# Coordinated and Non-Coordinated Routing Problems with Drones

Carlos Valverde Martín

Congreso Fuengirola 2021

Joint work with Justo Puerto and Lavinia Amorosi

# Contents

1. Non-Coordinated Models
    1. The Crossing Postman Problem with Neighborhoods (XPPN)
2. Coordinated Models
    1. The mothership and drone routing problem with Graphs (MDRPG)
    2. The all-terrain mothership and multiple drone routing problem with Graphs (AMMDRPG)

# Contents

- Problem Description
- Formulations
- Example
- Heuristic
- Strengthening results for the formulation
- Benders decomposition
- Computational Experiments

# Problem Description

- Our problem (XPPN) combines the Crossing Postman Problem (XPP) and Travelling Salesman Problem with Neighborhoods (TSPN).
- The XPP was worked by Garfinkel. It is a relaxation of the Rural Postman Problem because it is permitted to leave the edges of the network and cross from one to another at points other than the original vertices.
- The TSPN is a extension of the TSP because points are replaced by neighborhoods (that we assume to be convex).
- It is also related to the Minimum Spanning Tree with Neighborhoods (MSTN).
- This work can be applied to Routing Problems with Umanned Aerial Vehicles because their movement are not limited.

# Problem Description

We have two nature of sets where the points can be located:

- Second Order Cone representable sets:

$$x_v^i \in \mathcal{U}_v \iff \left\{ \begin{array}{lcll} \|A_v^{j_\ell} x_v^i + b_v^{j_\ell}\| & \leq & (c_v^{j_\ell})^T x_v^i + d_v^{j_\ell} + M_v^{j_\ell}(1-\chi_v^{i\ell}), & \ell = 1, \ldots, m_v, j_\ell = 1, \ldots, n_{\ell v}, \\ \sum_{\ell=1}^{m_v} \chi_v^{i\ell} & = & 1. & \end{array} \right. \tag{$\mathcal{U}$-C}$$

- Polygonal Chains where we have to cross a percentage $\alpha$ of its total length:

$$x_v^i \in \mathcal{P}_v \iff \left\{ \begin{array}{lcll} \lambda_v^i - j & \geq & \gamma_v^{ij} - (n_{S_v}+1)(1-\mu_v^{ij}), & j = 2, \ldots, n_{S_v}+1, \\ \lambda_v^i - j & \leq & \gamma_v^{ij} + (n_{S_v}+1)(1-\mu_v^{ij}), & j = 2, \ldots, n_{S_v}+1, \\ \gamma_v^{i1} & \leq & \mu_v^{i1}, & \\ \gamma_v^{ij} & \leq & \mu_v^{ij-1} + \mu_v^{ij}, & j = 2, \ldots, n_{S_v} \\ \gamma_v^{in_{S_v}} & \leq & \mu_v^{in_{S_v}}, & \\ \sum_{j=1}^{n_{S_v}} \mu_v^{ij} & = & 1, & \\ \sum_{j=1}^{n_{S_v}+1} \gamma_v^{ij} & = & 1, & \\ x_v^i & = & \sum_{j=1}^{n_{S_v}+1} \gamma_v^{ij} A_v^j. & \end{array} \right. \tag{$\mathcal{P}$-C}$$

$$|\lambda_v^1 - \lambda_v^2| \geq \alpha_v n_{S_v} \iff \left\{ \begin{array}{lcl} \lambda_v^1 - \lambda_v^2 & = & \lambda_v^{\max} - \lambda_v^{\min}, \\ \lambda_v^{\max} + \lambda_v^{\min} & \geq & \alpha_v n_{S_v}, \\ \lambda_v^{\max} & \leq & n_{S_v}(1-u_v), \\ \lambda_v^{\min} & \leq & n_{S_v} u_v. \end{array} \right. \tag{$\alpha$-C}$$

# Time dependent formulation for the XPPN

The idea is to make variables dependent on the index of the stage when an element is visited in the sequence of visited elements.

$$
\begin{aligned}
\min \quad & \sum_{t=1}^{|V|} \sum_{v \neq w} d_{vw}^t z_{vw}^t + \sum_{t=1}^{|V|} \sum_{v \in V} f_v^t d_v^t & & \text{(1a)}\\
\text{s.t.} \quad & d_{vw}^t \geq \beta_{vw}^t \|x_v^2 - x_w^1\|, & \forall v \neq w, & \quad \text{(1b)}\\
& d_v^t \geq \beta_v^t \|x_v^1 - x_v^2\|, & \forall v \in V, & \quad \text{(1c)}\\
& \sum_{v \in V} y_v^t = 1, & \forall t, & \quad \text{(1d)}\\
& \sum_{t=1}^{|V|} y_v^t = 1, & \forall v \in V, & \quad \text{(1e)}\\
& y_v^t + y_w^{t+1} - 1 \leq z_{vw}^t, & \forall e = (v, w) \in E_{\text{out}},\ t = 1, \ldots, |\mathcal{C}| - 1, & \quad \text{(1f)}\\
& (\mathcal{U}\text{-C}), (\mathcal{P}\text{-C}), (\alpha\text{-C}) & & \quad \text{(1g)}
\end{aligned}
$$

# Non-time dependent formulations for the XPPN

$$
\begin{array}{llll}
\min & P = \displaystyle\sum_{e\in E_{\text{out}}} p_e + \sum_{v\in V} f_v d_v & & \text{(SEC-XPPN)} \\
\text{s.t.} & p_e \geq d_e - M_e(1-z_e) & \forall e \in E_{\text{out}}, & \text{(LIN-Mc)} \\
 & p_e \geq m_e z_e & \forall e \in E_{\text{out}}, & \text{(VI-1)} \\
 & d_v \leq M_v & \forall v \in V, & \text{(VI-2)} \\
 & \displaystyle\sum_{w\in V\setminus\{v\}} z_{vw} = 1, & \forall v \in V, & (\mathrm{C}_1) \\
 & \displaystyle\sum_{w\in V\setminus\{v\}} z_{wv} = 1, & \forall v \in V, & (\mathrm{C}_2) \\
 & \displaystyle\sum_{e=(v,w):v,w\in S} z_e \leq |S| - 1, & \forall S \subsetneq V, & \text{(SEC)} \\
 & d_e \geq \|x_v^1 - x_w^2\|_2 & \forall e = (v,w) \in E_{\text{out}}, & (\mathrm{D}_1) \\
 & d_v \geq \|x_v^1 - x_v^2\|_2 & \forall v \in V, & (\mathrm{D}_2) \\
 & (\mathcal{U}\text{-C}), (\mathcal{P}\text{-C}), (\alpha\text{-C}) & &
\end{array}
$$

# Non-time dependent formulations for the XPPN

$$\min \quad P = \sum_{e\in E_{\text{out}}} p_e + \sum_{v\in V} f_v d_v \tag{sSEC-XPPN}$$

$$\text{s.t.} \quad p_e \geq d_e - M_e(1 - z_e) \quad \forall e \in E_{\text{out}}, \tag{LIN-Mc}$$

$$p_e \geq m_e z_e \quad \forall e \in E_{\text{out}}, \tag{VI-1}$$

$$d_v \leq M_v \quad \forall v \in V, \tag{VI-2}$$

$$\sum_{w\in V\setminus\{v\}} z_{vw} = 2, \quad \forall v \in V,$$

$$\sum_{e=(v,w):v,w\in S} z_e \leq |S| - 1, \quad \forall S \subsetneq V, \tag{SEC}$$

$$d_e \geq \|x_v^1 - x_w^2\|_2 \quad \forall e = (u,v) \in E_{\text{out}}, \tag{D$_1$}$$

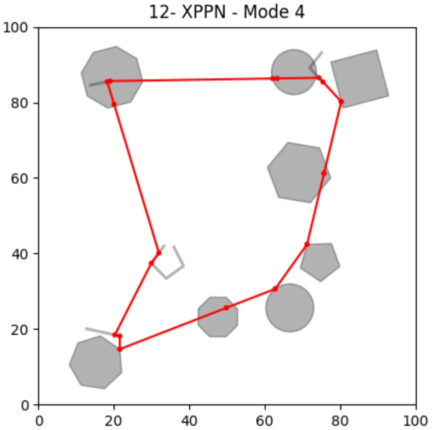
$$d_v \geq \|x_v^1 - x_v^2\|_2 \quad \forall v \in V, \tag{D$_2$}$$

$$(\mathcal{U}\text{-C}), (\mathcal{P}\text{-C}), (\alpha\text{-C})$$

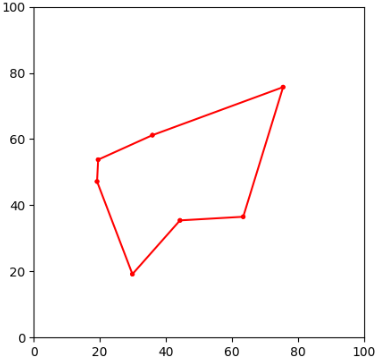
# Non-time dependent formulations for the XPPN

$$
\begin{aligned}
\min \quad & P = \sum_{e\in E_{\text{out}}} p_e + \sum_{v\in V} f_v d_v && && \text{(MTZ-XPPN)}\\
\text{s.t.} \quad & p_e \geq d_e - M_e(1 - z_e) && \forall e \in E_{\text{out}}, && \text{(LIN-Mc)}\\
& p_e \geq m_e z_e && \forall e \in E_{\text{out}}, && \text{(VI-1)}\\
& d_v \leq M_v && \forall v \in V, && \text{(VI-2)}\\
& \sum_{w\in V\setminus\{v\}} z_{vw} = 1, && \forall v \in V, && (\text{C}_1)\\
& \sum_{w\in V\setminus\{v\}} z_{wv} = 1, && \forall v \in V, && (\text{C}_2)\\
& |V| z_{vw} + s_v - s_w \leq |V| - 1, && \forall e = (v, w) \in E_{\text{out}}, && (\text{MTZ}_1)\\
& s_1 = 1, && && (\text{MTZ}_2)\\
& 2 \leq s_v \leq |V|, && \forall v \in V, && (\text{MTZ}_3)\\
& s_v - s_w + |V| z_{wv} \leq |V| - 1, && \forall e = (v, w) \in E_{\text{out}}, w > 1, && (\text{MTZ}_4)\\
& s_v - s_w + (|V| - 2) z_{wv} \leq |V| - 1, && \forall e = (v, w) \in E_{\text{out}}, v > 1, && (\text{MTZ}_5)\\
& d_e \geq \|x_v^1 - x_w^2\|_2 && \forall e \in E_{\text{out}}, && (\text{D}_1)\\
& d_v \geq \|x_v^1 - x_v^2\|_2 && \forall v \in V, && (\text{D}_2)\\
& (\mathcal{U}\text{-C}), (\mathcal{P}\text{-C}), (\alpha\text{-C}) && &&
\end{aligned}
$$

# Example



# Example 2



# Heuristic Algorithm

**Algorithm 1:** Heuristic for solving XPPN.

Let $\{\mathcal{N}_v : v \in V\}$ be the neighborhood set. Set *attempts* $= 25$, *neigh_size* $= 5$, *iter* $= 10$.

1. Solve

$$\begin{aligned} \min \quad & \sum_{v \in V} \|x_v^i - \mathrm{Med}\| \qquad \text{(Weber)} \\ \text{s.t.} \quad & (\mathcal{U}\text{-C}), (\mathcal{P}\text{-C}), (\alpha\text{-C}) \end{aligned}$$

   for $\{\mathcal{N}_v : v \in V\}$ to get $\bar{x}$.

2. Consider the VNS approach with parameters *attempts*, *neigh_size* and *iter* and points $\bar{x}$ to obtain the order of visit to the neighborhoods $\bar{z}$.

# Heuristic Algorithm: Clustering Phase

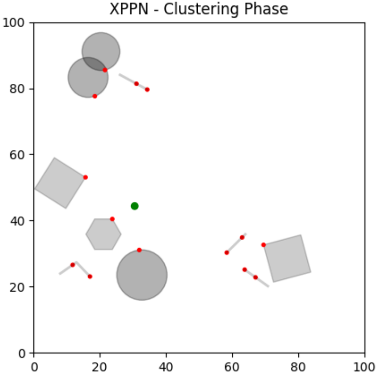



Figure: Illustration of the first phase of the heuristic algorithm

# Heuristic Algorithm: VNS Phase

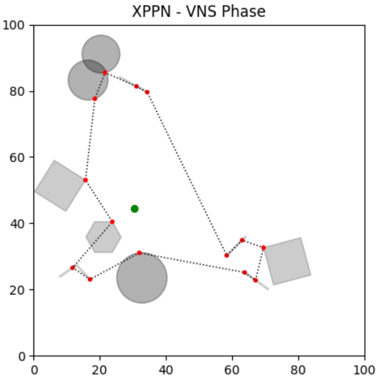



Figure: Application of the VNS phase to the example of Figure 1

# Strengthening results for the formulation: Preprocessing

**Remark 1**

If the problem verifies that $f_v = 0$ for all $v \in V_{\mathcal{C}}$, then the entry and exit points $x_v^1$ and $x_v^2$ selected in each neighborhood are the same that the ones obtained by minimizing the distance between the neighborhoods.

**Remark 2**

If $f_v \geq 1$ for some $v \in V_{\mathcal{C}}$, then, there exists an optimal solution verifying $x_v^1 = x_v^2$.

**Proposition 1**

Given two neighborhoods $A$ and $B$, if $B \supset A$, then $B$ can be removed in the problem.

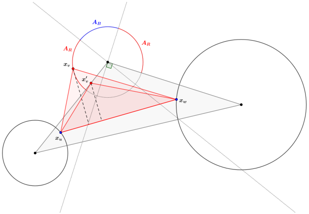
**Proposition 2**

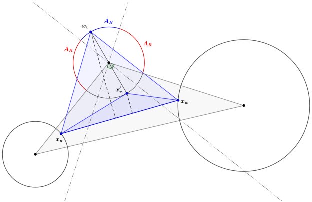
There exists always an optimal solution of the XPPN whose selected points are placed in the boundary of the neighborhoods.

# Strengthening results for the formulation: Preprocessing

## Corollary 1

Any point selected in an optimal solution of the XPPN when all the neighborhoods are circles is placed in some arc of one of the circumferences inside of the convex hull generated by the center of the circles.



# Strengthening results for the formulation: Valid Inequalities

Adjusting $M_e$ (resp. $m_e$) that denotes an upper (resp. lower) bound of the distance between the sets joined by an edge $e \in E_{\text{out}}$.

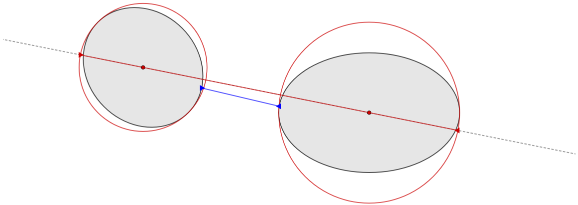


Figure: Upper and lower bound when both sets are ellipsoids

# Strengthening results for the formulation: Valid Inequalities

Adjusting $M_e$ (resp. $m_e$) that denotes an upper (resp. lower) bound of the distance between the sets joined by an edge $e \in E_{\text{out}}$.

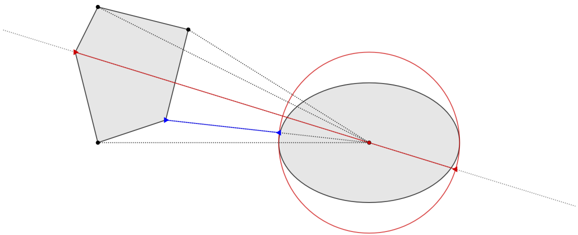


Figure: Upper and lower bound when a set is a polygon and the other is an ellipsoid

# Strengthening results for the formulation: Valid Inequalities

Adjusting $M_e$ (resp. $m_e$) that denotes an upper (resp. lower) bound of the distance between the sets joined by an edge $e \in E_{\text{out}}$.

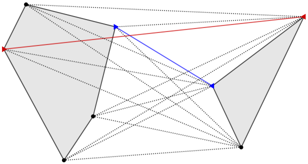


Figure: Upper and lower bound when both sets are ellipsoids

# Strengthening results for the formulation: Valid Inequalities

Adjusting $m_v$ that denotes the maximum distance between two points within a given neighborhood.

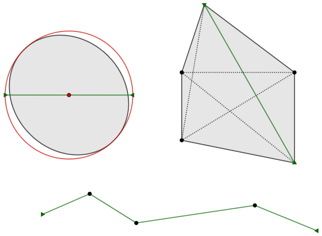


Figure: Upper bound on the maximal distance within a set

# Benders decomposition

**Algorithm 2:** Decomposition Algorithm for solving XPPN.

**Initialization:** Let $z^0 \in \mathcal{T}_G$ be an initial solution and $\varepsilon$ a given threshold value.
Set $LB = 0$, $UB = +\infty$, $\bar{z} = z^0$.

**while** $|UB - LB| > \varepsilon$ **do**

1. Solve

$$\begin{aligned} \min \quad & d(\bar{z}) = \sum_{e \in E_{\text{out}}} d_e \bar{z}_e + \sum_{v \in V} f_v d_v \\ \text{s.t.} \quad & d_e \in \mathcal{D}_e,\ d_v \in \mathcal{D}_v. \end{aligned} \tag{Pd$\bar{z}$}$$

   for $\bar{z}$ to get $d(\bar{z})$.

2. Add the cut $P \geq d(\bar{z}) + \sum_{e:\bar{z}_e=1} M_e(z_e - 1) + \sum_{e:\bar{z}_e=0} m_e z_e$ to the current master problem.
3. Obtain the optimal value $\bar{P}$ to the current master problem, and its associated solution $\bar{z}$.
4. Update $LB = \max\{LB, \bar{P}\}$ and $UB = \min\{UB, \sum_{e \in E} d_e(\bar{z})\bar{z}_e + \sum_{v \in V} f_v d_v\}$

**end**

# Computational Experiments: Data Generation

Five instances with a number $|V| \in \{5, 10, 15, 20\}$ of neighborhoods. We have considered three different types of neighborhoods to be visited:

- Circles of radii $r$.
- Regular polygons of radii $r$ with a random number of vertices in the interval $[3, 10]$.
- Polygonal chains parameterized by its breakpoints that are at a distance of in $r$ from one another and some random percentage $\alpha \in [0, 1]$ of their length to be visited.

The centers or breakpoints of these elements have been generated uniformly in the square [0, 100]. On the one hand, we have studied four different scenarios to generate the radii to define the elements:

- **Small size Neighborhoods** $(r = 1)$: Radii randomly generated in $[0, 5]$.
- **Small-Medium Neighborhoods** $(r = 2)$: Radii randomly generated in $[5, 10]$.
- **Medium-Large size Neighborhoods** $(r = 3)$: Radii randomly generated in $[10, 15]$.
- **Large size Neighborhoods** $(r = 4)$: Radii randomly generated in $[15, 20]$.

We have also considered four modes depending on the nature of the neighborhoods:

- Mode 1: All neighborhoods are circles.
- Mode 2: All neighborhoods are regular polygons.
- Mode 3: All neighborhoods are polygonal chains.
- Mode 4: Mixture of the three previously considered neighborhoods.

All the formulations were coded in Python 3.7, and solved using Gurobi 9.0 in a Intel(R) Xeon(R) E-2146G CPU @ 3.50 GHz and 64GB of RAM. A time limit of 2 hours was set in all the experiments.

# Comparing time and non-time dependent formulations

- Easiest configuration corresponding to neighborhoods given by circles of small radius (Mode 1 and $r = 1$).
- Size $n = 10$ neighborhoods: instances solved to optimality on average in 1800 sec.
- Size $n = 12$ neighborhoods: instances not solved to optimality within 7200 sec.
    - 20 % does not find a feasible solution
    - The average gap of the rest was above 80 %.

Conclusion: We have restricted ourselves to the comparisons of the non-time dependent formulations presented

# Assessing the difficulty of the problems depending on the $\alpha$ parameter

- Batch of 5 instances involving only polygonal chains that are fixed.
- Varying only the $\alpha$ parameter in $\{0, 0.1, 0.2, \ldots, 0.9, 1\}$.

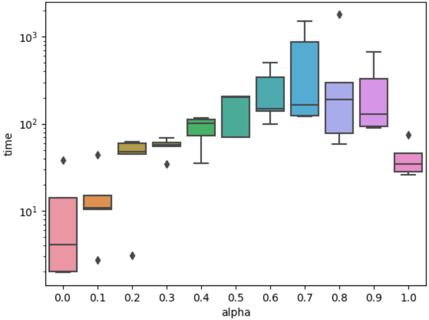



Figure: Comparison of the times of the MTZ formulation varying the $\alpha$ parameter

# Initializing the solver with a heuristic solution

- Comparing the MTZ formulation with and without initial solution provided by our proposed heuristic.
- Sizes 5, 10 and 15.
- Sizes 5 and 10 were solved to optimality.
- For size 15:

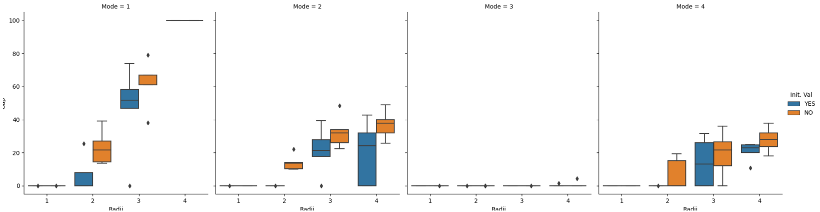



Figure: Comparison of the final gap between MTZ formulation with and without initial solution after 7200 seconds for instances with 15 neighborhoods

# Comparing Benders cuts with the MTZ formulation

- Comparing the decomposition algorithm described before with the MTZ formulation without initialization for size 10.

| Size | Radii | Mode | Final Gap (Benders) | Time (Benders) | #Cuts | Final Gap (MTZ) | Time (MTZ) |
|---|---|---|---|---|---|---|---|
| 10 | 1 | 1 | 0.0 | 15.72 | 19.0 | 0.0 | 1.93 |
| 10 | 1 | 2 | 0.0 | 23.64 | 55.4 | 0.0 | 0.75 |
| 10 | 1 | 3 | 0.0 | 13.22 | 25.8 | 0.0 | 0.72 |
| 10 | 1 | 4 | 0.0 | 33.96 | 29.8 | 0.0 | 1.52 |
| 10 | 2 | 1 | 76.1 | 6430.08 | 1209.0 | 0.0 | 38.83 |
| 10 | 2 | 2 | 56.14 | 4777.58 | 1009.6 | 0.0 | 14.14 |
| 10 | 2 | 3 | 0.0 | 1766.06 | 380.6 | 0.0 | 2.23 |
| 10 | 2 | 4 | 10.57 | 5993.57 | 804.6 | 0.0 | 2.52 |
| 10 | 3 | 1 | 96.21 | 7208.56 | 1481.2 | 0.0 | 487.94 |
| 10 | 3 | 2 | 92.16 | 7203.99 | 1352.2 | 0.0 | 35.81 |
| 10 | 3 | 3 | 9.29 | 5832.51 | 520.4 | 0.0 | 13.28 |
| 10 | 3 | 4 | 84.41 | 7214.86 | 921.8 | 0.0 | 133.81 |
| 10 | 4 | 1 | 98.79 | 7205.35 | 2283.0 | 19.28 | 3513.38 |
| 10 | 4 | 2 | 95.53 | 7207.51 | 1343.4 | 0.0 | 238.98 |
| 10 | 4 | 3 | 19.55 | 7220.14 | 499.0 | 0.0 | 20.25 |
| 10 | 4 | 4 | 82.69 | 7211.46 | 789.8 | 0.0 | 1142.17 |

Table: Computational comparison between MTZ formulation and Benders algorithm for problems with up to 10 neighborhoods

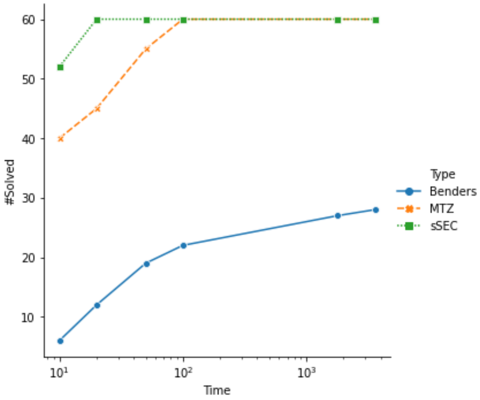
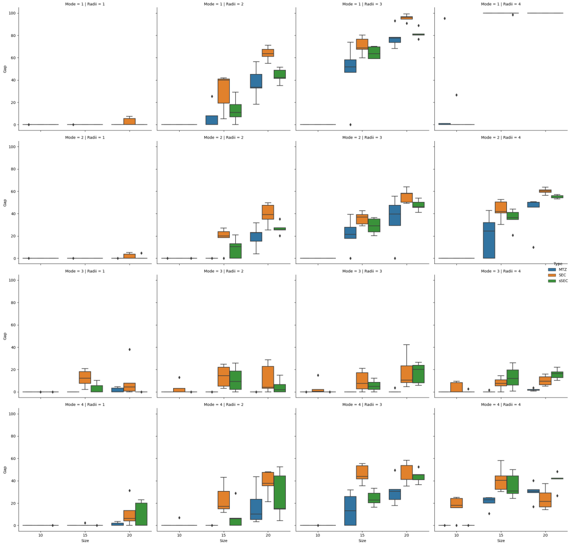



Figure: Performance profile: Time vs #Solved

# Comparing MTZ, SEC and sSEC with initialization



# Comparing MTZ, SEC and sSEC with initialization

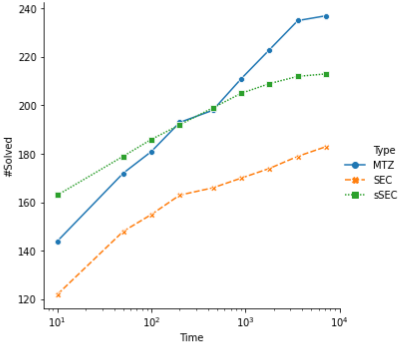


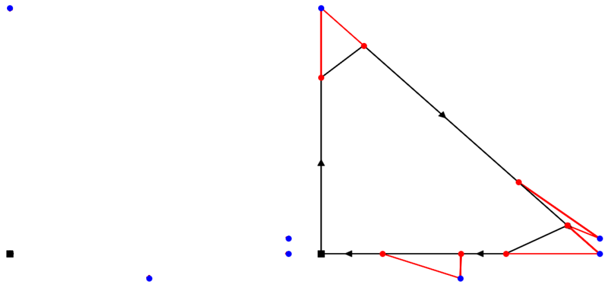

Figure: Performance profile: Time vs #Solved

# Contents

- Problem Description
- Formulations
  - The AMDRPG.
  - The PMDRPG.
  - The NMDRPG.
- Examples
- Matheuristic
- Computational Experiments

# Problem Description: Starting Point

In 2018, Stefan Poikonen and Bruce Golden defined The Mothership and Drone Routing Problem (MDRP):



# Problem Description

- There is one mothership and one drone in coordination to visit a set of target graphs, whose locations are given.
- For each graph $g \in \mathcal{G}$ the drone performs the following task:
  1. It is launched from the current mothership location (to be determined).
  2. It flies to the graph $g$ that has to be visited.
  3. It traverses the required edges of graph $g$.
  4. It returns to the current position of the mothership (to be determined).

We assume wlog that the mothership and the drone do not need to arrive at each rendezvous location at the same time: the fastest arriving vehicle may wait for the other at the rendezvous point.

# Problem Description

It is required to determine:

- The tour of the mothership starting at *orig*, deciding the different launching and rendezvous points, and returning to *dest*.
- The order of visits of the target graphs followed by the drone, determining the corresponding launching and rendezvous points of the drone on each visited graph.
- The tour followed by the drone on each target graph $g \in \mathcal{G}$.

# Problem Description

Depending on the assumptions made on the movements of the mothership vehicle, this problem gives rise to two different versions:

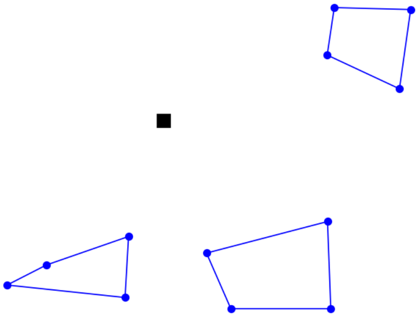
- The mothership vehicle can move freely on the continuous space (all terrain ground vehicle, boat on the water or aircraft vehicle), called the All terrain Mothership-Drone Routing Problem with Graphs (AMDRPG).
- The mothership can move on a connected piecewise linear polygonal chain, called the Network Mothership-Drone Routing Problem on a Polygonal with Graphs (PMDRPG).
- The mothership can move on a road network (that is, it is a normal truck or van), called the Network Mothership-Drone Routing Problem with Graphs (NMDRPG).

# Parameters of the AMDRPG

The known parameters of the problem are:

- $orig$: coordinates of the point defining the origin of the mothership path (or tour).
- $dest$: coordinates of the point defining the destination of the mothership path (or tour).
- $\mathcal{G}$: set of the target graphs.
- $g = (V_g, E_g)$: set of nodes and edges of each target graph $g \in \mathcal{G}$.
- $\mathcal{L}(e_g)$: length of edge $e$ of graph $g \in \mathcal{G}$.
- $B^{e_g}, C^{e_g}$: coordinates of the endpoints of edge $e$ of graph $g \in \mathcal{G}$.
- $\alpha^{e_g}$: percentage of edge $e$ of graph $g \in \mathcal{G}$ that must be visited.
- $\alpha^g$: percentage of graph $g \in \mathcal{G}$ that must be visited.
- $v_D$: drone speed.
- $v_M$: mothership speed.
- $M$: big-M constant.

# Initial data for the AMDRPG



# Binary and Integer Decision Variables for the AMDRPG-ST

- $\mu^{e_g} \in \{0,1\}\ \forall e_g \in E_g\ (g \in \mathcal{G})$: equal to 1 if edge $e$ of graph $g$ (or a portion of it) is visited by the drone, and 0 otherwise.
- $entry^{e_g} \in \{0,1\}\ \forall e_g \in E_g\ (g \in \mathcal{G})$: auxiliary binary variable used for linearizing expressions.
- $u^{e_g t} \in \{0,1\}\ \forall e_g \in E_g\ (g \in \mathcal{G})\ \forall t \in T$: equal to 1 if the drone enters in graph $g$ by $e_g$ at stage $t$, 0 otherwise.
- $z^{e_g e_g^{'}} \in \{0,1\}\ \forall e_g, e_g' \in E_g\ (g \in \mathcal{G})$: equal to 1 if the drone goes from $e_g$ to $e_g^{'}$, 0 otherwise.
- $v^{e_g t} \in \{0,1\}\ \forall e_g \in E_g\ (g \in \mathcal{G})\ \forall t \in T$: equal to 1 if the drone exits from graph $g$ by $e_g$ at stage $t$, 0 otherwise.
- $s^{e_g},\ \forall e_g \in E_g\ (g \in \mathcal{G})$: integer non negative variable representing the order of visit of edge $e$.

# Continuous Decision Variables for the AMDRPG-ST

**Location variables**

- $\rho^{e_g} \in [0,1]$ and $\lambda^{e_g} \in [0,1]$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: defining the entry and exit points on $e_g$.
- $\nu_{\min}^{e_g}$ and $\nu_{\max}^{e_g} \in [0,1] \forall e_g \in E_g$ $(g \in \mathcal{G})$: auxiliary variables used for linearizing expressions.
- $x_L^t$ $\forall t \in T$: coordinates representing the point where the mothership launches the drone at stage $t$.
- $x_R^t$ $\forall t \in T$: coordinates representing the point where the mothership retrieves the drone at stage $t$.
- $R^{e_g}$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: coordinates representing the entry point on edge $e$ of graph $g$.
- $L^{e_g}$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: coordinates representing the exit point on edge $e$ of graph $g$.

# Continuous Decision Variables for the AMDRPG-ST

## Distance variables

- $d_L^{e_g t} \geq 0, \ \forall e_g \in E_g \ (g \in \mathcal{G}) \ \forall t \in T$: representing the distance travelled by the drone from the launching point $x_L^t$ on the mothership at stage $t$ to the first visiting point $R^{e_g}$ on $e_g$.
- $d^{e_g e_g'} \geq 0, \ \forall e_g, e_g' \in E_g \ (g \in \mathcal{G})$: representing the distance travelled by the drone from the launching point $L^{e_g}$ on $e_g$ to the rendezvous point $R^{e_g'}$ on $e_g'$.
- $d^{e_g} \geq 0, \ \forall e_g \in E_g \ (g \in \mathcal{G})$: representing the distance travelled by the drone from the rendezvous point $R^{e_g}$ to the launching point $L^{e_g}$ on $e_g$.
- $d_R^{e_g t} \geq 0 \ \forall e_g \in E_g \ (g \in \mathcal{G}) \ \forall t \in T$: representing the distance travelled by the drone from the last visiting point $L^{e_g}$ on $e_g$ to the rendezvous point $x_R^t$ on the mothership at stage $t$.
- $d_{LR}^t \geq 0 \ \forall t \in T$: representing the distance travelled by the mothership from the launching point $x_L^t$ to the rendezvous point $x_R^t$ at stage $t$.
- $d_{RL}^t \geq 0 \ \forall t \in T$: representing the distance travelled by the mothership from the rendezvous point $x_R^t$ at stage $t$ to the launching point $x_L^{(t+1)}$ at the stage $t+1$.

# Visit of the graphs

We have considered two modes of visit to the target graphs $g \in \mathcal{G}$ that must be represented by their corresponding constraints:

- Visiting a percentage $\alpha^{e_g}$ of each edge $e_g$ which can be modeled by:

$$|\lambda^{e_g} - \rho^{e_g}|\mu^{e_g} \geq \alpha^{e_g}, \quad \forall e_g \in E_g. \tag{$\alpha$-E}$$

- Visiting a percentage $\alpha^g$ of the total length $\mathcal{L}(g)$ of the graph $g$ modeled by:

$$\sum_{e_g \in E_g} \mu^{e_g}|\lambda^{e_g} - \rho^{e_g}|\mathcal{L}(e_g) \geq \alpha^g \mathcal{L}(g). \tag{$\alpha$-G}$$

# Visit of the graphs

In both cases the corresponding constraints are nonlinear. For each edge $e_g$, we linearize the absolute value constraint ($\alpha$-E) by introducing a binary variable:

$$\mu^{e_g}|\rho^{e_g} - \lambda^{e_g}| \geq \alpha^{e_g} \iff \left\{ \begin{array}{ccl} \rho^{e_g} - \lambda^{e_g} & = & \nu_{\text{max}}^{e_g} - \nu_{\text{min}}^{e_g} \\ \nu_{\text{max}}^{e_g} & \leq & 1 - \text{entry}^{e_g} \\ \nu_{\text{min}}^{e_g} & \leq & \text{entry}^{e_g}, \\ \mu^{e_g}(\nu_{\text{max}}^{e_g} + \nu_{\text{min}}^{e_g}) & \geq & \alpha^{e_g}. \end{array} \right. \tag{$\alpha$-E}$$

The linearization of ($\alpha$-G) is similar to ($\alpha$-E) and only requires changing the last inequality in ($\alpha$-E) for

$$\sum_{e_g \in E_g} \mu^{e_g}(\nu_{\text{max}}^{e_g} + \nu_{\text{min}}^{e_g})\mathcal{L}(e_g) \geq \alpha_g \mathcal{L}(g). \tag{$\alpha$-G}$$

# Modeling the Drone Route

$$\sum_{g\in\mathcal{G}}\sum_{e_g\in E_g} u^{e_g t} = 1, \qquad \forall t \in T, \tag{2}$$

$$\sum_{g\in\mathcal{G}}\sum_{e_g\in E_g} v^{e_g t} = 1, \qquad \forall t \in T, \tag{3}$$

$$\sum_{e_g\in E_g}\sum_{t\in T} u^{e_g t} = 1, \qquad \forall g \in \mathcal{G}, \tag{4}$$

$$\sum_{e_g\in E_g}\sum_{t\in T} v^{e_g t} = 1, \qquad \forall g \in \mathcal{G}, \tag{5}$$

$$\sum_{e_g\in E_g} u^{e_g t} = \sum_{e_g\in E_g} v^{e_g t}, \qquad \forall g \in \mathcal{G}, \forall t \in T, \tag{6}$$

$$\sum_{e'_g\in E_g} z_g^{e'_g e_g} + \sum_{t\in T} u^{e_g t} = \mu^{e_g}, \qquad \forall e_g \in E_g : g \in \mathcal{G}, \tag{7}$$

$$\sum_{e'_g\in E_g} z_g^{e_g e'_g} + \sum_{t\in T} v^{e_g t} = \mu^{e_g}, \qquad \forall e_g \in E_g : g \in \mathcal{G}. \tag{8}$$

# Subtour elimination inside the graph

To prevent the existence of subtours within each graph $g \in \mathcal{G}$ that the drone must visit:

- One can add the Miller-Tucker-Zemlin constraints, given by:

$$s^{e_g} - s^{e'_g} + |E_g| z^{e_g e'_g} \leq |E_g| - 1, \qquad \forall e_g \neq e'_g \in E_g, \qquad (\mathrm{MTZ}_1)$$

$$0 \leq s^{e_g} \leq |E_g| - 1 \qquad \forall e_g \in E_g, \qquad (\mathrm{MTZ}_2)$$

- It is also possible to include the subtour elimination constraints:

$$\sum_{e_g, e'_g \in S} z_g^{e_g e'_g} \leq |S| - 1, \quad \forall S \subset E_g : g \in \mathcal{G}. \qquad (\mathrm{SEC})$$

# Distance constraints

To account for the different distances among the decision variables of the model we need to set the continuous variables $d_L^{e_g t}$, $d^{e_g}$, $d^{e_g e_g'}$, $d_R^{e_g t}$, $d_{RL}^t$ and $d_{LR}^t$. This can be done by means of the following constraints:

$$
\begin{aligned}
\|x_L^t - R^{e_g}\| &\leq d_L^{e_g t}, & \forall e_g : g \in \mathcal{G}, \forall t \in T, && (\text{DIST}_1\text{-t})\\
\|R^{e_g} - L^{e_g}\| &\leq d^{e_g}, & \forall e_g : g \in \mathcal{G}, \forall t \in T, && (\text{DIST}_2\text{-t})\\
\|R^{e_g} - L^{e_g'}\| &\leq d^{e_g e_g'}, & \forall e_g \neq e_g' \in E_g : g \in \mathcal{G}, && (\text{DIST}_3\text{-t})\\
\|L^{e_g} - x_R^t\| &\leq d_R^{e_g t}, & \forall e_g : g \in \mathcal{G}, \forall t \in T, && (\text{DIST}_4\text{-t})\\
\|x_R^t - x_L^{t+1}\| &\leq d_{RL}^t, & \forall t \in T, && (\text{DIST}_5\text{-t})\\
\|x_L^t - x_R^t\| &\leq d_{LR}^t, & \forall t \in T. && (\text{DIST}_6\text{-t})
\end{aligned}
$$

# Coordination constraint

To ensure that the time spent by the drone to visit graph $g$ at stage $t$ is less than or equal to the time that the mothership needs to move from the launching point to the rendezvous point at stage $t$, we need to define the following coordination constraint for each $g \in \mathcal{G}$ and $t \in T$:

$$\frac{1}{v_D}\left(\sum_{e_g \in E_g} u^{e_g t} d_L^{e_g t} + \sum_{e_g, e'_g \in E_g} z^{e_g e'_g} d^{e_g e'_g} + \sum_{e_g \in E_g} \mu^{e_g} d^{e_g} + \sum_{e_g \in E_g} v^{e_g t} d_R^{e_g t}\right) \leq \frac{d_{RL}^t}{v_M} + M\left(1 - \sum_{e_g \in E_g} u^{e_g t}\right). \tag{DCW-t}$$

# Setting the origin and the destination

Eventually, we have to impose that the tour of the mothership, together with the drone, starts from the origin *orig* and ends at the destination *dest*. To this end, we define the following constraints:

$$x_L^0 = orig, \tag{ORIG$_1$}$$

$$x_R^0 = orig, \tag{ORIG$_2$}$$

$$x_L^{|\mathcal{G}|+1} = dest, \tag{DEST$_1$}$$

$$x_R^{|\mathcal{G}|+1} = dest. \tag{DEST$_2$}$$

# Formulation for the AMDRPG-ST

$$
\begin{aligned}
\min \quad & \sum_{g\in\mathcal{G}} \sum_{e_g\in E_g} \sum_{t\in T} (u^{e_g t} d_L^{e_g t} + v^{e_g t} d_R^{e_g t}) + \sum_{g\in\mathcal{G}} \sum_{e_g\in E_g} \mu^{e_g} d^{e_g} + \\
& + \sum_{g\in\mathcal{G}} \sum_{e_g, e'_g\in E_g} z^{e_g e'_g} d^{e_g e'_g} + \sum_{t\in T} (d_{RL}^t + d_{LR}^t) \\
\text{s.t.} \quad & (2) - (8), \\
& (\mathrm{MTZ}_1) - (\mathrm{MTZ}_2) \text{ or } (\mathrm{SEC}), \\
& (\alpha\text{-E}) \text{ or } (\alpha\text{-G}), \\
& (\text{DCW-t}), \\
& (\mathrm{DIST}_1\text{-t}) - (\mathrm{DIST}_6\text{-t}), \\
& (\mathrm{ORIG}_1) - (\mathrm{DEST}_2).
\end{aligned}
$$

# Alternative formulations based on enforcing connectivity

In this family of formulations we replace the variables $u^{\cdot t}, v^{\cdot t}$ and constraints that model the tour using stages, namely (1)-(7), by constraints that ensure connectivity.

We will distinguish two different approaches:

- Using Miller-Tucker-Zemlin compact formulation.
- Using the known subtour elimination constraints.

## Binary and Integer Decision Variables for the AMDRPG-MTZ

- $\mu^{e_g} \in \{0,1\}$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: equal to 1 if edge $e$ of graph $g$ (or a portion of it) is visited by the drone, and 0 otherwise.
- $entry^{e_g} \in \{0,1\}$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: auxiliary binary variables for linearization.
- $u^{e_g} \in \{0,1\}$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: equal to 1 if the drone enters in graph $g$ by $e_g$, 0 otherwise.
- $z^{e_g e_g^{'}} \in \{0,1\}$ $\forall e_g, e_g' \in E_g$ $(g \in \mathcal{G})$: equal to 1 if the drone goes from $e_g$ to $e_g^{'}$, 0 otherwise.
- $v^{e_g} \in \{0,1\}$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: equal to 1 if the drone exits from graph $g$ by $e_g$, 0 otherwise.
- $w^{gg'} \in \{0,1\}$ $\forall g, g' \in \mathcal{G}$: equal to 1 if the mothership moves from $x_R^g$ to $x_L^{g^{'}}$, 0 otherwise.
- $s^{e_g}$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: integer non negative variables representing the order of visit of edge $e$ of graph $g$.

# Continuous Decision Variables for the AMDRPG-MTZ

**Location variables**

- $\rho^{e_g} \in [0,1]$ and $\lambda^{e_g} \in [0,1]$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: defining the entry and exit points on $e_g$.
- $\nu_{\text{min}}^{e_g}$ and $\nu_{\text{max}}^{e_g} \in [0,1]$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: auxiliary variables for linearization.
- $x_L^g$ $\forall g \in \mathcal{G}$: pairs of coordinates representing the point where the mothership launches the drone to visit graph $g$.
- $x_R^g$ $\forall g \in \mathcal{G}$: pairs of coordinates representing the point where the mothership retrieves the drone after visit graph $g$.
- $R^{e_g}$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: coordinates representing the entry point on edge $e$ of graph $g$.
- $L^{e_g}$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: coordinates representing the exit point on edge $e$ of graph $g$.

# Continuous Decision Variables for the AMDRPG-MTZ

## Distance variables

- $d_L^{e_g} \geq 0 \; \forall e_g \in E_g \; (g \in \mathcal{G})$: representing the distance travelled by the drone from the launching point on the mothership $x_L^g$ to the first visiting point $R^{e_g}$ on edge $e_g$.
- $d^{e_g e_g'} \geq 0 \; \forall e_g, e_g' \in E_g \; (g \in \mathcal{G})$: representing the distance travelled by the drone from launching point
- $L^{e_g}$ on $e_g$ to the rendezvous point $R^{e_g'}$ on $e_g'$.
- $d^{e_g} \geq 0 \; \forall e_g \in E_g \; (g \in \mathcal{G})$: representing the distance travelled by the drone from the rendezvous point $R^{e_g}$ to the launching point $L^{e_g}$ on $e_g$.
- $d_R^{e_g} \geq 0 \; \forall e_g \in E_g \; (g \in \mathcal{G})$: representing the distance travelled by the drone from the last visiting point $L^{e_g}$ on $e_g$ to the rendezvous point $x_R^g$ on the mothership.
- $d_{LR}^g \geq 0 \; \forall g \in \mathcal{G}$: representing the distance travelled by the mothership from the launching point $x_L^g$ to the rendezvous point $x_R^g$ while the drone is visiting $g$.
- $d_{RL}^{gg\prime} \geq 0 \; \forall g, g' \in \mathcal{G}$: representing the distance travelled by the mothership from the rendezvous point $x_R^g$ for graph $g$ to the launching point $x_L^{g\prime}$ for graph $g\prime$.

# Modeling the Drone Route

We can model the route that the drone follows in each particular graph $g \in \mathcal{G}$:

$$\sum_{e_g \in E_g} u^{e_g} = 1, \qquad \forall g \in \mathcal{G}, \tag{9}$$

$$\sum_{e_g \in E_g} v^{e_g} = 1, \qquad \forall g \in \mathcal{G}, \tag{10}$$

$$\sum_{e'_g \in E_g} z^{e'_g e_g} + u^{e_g} = \mu^{e_g}, \qquad \forall e_g \in E_g : g \in \mathcal{G}, \tag{11}$$

$$\sum_{e'_g \in E_g} z^{e_g e'_g} + v^{e_g} = \mu^{e_g}, \qquad \forall e_g \in E_g : g \in \mathcal{G}. \tag{12}$$

# Modeling the Mothership Route

On the other hand, to Mode the tour followed by the mothership, we have to include the following new constraints:

$$\sum_{g\in\mathcal{G}} w^{g0} = 0, \tag{13}$$

$$\sum_{g'\in\mathcal{G}} w^{(n_G+1)g'} = 0, \tag{14}$$

$$\sum_{g'\in\mathcal{G}\setminus\{g\}} w^{gg'} = 1, \qquad \forall g \in \mathcal{G}, \tag{15}$$

$$\sum_{g\in\mathcal{G}\setminus\{g'\}} w^{gg'} = 1, \qquad \forall g' \in \mathcal{G}, \tag{16}$$

$$s_g - s_{g'} + |\mathcal{G}|w^{gg'} \leq |\mathcal{G}| - 1, \qquad \forall g \neq g', \tag{MTZ$_3$}$$

$$0 \leq s_g \leq |\mathcal{G}| - 1 \qquad \forall g \in \mathcal{G}, \tag{MTZ$_4$}$$

$$s_0 = 0, \tag{MTZ$_5$}$$

$$s_{n_G+1} = n_G + 1. \tag{MTZ$_6$}$$

# Distance Constraints

$$
\begin{aligned}
\|x_L^g - R^{e_g}\| &\leq d_L^{e_g}, & \forall e_g : g \in \mathcal{G}, && \text{(DIST}_1\text{-g)}\\
\|R^{e_g} - L^{e_g}\| &\leq d^{e_g}, & \forall e_g : g \in \mathcal{G}, && \text{(DIST}_2\text{-g)}\\
\|R^{e_g} - L^{e_g'}\| &\leq d^{e_g e_g'}, & \forall e_g \neq e_g' : g \in \mathcal{G}, && \text{(DIST}_3\text{-g)}\\
\|L^{e_g} - x_R^g\| &\leq d_R^{e_g}, & \forall e_g : g \in \mathcal{G}, && \text{(DIST}_4\text{-g)}\\
\|x_R^g - x_L^{g'}\| &\leq d_{RL}^{gg'}, & \forall g, g' \in \mathcal{G}, && \text{(DIST}_5\text{-g)}\\
\|x_L^g - x_R^g\| &\leq d_{LR}^g, & \forall g \in \mathcal{G}. && \text{(DIST}_6\text{-g)}
\end{aligned}
$$

# Coordination constraint

Again, we need to be sure that the time spent by the drone to visit the graph $g$ is less than or equal to the time that the mothership needs to move from the launching point to the rendezvous point associated to this graph $g$. Hence, by using the same argument, as the one used in (DCW-t), we define for each $g \in \mathcal{G}$:

$$\frac{1}{v_D}\left(\sum_{e_g\in E_g} u^{e_g} d_L^{e_g} + \sum_{e_g, e'_g\in E_g} z^{e_g e'_g} d^{e_g e'_g} + \sum_{e_g\in E_g} \mu^{e_g} d^{e_g} + \sum_{e_g\in E_g} v^{e_g} d_R^{e_g}\right) \leq \frac{d_{RL}^g}{v_M}, \quad \forall g \in \mathcal{G}. \tag{DCW-g}$$

# Formulation for the AMDRPG-MTZ (resp. SEC)

$$
\begin{aligned}
\min \quad & \sum_{g\in\mathcal{G}} \sum_{e_g\in E_g} (u^{e_g} d_L^{e_g} + v^{e_g} d_R^{e_g}) + \sum_{g\in\mathcal{G}} \sum_{e_g\in E_g} \mu^{e_g} d^{e_g} + \\
& + \sum_{g\in\mathcal{G}} \sum_{e_g, e_g'\in E_g} z^{e_g e_g'} d^{e_g e_g'} + \sum_{g\in\mathcal{G}} d_{LR}^{g} + \sum_{g,g'\in\mathcal{G}} d_{RL}^{gg'} w^{gg'} \\
\text{s.t.} \quad & (9) - (16), \\
& (\text{MTZ}_1) - (\text{MTZ}_2) \text{ or } (\text{SEC}), \\
& (\text{MTZ}_3) - (\text{MTZ}_6), \\
& (\alpha\text{-E}) \text{ or } (\alpha\text{-G}), \\
& (\text{DCW-g}), \\
& (\text{DIST}_1\text{-g}) - (\text{DIST}_6\text{-g}), \\
& (\text{ORIG}_1) - (\text{DEST}_2).
\end{aligned}
$$

The formulation above can be slightly modified replacing constraints $(\text{MTZ}_3) - (\text{MTZ}_6)$ by

$$
\sum_{g,g'\in\mathcal{G}} w^{gg'} \leq |S| - 1, \quad \forall S \subseteq \{1, \ldots, |\mathcal{G}|\}. \tag{17}
$$

# Example 1

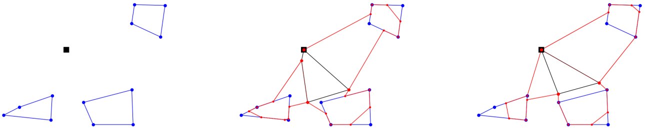
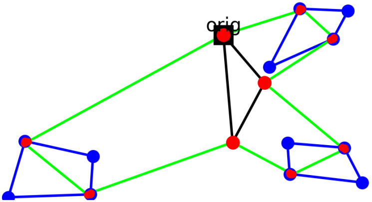
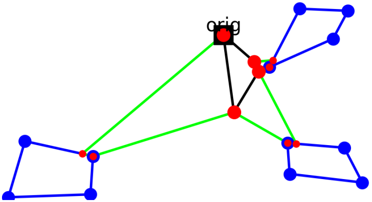


Figure: (a) Origin and target graphs, (b) Visit of $\alpha^{e_g}$% of each edge, (c) Visit of $\alpha_g$% of each graph.

# Example 2: Percentage of each edge



# Example 3: Percentage of each graph



# Parameters of the NMDRPG

- $\mathcal{N} = (V, E)$: set of nodes and edges of the network representing the road system where the mothership can move.
- $e = (i, j)$,$e' = (i', j')$: starting edge and ending edge of the mothership tour.
- $\mathcal{G}$: set of the target graphs.
- $g = (V_g, E_g)$: set of nodes and edges of each target graph $g \in \mathcal{G}$.
- $\mathcal{L}(e_g)$: length of edge $e$ of graph $g \in \mathcal{G}$.
- $B^{e_g}, C^{e_g}$: coordinates of the endpoints of edge $e$ of graph $g \in \mathcal{G}$.
- $\alpha^{e_g}$: percentage of edge $e$ of graph $g \in \mathcal{G}$ that must be visited.
- $\alpha^{g}$: percentage of graph $g \in \mathcal{G}$ that must be visited.
- $v_D$: drone speed.
- $v_M$: mothership speed.
- $M$: big-M constant.

# Binary and Integer Decision Variables for the NMDRPG-ST

**Drone tour variables**

- $\mu^{e_g} \in \{0,1\}\ \forall e_g \in E_g\ (g \in \mathcal{G})$: equal to 1 if edge $e$ of graph $g$ (or a portion of it) is visited by the drone, and 0 otherwise.
- $entry^{e_g} \in \{0,1\}\ \forall e_g \in E_g\ (g \in \mathcal{G})$: auxiliary binary variables for linearization.
- $u^{e_g t} \in \{0,1\}\ \forall e_g \in E_g\ (g \in \mathcal{G}),\ \ \forall t \in T$: equal to 1 if the drone enters in graph $g$ by $e_g$ at stage $t$, 0 otherwise.
- $z^{e_g e_g^{'}} \in \{0,1\}\ \forall e_g, e_{g'} \in E_g\ (g \in \mathcal{G})$: equal to 1 if the drone goes from $e_g$ to $e_g^{'}$, 0 otherwise.
- $v^{e_g t} \in \{0,1\}\ \forall e_g \in E_g\ (g \in \mathcal{G}),\ \ \forall t \in T$: equal to 1 if the drone exits from graph $g$ by $e_g$ at stage $t$, 0 otherwise.
- $s^{e_g}\ \ \forall e_g \in E_g\ (g \in \mathcal{G})$: integer non negative variable representing the order of visit of edge $e$ of graph $g$.

# Binary and Integer Decision Variables for the NDMRPG-ST

**Mothership tour variables inside the network**

- $\mu_L^{et} \in \{0,1\}, \ \forall e \in E, \quad \forall t \in T$: equal to 1 if the launching point $x_L^t$ is located on $e$ at stage $t$.
- $\mu_R^{et} \in \{0,1\}, \ \forall e \in E, \quad \forall t \in T$: equal to 1 if the rendezvous point $x_R^t$ is located on $e$ at stage $t$.
- $z_{LR}^{ee't} \in \{0,1\} \quad \forall e, e' \in E, \quad \forall t \in T$: equal to 1 if the launching point $x_L^t$ is located on $e$ and the rendezvous point $x_R^t$ is located on $e'$ at stage $t$.
- $b_{LR}^{it} \in \{0,1\}, \ \forall i : e = (i,j) \in E, \quad \forall t \in T$, equal to 1 if the mothership exits from $x_L^t$ by the vertex $V^i$ of the edge $e$.
- $c_{LR}^{it} \in \{0,1\}, \ \forall i : e = (i,j) \in E, \quad \forall t \in T$, equal to 1 if the mothership enters in $x_R^t$ by the vertex $V^i$ of the edge $e$.
- $q_{LR}^{et} \geq 0 \quad \forall e \in E, \ \forall t \in T$, integer variable counting the number of times the mothership fully traverses edge $e$ to move between the launching point $x_L^t$ on $e$ to the rendezvous point $x_R^t$ on $e'$ at stage $t$.

# Binary and Integer Decision Variables for the NMDRPG-ST

**Mothership tour variables inside the network**

- $z_{RL}^{ee't} \in \{0,1\} \quad \forall e, e' \in E, \quad \forall t \in T$: equal to 1 if the rendezvous point $x_R^t$ is located on $e$ at stage $t$ and the launching point $x_L^{t+1}$ is located on $e'$ at stage $t+1$.
- $b_{RL}^{it} \in \{0,1\}, \ \forall i : e = (i,j) \in E, \quad \forall t \in T$, equal to 1 if the mothership exits from $x_R^t$ by the vertex $V^i$ of the edge $e$.
- $c_{RL}^{it} \in \{0,1\}, \ \forall i : e = (i,j) \in E, \quad \forall t \in T$, equal to 1 if the mothership enters in $x_L^{t+1}$ by the vertex $V^i$ of the edge $e$.
- $q_{RL}^{et} \geq 0 \quad \forall e \in E, \ \forall t \in T$, integer variable counting the number of times the mothership fully traverses edge $e$ to move between the rendezvous point $x_R^t$ on $e$ to the launch point for the next stage $x_L^{t+1}$.

# Continuous Decision Variables for the NMDRPG-ST

**Location variables**

- $\rho^{e_g} \in [0,1]$ and $\lambda^{e_g} \in [0,1]$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: defining the entry and exit points on $e_g$.
- $\gamma_L^{et} \in [0,1]$ and $\gamma_R^{et} \in [0,1]$ $\forall e \in E$, $\forall t \in T$: defining the launching and rendezvous points on edge $e$.
- $\nu_{\min}^{e_g}$ and $\nu_{\max}^{e_g} \in [0,1] \forall e_g \in E_g$ $(g \in \mathcal{G})$: auxiliary variables used for linearizing expressions.
- $x_L^t$ $\forall t \in T$: coordinates representing the point where the mothership launches the drone at stage $t$.
- $x_R^t$ $\forall t \in T$: coordinates representing the point where the mothership retrieves the drone at stage $t$.
- $R^{e_g}$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: coordinates representing the entry point on edge $e$ of graph $g$.
- $L^{e_g}$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: coordinates representing the exit point on edge $e$ of graph $g$.

# Continuous Decision Variables for the NMDRPG-ST

## Distance variables

- $d_L^{e_g t} \geq 0, \ \forall e_g \in E_g \ (g \in \mathcal{G}) \ \forall t \in T$: representing the distance travelled by the drone from the launching point $x_L^t$ on the mothership at stage $t$ to the first visiting point $R^{e_g}$ on $e_g$.
- $d^{e_g e'_g} \geq 0, \ \forall e_g, e'_g \in E_g \ (g \in \mathcal{G})$: representing the distance travelled by the drone from the launching point $L^{e_g}$ on $e_g$ to the rendezvous point $R^{e'_g}$ on $e'_g$.
- $d^{e_g} \geq 0, \ \forall e_g \in E_g \ (g \in \mathcal{G})$: representing the distance travelled by the drone from the rendezvous point $R^{e_g}$ to the launching point $L^{e_g}$ on $e_g$.
- $d_R^{e_g t} \geq 0 \ \forall e_g \in E_g \ (g \in \mathcal{G}) \ \forall t \in T$: representing the distance travelled by the drone from the last visiting point $L^{e_g}$ on $e_g$ to the rendezvous point $x_R^t$ on the mothership at stage $t$.
- $d_{LR}^t \geq 0 \ \forall t \in T$: representing the distance travelled by the mothership from the launching point $x_L^t$ to the rendezvous point $x_R^t$ at stage $t$.
- $d_{RL}^t \geq 0 \ \forall t \in T$: representing the distance travelled by the mothership from the rendezvous point $x_R^t$ at stage $t$ to the launching point $x_L^{(t+1)}$ at the stage $t+1$.

# Modeling the distance inside the graph

The distance traveled by the mothership, between two consecutive launching and rendezvous points in two edges, not necessarily distinct, of the graph can be represented as:

$$d_{LR}^{ee't} = \begin{cases} |\gamma_L^{et} - \gamma_R^{et}|\mathcal{L}(e), & \text{if } e = e', \\ \left[b_{LR}^{it}\gamma_L^{et} + b_{LR}^{jt}(1-\gamma_L^{et})\right]\mathcal{L}(e) + \sum\limits_{e''\in\mathcal{N}} q_{LR}^{e''t}\mathcal{L}(e'') + \left[c_{LR}^{i't}\gamma_R^{e't} + c_{LR}^{j't}(1-\gamma_R^{e't})\right]\mathcal{L}(e'), & \text{if } e \neq e'. \end{cases} \tag{$d_{LR}^{t\mathcal{N}}$}$$

Similarly, the distance covered by the mothership along the path on the network from the rendezvous point $x_R^t$ to the next launching point $x_L^{t+1}$ can be modeled using the following definition of distance:

$$d_{RL}^{ee't} = \begin{cases} |\gamma_R^{et} - \gamma_L^{et+1}|\mathcal{L}(e), & \text{if } e = e', \\ \left[b_{RL}^{it}\gamma_R^{et} + b_{RL}^{jt}(1-\gamma_R^{et})\right]\mathcal{L}(e) + \sum\limits_{e''\in\mathcal{N}} q_{RL}^{e''t}\mathcal{L}(e'') + \left[c_{RL}^{i't}\gamma_L^{e't+1} + c_{RL}^{j't}(1-\gamma_L^{e't+1})\right]\mathcal{L}(e'), & \text{if } e \neq e'. \end{cases} \tag{$d_{RL}^{t\mathcal{N}}$}$$

# Example of parameterization

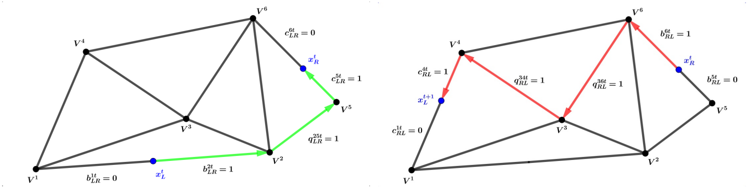



Figure: An example of a parameterization of a mothership route for a stage $t$. In the first picture, the mothership launches the drone at the point $x_L^t$ on the edge $\overline{V^1V^2}$, then traverses the edge $\overline{V^2V^3}$ and finally moves to $x_R^t$ on the edge $\overline{V^4V^5}$ to retrieve it. In the second one, the mothership retrieves the drone at the point $x_R^t$ on the edge $\overline{V^4V^5}$, then traverses the edge $\overline{V^3V^6}$, the edge $\overline{V^3V^4}$ and finally moves to $x_L^{t+1}$ on the edge $\overline{V^1V^4}$ to launch the drone for the next stage.

# Modeling the distance inside the graph

With the definition one can account for the movement of the mothership at each stage $t \in T$ from a launching to a rendezvous point:

$$x_L^t = \sum_{e=(i,j)\in E} \mu_L^{et}\left[v^i + \gamma_L^{et}(v^j - v^i)\right], \qquad \forall t \in T, \tag{18}$$

$$x_R^t = \sum_{e=(i,j)\in E} \mu_R^{et}\left[v^i + \gamma_R^{et}(v^j - v^i)\right], \qquad \forall t \in T, \tag{19}$$

$$z_{LR}^{ee't} = \mu_L^{et}\mu_R^{e't}, \qquad \forall e, e' \in E, \ \forall t \in T, \tag{20}$$

$$b_{LR}^{it} \leq \sum_{e\in\delta(i)} \mu_L^{et}, \qquad \forall i \in V, \ \forall t \in T, \tag{21}$$

$$c_{LR}^{it} \leq \sum_{e\in\delta(i)} \mu_R^{et}, \qquad \forall i \in V, \ \forall t \in T, \tag{22}$$

$$b_{LR}^{it} + b_{LR}^{jt} \geq \mu_L^{et}, \qquad \forall e = (i,j) \in E, \ \forall t \in T, \tag{23}$$

$$c_{LR}^{it} + c_{LR}^{jt} \geq \mu_R^{et}, \qquad \forall e = (i,j) \in E, \ \forall t \in T, \tag{24}$$

$$b_{LR}^{it} + \sum_{\{j:(i,j)\in E\}} q_{LR}^{jit} = \sum_{\{j:(i,j)\in E\}} q_{LR}^{ijt} + c_{LR}^{it}, \qquad \forall i \in V, \ \forall t \in T, \tag{25}$$

$$\sum_{e\in E} \mu_L^{et} = 1, \qquad \forall t \in T, \tag{26}$$

$$\sum_{e\in E} \mu_R^{et} = 1, \qquad \forall t \in T, \tag{27}$$

$$d_{LR}^t = \sum_{e,e'\in E} z_{LR}^{ee't} d_{LR}^{ee't}, \qquad \forall t \in T. \tag{28}$$

# Modeling the distance inside the graph

We can use a set of constraints similar to those used above and the distance from $x_R^t$ to $x_L^{t+1}$ can be computed by means of the following additional constraints:

$$z_{RL}^{ee't} = \mu_R^{et}\mu_L^{e't+1}, \tag{29}$$

$$b_{RL}^{it} \leq \sum_{e\in\delta(i)} \mu_R^{et}, \quad \forall i \in V, \quad \forall t \in T, \tag{30}$$

$$c_{RL}^{it} \leq \sum_{e\in\delta(i)} \mu_L^{et}, \quad \forall e \in \delta(i), \forall i \in V, \quad \forall t \in T, \tag{31}$$

$$b_{RL}^{it} + b_{RL}^{jt} \geq \mu_R^{et}, \quad \forall e = (i,j) \in E, \quad \forall t \in T, \tag{32}$$

$$c_{RL}^{it} + c_{RL}^{jt} \geq \mu_L^{et+1}, \quad \forall e = (i,j) \in E, \quad \forall t \in T, \tag{33}$$

$$b_{RL}^{it} + \sum_{\{j:(i,j)\in E\}} q_{RL}^{jit} = \sum_{\{j:(i,j)\in E\}} q_{RL}^{ijt} + c_{RL}^{it}, \quad \forall i \in V, \quad \forall t \in T, \tag{34}$$

$$\sum_{e\in E} \mu_L^{et} = 1, \quad \forall t \in T, \tag{35}$$

$$\sum_{e\in E} \mu_R^{et} = 1, \quad \forall t \in T, \tag{36}$$

$$d_{RL}^{t} = \sum_{e,e'\in E} z_{RL}^{ee't} d_{RL}^{ee't}, \quad \forall t \in T. \tag{37}$$

# Formulation for the NMDRPG-ST

$$
\begin{aligned}
\min \quad & \sum_{g\in\mathcal{G}} \sum_{e_g\in E_g} \sum_{t\in T} (u^{e_g t} d_L^{e_g t} + v^{e_g t} d_R^{e_g t}) + \sum_{g\in\mathcal{G}} \sum_{e_g\in E_g} \mu^{e_g} d^{e_g} + \\
& + \sum_{g\in\mathcal{G}} \sum_{e_g, e'_g\in E_g} z^{e_g e'_g} d^{e_g e'_g} + \sum_{t\in T} (d_{RL}^t + d_{LR}^t) \\
\text{s.t.} \quad & (2) - (8), \\
& (18) - (37), \\
& (\mathrm{MTZ}_1) - (\mathrm{MTZ}_2) \text{ or } (\mathrm{SEC}), \\
& (\alpha\text{-E}) \text{ or } (\alpha\text{-G}), \\
& (\text{DCW-t}), \\
& (d_{LR}^{t\mathcal{N}}), (d_{RL}^{t\mathcal{N}}), \\
& (\mathrm{DIST}_1\text{-t}) - (\mathrm{DIST}_6\text{-t}), \\
& (\mathrm{ORIG}_1) - (\mathrm{DEST}_2).
\end{aligned}
$$

# Binary and Integer Decision Variables for the NMDRPG-MTZ

## Drone tour variables

- $\mu^{e_g} \in \{0,1\}\ \forall e_g \in E_g\ (g \in \mathcal{G})$: equal to 1 if edge $e$ of graph $g$ (or a portion of it) is visited by the drone, and 0 otherwise.
- $entry^{e_g} \in \{0,1\}\ \forall e_g \in E_g\ (g \in \mathcal{G})$: auxiliary binary variables for linearization.
- $u^{e_g} \in \{0,1\}\ \forall e_g \in E_g\ (g \in \mathcal{G})$: equal to 1 if the drone enters in graph $g$ by $e_g$, 0 otherwise.
- $z^{e_g e_g^{'}} \in \{0,1\}\ \forall e_g, e_g' \in E_g\ (g \in \mathcal{G})$: equal to 1 if the drone goes from $e_g$ to $e_g^{'}$, 0 otherwise.
- $v^{e_g} \in \{0,1\}\ \forall e_g \in E_g\ (g \in \mathcal{G})$: equal to 1 if the drone exits from graph $g$ by $e_g$, 0 otherwise.
- $w^{gg'} \in \{0,1\}\ \ \forall g, g' \in \mathcal{G}$: equal to 1 if the mothership moves from $x_R^g$ to $x_L^{g^{'}}$, 0 otherwise.
- $s^{e_g}\ \ \forall e_g \in E_g\ (g \in \mathcal{G})$: integer non negative variables representing the order of visit of edge $e$ of graph $g$.

# Binary and Integer Decision Variables for the NMDRPG-MTZ

## Mothership tour variables inside the network

- $\mu_L^{eg} \in \{0,1\}$ $\forall e \in E, \quad \forall g \in \mathcal{G}$: equal to 1 if the launching point $x_L^g$ to visit graph $g$ is located on $e$.
- $\mu_R^{eg} \in \{0,1\}$ $\forall e \in E, \quad \forall g \in \mathcal{G}$: equal to 1 if the rendezvous point to visit graph $g$ is located on $e$.
- $z_{LR}^{ee'g} \in \{0,1\}$ $\forall e, e' \in E, \ \forall g \in \mathcal{G}$: equal to 1 if the launching point $x_L^g$ to visit graph $g$ is located on $e$ and the rendezvous point $x_R^g$ is located on $e'$.
- $b_{LR}^{ig} \in \{0,1\}, \ \forall i : e = (i,j) \in E, \quad \forall g \in \mathcal{G}$, equal to 1 if the mothership exits from $x_L^g$ by the vertex $V^i$ of the edge $e$.
- $c_{LR}^{ig} \in \{0,1\}, \ \forall i : e = (i,j) \in E, \quad \forall g \in \mathcal{G}$, equal to 1 if the mothership enters in $x_R^g$ by the vertex $V^i$ of the edge $e$.
- $q_{LR}^{eg} \geq 0, \quad \forall e \in E, \ \forall g \in \mathcal{G}$, integer variable counting the number of times the mothership fully traverses edge $e$ to move between the launching point $x_L^g$ on $e$ to the rendezvous point $x_R^g$ on $e'$ for graph $g$.

# Binary and Integer Decision Variables for the NMDRPG-MTZ

**Mothership tour variables inside the network**

- $z_{RL}^{ee'gg'} \in \{0,1\}\ \forall e, e' \in E,\ \forall g, g' \in G$: equal to 1 if the rendezvous point $x_R^g$, for graph $g$, is located on $e$ and the launching point $x_L^{g'}$, for graph $g'$, is located on $e'$.
- $b_{RL}^{ig} \in \{0,1\},\ \forall i : e = (i,j) \in E,\ \ \forall g \in \mathcal{G}$, equal to 1 if the mothership exits from $x_R^g$ by the vertex $V^i$ of the edge $e$.
- $c_{RL}^{ig} \in \{0,1\},\ \forall i : e = (i,j) \in E,\ \ \forall g \in \mathcal{G}$, equal to 1 if the mothership enters in $x_L^g$ by the vertex $V^i$ of the edge $e$.
- $q_{RL}^{egg'} \geq 0,\ \ \forall e \in E,\ \forall g \in \mathcal{G}$, integer variable counting the number of times the mothership fully traverses edge $e$ to move between the rendezvous point $x_R^g$ to the launching point $x_L^{g'}$ for graph $g$.

# Continuous Decision Variables for the NMDRPG-MTZ

## Location variables

- $\rho^{e_g} \in [0,1]$ and $\lambda^{e_g} \in [0,1]$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: defining the entry and exit points on edge $e_g$.
- $\gamma_R^{eg} \in [0,1]$ and $\gamma_L^{eg} \in [0,1]$ $\forall e \in E, \ \forall g \in \mathcal{G}$: defining the launching and rendezvous points, associated with graph $g$, located on edge $e$ of the network $\mathcal{N}$.
- $\nu_{\min}^{e_g}$ and $\nu_{\max}^{e_g} \in [0,1]$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: auxiliary variables for linearization.
- $x_L^g \quad g \in \mathcal{G}$: coordinates representing the point where the mothership launches the drone to visit graph $g$.
- $x_R^g \quad g \in \mathcal{G}$: coordinates representing the point where the mothership retrieves the drone to visit graph $g$.
- $R^{e_g}$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: coordinates representing the entry point on edge $e$ of graph $g$.
- $L^{e_g}$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: coordinates representing the exit point on edge $e$ of graph $g$.

# Continuous Decision Variables for the NMDRPG-MTZ

## Distance variables

- $d_L^{e_g} \geq 0 \ \forall e_g \in E_g \ (g \in \mathcal{G})$: representing the distance travelled by the drone from the launching point $x_L^g$ on the mothership to the first visiting point $R^{e_g}$ on $e_g$.
- $d^{e_g e'_g} \geq 0 \ \forall e_g, e'_g \in E_g \ (g \in \mathcal{G})$: representing the distance travelled by the drone from launching point $L^{e_g}$ on $e_g$ to the rendezvous point $R^{e'_g}$ on $e'_g$.
- $d^{e_g} \geq 0 \ \forall e_g \in E_g \ (g \in \mathcal{G})$: representing the distance travelled by the drone from the rendezvous point $R^{e_g}$ to the launching point $L^{e_g}$ on $e_g$.
- $d_R^{e_g} \geq 0 \ \forall e_g \in E_g \ (g \in \mathcal{G})$: representing the distance travelled by the drone from the last visiting point for graph $g$ $L^{e_g}$ to the rendezvous point $x_R^g$ on the mothership.
- $d_{LR}^{ee'g} \geq 0 \ \forall e, \ e' \in E, \ \forall g \in \mathcal{G}$: representing the distance travelled by the mothership from the launching point $x_L^g$ on $e$ and the rendezvous point $x_R^g$ on $e'$ to visit graph $g$.
- $d_{RL}^{ee'gg'} \geq 0 \ \forall e, \ e' \in E \ \forall g, \ g' \in \mathcal{G}$: representing the distance travelled by the mothership from the rendezvous point $x_R^g$, for graph $g$, on $e$ to the launching point $x_L^{g'}$, for graph $g'$, on $e'$.
- $d_{LR}^g \geq 0 \ \forall g \in \mathcal{G}$: representing the total distance travelled by the mothership between the launching point $x_L^g$ and the rendezvous point $x_R^g$ to visit graph $g$.
- $d_{RL}^{gg'} \geq 0 \ \forall g, \ g' \in \mathcal{G}$, representing the total distance travelled by the mothership between the rendezvous point $x_R^g$, for graph $g$, and the launching point $x_L^{g'}$, for graph $g'$.

# Modeling the distance inside the graph

In this case, we observe that the distance between launching and rendezvous points in two edges $e = (i, j)$, $e' = (i', j') \in E$, not necessarily distinct, of the graph can be represented as:

$$d_{LR}^{ee'g} = \begin{cases} |\gamma_L^{eg} - \gamma_R^{eg}|\mathcal{L}(e), & \text{if } e = e', \\ \left[b_{LR}^{jg}\gamma_L^{eg} + b_{LR}^{jg}(1-\gamma_L^{eg})\right]\mathcal{L}(e) + \sum_{e''\in\mathcal{N}} q_{LR}^{e''g}\mathcal{L}(e'') + \left[c_{LR}^{i'g}\gamma_R^{e'g} + c_{LR}^{j'g}(1-\gamma_R^{e'g})\right]\mathcal{L}(e'), & \text{if } e \neq e', \end{cases} \tag{$d_{LR}^{g\mathcal{N}}$}$$

Similarly, the distance covered by the mothership along the path on the network $\mathcal{N}$, from the rendezvous point $x_R^g \in e$, after the visit to $g \in \mathcal{G}$ to the next launching point $x_L^{g'} \in e'$ (to go to the graph $g' \in \mathcal{G}$), can be modeled using the following definition of distance:

$$d_{RL}^{ee'gg'} = \begin{cases} |\gamma_R^{eg} - \gamma_L^{eg'}|\mathcal{L}(e), & \text{if } e = e', \\ \left[b_{RL}^{ig}\gamma_R^{eg} + b_{RL}^{jg}(1-\gamma_R^{eg})\right]\mathcal{L}(e) + \sum_{e''\in\mathcal{N}} q_{RL}^{e''gg'}\mathcal{L}(e'') + \left[c_{RL}^{i'g'}\gamma_L^{e'g'} + c_{RL}^{j'g'}(1-\gamma_L^{e'g'})\right]\mathcal{L}(e'), & \text{if } e \neq e'. \end{cases} \tag{$d_{RL}^{g\mathcal{N}}$}$$

# Modeling the distance inside the graph

All the above arguments give the necessary elements to account for the movement of the mothership on the network $\mathcal{N} = (V, E)$ from $x_L^g$ to $x_R^g$:

$$x_L^g = \sum_{e=(i,j)\in E} \mu_L^{eg} \left[ v^i + \gamma_L^{eg}(v^j - v^i) \right], \quad \forall g \in \mathcal{G}, \tag{38}$$

$$x_R^g = \sum_{e=(i,j)\in E} \mu_R^{eg} \left[ v^i + \gamma_R^{eg}(v^j - v^i) \right], \quad \forall g \in \mathcal{G}, \tag{39}$$

$$z_{LR}^{ee'g} = \mu_L^{eg} \mu_R^{e'g}, \quad \forall e, e' \in E, \ \forall g \in \mathcal{G}, \tag{40}$$

$$b_{LR}^{ig} \leq \sum_{e\in\delta(i)} \mu_L^{eg}, \quad \forall i \in V, \ \forall g \in \mathcal{G}, \tag{41}$$

$$c_{LR}^{ig} \leq \sum_{e\in\delta(i)} \mu_R^{eg}, \quad \forall i \in V, \ \forall g \in \mathcal{G}, \tag{42}$$

$$b_{LR}^{ig} + b_{LR}^{jg} \geq \mu_L^{eg}, \quad \forall e = (i, j) \in E, \ \forall g \in \mathcal{G}, \tag{43}$$

$$c_{LR}^{ig} + c_{LR}^{jg} \geq \mu_R^{eg}, \quad \forall e = (i, j) \in E, \ \forall g \in \mathcal{G}, \tag{44}$$

$$b_{LR}^{ig} + \sum_{\{j:(i,j)\in E\}} q_{LR}^{jig} = \sum_{\{j:(i,j)\in E\}} q_{LR}^{ijg} + c_{LR}^{ig}, \quad \forall i \in V, \ \forall g \in \mathcal{G}, \tag{45}$$

$$\sum_{e\in E} \mu_L^{eg} = 1, \quad \forall g \in \mathcal{G}, \tag{46}$$

$$\sum_{e\in E} \mu_R^{eg} = 1, \quad \forall g \in \mathcal{G}, \tag{47}$$

$$d_{LR}^g = \sum_{e,e'\in E} z_{LR}^{ee'g} d_{LR}^{ee'g}. \quad \forall g \in \mathcal{G}. \tag{48}$$

# Modeling the distance inside the graph

We can use a set of constraints similar to those used above and the distance from $x_R^g$ to $x_L^{g'}$ can be computed by means of the following additional constraints:

$$
\begin{aligned}
z_{RL}^{ee'gg'} &= \mu_R^{eg}\mu_L^{e'g'}, && \forall e, e' \in E, \quad \forall g, g' \in \mathcal{G}, && (49)\\
b_{RL}^{ig} &\leq \sum_{e\in\delta(i)} \mu_R^{eg}, && \forall i \in V, \quad \forall g \in \mathcal{G}, && (50)\\
c_{RL}^{ig} &\leq \sum_{e\in\delta(i)} \mu_L^{eg}, && \forall i \in V, \quad \forall g \in \mathcal{G}, && (51)\\
b_{RL}^{ig} + b_{RL}^{jg} &\geq \mu_R^{eg}, && \forall e = (i,j) \in E, \quad \forall g \in \mathcal{G}, && (52)\\
c_{RL}^{ig} + c_{RL}^{jg} &\geq \mu_L^{eg}, && \forall e = (i,j) \in E, \quad \forall g \in \mathcal{G}, && (53)\\
b_{RL}^{ig} + \sum_{\{j:(i,j)\in E\}} q_{RL}^{jigg'} &= \sum_{\{j:(i,j)\in E\}} q_{RL}^{ijgg'} + c_{RL}^{ig'}, && \forall i \in V, \quad \forall g \in \mathcal{G}, && (54)\\
d_{RL}^{gg'} &= \sum_{e,e'} z_{RL}^{ee'gg'} d_{RL}^{ee'gg'}, && \forall g, g' \in \mathcal{G}. && (55)
\end{aligned}
$$

# Formulation for the NMDRPG-MTZ

$$
\begin{aligned}
\min \quad & \sum_{g\in\mathcal{G}} \sum_{e_g\in E_g} (u^{e_g} d_L^{e_g} + v^{e_g} d_R^{e_g}) + \sum_{g\in\mathcal{G}} \sum_{e_g\in E_g} \mu^{e_g} d^{e_g} + \\
& + \sum_{g\in\mathcal{G}} \sum_{e_g, e_g'\in E_g} z^{e_g e_g'} d^{e_g e_g'} + \sum_{g\in\mathcal{G}} d_{LR}^{g} + \sum_{g,g'\in\mathcal{G}} d_{RL}^{gg'} w^{gg'} \\
\text{s.t.} \quad & (9) - (16), \\
& (38) - (55), \\
& (\text{MTZ}_1) - (\text{MTZ}_2) \text{ or } (\text{SEC}), \\
& (\text{MTZ}_3) - (\text{MTZ}_6), \\
& (\alpha\text{-E}) \text{ or } (\alpha\text{-G}), \\
& (\text{DCW-g}), \\
& (d_{LR}^{g\mathcal{N}}), (d_{RL}^{g\mathcal{N}}), \\
& (\text{DIST}_1\text{-g}) - (\text{DIST}_6\text{-g}), \\
& (\text{ORIG}_1) - (\text{DEST}_2).
\end{aligned}
$$

# Example

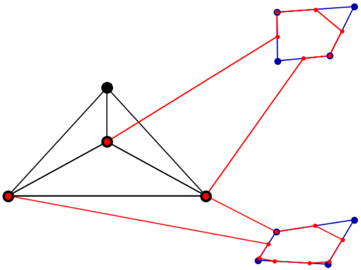


Figure: NMDRPG-MTZ example for a drone that visits the 50% of each edge

# Matheuristic for the MDRPG

**Pseudo-code of this algorithm:**

STEP 1 (Centroids of the graphs)
- Let *orig* be the origin/destination of the mothership tour

For each graph $g \in \mathcal{G}$:
- identify its centroid $c_g$ and consider its neighborhood defined as the circle $\rho(c_g, 2)$ centered at $c_g$ and with radius 2

STEP 2 (Order of visit of the graphs) Determine an order of visit for the graphs in $\mathcal{G}$ by solving the XPPN of the mothership over the set of the neighborhoods associated with the centroids of those graphs

# Matheuristic for the MDRPG

**Pseudo-code of this algorithm:**

STEP 3 (Determining the location of launching/rendezvous points) Let $\bar{w}_{gg'}$, $\forall g, g' \in \mathcal{G}$ be the optimal values of the variables $w_{gg'}$ generated by STEP 2.
Following this order of visit, set the launching point for the first graph as the depot, then solve the resulting (MDRPG) limited to the first graph. Repeat the same procedure for the remaining graphs to be visited, by solving on one single graph each time, by fixing as launching point of the current graph the rendezvous point of the previous graph.

STEP 4 (Solution update) Let $\bar{z}$ be the solution obtained by STEP 3, consisting of the tour of the drone on each target, and let $\bar{x}_L^g$, $\bar{x}_R^g$ $\forall g \in \mathcal{G}$ be the associated launching/rendezvous points Solve the model (MDRPG) with these launching/rendezvous points but leaving free the $w_{gg'}$ $\forall g, g' \in \mathcal{G}$ variables and providing to the solver $\bar{z}$ as initial partial solution.

STEP 5 Let $\hat{z}$ be the updated solution obtained by STEP 4 and $\hat{w}_{gg'}$ the associated order of visit of the graphs.
If the $\hat{w}_{gg'} \neq \bar{w}_{gg'}$ repeat from STEP 3, otherwise stop.

# Example of the Matheuristic

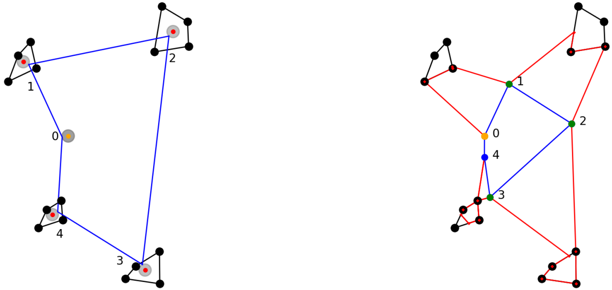



Figure: Illustrative Example

# Computational Experiments: Data Generation

We consider two typology of planar graphs:

- Grid graphs
- Delaunay triangulation (computed by using the Python class scipy.spatial.Delaunay)

# Computational Experiments: Grid Generation

To simulate graphs that are similar to the one found in a road network, we design grid graphs following these steps:

1. We consider a square of side 100 units that we divide in subsquares of side 5 and we randomly select among them the locations for each graph.
2. Each subsquare of side 5 selected is further partitioned in subsquares of side $\frac{5}{n}$ where $n$ is the cardinality of the set of nodes of the graph to build.
3. Two opposite corner subsquares are considered and one point inside each of them is randomly selected.
4. A grid of $n$ points is identified by locating $\frac{n}{m}$ equally spaced points on the two sides square rectangle whose diagonal joins these two points is built, where $m$ is randomly selected in the set of divisors of the number of points of the graph.

# Computational Experiments: Grid Generation

To simulate graphs that are similar to the one found in a road network, we design grid graphs following these steps:

5. The links of the graphs connect each point to its adjacent ones lying on the same side and with the one located on the opposite side of the square.
6. Let $width_x$ and $width_y$ be the lengths of these edges. In order to perturb the coordinates of these points, we randomly add a value, ranging between $-\frac{width_x}{3}$ and $\frac{width_x}{3}$ to the $x$ coordinate and between $-\frac{width_y}{3}$ and $\frac{width_y}{3}$ to the $y$ coordinate, always imposing that the perturbed point still belongs to the square.
7. The resulting grid graph is obtained connecting the same pairs of points but with perturbed coordinates.

# Computational Experiments: Grid Generation

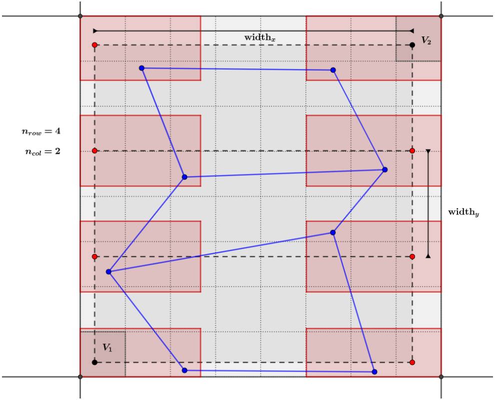



Figure: Example of generation of a grid graph

# Experiment 1: Comparing Gurobi and Cplex

We generate five instances with a number $|\mathcal{G}| = 10$ of target graphs:

- 3 graphs with 4 nodes, 3 graphs with 6 nodes, 3 graphs with 8 nodes and 1 graph with 10 nodes.
- $v_D = 2v_M$.
- 80% of each target must be visited by the drone.

The experiment consists on:

- Running the three formulations proposed for the (AMDRPG): Stages, MTZ and SEC.
- Using two commercial solvers, Cplex 12.8 and Gurobi 9.03.
- Time Limit: 1 hour.

# Experiment 1: Results

Table: Comparison between formulations for grid instances

| **Gap %** | Average | | Min | | Max | |
|---|---|---|---|---|---|---|
| **Solver** | Cplex | Gurobi | Cplex | Gurobi | Cplex | Gurobi |
| **Formulation** | | | | | | |
| Stages | 0,87 | 0,87 | 0,85 | 0,84 | 0,88 | 0,88 |
| MTZ | 0,66 | 0,62 | 0,59 | 0,58 | 0,72 | 0,65 |
| SEC | 0,65 | 0,61 | 0,59 | 0,57 | 0,70 | 0,64 |

Table: Comparison between formulations for Delauney instances

| **Gap %** | Average | | Min | | Max | |
|---|---|---|---|---|---|---|
| **Solver** | Cplex | Gurobi | Cplex | Gurobi | Cplex | Gurobi |
| **Formulation** | | | | | | |
| Stages | 0,91 | 0,91 | 0,90 | 0,89 | 0,93 | 0,93 |
| MTZ | 0,78 | 0,74 | 0,74 | 0,70 | 0,82 | 0,79 |
| SEC | 0,77 | 0,75 | 0,73 | 0,69 | 0,82 | 0,81 |

# Experiment 2: Testing the performance of the Matheuristic

We generate five instances for each number $|\mathcal{G}| \in \{5, 10, 15, 20\}$ of target graphs:

- The same percentage of graphs (20%) has respectively 4, 6, 8, 10 and 12 nodes.
- $v_D = 3v_M$.
- $\alpha^g$ and $\alpha^{e_g}$ randomly generated in $[0, 1]$.

The experiment consists on:

- Running the MTZ formulation for (AMDRPG) with and without initial solution provided by the matheuristic.
- Using Gurobi 9.03.
- Time Limit: 2 hour.

# Experiment 2: Results

Table: Comparison between exact resolution with and without initialization

| | | Grid | | | Delauney | | |
|---|---|---|---|---|---|---|---|
| List | % | % Gap (i) | Time_h | % Gap (wi) | % Gap (i) | Time_h | % Gap (wi) |
| 0 | e | 0.72 | 105.12 | 0.73 | 0.78 | 154.92 | 0.74 |
| | g | 0.55 | 58.92 | 0.54 | 0.62 | 92.64 | 0.67 |
| 1 | e | 0.76 | 241.99 | 0.76 | 0.80 | 314.69 | 0.79 |
| | g | 0.71 | 182.61 | 0.70 | 0.74 | 353.04 | 0.75 |
| 2 | e | 0.76 | 367.69 | 0.76 | 0.80 | 447.61 | 0.80 |
| | g | 0.71 | 326.49 | 0.72 | 0.76 | 429.16 | 0.76 |
| 3 | e | 0.75 | 481.68 | 0.74 | 0.80 | 514.98 | 0.76* |
| | g | 0.71 | 492.27 | 0.70 | 0.77 | 582.90 | 0.77 |

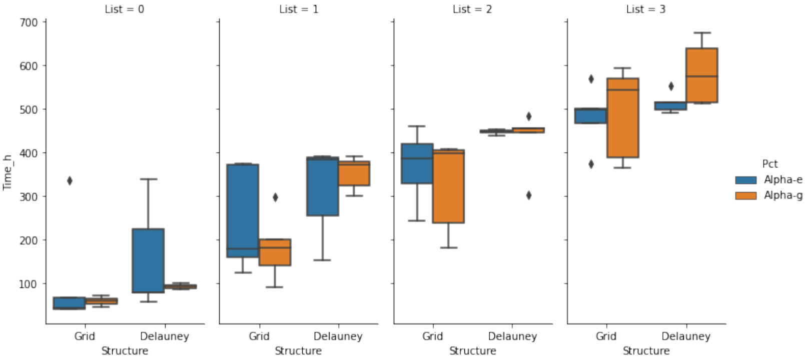



Figure: Matheuristic running time

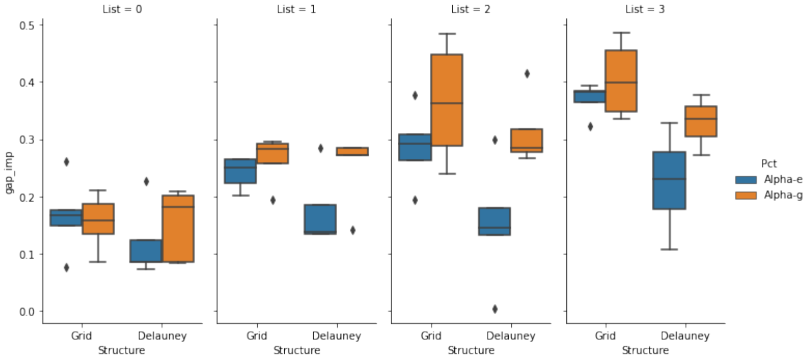



Figure: Matheuristic improved gap

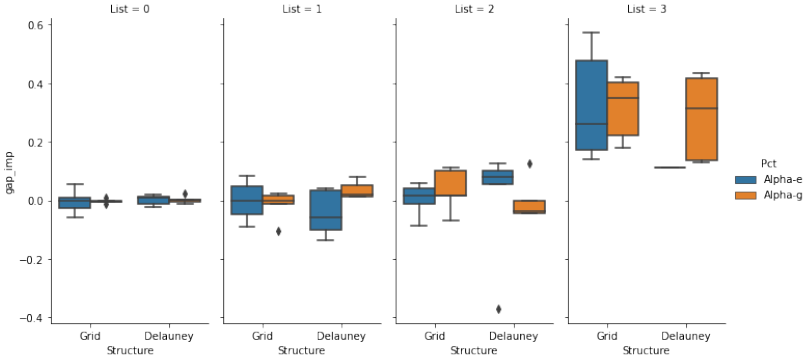



Figure: Improved gap of MTZ formulation with and without initialization

# Experiment 3: Solving the NMDRPG

We generate five instances for each number $|\mathcal{G}| \in \{5, 10\}$ of target graphs. For each instance:

- The same percentage of graphs (20%) has respectively 4, 6, 8, 10 and 12 nodes.
- $v_D = 3v_M$.
- $\alpha^g$ and $\alpha^{e_g}$ randomly generated in $[0, 1]$.
- Three different network structures are considered:
  - Graph of 6 nodes with a tree structure with origin of the path of the base vehicle different from the destination.
  - Complete graph of 4 vertices with origin of the path of the base vehicle different from the destination.
  - Star graphs of 7 nodes representing the mothership network, where the origin coincides with the destination and it is located at the centre of the star.

The experiment consists on:

- Running the MTZ and Stages formulation for (NMDRPG) with and without initial solution provided by the matheuristic.
- Using Gurobi 9.03.
- Time Limit: 2 hour.

# Experiment 3: Network Structures

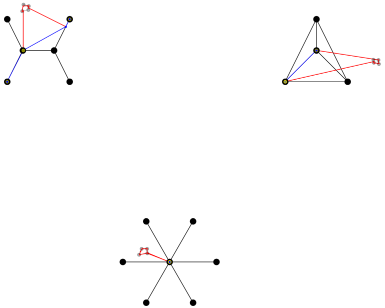


Figure: Network Structures

# Experiment 3: Results

Table: Comparison between formulations of the (NMDRPG)

| | Net Struct | 1 | | 2 | | 3 | |
|---|---|---|---|---|---|---|---|
| List | % | Stages | MTZ | Stages | MTZ | Stages | MTZ |
| 0 | e | 0.89 | 0.33 | 0.88 | 0.24 | 0.87 | 0.39 |
| | g | 0.86 | 0.29 | 0.89 | 0.18 | 0.90 | 0.42 |
| 1 | e | 0.92 | 0.43 | 0.92 | 0.33 | 0.92 | 0.46 |
| | g | 0.91 | 0.36 | 0.92 | 0.23 | 0.92 | 0.39 |

Table: Comparison between exact resolution with and without initialization of the (NMDRPG)

| | Net Struct | 1 | | | 2 | | | 3 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| List | % | % Gap (i) | T_h | % Gap (wi) | % Gap (i) | T_h | % Gap (wi) | % Gap (i) | T_h | % Gap (wi) |
| 0 | e | 0.32 | 109.96 | 0.33 | 0.24 | 207 | 0.24 | 0.39 | 177.57 | 0.39 |
| | g | 0.30 | 110.92 | 0.29 | 0.18 | 163.36 | 0.18 | 0.45 | 149.68 | 0.42 |
| 1 | e | 0.48 | 1030.64 | 0.43 | 0.39 | 802.3 | 0.33 | 0.53 | 770.05 | 0.46 |
| | g | 0.33 | 479.36 | 0.36 | 0.35 | 639.09 | 0.23 | 0.42 | 689.51 | 0.39 |

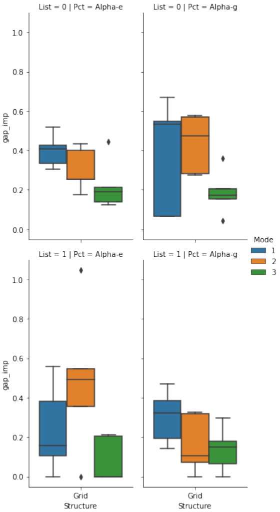



Figure: Matheuristic improved gap for the NMDRPG

# Contents

- Problem Description
- Formulations
- Example
- Strengthening the formulation
- Matheuristic
- Computational Experiments
- Case Study

# Problem Description

- There is one mothership and a fleet of drones in coordination to visit a set of target graphs, whose locations are given. The drone has a limited endurance $N^d$ in this case.
- For each graph $g \in \mathcal{G}$ the assigned drone performs the following task:
  1. It is launched from the current mothership location (to be determined).
  2. It flies to the graph $g$ that has to be visited.
  3. It traverses the required edges of graph $g$.
  4. It returns to the current position of the mothership (to be determined).

  We assume wlog that the mothership and the drone do not need to arrive at each rendezvous location at the same time: the fastest arriving vehicle may wait for the other at the rendezvous point.

# Problem Description

It is required to determine:

- The tour of the mothership starting at *orig*, deciding the different launching and rendezvous points, and returning to *dest*.
- The optimal assignment of drones for visiting graphs in a given stage $t$.
- The order of visits of the target graphs followed by the drone, determining the corresponding launching and rendezvous points of the drone on each visited graph.
- The tour followed by the drone on each target graph $g \in \mathcal{G}$.

# Parameters of the AMMDRPG

The known parameters of the problem are:

- *orig*: coordinates of the point defining the origin of the mothership path (or tour).
- *dest*: coordinates of the point defining the destination of the mothership path (or tour).
- $\mathcal{G}$: set of the target graphs.
- $g = (V_g, E_g)$: set of nodes and edges of each target graph $g \in \mathcal{G}$.
- $\mathcal{L}(e_g)$: length of edge $e$ of graph $g \in \mathcal{G}$.
- $B^{e_g}, C^{e_g}$: coordinates of the endpoints of edge $e$ of graph $g \in \mathcal{G}$.
- $\alpha^{e_g}$: percentage of edge $e$ of graph $g \in \mathcal{G}$ that must be visited.
- $\alpha^g$: percentage of graph $g \in \mathcal{G}$ that must be visited.
- $v_D$: drone speed.
- $v_M$: mothership speed.
- $N^d$: drone endurance.
- $M$: big-M constant.

# Binary and Integer Decision Variables for the AMMDRPG

- $\mu^{e_g} \in \{0,1\}\ \forall e_g \in E_g\ (g \in \mathcal{G})$: equal to 1 if edge $e$ of graph $g$ (or a portion of it) is visited by the drone, and 0 otherwise.
- $entry^{e_g} \in \{0,1\}\ \forall e_g \in E_g\ (g \in \mathcal{G})$: auxiliary binary variable used for linearizing expressions.
- $u^{e_g td} \in \{0,1\}\ \forall e_g \in E_g\ (g \in \mathcal{G})\ \forall t \in \mathcal{T}$: equal to 1 if the drone $d$ enters in graph $g$ by the edge $e_g$ at stage $t$, 0 otherwise.
- $z^{e_g e_g^{'}} \in \{0,1\}\ \forall e_g, e_g' \in E_g\ (g \in \mathcal{G})$: equal to 1 if the drone goes from $e_g$ to $e_g^{'}$, 0 otherwise.
- $v^{e_g td} \in \{0,1\}\ \forall e_g \in E_g\ (g \in \mathcal{G})\ \forall t \in \mathcal{T}$: equal to 1 if the drone $d$ exits from graph $g$ by $e_g$ at stage $t$, 0 otherwise.
- $s^{e_g},\ \forall e_g \in E_g\ (g \in \mathcal{G})$: integer non negative variable representing the order of visit of edge $e$.

# Continuous Decision Variables for the AMMDRPG

**Location variables**

- $\rho^{e_g} \in [0, 1]$ and $\lambda^{e_g} \in [0, 1]$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: defining the entry and exit points on $e_g$.
- $\nu_{\min}^{e_g}$ and $\nu_{\max}^{e_g} \in [0, 1] \forall e_g \in E_g$ $(g \in \mathcal{G})$: auxiliary variables used for linearizing expressions.
- $x_L^t$ $\forall t \in \mathcal{T}$: coordinates representing the point where the mothership launches the drone at stage $t$.
- $x_R^t$ $\forall t \in \mathcal{T}$: coordinates representing the point where the mothership retrieves the drone at stage $t$.
- $R^{e_g}$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: coordinates representing the entry point on edge $e$ of graph $g$.
- $L^{e_g}$ $\forall e_g \in E_g$ $(g \in \mathcal{G})$: coordinates representing the exit point on edge $e$ of graph $g$.

# Continuous Decision Variables for the AMMDRPG

## Distance variables

- $d_L^{e_g td} \geq 0, \ \forall e_g \in E_g \ (g \in \mathcal{G}) \ \forall t \in \mathcal{T}$: representing the distance travelled by the drone $d$ from the launching point $x_L^t$ on the mothership at stage $t$ to the first visiting point $R^{e_g}$ on $e_g$.
- $d^{e_g e_g'} \geq 0, \ \forall e_g, e_g' \in E_g \ (g \in \mathcal{G})$: representing the distance travelled by the drone from the launching point $L^{e_g}$ on $e_g$ to the rendezvous point $R^{e_g'}$ on $e_g'$.
- $d^{e_g} \geq 0, \ \forall e_g \in E_g \ (g \in \mathcal{G})$: representing the distance travelled by the drone from the rendezvous point $R^{e_g}$ to the launching point $L^{e_g}$ on $e_g$.
- $d_R^{e_g td} \geq 0 \ \forall e_g \in E_g \ (g \in \mathcal{G}) \ \forall t \in \mathcal{T}$: representing the distance travelled by the drone $d$ from the last visiting point $L^{e_g}$ on $e_g$ to the rendezvous point $x_R^t$ on the mothership at stage $t$.
- $d_{LR}^t \geq 0 \ \forall t \in \mathcal{T}$: representing the distance travelled by the mothership from the launching point $x_L^t$ to the rendezvous point $x_R^t$ at stage $t$.
- $d_{RL}^t \geq 0 \ \forall t \in \mathcal{T}$: representing the distance travelled by the mothership from the rendezvous point $x_R^t$ at stage $t$ to the launching point $x_L^{(t+1)}$ at the stage $t+1$.

# Visit of the graphs

We have considered two modes of visit to the target graphs $g \in \mathcal{G}$ that must be represented by their corresponding constraints:

- Visiting a percentage $\alpha^{e_g}$ of each edge $e_g$ which can be modeled by:

$$|\lambda^{e_g} - \rho^{e_g}|\mu^{e_g} \geq \alpha^{e_g}, \quad \forall e_g \in E_g. \tag{$\alpha$-E}$$

- Visiting a percentage $\alpha^g$ of the total length $\mathcal{L}(g)$ of the graph $g$ modeled by:

$$\sum_{e_g \in E_g} \mu^{e_g}|\lambda^{e_g} - \rho^{e_g}|\mathcal{L}(e_g) \geq \alpha^g \mathcal{L}(g). \tag{$\alpha$-G}$$

# Visit of the graphs

In both cases the corresponding constraints are nonlinear. For each edge $e_g$, we linearize the absolute value constraint ($\alpha$-E) by introducing a binary variable:

$$\mu^{e_g}|\rho^{e_g}-\lambda^{e_g}| \geq \alpha^{e_g} \Longleftrightarrow \left\{ \begin{array}{rcl} \rho^{e_g}-\lambda^{e_g} & = & \nu^{e_g}_{\text{max}}-\nu^{e_g}_{\text{min}} \\ \nu^{e_g}_{\text{max}} & \leq & 1-\text{entry}^{e_g} \\ \nu^{e_g}_{\text{min}} & \leq & \text{entry}^{e_g}, \\ \mu^{e_g}(\nu^{e_g}_{\text{max}}+\nu^{e_g}_{\text{min}}) & \geq & \alpha^{e_g}. \end{array} \right. \qquad (\alpha\text{-E})$$

The linearization of ($\alpha$-G) is similar to ($\alpha$-E) and only requires changing the last inequality in ($\alpha$-E) for

$$\sum_{e_g\in E_g} \mu^{e_g}(\nu^{e_g}_{\text{max}}+\nu^{e_g}_{\text{min}})\mathcal{L}(e_g) \geq \alpha_g\mathcal{L}(g). \qquad (\alpha\text{-G})$$

# Modeling the Drone Route

$$\sum_{g\in\mathcal{G}}\sum_{e_g\in E_g}\sum_{d\in\mathcal{D}} u^{e_g td} \leq 1, \qquad \forall t\in\mathcal{T}, \tag{56}$$

$$\sum_{g\in\mathcal{G}}\sum_{e_g\in E_g}\sum_{d\in\mathcal{D}} v^{e_g td} \leq 1, \qquad \forall t\in\mathcal{T}, \tag{57}$$

$$\sum_{e_g\in E_g}\sum_{t\in\mathcal{T}}\sum_{d\in\mathcal{D}} u^{e_g td} = 1, \qquad \forall g\in\mathcal{G}, \tag{58}$$

$$\sum_{e_g\in E_g}\sum_{t\in\mathcal{T}}\sum_{d\in\mathcal{D}} v^{e_g td} = 1, \qquad \forall g\in\mathcal{G}, \tag{59}$$

$$\sum_{e_g\in E_g} u^{e_g td} = \sum_{e_g\in E_g} v^{e_g td}, \qquad \forall g\in\mathcal{G}, \forall t\in\mathcal{T}, \forall d\in\mathcal{D}, \tag{60}$$

$$\sum_{t\in\mathcal{T}}\sum_{d\in\mathcal{D}} u^{e_g td} + \sum_{e'_g\in E_g} z_g^{e'_g e_g} = \mu^{e_g}, \qquad \forall e_g\in E_g : g\in\mathcal{G}, \tag{61}$$

$$\sum_{t\in\mathcal{T}}\sum_{d\in\mathcal{D}} v^{e_g td} + \sum_{e'_g\in E_g} z_g^{e_g e'_g} = \mu^{e_g}, \qquad \forall e_g\in E_g : g\in\mathcal{G}. \tag{62}$$

# Subtour elimination inside the graph

To prevent the existence of subtours within each graph $g \in \mathcal{G}$ that the drone must visit:

- One can add the Miller-Tucker-Zemlin constraints, given by:

$$
\begin{aligned}
s^{e_g} - s^{e'_g} + |E_g| z^{e_g e'_g} &\leq |E_g| - 1, && \forall e_g \neq e'_g \in E_g, && (\mathrm{MTZ}_1)\\
0 \leq s^{e_g} &\leq |E_g| - 1, && \forall e_g \in E_g, && (\mathrm{MTZ}_2)
\end{aligned}
$$

- It is also possible to include the subtour elimination constraints:

$$
\sum_{e_g, e'_g \in S} z_g^{e_g e'_g} \leq |S| - 1, \quad \forall S \subset E_g : g \in \mathcal{G}. \qquad (\mathrm{SEC})
$$

# Distance constraints

To account for the different distances among the decision variables of the model we need to set the continuous variables $d_L^{e_g t}$, $d^{e_g}$, $d^{e_g e_g'}$, $d_R^{e_g t}$, $d_{RL}^t$ and $d_{LR}^t$. This can be done by means of the following constraints:

$$\begin{aligned}
\|x_L^t - R^{e_g}\| &\leq d_L^{e_g td}, && \forall e_g : g \in \mathcal{G}, \forall t \in \mathcal{T}, \forall d \in \mathcal{D}, && (\text{DIST}_1\text{-t})\\
\|R^{e_g} - L^{e_g}\| &\leq d^{e_g}, && \forall e_g : g \in \mathcal{G}, && (\text{DIST}_2\text{-t})\\
\|R^{e_g} - L^{e_g'}\| &\leq d^{e_g e_g'}, && \forall e_g \neq e_g' \in E_g : g \in \mathcal{G}, && (\text{DIST}_3\text{-t})\\
\|L^{e_g} - x_R^t\| &\leq d_R^{e_g td}, && \forall e_g : g \in \mathcal{G}, \forall t \in T, \forall d \in \mathcal{D}, && (\text{DIST}_4\text{-t})\\
\|x_R^t - x_L^{t+1}\| &\leq d_{RL}^t, && \forall t \in \mathcal{T}, && (\text{DIST}_5\text{-t})\\
\|x_L^t - x_R^t\| &\leq d_{LR}^t, && \forall t \in \mathcal{T}. && (\text{DIST}_6\text{-t})
\end{aligned}$$

# Coordination constraint

The coordination between the drones and the mothership must ensure that the time spent by the drone $d$ to visit the graph $g$ at the stage $t$ is less than or equal to the time that the mothership needs to move from the launching point to the retrieving point during the stage $t$. To this end, we need to define the following coordination constraint for each graph $g \in \mathcal{G}$, stage $t \in \mathcal{T}$ and drone $d \in \mathcal{D}$:

$$\frac{1}{v_D}\left(\sum_{e_g \in E_g} u^{e_g td} d_L^{e_g td} + \sum_{e_g, e'_g \in E_g} z^{e_g e'_g} d^{e_g e'_g} + \sum_{e_g \in E_g} \mu^{e_g} d^{e_g} + \sum_{e_g \in E_g} v^{e_g td} d_R^{e_g td}\right) \leq \frac{d_{LR}^t}{v_M} + M\Big(1 - \sum_{e_g \in E_g} u^{e_g td}\Big). \tag{DCW}$$

# Endurance constraint

Note that, since the objective function of this problem minimizes the right-hand-side of (DCW), this constraint will become an equality and we can model the time capacity constraint for a particular stage $t \in \mathcal{T}$ by limiting the distance traveled by the mothership for this task $t$:

$$d_{LR}^t \leq N^d. \tag{Capacity}$$

# Setting the origin and the destination

Eventually, we have to impose that the tour of the mothership, together with the drone, starts from the origin *orig* and ends at the destination *dest*. To this end, we define the following constraints:

$$x_L^0 = orig, \tag{ORIG$_1$}$$

$$x_R^0 = orig, \tag{ORIG$_2$}$$

$$x_L^{|\mathcal{G}|+1} = dest, \tag{DEST$_1$}$$

$$x_R^{|\mathcal{G}|+1} = dest. \tag{DEST$_2$}$$

# Formulation for the AMMDRPG

$$
\begin{array}{rl}
\min & \displaystyle\sum_{t\in\mathcal{T}}(d_{RL}^t + d_{LR}^t) \\
\text{s.t.} & (56) - (62), \\
& (\text{MTZ}_1) - (\text{MTZ}_2) \text{ or } (\text{SEC}), \\
& (\alpha\text{-E}) \text{ or } (\alpha\text{-G}), \\
& (\text{DCW}), (\text{Capacity}), \\
& (\text{DIST}_1\text{-t}) - (\text{DIST}_6\text{-t}), \\
& (\text{ORIG}_1) - (\text{DEST}_2).
\end{array}
$$

# Example

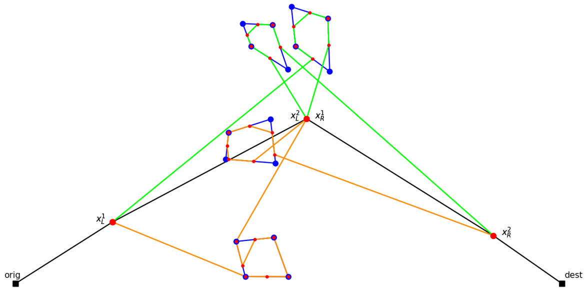



Figure: Example illustrating the meaning of the launching (L) and retrieving (R) points.

# The AMMDRPG without synchronisation

We consider a variant of the model presented before, in which we assume that the mothership can retrieve one drone in a stage different from the one in which it has been launched. That is, the mothership can move to another point to launch a new drone without having retrieved the one that was launched before.

To deal with this extension, we do not need to define new variables since it is possible to use the same variables that were used in the previous model.

First of all, constraint (60) must be changed to:

$$\sum_{e_g \in E_g} u^{e_g td} - \sum_{e_g \in E_g} \sum_{t' \geq t} v^{e_g t' d} = 0, \quad \forall g \in \mathcal{G}, \forall t \in \mathcal{T}, \forall d \in \mathcal{D}. \tag{63}$$

# The AMMDRPG without synchronisation

Moreover, the coordination constraint (DCW) must be modified to consider the general case in which the stages of launching and retrieving can be different.
For each $t_1 < t_2$ and $\forall g \in \mathcal{G}, \forall d \in \mathcal{D}$:

- Time spent by the drone $d$ to visit the graph $g$:

$$T_D = \frac{1}{v_D}\left(\sum_{e_g \in E_g} u^{e_g t_1 d} d_L^{e_g t_1 d} + \sum_{e_g, e_g' \in E_g} z^{e_g e_g'} d^{e_g e_g'} + \sum_{e_g \in E_g} \mu^{e_g} d^{e_g} + \sum_{e_g \in E_g} v^{e_g t_2 d} d_R^{e_g t_2 d}\right).$$

- Time spent by the mothership:

$$T_M = \frac{\sum_{t=t_1}^{t_2} d_{LR}^t}{v_M} + \frac{\sum_{t=t_1}^{t_2-1} d_{RL}^t}{v_M} + M(2 - \sum_{e_g \in E_g} u^{e_g t_1 d} - \sum_{e_g \in E_g} v^{e_g t_2 d}).$$

The time spent by the drone must be lower than the one spent by the mothership:

$$T_D \leq T_M. \qquad \text{(DCW-NS)}$$

# Link between AMMDRPG with and without synchronisation

**Theorem**

*Let $x_L^1$, $x_L^2$ (resp. $x_R^1$, $x_R^2$) be the launching (resp. rendezvous) points associated to the visit of the target points $P_1$ and $P_2$. If there exist two points $x_L$ and $x_R$ verifying*

$$\left\{\begin{array}{lcl} \dfrac{\|x_L - x_R\|}{v_C} & \leq & \dfrac{\|x_L - P_1\| + \|P_1 - x_R\|}{v_D}, \\ \dfrac{\|x_L - x_R\|}{v_C} & \leq & \dfrac{\|x_L - P_2\| + \|P_2 - x_R\|}{v_D}, \\ \dfrac{\|x_L - x_R\|}{v_C} & \leq & N^d, \\ \|x_L - x_R\| & \leq & \|x_L^1 - x_L^2\| + \|x_L^2 - x_R^1\| + \|x_R^1 - x_R^2\|, \end{array}\right.$$

*then the contribution of this partial route to the optimal objective value will be the same in both models.*

# Valid inequalities for the AMMDRPG

- Let $\beta^t$ be a binary variable that assumes the value one if all the target graphs are visited when the operation $t$ begins, and zero, otherwise.

$$\beta^t \leq \beta^{t+1}, \text{ for all } t = 1, \ldots, |\mathcal{G}| - 1. \qquad \text{(Monotonicity)}$$

- Let $k^t$ denote the number of graphs that are visited in the stage $t$.

$$k^t = \sum_{e_g \in g : g \in \mathcal{G}} \sum_{d \in \mathcal{D}} u^{e_g t d}.$$

# Valid inequalities for the AMMDRPG

## Valid Inequalities I

If $\beta^t$ is one, the entire set of graphs in $\mathcal{G}$ must have been visited before the stage $t$:

$$\sum_{t'=1}^{t-1} k^{t'} \geq |\mathcal{G}|\beta^t. \tag{VI-1}$$

## Valid Inequalities II

It is not permitted to have a stage $t$ without any operation if some graphs are still to be visited:

$$k^t \geq 1 - \beta^t. \tag{VI-2}$$

# Valid inequalities for the AMMDRPG

## Valid Inequalities III and IV

We are assuming that drones are indistinguishable, we can assume that given an arbitrary order on them, we always assign drones to operations in that given order. For all $t = 1, \ldots, |\mathcal{G}|$ :

$$\sum_{e_g \in \mathcal{G}} u^{e_g td} \leq \sum_{e_g : g \in \mathcal{G}} u^{e_g td-1}, \ \forall d = 2, \ldots |\mathcal{D}|, \qquad \text{(VI-3)}$$

$$\sum_{e_g \in \mathcal{G}} v^{e_g td} \leq \sum_{e_g : g \in \mathcal{G}} v^{e_g td-1}, \ \forall d = 2, \ldots |\mathcal{D}|. \qquad \text{(VI-4)}$$

# Adjusting the Big-M constants

The model that we have proposed includes big-M constants. We have defined different big-M constants along this work. In order to strengthen the formulations we provide tight upper and lower bounds for those constants:

## Big-M constants bounding $d_L^{e_g td}$ and $d_R^{e_g td}$

We model the product $d_L^{e_g td} u_L^{e_g td}$ by defining the auxiliar non-negative continuous variables $p_L^{e_g td}$ (resp. $p_R^{e_g td}$) and including the following constraints:

$$
\begin{aligned}
p_L^{e_g td} &\geq m_L^{e_g} u^{e_g td}, \\
p_L^{e_g td} &\leq d_L^{e_g} - M_L^{e_g td}(1 - u^{e_g t}).
\end{aligned}
$$

The best upper bound $M_L^{e_g td}$ or $M_R^{e_g td}$ that we can consider is the maximum distance between every pair of vertices of the graphs $g \in \mathcal{G}$:

$$M_R^{e_g td} = \max_{\{v \in V_g, v' \in V_{g'} : g, g' \in \mathcal{G}\}} \|v - v'\| = M_L^{e_g td}.$$

# Adjusting the Big-M constants

The model that we have proposed includes big-M constants. We have defined different big-M constants along this work. In order to strengthen the formulations we provide tight upper and lower bounds for those constants:

## Big-M constants bounding $d^{e_g e'_g}$

We model the product $d^{e_g e'_g} z^{e_g e'_g}$ by defining the auxiliar non-negative continuous variables $p^{e_g e'_g}$ and including the following constraints:

$$p^{e_g e'_g} \geq m^{e_g e'_g} d_{RL}^{gg'},$$
$$p^{e_g e'_g} \leq d^{e_g e'_g} - M^{e_g e'_g}(1 - z^{e_g e'_g}).$$

Since we are taking into account the distance between two edges $e = (B^{e_g}, C^{e_g})$, $e' = (B^{e'_g}, C^{e'_g}) \in E_g$, the maximum and minimum distances between their vertices give us the upper and lower bounds:

$$M^{e_g e'_g} = \max\{\|B^{e_g} - C^{e'_g}\|, \|B^{e_g} - B^{e'_g}\|, \|C^{e_g} - B^{e'_g}\|, \|C^{e_g} - C^{j_g}\|\},$$
$$m^{e_g e'_g} = \min\{\|B^{e_g} - C^{e'_g}\|, \|B^{e_g} - B^{e'_g}\|, \|C^{e_g} - B^{e'_g}\|, \|C^{e_g} - C^{e'_g}\|\}.$$

# Adjusting the Big-M constants

The model that we have proposed includes big-M constants. We have defined different big-M constants along this work. In order to strengthen the formulations we provide tight upper and lower bounds for those constants:

## Big-M constants bounding the total distance made by a drone in a stage

To link the drone operation with the trip followed by the mothership, we have defined the constraint (DCW) that includes another big-M constant.

To obtain an upper bound on $M$ we add to the length of the graph $\mathcal{L}(g)$ the big-Ms computed for $u^{e_g td}$ and $v^{e_g td}$, namely $M_L^{e_g td}$ and $M_R^{e_g td}$, respectively, and the maximum distance that can be traveled by the drone to move from one edge to another one. This results in a valid value for this $M$ constant:

$$M = \mathcal{L}(g) + M_L^{e_g td} + M_R^{e_g td} + \sum_{e_g, e'_g \in E_g} M^{e_g e'_g}.$$

# Matheuristic for the AMMDRPG

**Pseudo-code of this algorithm:**

STEP 1 (First entry and last exit points for each target graph)

Compute the route on each target graph $g \in \mathcal{G}$. Let $L^{e_g}$ and $R^{e_g^{'}}$ be the pair of entry and exit points on $g$ closest to the origin and let $\mathcal{L}(e_g, e_g^{'})$ be the associated length computed as the sum of the distances travelled by the drone to visit the graph $g$, excluding the distance between $L^{e_g}$ and $R^{e_g^{'}}$.

# Matheuristic for the AMMDRPG

## Pseudo-code of this algorithm:

STEP 2 (Clustering procedure)

Initialization: set $it = 1$, define one cluster for each target graph and set $nit = 1$.

Select randomly two clusters $K_i$ and $K_j$ (where $i < j$).

Check if the number of graphs belonging to the union of $K_i$ and $K_j$ is less than the number of available drones $n_D$.

If this condition is satisfied:

search for point $P$ satisfying the following capacity constraint:

$$\frac{d(P, R^{e_g}) + \mathcal{L}(e_g, e_g^{'}) + d(L^{e_g^{'}}, P)}{v_D} \leq N^d, \quad \forall R^{e_g}, L^{e_g^{'}} \in K_i, \ K_j.$$

If such a point exists, merge the two clusters and label the new one as $K_i$.

Set $nit = nit + 1$.

Repeat the same procedure on the new cluster structure while $nit < maxit$.

# Matheuristic for the AMMDRPG

**Pseudo-code of this algorithm:**

STEP 3 (Computation of Reference Points) Compute a reference point for each cluster generated at STEP 2. This computation seeks for the minimization of the distance between each pair of reference points and the distance between them and the origin, always imposing that the (Capacity) constraint is satisfied.

STEP 4 (Setting the order of visits to the graphs: route via the reference points and the origin/destination points)
Compute the TSP of the mothership among the reference points of the clusters and let $\mathcal{L}(TSP)$ be the associated length.
Set $it = it + 1$.
if(it¡ maxseed) go to STEP 2
else go to STEP 5

STEP 5 (Solution of the AMMDRPG model fixing an initial partial solution)
Set the values of the binary variables $u^{e_g td}$ and $v^{e_g td}$ and solve the model to obtain a feasible solution.

# Experiment 1

We generate five instances for each setting:

- Number of target graphs $|\mathcal{G}| \in \{5, 10\}$.
- The same percentage of graphs (20%) has respectively 4, 6, 8, 10 and 12 nodes.
- Number of drones $n_D \in \{1, 2, 3\}$.
- Drone endurance $N^d \in \{20, 30, 40, 50, 60\}$.
- $v_D = 2v_M$.
- $\alpha^g$ and $\alpha^{e_g}$ randomly generated in $[0, 1]$.

The experiment consists on:

- Running the formulation for (AMMDRPG) with and without initial solution provided by the matheuristic.
- Using Gurobi 10.1.
- Time Limit: 2 hours.

# Experiment 1: Results

Table: Comparison between exact solution with and without initialization by the matheuristic solution

| —$\mathcal{G}$— | N$^d$ | v.t. | # drones | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1 | | | 2 | | | 3 | | |
| | | | %Gap (i) | TimeH | %Gap (wi) | %Gap (i) | TimeH | %Gap (wi) | %Gap (i) | TimeH | %Gap (wi) |
| 5 | 20 | e | 82,63 | 61,56 | 81,70 | 91,57 | 63,80 | 90,61 | 93,06 | 60,87 | 90,93 |
| | | g | 79,09 | 44,97 | 79,63 | 89,03 | 37,32 | 91,85 | 94,00 | 39,05 | 95,80 |
| | 30 | e | 82,70 | 65,21 | 80,17 | 85,14 | 64,41 | 82,21 | 91,90 | 63,34 | 90,12 |
| | | g | 75,80 | 55,77 | 71,19 | 84,36 | 44,36 | 88,27 | 91,02 | 44,59 | 91,39 |
| | 40 | e | 80,94 | 68,81 | 77,98 | 83,44 | 64,80 | 82,16 | 91,24 | 63,19 | 86,25 |
| | | g | 74,47 | 43,92 | 73,46 | 81,21 | 38,27 | 84,35 | 85,34 | 37,51 | 89,63 |
| | 50 | e | 76,87 | 66,67 | 74,41 | 81,12 | 63,86 | 79,57 | 85,11 | 63,51 | 86,16 |
| | | g | 70,58 | 43,42 | 66,90 | 80,96 | 43,98 | 88,84 | 80,49 | 44,35 | 82,81 |
| | 60 | e | 76,39 | 67,78 | 71,61 | 81,63 | 66,08 | 79,84 | 83,82 | 64,40 | 82,06 |
| | | g | 78,17 | 44,69 | 72,79 | 79,35 | 40,63 | 86,55 | 81,74 | 50,01 | 84,66 |
| 10 | 20 | e | 82,56 | 137,93 | 84,91 | 92,30 | 128,53 | - | 94,73 | 124,44 | - |
| | | g | 81,00 | 119,20 | 84,08 (2) | 89,88 | 83,50 | 96,64 (2) | 96,44 | 70,00 | 97,43 (3) |
| | 30 | e | 80,60 | 159,00 | 80,93 | 87,11 | 132,15 | 87,58 (3) | 94,56 | 127,35 | 92,85 (2) |
| | | g | 79,93 | 132,67 | 82,70 (1) | 86,32 | 80,29 | 86,13 (3) | 91,12 | 76,72 | 89,74 (1) |
| | 40 | e | 79,05 | 191,37 | 78,07 | 85,11 | 131,26 | 84,33 | 91,88 | 132,10 | 88,61 (1) |
| | | g | 80,23 | 115,00 | 79,64 | 87,31 | 68,39 | 84,57 (3) | 96,09 | 69,40 | 91,86 (1) |
| | 50 | e | 81,49 | 188,32 | 77,81 | 87,72 | 134,01 | 85,51 (1) | 92,68 | 132,82 | 90,79 (3) |
| | | g | 79,92 | 87,23 | 80,38 | 82,80 | 66,14 | 84,00 (3) | 92,48 | 64,94 | 91,96 (2) |
| | 60 | e | 83,79 | 155,27 | 81,57 | 85,91 | 131,94 | 82,96 (2) | 92,24 | 130,11 | 86,58 (3) |
| | | g | 77,57 | 97,89 | 78,46 | 86,94 | 76,53 | 88,29 (2) | 94,31 | 69,53 | 92,23 (3) |

# Experiment 1: Results

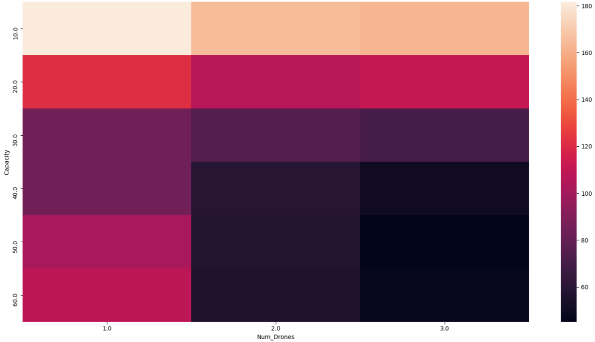



Figure: Heatmap of objective function values depending on number of drones and drone capacities. The darker the color intensity the smaller the objective value.

# Case Study: Los Patios de Córdoba

Considering the current COVID-19 restrictions, we focus on the problem of preventing and identifying possible concentrations of people during events such as popular or religious festivals. In particular we consider the Courtyards Festival of Cordoba (`https://patios.cordoba.es/es/`).


# Case Study

We run the model on this scenario starting from the initial solution provided by the matheuristic, where the 6 coloured paths reported in the map of Figure 25 represent the 6 target graphs to be visited, in this case inspected, by the fleet of drones. In addition, we suppose that the drones' speed is 100 km/h while that of the helicopter is 50 km/h aiming to minimize costs.

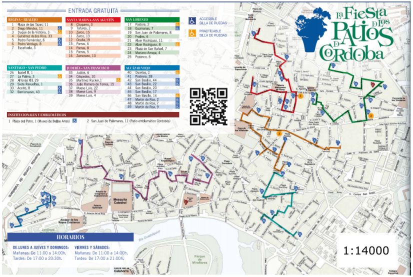



Figure: Map of the Courtyards Festival in Cordoba.

# Case Study

We assume that the fleet is composed by three drones with an endurance equal to 2 hours, and we impose that each target graph must be fully visited (inspected) and the origin of the mothership tour coincides with the destination and it is located in an area of the city where it is possible to assume the take-off and landing of an helicopter.

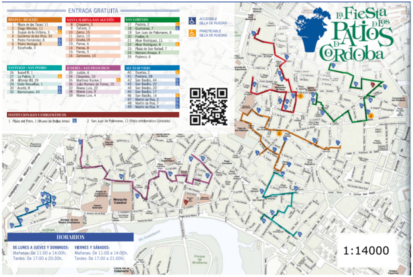



Figure: Map of the Courtyards Festival in Cordoba.

# Case Study: Solution

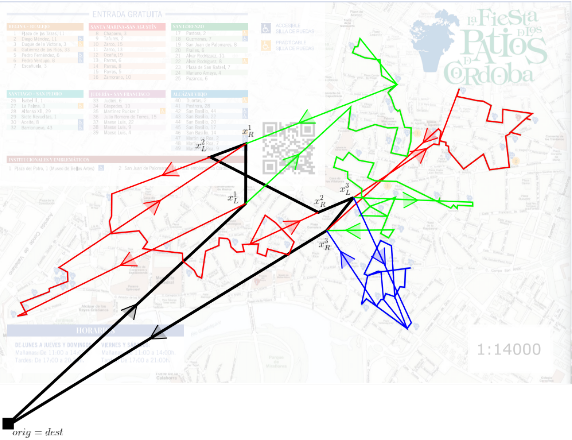



Figure: The complete solution.

# Acknowledgements

This research has been partially supported by Spanish Ministry of Education and Science/FEDER grant number MTM2016-74983-C02-(01-02), and projects FEDER-US-1256951, Junta de Andalucía P18-FR-1422, CEI-3-FQM331 and *NetmeetData*: Ayudas Fundación BBVA a equipos de investigación científica 2019.


Thank You!