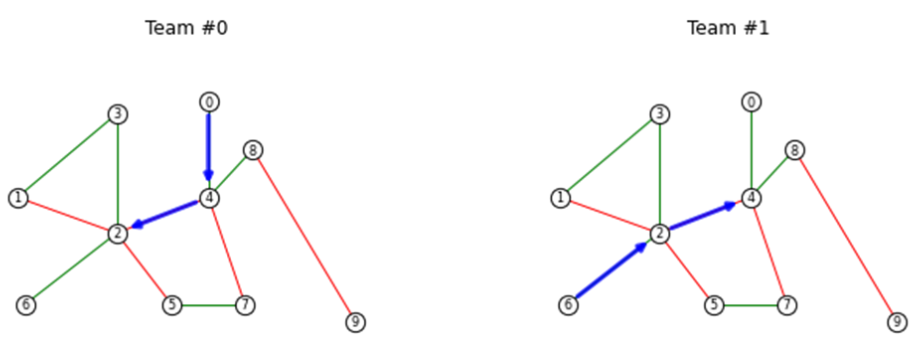


Figure 36 – Road clearing teams' routes in '*Two Teams Test B*'

The results of this test are available in Figure 37. Road clearing team 1 is the first to arrive at edge (2,4), with ($w_{241} = 1$ and $w_{420} = 0$). Notice that team 1 enters edge (2,4) at time 0.78 (s_{241}), whereas team 0 enters (4,2) at time 0.8 (s_{420}). Therefore, team 1 clears edge (2,4) for 5.01h (a_{241}) and team 0 clears for 4.99h (a_{420}). The total clearing time of edge (2,4) is 5.01 + 4.99 = 10 hours. After clearing the edge, the teams traverse it (0.58h) and go to the sink vertex. The schedule of this test is presented in Figure 38, and all the decision variables have the expected value. It is concluded that the E-ARCP software passes this test.

```
Processing time = 0.8325408999999997 seconds
--------
The problem is Optimal
Objective Function Value =  10.0
--------
a_(2,_4,_1) = 5.010000000000002
a_(4,_2,_0) = 4.990000000000002
f_(0,_4,_0) = 3.0
f_(2,_10,_0) = 1.0
f_(2,_4,_1) = 2.0
f_(4,_10,_1) = 1.0
f_(4,_2,_0) = 2.0
f_(6,_2,_1) = 3.0
s_(2,_4,_1) = 0.7799999999999985
s_(4,_2,_0) = 0.799999999999998
t_(0,_4,_2,_0) = 0.7999999999999989
t_(2,_4,_10,_1) = 6.37
t_(4,_2,_10,_0) = 6.37
t_(6,_2,_4,_1) = 0.7799999999999994
v_(10,_0) = 1.0
v_(10,_1) = 1.0
v_(2,_0) = 1.0
v_(2,_1) = 1.0
v_(4,_0) = 1.0
v_(4,_1) = 1.0
w_(2,_4,_1) = 1.0
x_(0,_4,_2,_0) = 1.0
x_(2,_4,_10,_1) = 1.0
x_(4,_2,_10,_0) = 1.0
x_(6,_2,_4,_1) = 1.0
y_(0,_1) = 1.0
y_(1,_0) = 1.0
z_(2,_4,_1) = 1.0
z_(4,_2,_0) = 1.0
```

Figure 37 – Results: '*Two Teams Test B*'

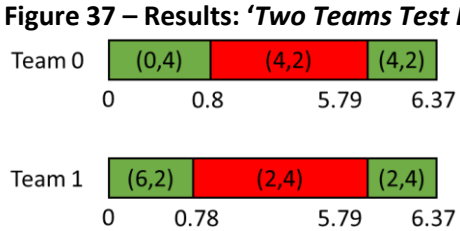


Figure 38 – Schedule of road clearing teams – '*Two Teams Test B*'

3.3.3) Two Teams Test C

In this test, teams 0 and 1 collaborate to clear edge (2,4), then continue their routes clearing other edges. The purpose of this test is to check whether the synchronization times are being calculated properly. The constraints added to the model are given in Figure 39, and the teams' routes are depicted in Figure 40. The time horizon was set to 55 hours, to force the teams to collaborate.

```
MPC_ARCP += x[0,4,2,0] == 1
MPC_ARCP += x[4,2,5,0] == 1
MPC_ARCP += x[2,5,7,0] == 1
MPC_ARCP += x[5,7,4,0] == 1
MPC_ARCP += x[7,4,10,0] == 1

MPC_ARCP += x[6,2,4,1] == 1
MPC_ARCP += x[2,4,8,1] == 1
MPC_ARCP += x[4,8,9,1] == 1
MPC_ARCP += x[8,9,10,1] == 1
```

Figure 39 – Constraints included in '*Two Teams Test C*'

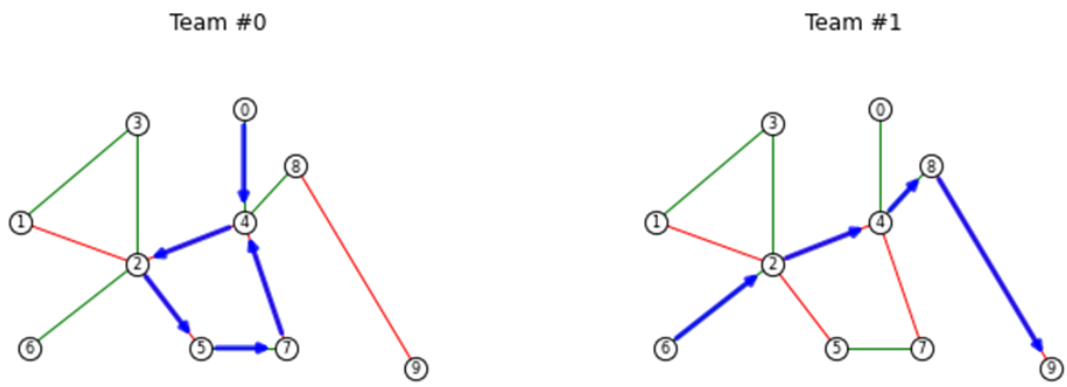


Figure 40 – Road clearing teams' routes in '*Two Teams Test C*'

The routes of teams 0 and 1 start as in '*Two Teams Test B*' (Subsection 3.3.2), and teams 0 and 1 collaborate to clear edge (2,4). Then, team 0 clears edges (2,5) and (7,4). Team 1 clears edge (8,9) after (2,4). As depicted in Figure 41, team 1 clears edge (2,5) for 5.01 hours (a_{241}) and team 0 for 4.99 hours (a_{420}). The teams clear the other edges independently ($a_{250} = 15\ hours$, $a_{740} = 30\ hours$ and $a_{891} = 35\ hours$). The model recognized that team 1 is the first to arrive at edge (2,4) ($w_{241} = 1$). Note that since the teams collaborate to clear the same edge, $z_{241} = 1$ and $z_{420} = 1$. All other decision variables are correct, and the schedule of the road clearing teams is available in Figure 42.

```
Processing time = 0.8067875000000004 seconds
--------
The problem is Optimal
Objective Function Value =  55.0
--------
a_(2,_4,_1) = 5.010000000000055
a_(2,_5,_0) = 15.0
a_(4,_2,_0) = 4.990000000000059
a_(7,_4,_0) = 30.0
a_(8,_9,_1) = 35.0
f_(0,_4,_0) = 6.0
f_(2,_4,_1) = 4.0
f_(2,_5,_0) = 4.0
f_(4,_10,_0) = 1.0
f_(4,_2,_0) = 5.0
f_(4,_8,_1) = 3.0
f_(5,_7,_0) = 3.0
f_(6,_2,_1) = 5.0
f_(7,_4,_0) = 2.0
f_(8,_9,_1) = 2.0
f_(9,_10,_1) = 1.0
z_(2,_4,_1) = 1.0
z_(2,_5,_0) = 1.0
z_(4,_2,_0) = 1.0
z_(7,_4,_0) = 1.0
z_(8,_9,_1) = 1.0
s_(2,_4,_1) = 0.7799999999999869
s_(2,_5,_0) = 6.370000000000042
s_(4,_2,_0) = 0.799999999999983
s_(7,_4,_0) = 22.44000000000004
s_(8,_9,_1) = 6.840000000000046
t_(0,_4,_2,_0) = 0.799999999999983
t_(2,_4,_8,_1) = 6.370000000000042
t_(2,_5,_7,_0) = 22.040000000000042
t_(4,_2,_5,_0) = 6.370000000000042
t_(4,_8,_9,_1) = 6.840000000000046
t_(5,_7,_4,_0) = 22.44000000000004
t_(6,_2,_4,_1) = 0.7799999999999869
t_(7,_4,_10,_0) = 53.36000000000004
t_(8,_9,_10,_1) = 43.39000000000004
y_(0,_1) = 1.0
y_(0,_2) = 1.0
y_(0,_3) = 1.0
y_(1,_0) = 1.0
y_(1,_2) = 1.0
y_(2,_0) = 1.0
y_(2,_1) = 1.0
y_(3,_0) = 1.0
v_(10,_0) = 1.0
v_(10,_1) = 1.0
v_(2,_0) = 1.0
v_(2,_1) = 1.0
v_(4,_0) = 2.0
v_(4,_1) = 1.0
v_(5,_0) = 1.0
v_(7,_0) = 1.0
v_(8,_1) = 1.0
v_(9,_1) = 1.0
w_(2,_4,_1) = 1.0
x_(0,_4,_2,_0) = 1.0
x_(2,_4,_8,_1) = 1.0
x_(2,_5,_7,_0) = 1.0
x_(4,_2,_5,_0) = 1.0
x_(4,_8,_9,_1) = 1.0
x_(5,_7,_4,_0) = 1.0
x_(6,_2,_4,_1) = 1.0
x_(7,_4,_10,_0) = 1.0
x_(8,_9,_10,_1) = 1.0
```

Figure 41 – Results: '*Two Teams Test C*'

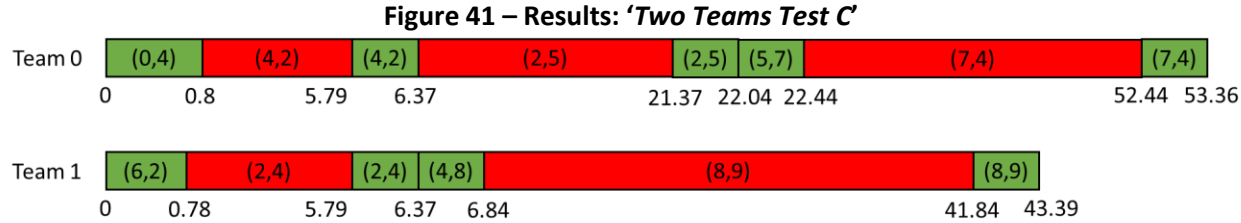


Figure 42 – Schedule of road clearing teams – '*Two Teams Test C*'

3.3.4) Two Teams Test D

In this test, teams 0 and 1 collaborate to clear two edges (i.e. (2,4) and (4,7)). A time horizon of 31 hours was used to force the teams to collaborate. The constraints added to the model are presented in Figure 43, and the teams' routes are shown in Figure 44.

```
MPC_ARCP += x[0,4,2,0] == 1
MPC_ARCP += x[4,2,5,0] == 1
MPC_ARCP += x[2,5,7,0] == 1
MPC_ARCP += x[5,7,4,0] == 1
MPC_ARCP += x[7,4,10,0] == 1

MPC_ARCP += x[6,2,4,1] == 1
MPC_ARCP += x[2,4,7,1] == 1
MPC_ARCP += x[4,7,10,1] == 1
```

Figure 43 – Constraints included in '*Two Teams Test D*'

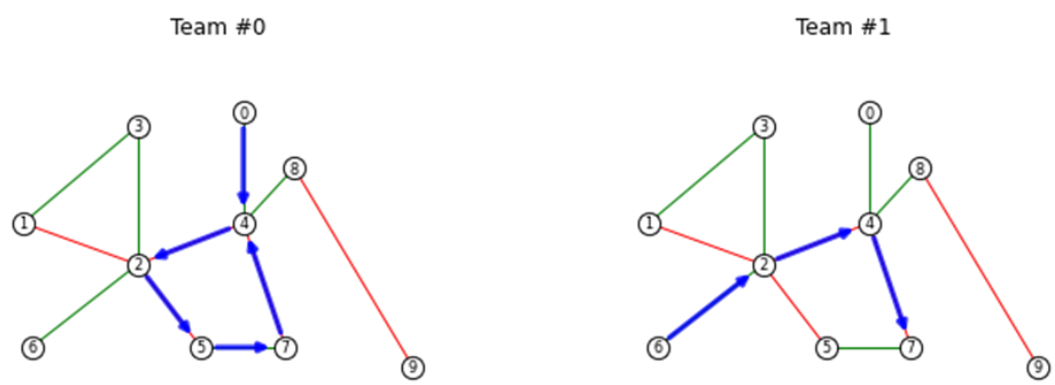


Figure 44 – Road clearing teams' routes in '*Two Teams Test D*'

In this test, team 1 is the first to arrive to clear both edges (2,4) and (4,7), thus w_{241} and w_{471} are equal one. Edge (2,4) requires a total unblocking time of 10 hours, which is equivalent to the sum of a_{241} and a_{420}. Similarly, edge (4,7) requires a total unblocking time of 30 hours with team 1 clearing 23.035 (a_{471}) and team 0 clearing 6.965 (a_{740}). All the other decision variables have the expected value, as shown in Figure 45. The schedule of the teams' routes is presented in Figure 46.

```
Processing time = 0.9395664999999997 seconds
--------
The problem is Optimal
Objective Function Value =  40.0
--------
a_(2,_4,_1) = 5.010000000000057
a_(2,_5,_0) = 15.0
a_(4,_2,_0) = 4.990000000000057
a_(4,_7,_1) = 23.03500000000006
a_(7,_4,_0) = 6.965000000000057
f_(0,_4,_0) = 6.0
f_(2,_4,_1) = 3.0
f_(2,_5,_0) = 4.0
f_(4,_10,_0) = 1.0
f_(4,_2,_0) = 5.0
f_(4,_7,_1) = 2.0
f_(5,_7,_0) = 3.0
f_(6,_2,_1) = 4.0
f_(7,_10,_1) = 1.0
f_(7,_4,_0) = 2.0
s_(2,_4,_1) = 0.779999999999994
s_(2,_5,_0) = 6.370000000000051
s_(4,_2,_0) = 0.7999999999999936
s_(4,_7,_1) = 6.370000000000047
s_(7,_4,_0) = 22.440000000000055
t_(0,_4,_2,_0) = 0.7999999999999936
t_(2,_4,_7,_1) = 6.370000000000051
t_(2,_5,_7,_0) = 22.04000000000005
t_(4,_2,_5,_0) = 6.370000000000051
t_(4,_7,_10,_1) = 30.32500000000011
t_(5,_7,_4,_0) = 22.44000000000005
t_(6,_2,_4,_1) = 0.779999999999994
t_(7,_4,_10,_0) = 30.32500000000011
v_(10,_0) = 1.0
v_(10,_1) = 1.0
v_(2,_0) = 1.0
v_(2,_1) = 1.0
v_(4,_0) = 2.0
v_(4,_1) = 1.0
v_(5,_0) = 1.0
v_(7,_0) = 1.0
v_(7,_1) = 1.0
w_(2,_4,_1) = 1.0
w_(4,_7,_1) = 1.0
x_(0,_4,_2,_0) = 1.0
x_(2,_4,_7,_1) = 1.0
x_(2,_5,_7,_0) = 1.0
x_(4,_2,_5,_0) = 1.0
x_(4,_7,_10,_1) = 1.0
x_(5,_7,_4,_0) = 1.0
x_(6,_2,_4,_1) = 1.0
x_(7,_4,_10,_0) = 1.0
y_(0,_1) = 1.0
y_(0,_2) = 1.0
y_(1,_0) = 1.0
y_(1,_2) = 1.0
y_(2,_0) = 1.0
y_(2,_1) = 1.0
z_(2,_4,_1) = 1.0
z_(2,_5,_0) = 1.0
z_(4,_2,_0) = 1.0
z_(4,_7,_1) = 1.0
z_(7,_4,_0) = 1.0
```

Figure 45 – Results: '*Two Teams Test D*'

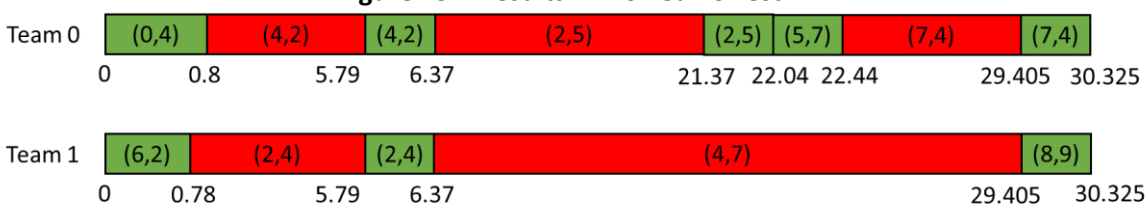


Figure 46 – Schedule of road clearing teams – '*Two Teams Test D*'

3.4) Three Teams Test

The '*Three Teams*' test consider that two teams unblock one edge, while another team must wait to traverse it. The time horizon considered is 7.75 hours to force the teams to collaborate while clearing edge (2,4). The constraints with the teams' routes are depicted in Figure 47, and the routes are displayed in Figure 48. This problem considers three teams: team 0 is positioned in vertex 0, and teams 1 and 2 are in vertex 6. The instance used is MPC-10, but since one team was added, this instance was named "MPC-10-2".

```
MPC_ARCP += x[0,4,2,0] == 1
MPC_ARCP += x[4,2,10,0] == 1

MPC_ARCP += x[6,2,4,1] == 1
MPC_ARCP += x[2,4,0,1] == 1
MPC_ARCP += x[4,0,10,1] == 1

MPC_ARCP += x[6,2,4,2] == 1
MPC_ARCP += x[2,4,10,2] == 1
```

Figure 47 – Constraints included in '*Three Teams Test*'

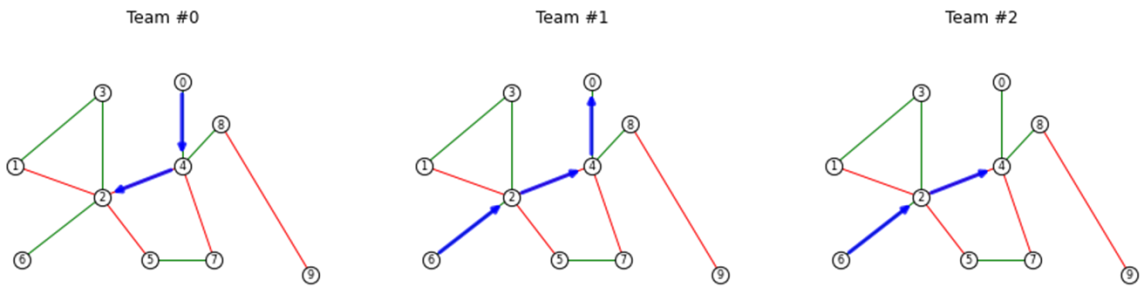


Figure 48 – Road clearing teams' routes in '*Three Teams Test*'

The results are displayed in Figure 49, with the schedule of the road clearing teams shown in Figure 50. As demonstrated in Figure 50, team 2 waits, while teams 0 and 1 clear edge (2,4). Notice that since this is a maximization problem, GUROBI tries to maximize all decision variables, and therefore it schedules for team 1 to traverse edge (0,4) arriving exactly at the time horizon ($t_{4_0_10_1} = 7.75$). Thus, there is a slack between edges (2,4) and (4,0) in team 1.

```
Processing time = 1.0446064000000002 seconds
--------
The problem is Optimal
Objective Function Value =  10.0
--------
a_(2,_4,_1) = 5.010000000000055
a_(4,_2,_0) = 4.990000000000059
f_(0,_10,_1) = 1.0
f_(0,_4,_0) = 3.0
f_(2,_10,_0) = 1.0
f_(2,_4,_1) = 3.0
f_(2,_4,_2) = 2.0
f_(4,_0,_1) = 2.0
f_(4,_10,_2) = 1.0
f_(4,_2,_0) = 2.0
f_(6,_2,_1) = 4.0
f_(6,_2,_2) = 3.0
s_(2,_4,_1) = 0.779999999999994
s_(2,_4,_2) = 6.370000000000077
s_(4,_2,_0) = 0.79999999999999
t_(0,_4,_2,_0) = 0.79999999999999
t_(2,_4,_0,_1) = 6.370000000000049
t_(2,_4,_10,_2) = 6.9500000000000774
t_(4,_0,_10,_1) = 7.75
t_(4,_2,_10,_0) = 6.370000000000049
t_(6,_2,_4,_1) = 0.779999999999994
t_(6,_2,_4,_2) = 0.779999999999994
v_(0,_1) = 1.0
v_(10,_0) = 1.0
v_(10,_1) = 1.0
v_(10,_2) = 1.0
v_(2,_0) = 1.0
v_(2,_1) = 1.0
v_(2,_2) = 1.0
v_(4,_0) = 1.0
v_(4,_1) = 1.0
v_(4,_2) = 1.0
w_(2,_4,_1) = 1.0
x_(0,_4,_2,_0) = 1.0
x_(2,_4,_0,_1) = 1.0
x_(2,_4,_10,_2) = 1.0
x_(4,_0,_10,_1) = 1.0
x_(4,_2,_10,_0) = 1.0
x_(6,_2,_4,_1) = 1.0
x_(6,_2,_4,_2) = 1.0
y_(0,_1) = 1.0
y_(1,_0) = 1.0
z_(2,_4,_1) = 1.0
z_(4,_2,_0) = 1.0
```

Figure 49 – Results: '*Three Teams*'

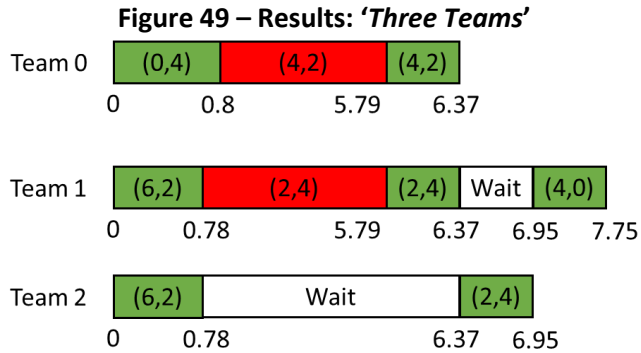


Figure 50 – Schedule of road clearing teams – '*Three Teams*' test

3.5) MPC-10

The '*MPC-10*' test consists of running the model without forcing a known solution. The purpose of this test is to check if the model behaviour is as expected when the time horizon is reduced. For example, a shorter time horizon leads to less clearance. Three cases were tested: A) time horizon of 72 hours, B) time horizon of 20 hours, and C) time horizon of 10 hours. The results of tests '*MPC-10 Test A*', '*MPC-10 Test B*', and '*MPC-10 Test C*' are presented in Subsections 3.5.1, 3.5.2, and 3.5.3, respectively. As presented in Table 1, E-ARCP passed these three tests.

3.5.1) MPC-10 Test A

In this test, the MPC-10 instance is solved considering a time horizon of 72 hours. Notice that no solution is forced in this case. This test aims to check if the model will find an optimal solution to MPC-10 and if all the constraints are respected. For example, this test checked (i) if the model presented a solution with a subtour to the teams' routes, (ii) if the total unblocking times of the edges were respected, and (iii) if all decision variables' values are correct. The results are displayed in Figure 51, and the teams' routes are shown in Figure 52. As mentioned, since this is a maximization problem, GUROBI tries to maximize all decision variables. Thus, since 72 hours is a lengthy time horizon for this case, the teams keep going back and forth on the edges to have routes with a duration of 72 hours. All components were connected, and there are no subtours on the teams' routes.

```
Processing time = 0.9279960000000003 seconds
--------
The problem is Optimal
Objective Function Value =  55.0
--------
a_(2,_4,_1) = 10.0
a_(2,_5,_0) = 15.0
a_(4,_7,_1) = 30.0
a_(8,_9,_0) = 35.0
f_(0,_4,_0) = 10.0
f_(0,_4,_1) = 4.0
f_(2,_4,_1) = 6.0
f_(2,_5,_0) = 4.0
f_(2,_6,_0) = 2.0
f_(4,_0,_1) = 5.0
f_(4,_10,_1) = 1.0
f_(4,_2,_0) = 5.0
f_(4,_7,_1) = 3.0
f_(4,_8,_0) = 9.0
f_(5,_2,_0) = 3.0
f_(6,_10,_0) = 1.0
f_(6,_2,_1) = 7.0
f_(7,_4,_1) = 2.0
f_(8,_4,_0) = 6.0
f_(8,_9,_0) = 8.0
f_(9,_8,_0) = 7.0
v_(0,_1) = 1.0
v_(10,_0) = 1.0
v_(10,_1) = 1.0
v_(2,_0) = 2.0
v_(2,_1) = 1.0
v_(4,_0) = 2.0
v_(4,_1) = 3.0
v_(5,_0) = 1.0
v_(6,_0) = 1.0
v_(7,_1) = 1.0
v_(8,_0) = 2.0
v_(9,_0) = 1.0
x_(0,_4,_7,_1) = 1.0
x_(0,_4,_8,_0) = 1.0
x_(2,_4,_0,_1) = 1.0
x_(2,_5,_2,_0) = 1.0
x_(2,_6,_10,_0) = 1.0
x_(4,_0,_4,_1) = 1.0
x_(4,_2,_5,_0) = 1.0
x_(4,_7,_4,_1) = 1.0
x_(4,_8,_9,_0) = 1.0
x_(5,_2,_6,_0) = 1.0
x_(6,_2,_4,_1) = 1.0
x_(7,_4,_10,_1) = 1.0
x_(8,_4,_2,_0) = 1.0
x_(8,_9,_8,_0) = 1.0
x_(9,_8,_4,_0) = 1.0
s_(2,_4,_1) = 27.980000000000004
s_(2,_5,_0) = 50.419999999999995
s_(4,_2,_0) = 39.839999999999996
s_(4,_7,_1) = 40.16
s_(5,_2,_0) = 66.08999999999999
s_(7,_4,_1) = 71.08
s_(8,_9,_0) = 1.27
s_(9,_8,_0) = 37.82
t_(0,_4,_7,_1) = 40.16
t_(0,_4,_8,_0) = 0.8
t_(2,_4,_0,_1) = 38.56
t_(2,_5,_2,_0) = 66.08999999999999
t_(2,_6,_10,_0) = 72.0
t_(4,_0,_4,_1) = 39.36
t_(4,_2,_5,_0) = 40.419999999999995
t_(4,_7,_4,_1) = 71.08
t_(4,_8,_9,_0) = 1.27
t_(5,_2,_6,_0) = 66.75999999999999
t_(6,_2,_4,_1) = 27.980000000000004
t_(7,_4,_10,_1) = 72.0
t_(8,_4,_2,_0) = 39.84
t_(8,_9,_8,_0) = 37.82
t_(9,_8,_4,_0) = 39.37
y_(0,_1) = 1.0
y_(0,_2) = 1.0
y_(0,_3) = 1.0
y_(1,_0) = 1.0
y_(1,_2) = 1.0
y_(2,_0) = 1.0
y_(2,_1) = 1.0
y_(3,_0) = 1.0
z_(2,_4,_1) = 1.0
z_(2,_5,_0) = 1.0
z_(4,_7,_1) = 1.0
z_(8,_9,_0) = 1.0
```

Figure 51 – Results: '*MPC-10 Test A*'

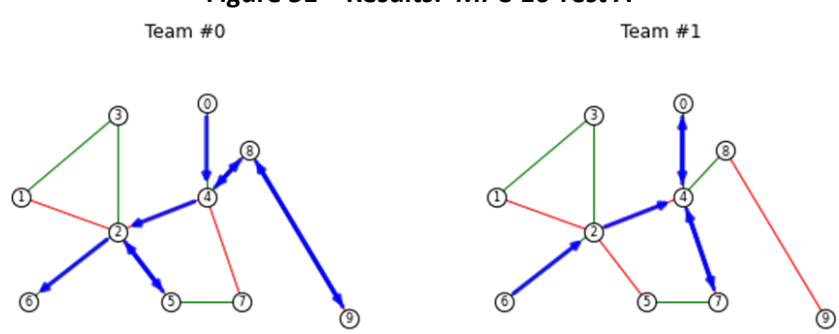


Figure 52 – Road clearing teams' routes in '*MPC-10 Test A*'

3.5.2) MPC-10 Test B

In this test, MPC-10 instance is solved considering a time horizon of 20 hours. The results are displayed in Figure 53, and the teams' routes are shown in Figure 54. In this case, almost all components are reconnected (except $q_3 = \{9\}$), and the teams do not collaborate. As expected, when the time horizon was reduced, all the teams' routes changed and the their duration are within the new time horizon. Thus, the model behaved properly, and E-ARCP passed this test.

```
Processing time = 14.476824599999999 seconds
--------
The problem is Optimal
Objective Function Value =  25.0
--------
a_(2,_5,_1) = 15.0
a_(4,_2,_0) = 10.0
f_(0,_4,_0) = 3.0
f_(2,_10,_0) = 1.0
f_(2,_5,_1) = 4.0
f_(4,_2,_0) = 2.0
f_(5,_10,_1) = 1.0
f_(5,_7,_1) = 3.0
f_(6,_2,_1) = 5.0
f_(7,_5,_1) = 2.0
s_(2,_5,_1) = 3.5300000000000633
s_(4,_2,_0) = 0.7999999999999688
t_(0,_4,_2,_0) = 0.7999999999999972
t_(2,_5,_7,_1) = 19.200000000000063
t_(4,_2,_10,_0) = 11.379999999999969
t_(5,_7,_5,_1) = 19.600000000000062
t_(6,_2,_5,_1) = 0.7815631262524967
t_(7,_5,_10,_1) = 20.0
v_(10,_0) = 1.0
v_(10,_1) = 1.0
v_(2,_0) = 1.0
v_(2,_1) = 1.0
v_(4,_0) = 1.0
v_(5,_1) = 2.0
v_(7,_1) = 1.0
x_(0,_4,_2,_0) = 1.0
x_(2,_5,_7,_1) = 1.0
x_(4,_2,_10,_0) = 1.0
x_(5,_7,_5,_1) = 1.0
x_(6,_2,_5,_1) = 1.0
x_(7,_5,_10,_1) = 1.0
y_(0,_1) = 1.0
y_(1,_0) = 1.0
y_(1,_2) = 1.0
y_(2,_1) = 1.0
z_(2,_5,_1) = 1.0
z_(4,_2,_0) = 1.0
```

Figure 53 – Results: '*MPC-10 Test B*'

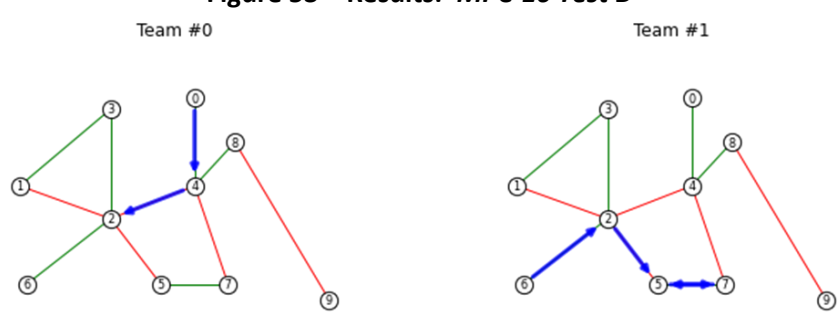


Figure 54 – Road clearing teams' routes in '*MPC-10 Test B*'

3.5.3) MPC-10 Test C

In this test, MPC-10 instance is solved considering a time horizon of 10 hours. The results are displayed in Figure 55, and the teams' routes are shown in Figure 56. In this test, only edge (2,4) is unblocked and teams 0 and 1 collaborate to clear it. The unblocking time of edge (2,4) is being respected ($b_{24} = a_{420} + a_{241} = 10\ hours$, and there are no subtours. Therefore, E-ACRP passed this test.

```
Processing time = 1.888712 seconds
--------
The problem is Optimal
Objective Function Value =  10.0
--------
a_(2,_4,_1) = 4.999999999999943
a_(4,_2,_0) = 5.0000000000001705
f_(0,_4,_0) = 4.0
f_(2,_3,_0) = 2.0
f_(2,_4,_1) = 2.0
f_(3,_10,_0) = 1.0
f_(4,_10,_1) = 1.0
f_(4,_2,_0) = 3.0
f_(6,_2,_1) = 3.0
s_(2,_4,_1) = 0.8000000000002273
s_(4,_2,_0) = 0.8
t_(0,_4,_2,_0) = 0.8
t_(2,_3,_10,_0) = 10.0
t_(2,_4,_10,_1) = 6.38000000000017
t_(4,_2,_3,_0) = 6.38000000000017
t_(6,_2,_4,_1) = 0.8000000000002274
v_(10,_0) = 1.0
v_(10,_1) = 1.0
v_(2,_0) = 1.0
v_(2,_1) = 1.0
v_(3,_0) = 1.0
v_(4,_0) = 1.0
v_(4,_1) = 1.0
w_(4,_2,_0) = 1.0
x_(0,_4,_2,_0) = 1.0
x_(2,_3,_10,_0) = 1.0
x_(2,_4,_10,_1) = 1.0
x_(4,_2,_3,_0) = 1.0
x_(6,_2,_4,_1) = 1.0
y_(0,_1) = 1.0
y_(1,_0) = 1.0
z_(2,_4,_1) = 1.0
z_(4,_2,_0) = 1.0
```

Figure 55 – Results: '*MPC-10 Test C*'

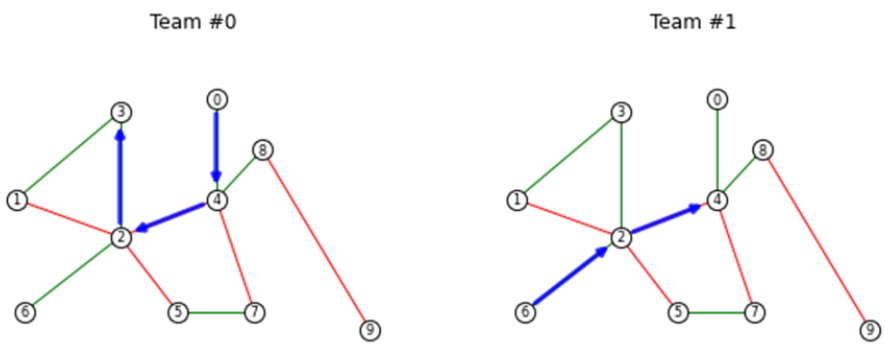


Figure 56 – Road clearing teams' routes in '*MPC-10 Test C*'

3.6) MPC-15

The '*MPC-15*' test consists of running the model on instance MPC-15 without forcing a known solution. Two cases were tested: A) time horizon of 72 hours ('*MPC-15 Test A*'), B) time horizon of 25 hours ('*MPC-15 Test B*'). The results of tests A and B are presented in Subsections 3.6.1 and 3.6.2, respectively. The purpose of this test was to check if E-ARCP behaved properly on different instances when the parameter "t_max" changed. E-ARCP passed those two tests.

3.6.1) MPC-15 Test A

In this test, the MPC-15 instance is solved with a time horizon of 72 hours. The results are displayed in Figure 57, and the teams' routes are shown in Figure 58. In this time horizon, the teams do not need to collaborate to open the edges. The result presents no subtours, the unblocking times (variable "a") are correct, as well as the time variables ("s" and "t"). Therefore, E-ARCP passed this test.

```
Processing time = 2.9529198 seconds
--------
The problem is Optimal
Objective Function Value =  49.0
--------
a_(12,_11,_0) = 22.0
a_(12,_13,_0) = 10.0
a_(2,_3,_0) = 15.0
a_(2,_4,_1) = 20.0
a_(6,_9,_2) = 31.0
a_(9,_10,_2) = 5.0
f_(0,_2,_0) = 7.0
f_(0,_2,_1) = 3.0
f_(10,_15,_2) = 1.0
f_(11,_15,_0) = 1.0
f_(12,_11,_0) = 2.0
f_(12,_13,_0) = 4.0
f_(13,_12,_0) = 3.0
f_(2,_3,_0) = 6.0
f_(2,_4,_1) = 2.0
f_(3,_12,_0) = 5.0
f_(4,_15,_1) = 1.0
f_(5,_6,_2) = 4.0
f_(6,_9,_2) = 3.0
f_(9,_10,_2) = 2.0
s_(12,_11,_0) = 44.21999999999936
s_(12,_13,_0) = 28.21999999999936
s_(13,_12,_0) = 41.21999999999936
s_(2,_3,_0) = 5.829999999999359
s_(2,_4,_1) = 47.00000000000028
s_(6,_9,_2) = 1.9999999999995737
s_(9,_10,_2) = 65.00000000000011
t_(0,_2,_3,_0) = 5.8299999999994725
t_(0,_2,_4,_1) = 5.83
t_(12,_11,_15,_0) = 68.45999999999935
t_(12,_13,_12,_0) = 41.21999999999936
t_(13,_12,_11,_0) = 44.21999999999936
t_(2,_3,_12,_0) = 22.82999999999936
t_(2,_4,_15,_1) = 72.0
t_(3,_12,_13,_0) = 28.21999999999936
t_(5,_6,_9,_2) = 1.9999999999996874
t_(6,_9,_10,_2) = 39.31999999999923
t_(9,_10,_15,_2) = 72.0
v_(10,_2) = 1.0
v_(11,_0) = 1.0
v_(12,_0) = 2.0
v_(13,_0) = 1.0
v_(15,_0) = 1.0
v_(15,_1) = 1.0
v_(15,_2) = 1.0
v_(2,_0) = 1.0
v_(2,_1) = 1.0
v_(3,_0) = 1.0
v_(4,_1) = 1.0
v_(6,_2) = 1.0
v_(9,_2) = 1.0
w_(12,_11,_0) = 1.0
x_(0,_2,_3,_0) = 1.0
x_(0,_2,_4,_1) = 1.0
x_(12,_11,_15,_0) = 1.0
x_(12,_13,_12,_0) = 1.0
x_(13,_12,_11,_0) = 1.0
x_(2,_3,_12,_0) = 1.0
x_(2,_4,_15,_1) = 1.0
x_(3,_12,_13,_0) = 1.0
x_(5,_6,_9,_2) = 1.0
x_(6,_9,_10,_2) = 1.0
x_(9,_10,_15,_2) = 1.0
y_(0,_1) = 1.0
y_(0,_3) = 1.0
y_(1,_0) = 1.0
y_(1,_2) = 1.0
y_(1,_3) = 1.0
y_(2,_1) = 1.0
y_(3,_0) = 1.0
y_(3,_1) = 1.0
y_(3,_4) = 1.0
y_(4,_3) = 1.0
y_(4,_5) = 1.0
y_(5,_4) = 1.0
z_(12,_11,_0) = 1.0
z_(12,_13,_0) = 1.0
z_(2,_3,_0) = 1.0
z_(2,_4,_1) = 1.0
z_(6,_9,_2) = 1.0
z_(9,_10,_2) = 1.0
```

Figure 57 – Results: '*MPC-15 Test A*'

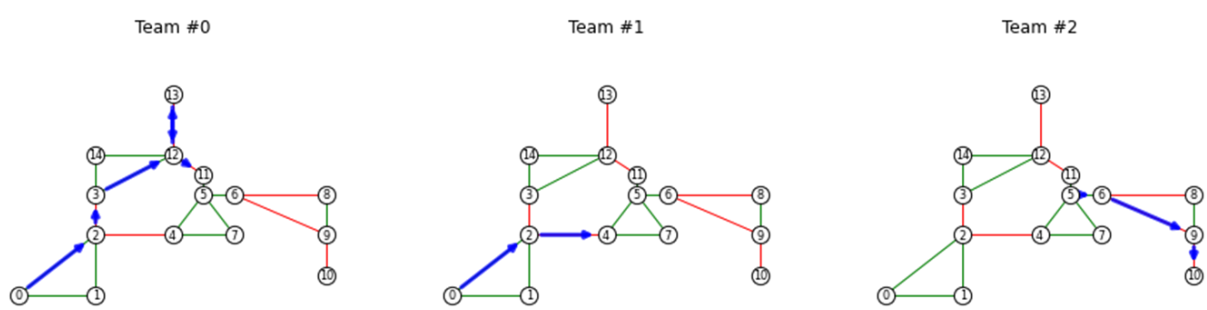


Figure 58 – Road clearing teams' routes in '*MPC-15 Test A*'

3.6.2) MPC-15 Test B

In this test, the instance MPC-15 is solved to optimality with a time horizon of 25 hours. The results are displayed in Figure 59, and the teams' routes are shown in Figure 60. Notice that teams 1 and 2 collaborate to clear edge (2,4). All decision variables' values are correct, and therefore, E-ARCP passed this test.

```
Processing time = 159.357497 seconds
--------
The problem is Optimal
Objective Function Value =  19.0
--------
a_(2,_3,_0) = 15.0
a_(2,_4,_1) = 8.499999999999996
a_(4,_2,_2) = 11.500000000000004
f_(0,_2,_0) = 3.0
f_(0,_2,_1) = 5.0
f_(11,_15,_1) = 1.0
f_(2,_15,_2) = 1.0
f_(2,_3,_0) = 2.0
f_(2,_4,_1) = 4.0
f_(3,_15,_0) = 1.0
f_(4,_2,_2) = 2.0
f_(4,_5,_1) = 3.0
f_(5,_11,_1) = 2.0
f_(5,_4,_2) = 3.0
s_(2,_3,_0) = 8.0
s_(2,_4,_1) = 7.670000000000005
s_(4,_2,_2) = 2.830000000000002
t_(0,_2,_3,_0) = 8.0
t_(0,_2,_4,_1) = 5.830000000000005
t_(2,_3,_15,_0) = 25.0
t_(2,_4,_5,_1) = 21.17
t_(4,_2,_15,_2) = 19.330000000000005
t_(4,_5,_11,_1) = 24.0
t_(5,_11,_15,_1) = 25.0
t_(5,_4,_2,_2) = 2.83
v_(11,_1) = 1.0
v_(15,_0) = 1.0
v_(15,_1) = 1.0
v_(15,_2) = 1.0
v_(2,_0) = 1.0
v_(2,_1) = 1.0
v_(2,_2) = 1.0
v_(3,_0) = 1.0
v_(4,_1) = 1.0
v_(4,_2) = 1.0
v_(5,_1) = 1.0
w_(4,_2,_2) = 1.0
x_(0,_2,_3,_0) = 1.0
x_(0,_2,_4,_1) = 1.0
x_(2,_3,_15,_0) = 1.0
x_(2,_4,_5,_1) = 1.0
x_(4,_2,_15,_2) = 1.0
x_(4,_5,_11,_1) = 1.0
x_(5,_11,_15,_1) = 1.0
x_(5,_4,_2,_2) = 1.0
y_(0,_1) = 1.0
y_(0,_3) = 1.0
y_(1,_0) = 1.0
y_(3,_0) = 1.0
z_(2,_3,_0) = 1.0
z_(2,_4,_1) = 1.0
z_(4,_2,_2) = 1.0
```

Figure 59 – Results: '*MPC-15 Test B*'

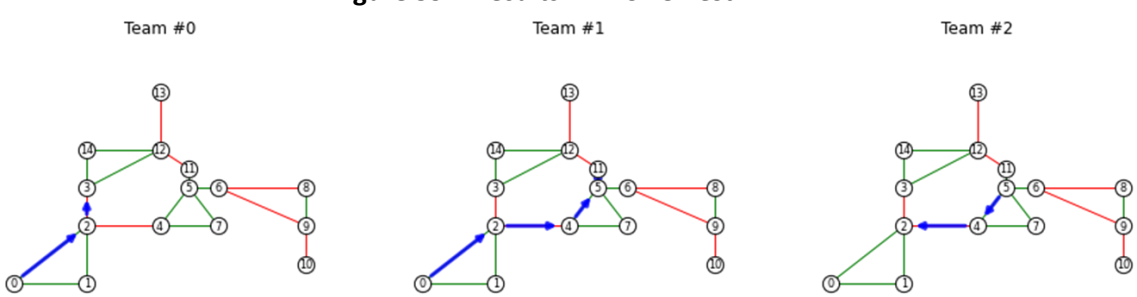


Figure 60 – Road clearing teams' routes in '*MPC-15 Test B*'

3.7) MPC-20

The '*MPC-20*' test consists of running the model on instance MPC-20 without forcing a known solution. The purpose of this test is (i) to check if the model will create routes with subtours, (ii) to check if the teams' routes will change when the time horizon is reduced, and (ii) to check if the decision variables are correct, specially the variables related with the time (i.e. "s", "t", and "a"). Two cases were tested: A) time horizon of 72 hours ('*MPC-20 Test A*'), B) time horizon of 25 hours ('*MPC-20 Test B*'). The results of tests A and B are presented in Subsections 3.7.1 and 3.7.2, and E-ARCP passed both tests.

3.7.1) MPC-20 Test A

In this test, instance MPC-20 is solved considering a time horizon of 72 hours. The results are displayed in Figure 61, and the teams' routes are shown in Figure 62. The teams do not need to collaborate to open the edges with a time horizon of 72 hours, and team 2 does not unblock any edge. The values of all decision variables are calculated properly, and therefore, E-ARCP passed this test.

```
Processing time = 26.1814808 seconds
--------
The problem is Optimal
Objective Function Value =  37.0
--------
a_(1,_8,_1) = 50.0
a_(19,_17,_3) = 40.0
a_(2,_10,_0) = 30.0
f_(0,_1,_1) = 5.0
f_(0,_2,_0) = 5.0
f_(0,_20,_1) = 1.0
f_(1,_0,_1) = 2.0
f_(1,_8,_1) = 4.0
f_(10,_11,_0) = 3.0
f_(10,_20,_0) = 1.0
f_(11,_10,_0) = 2.0
f_(16,_18,_2) = 134.0
f_(17,_20,_3) = 1.0
f_(18,_16,_2) = 135.0
f_(18,_19,_3) = 3.0
f_(18,_20,_2) = 1.0
f_(19,_17,_3) = 2.0
f_(2,_10,_0) = 4.0
f_(8,_1,_1) = 3.0
s_(1,_8,_1) = 15.400000000000091
s_(19,_17,_3) = 1.5
s_(2,_10,_0) = 2.0599999999998033
s_(8,_1,_1) = 67.12000000000009
t_(0,_1,_8,_1) = 3.16
t_(0,_2,_10,_0) = 2.059999999999917
t_(1,_0,_20,_1) = 72.0
t_(1,_8,_1,_1) = 67.12000000000009
t_(10,_11,_10,_0) = 34.46999999999932
t_(11,_10,_20,_0) = 72.0
t_(16,_18,_20,_2) = 72.0
t_(18,_16,_18,_2) = 2.0
t_(18,_19,_17,_3) = 1.5
t_(19,_17,_20,_3) = 42.5
t_(2,_10,_11,_0) = 33.059999999999526
t_(8,_1,_0,_1) = 68.84000000000009
v_(1,_1) = 2.0
v_(10,_0) = 2.0
v_(11,_0) = 1.0
v_(16,_2) = 1.0
v_(17,_3) = 1.0
v_(19,_3) = 1.0
v_(2,_0) = 1.0
v_(20,_0) = 1.0
v_(20,_1) = 1.0
v_(20,_2) = 1.0
v_(20,_3) = 1.0
v_(8,_1) = 1.0
x_(0,_1,_8,_1) = 1.0
x_(0,_2,_10,_0) = 1.0
x_(1,_0,_20,_1) = 1.0
x_(1,_8,_1,_1) = 1.0
x_(10,_11,_10,_0) = 1.0
x_(11,_10,_20,_0) = 1.0
x_(16,_18,_20,_2) = 1.0
x_(18,_16,_18,_2) = 1.0
x_(18,_19,_17,_3) = 1.0
x_(19,_17,_20,_3) = 1.0
x_(2,_10,_11,_0) = 1.0
x_(8,_1,_0,_1) = 1.0
y_(0,_1) = 1.0
y_(0,_3) = 1.0
y_(0,_4) = 1.0
y_(1,_0) = 1.0
y_(3,_0) = 1.0
y_(4,_0) = 1.0
z_(1,_8,_1) = 1.0
z_(19,_17,_3) = 1.0
z_(2,_10,_0) = 1.0
```

Figure 61 – Results: '*MPC-20 Test A*'

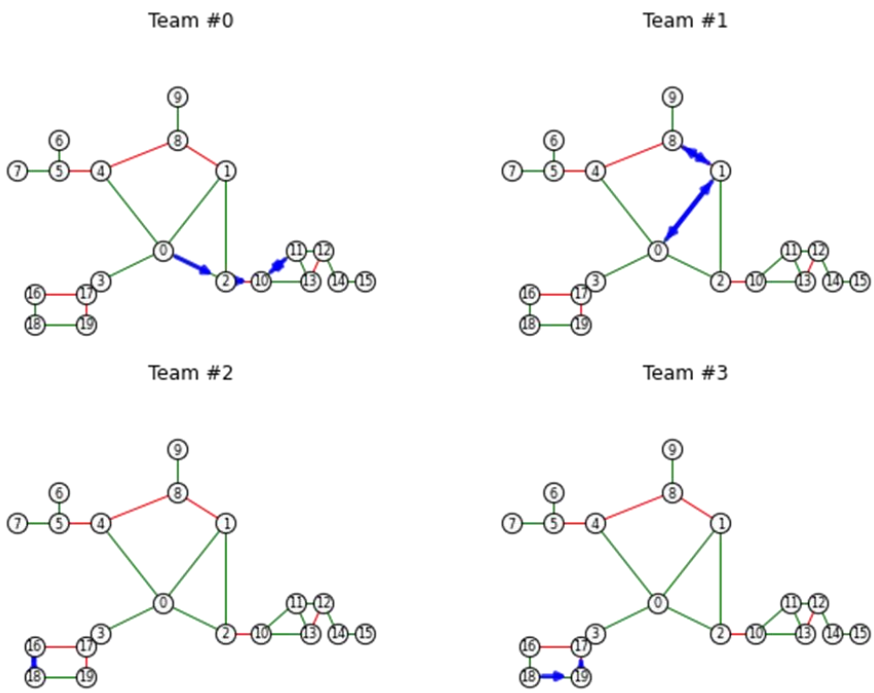


Figure 62 – Road clearing teams' routes in '*MPC-20 Test A*'

3.7.1) MPC-20 Test B

In this test, the MPC-20 instance is solved with a time horizon of 25 hours. The results are displayed in Figure 63, and the teams' routes are shown in Figure 64. Only one edge is open in this time horizon, and teams 0 and 2 must collaborate to open it. The total unblocking time of this edge is 40 hours, which is equivalent to the sum of a_{17190} and a_{19171}. The other time decision variables (i.e. "s" and "t") are also correct, and the teams are synchronized. Thus, E-ARCP passed this test.

```
Processing time = 99.2093243 seconds
--------
The problem is Optimal
Objective Function Value =  11.0
--------
a_(17,_19,_0) = 19.43500000000023
a_(19,_17,_2) = 20.56500000000023
f_(0,_20,_1) = 1.0
f_(0,_3,_0) = 4.0
f_(0,_4,_1) = 3.0
f_(17,_19,_0) = 2.0
f_(17,_20,_2) = 1.0
f_(18,_19,_2) = 3.0
f_(18,_19,_3) = 4.0
f_(18,_20,_3) = 1.0
f_(19,_17,_2) = 2.0
f_(19,_18,_3) = 3.0
f_(19,_20,_0) = 1.0
f_(3,_17,_0) = 3.0
f_(4,_0,_1) = 2.0
s_(17,_19,_0) = 2.63
s_(19,_17,_2) = 1.5
t_(0,_3,_17,_0) = 2.06
t_(0,_4,_0,_1) = 6.32
t_(17,_19,_20,_0) = 23.065000000000232
t_(18,_19,_17,_2) = 1.5
t_(18,_19,_18,_3) = 1.5
t_(19,_17,_20,_2) = 23.06500000000023
t_(19,_18,_20,_3) = 25.0
t_(3,_17,_19,_0) = 2.63
t_(4,_0,_20,_1) = 25.0
v_(17,_0) = 1.0
v_(17,_2) = 1.0
v_(19,_0) = 1.0
v_(19,_2) = 1.0
v_(19,_3) = 1.0
v_(20,_0) = 1.0
v_(20,_1) = 1.0
v_(20,_2) = 1.0
v_(20,_3) = 1.0
v_(3,_0) = 1.0
v_(4,_1) = 1.0
w_(19,_17,_2) = 1.0
x_(0,_3,_17,_0) = 1.0
x_(0,_4,_0,_1) = 1.0
x_(17,_19,_20,_0) = 1.0
x_(18,_19,_17,_2) = 1.0
x_(18,_19,_18,_3) = 1.0
x_(19,_17,_20,_2) = 1.0
x_(19,_18,_20,_3) = 1.0
x_(3,_17,_19,_0) = 1.0
x_(4,_0,_20,_1) = 1.0
y_(0,_3) = 1.0
y_(3,_0) = 1.0
z_(17,_19,_0) = 1.0
z_(19,_17,_2) = 1.0
```

Figure 63 – Results: '*MPC-20 Test B*'

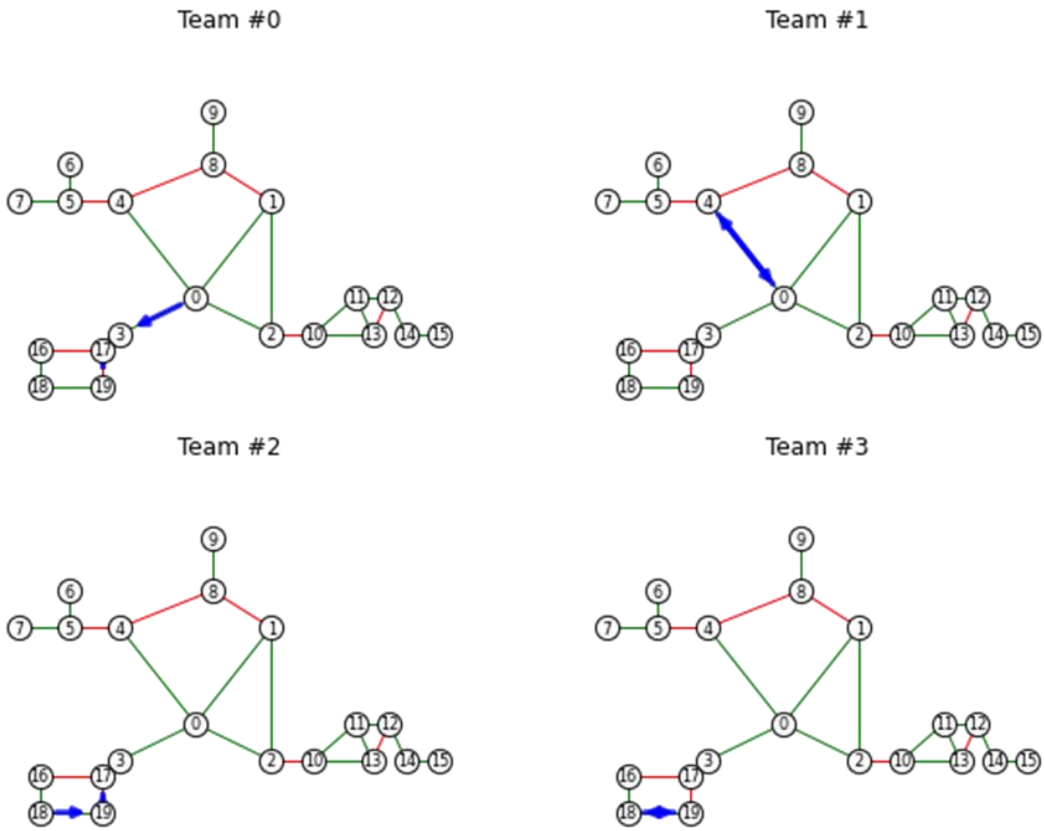


Figure 64 – Road clearing teams' routes in '*MPC-20 Test B*'

5) Converting MPC-ARCP to other Arc Routing for Connectivity Problems

The MPC-ARCP is a generalization of two other Arc Routing for Connectivity Problems: the 'Prize Collecting Arc Routing for Connectivity Problem' (PC-ARCP) and the 'Multi-Vehicle Prize Collecting Arc Routing for Connectivity Problem' (KPC-ARCP). Subsection 5.1 explains how to solve a PC-ARCP with E-ARCP, whereas Subsection 5.2 explains how to convert E-ARCP into KPC-ARCP. Examples are available in the folder "MPC-ARCP to PC and KPC-ARCP".

5.1) PC-ARCP

The 'Prize Collecting Arc Routing for Connectivity Problem' (PC-ARCP) is proposed by Kasaei and Salman (2016). The three differences between MPC-ARCP and PC-ARCP are: (i) in PC-ARCP there is only one team starting from a single depot, (ii) there are no relative prizes, (iii) and there is no collaboration between the teams to clear the roads.

The user must update the 'Instance Description' file (e.g. "MPC-10.txt"). First, the list "DEPOT_IDS" must have only one depot, as exemplified in Figure 65.

```
NAME: PC-ARCP-10
TYPE: PC-ARCP
COMMENT: Generated by Almeida, L., Ghasemi, A., Goerlandt, F.
DIMENSION: 10
EDGE_WEIGHT_TYPE: EUC_2D
DEPOT_IDS: [0]
SPEED: 50
AVERAGE_DEGREE: 2.5
COMPONENTS: 4
```

Figure 65 – Heading of PC-ARCP-10 instance

In PC-ARCP, the components must be connected to the road clearing depot (i.e. the depot's component). In the example of Figure 66, the depot is located in component 0, so only the prizes related to the depot's component have an assigned value (e.g. 0-1, 0-2, 0-3, 1-0). Another characteristic of PC-ARCP is that there is only one team. In the Section "TEAMS PER DEPOT (DEPOT_ID, TEAMS)", the depot must have only one team.

```
RELATIVE PRIZES
0       1       15
0       2       15
0       3       15
1       0       15
1       2       0
1       3       0
2       0       15
2       1       0
2       3       0
3       0       15
3       1       0
3       2       0
TEAMS PER DEPOT (DEPOT_ID, TEAMS)
0       1
```

Figure 66 – Relative Prizes and Teams per depot in the PC-ARCP-10 instance

Finally, the constraint in Figure 67, must be replaced with the constraint in Figure 68. Therefore, the user should delete (or comment) the constraint in Figure 67, and add the constraint in Figure 68.

$$\sum_{k \in K} z_{ijk} + \sum_{k \in K} z_{jik} \leq 2, \forall (i,j) \in B$$

```
for (i,j) in B:
    MPC_ARCP += lpSum([z[i,j,k] for k in K]) + lpSum([z[j,i,k] for k in K]) <= 2
```

Figure 67 – Constraint that should be replaced in MPC-ARCP to solve PC-ARCP

$$\sum_{k \in K} z_{ijk} + \sum_{k \in K} z_{jik} \leq 1, \forall (i,j) \in B$$

```
for (i,j) in B:
    MPC_ARCP += lpSum([z[i,j,k] for k in K]) + lpSum([z[j,i,k] for k in K]) <= 1
```

Figure 68 – Constraint required to solve PC-ARCP

5.2) KPC-ARCP

The 'Multi-vehicle Prize Collecting Arc Routing for Connectivity Problem' (KPC-ARCP) is proposed by Akbari and Salman (2017). The three differences between MPC-ARCP and KPC-ARCP are: (i) in KPC-ARCP there is only one depot, (ii) there are no relative prizes, (iii) and the road clearing teams do not collaborate.

The user must update the Instance Description file (e.g. “MPC-10.txt”) as explained in Subsection 5.1. Additionally, the number of teams per depot in the ‘Instance Description’ file must be greater than 1.

4) Legal Code License

GNU General Public License (GPL) (https://github.com/souzaalmeidaluana/E-ARCP/blob/main/LICENSE)

References

Akbari, V., & Salman, F. S. (2017). Multi-vehicle prize collecting arc routing for connectivity problem. *Computers & Operations Research*, *82*, 52–68. https://doi.org/10.1016/j.cor.2017.01.007

Kasaei, M., & Salman, F. S. (2016). Arc routing problems to restore connectivity of a road network. *Transportation Research. Part E, Logistics and Transportation Review*, *95*, 177–206. https://doi.org/10.1016/j.tre.2016.09.012

Souza Almeida, L., Ghasemi, A., & Goerlandt, F. (2022). Results of the numerical experiments of the proposed MIP to the Collaborative Multi-vehicle Prize Collecting Arc Routing for Connectivity Problem. *Mendeley Data*, *V1*.

Souza Almeida, L., Ghasemi, A., & Goerlandt, F. (2023). The collaborative multi-vehicle prize collecting arc routing for connectivity problem. *Submitted*.

Souza Almeida, L., Kodali, R., & Goerlandt, F. (2022). Benchmark instances for road network repair and restoration problems in the context of disaster response operations. *Data in Brief*, *45*, 108674. https://doi.org/10.1016/j.dib.2022.108674