Extending Coordination Models: The Mothership and Drone Routing Problem with Graphs

Carlos Valverde Martín

ISOLDE XV

Joint work with Justo Puerto and Lavinia Amorosi

Contents

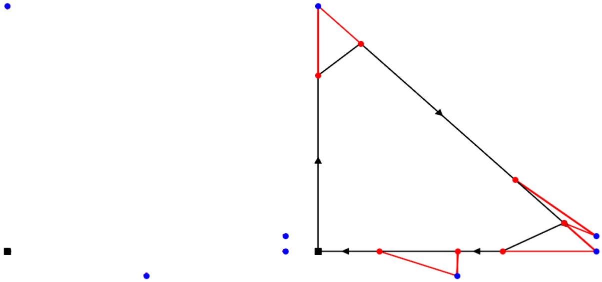
Literature Review

The number of references about ARPs with drones is limited.

- Tokekar, P., Hook, J. V., Mulla, D., and Isler, V. (2016). Sensor Planning for a Symbiotic UAV and UGV System for Precision Agriculture. IEEE Transactions on Robotics, 32(6):1498-1511.
- Campbell, J. F., Corberán, Á., Plana, I., and Sanchis, J. M. (2018). Drone arc routing problems. Networks, 72(4):543-559.
- Otto, A., Agatz, N., Campbell, J., Golden, B., and Pesch, E. (2018). Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey. Networks, 72:1-48.

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:

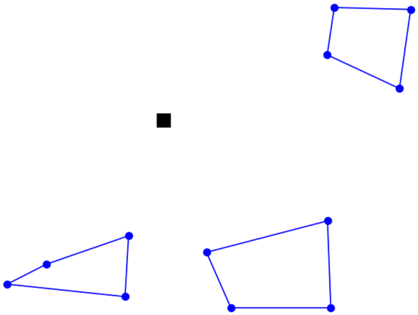
- The mothership vehicle can move freely on the continuous space (all terrain ground vehicle, boat on the water or aircraft vehicle), called the All terrain Mothership-Drone Routing Problem with Graphs (AMDRPG).
- The mothership can move on a road network (that is, it is a normal truck or van), called 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 $g \in \mathcal{G}$.
- α^{e_g}: percentage of edge e of graph $g \in \mathcal{G}$ that must be visited.
- α^{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 $t+1$.

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:

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

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 α^{e_g} of each edge e_g which can be modeled by:

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

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

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

Visit of the graphs

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

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

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

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

Modeling the Drone Route

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

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

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

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

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

$$\sum_{e'_g\in E_g} z_g^{e'_g e_g} + \sum_{t\in T} u^{e_g t} = \mu^{e_g}, \qquad \forall e_g\in E_g : g\in\mathcal{G}, \tag{6}$$

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

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:

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

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 $t \in T$:

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

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

$$\begin{aligned} \min \quad & \beta_D\Big(\sum_{g\in\mathcal{G}}\sum_{e_g\in E_g}\sum_{t\in T}(u^{e_g t} d_L^{e_g t} + v^{e_g t} d_R^{e_g t}) + \sum_{g\in\mathcal{G}}\sum_{e_g\in E_g} \mu^{e_g} d^{e_g} + \\ & + \sum_{g\in\mathcal{G}}\sum_{e_g, e'_g\in E_g} z^{e_g e'_g} d^{e_g e'_g}\Big) + \beta_M \sum_{t\in T}(d_{RL}^t + d_{LR}^t) \\ \text{s.t.} \quad & (1)-(7), \\ & (\text{MTZ}_1)-(\text{MTZ}_2) \text{ or } (\text{SEC}), \\ & (\alpha\text{-E}) \text{ or } (\alpha\text{-G}), \\ & (\text{DCW-t}), \\ & (\text{DIST}_1\text{-t})-(\text{DIST}_6\text{-t}), \\ & (\text{ORIG}_1)-(\text{DEST}_2). \end{aligned}$$

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.
- $w^{gg'} \in \{0,1\}\quad \forall g, g' \in \mathcal{G}$: 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.
- $\nu_{\min}^{e_g}$ 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 $\forall g \in \mathcal{G}$: 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 $g\prime$.

Distance Constraints

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

Modeling the Drone Route

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

$$\sum_{e_g \in E_g} u^{e_g} = 1, \qquad \forall g \in \mathcal{G}, \tag{8}$$

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

$$\sum_{e'_g \in E_g} z^{e'_g e_g} + u^{e_g} = \mu^{e_g}, \qquad \forall e_g \in E_g : g \in \mathcal{G}, \tag{10}$$

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

Modeling the Mothership Route

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

$$\sum_{g\in\mathcal{G}} w^{g0} = 0, \tag{12}$$

$$\sum_{g'\in\mathcal{G}} w^{(n_G+1)g'} = 0, \tag{13}$$

$$\sum_{g'\in\mathcal{G}\setminus\{g\}} w^{gg'} = 1, \qquad \forall g \in \mathcal{G}, \tag{14}$$

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

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

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

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

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

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 $g \in \mathcal{G}$:

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

Formulation for the AMDRPG-MTZ (resp. SEC)

$$\begin{aligned}
\min \quad & \beta_D\Big(\sum_{g\in\mathcal{G}}\sum_{e_g\in E_g}(u^{e_g}d_L^{e_g}+v^{e_g}d_R^{e_g})+\sum_{g\in\mathcal{G}}\sum_{e_g\in E_g}\mu^{e_g}d^{e_g}+ \\
& +\sum_{g\in\mathcal{G}}\sum_{e_g,e'_g\in E_g} z^{e_g e'_g}d^{e_g e'_g}\Big)+\beta_M\Big(\sum_{g\in\mathcal{G}} d_{LR}^{g}+\sum_{g,g'\in\mathcal{G}} d_{RL}^{gg'}w^{gg'}\Big) \\
\text{s.t.} \quad & (8)-(15), \\
& (\text{MTZ}_1)-(\text{MTZ}_2) \text{ or } (\text{SEC}), \\
& (\text{MTZ}_3)-(\text{MTZ}_6), \\
& (\alpha\text{-E}) \text{ or } (\alpha\text{-G}), \\
& (\text{DCW-g}), \\
& (\text{DIST}_1\text{-g})-(\text{DIST}_6\text{-g}), \\
& (\text{ORIG}_1)-(\text{DEST}_2).
\end{aligned}$$

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

$$\sum_{g,g'\in\mathcal{G}} w^{gg'} \leq |S|-1, \quad \forall S\subseteq\{1,\ldots,|\mathcal{G}|\}. \tag{16}$$

Example 1

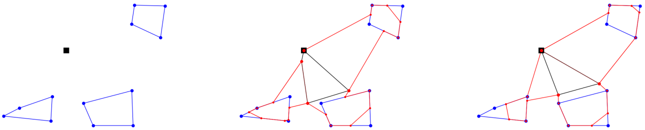
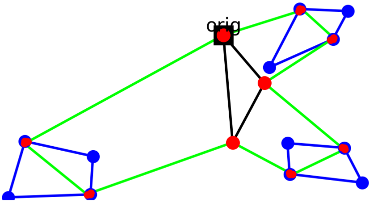
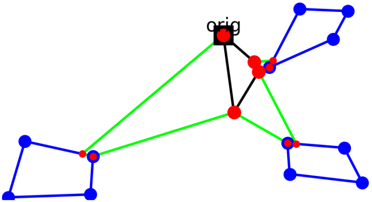


Figure: (a) Origin and target graphs, (b) Visit of α^{e_g}% of each edge, (c) Visit of α_g% of each graph.

Example 2: Percentage of each edge



Example 3: Percentage of each graph



Matheuristic for the AMDRPG

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 AMDRPG

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

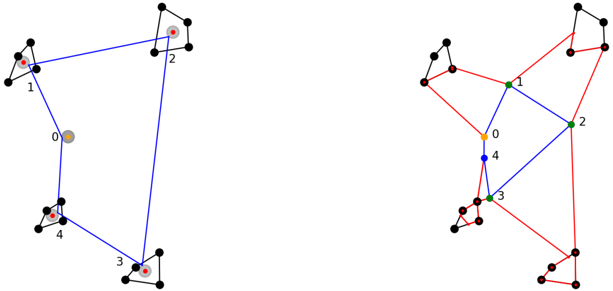


Figure: Illustrative Example

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

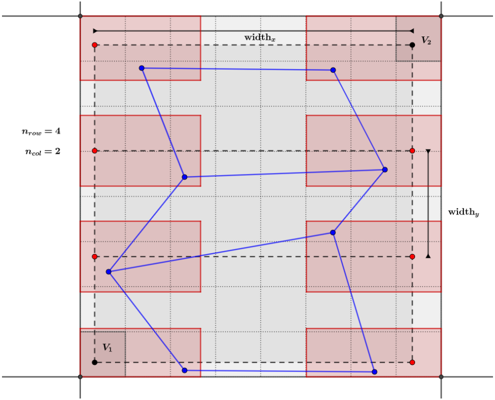


Figure: Example of generation of a grid graph

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.
- $v_D = 3v_M$.
- α^g and α^{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 hours.

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

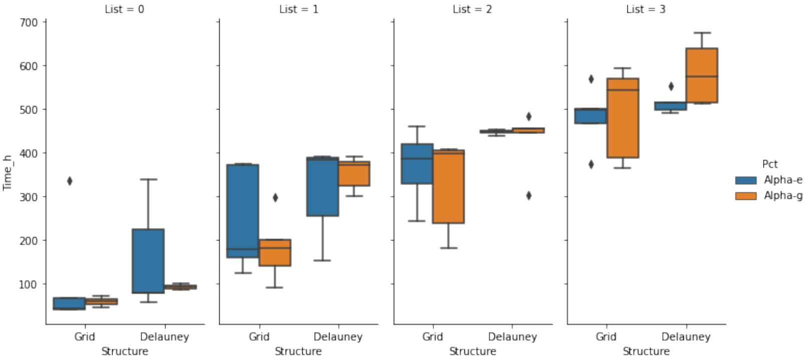


Figure: Matheuristic running time

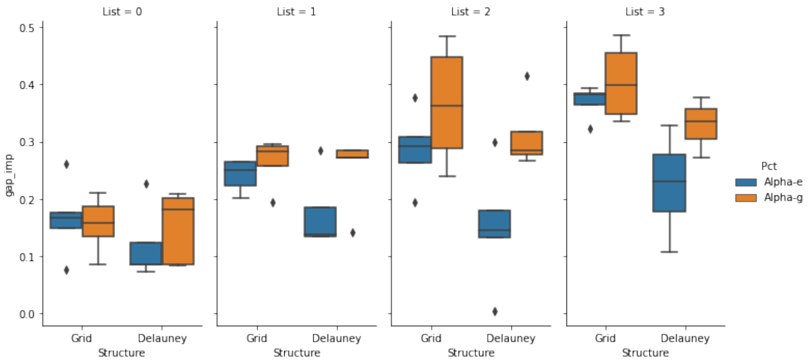


Figure: Matheuristic improved gap

Conclusions

- Coordination problem that arises between a mothership vehicle and a drone that must adjust their routes to minimize travel distances while visiting a set of targets modeled by graphs.
- Exact formulations for different versions of the problem depending on the constraints imposed to the mothership movement.
- The considered problem is rather hard and only small to medium size problems can be solved to optimality.
- Matheuristic algorithm that provides acceptable feasible solutions in short computing time.

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!

Bibliography I

- L. Amorosi, R. Caprari, T. Crainic, P. Dell'Olmo, and N. Ricciardi.
  CIRRELT-2020-17 An Integrated Routing-Scheduling Model for a Hybrid UAV-Based Delivery System.
  CIRRELT, 2020.

- L. Amorosi, L. Chiaraviglio, F. D'Andreagiovanni, and N. Blefari-Melazzi.
  Energy-efficient mission planning of UAVs for 5G coverage in rural zones.
  Proceedings of the IEEE International Conference on Environmental Engineering, EE 2018, 2018.

- L Amorosi, L Chiaraviglio, and J Galan-Jimenez.
  Optimal energy management of uav-based cellular networks powered by solar panels and batteries: Formulation and solutions.
  IEEE Access, Vol. 7:53698–53717, 2019.

Bibliography II

Víctor Blanco, Elena Fernández, and Justo Puerto.
Minimum spanning trees with neighborhoods: Mathematical programming formulations and solution methods.
European Journal of Operational Research, 262(3):863–878, nov 2017.

James F Campbell, Ángel Corberán, Isaac Plana, and José M Sanchis.
Drone arc routing problems.
Networks, 72(4):543–559, oct 2018.

James F Campbell, Don Sweeney, and Juan Zhang.
Strategic design for delivery with trucks and drones.
Computer Science, 2017.

John Gunnar Carlsson and Siyuan Song.
Coordinated logistics with a truck and a drone.
Management Science, 64(9):4052–4069, 2018.

Bibliography III

L. Chiaraviglio, L. Amorosi, N. Blefari-Melazzi, P. Dell'Olmo, C. Natalino, and P. Monti.
Optimal Design of 5G Networks in Rural Zones with UAVs, Optical Rings, Solar Panels and Batteries.
Proceedings of the 20th International Conference on Transparent Optical Networks, ICTON 2018, 2018.

Luca Chiaraviglio, Lavinia Amorosi, Nicola Blefari-Melazzi, Paolo Dell'olmo, Antonio Lo Mastro, Carlos Natalino, and Paolo Monti.
Minimum Cost Design of Cellular Networks in Rural Areas with UAVs, Optical Rings, Solar Panels, and Batteries.
IEEE Transactions on Green Communications and Networking, 3(4):901–918, 2019.

Bibliography IV

Luca Chiaraviglio, Lavinia Amorosi, Francesco Malandrino, Carla Fabiana Chiasserini, Paolo Dell'Olmo, and Claudio Casetti.
Optimal Throughput Management in UAV-based Networks during Disasters.
INFOCOM 2019 - IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2019, pages 307–312, 2019.

Joseph Y.J. Chow.
Dynamic UAV-based traffic monitoring under uncertainty as a stochastic arc-inventory routing policy.
International Journal of Transportation Science and Technology, 5(3):167–185, 2016.

Sung Hoon Chung, Bhawesh Sah, and Jinkun Lee.
Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions.
Computers & Operations Research, 123:105004, 2020.

Bibliography V

Luigi Di Puglia Pugliese and Francesca Guerriero.
Last-Mile Deliveries by Using Drones and Classical Vehicles BT - Optimization and Decision Science: Methodologies and Applications.
pages 557–565, Cham, 2017. Springer International Publishing.

Michael Dille and Sanjiv Singh.
Efficient aerial coverage search in road networks.
AIAA Guidance, Navigation, and Control (GNC) Conference, 2013.

Emanuele Garone, Roberto Naldi, Alessandro Casavola, and Emilio Frazzoli.
Cooperative mission planning for a class of carrier-vehicle systems.
Proceedings of the IEEE Conference on Decision and Control, pages 1354–1359, 2010.

Bibliography VI

Jaime Galan Jimenez, Luca Chiaraviglio, Lavinia Amorosi, and Nicola Blefari-Melazzi.
Multi-Period Mission Planning of UAVs for 5G Coverage in Rural Areas: A Heuristic Approach.
Proceedings of the 2018 9th International Conference on the Network of the Future, NOF 2018, pages 52–59, 2018.

Miao Li, Lu Zhen, Shuaian Wang, Wenya Lv, and Xiaobo Qu.
Unmanned aerial vehicle scheduling problem for traffic monitoring.
Computers and Industrial Engineering, 122:15–23, 2018.

Neil Mathew, Stephen L. Smith, and Steven L. Waslander.
Planning Paths for Package Delivery in Heterogeneous Multirobot Teams.
IEEE Transactions on Automation Science and Engineering, 12(4):1298–1308, 2015.

Bibliography VII

M Moshref-Javadi and S Lee.
Using drones to minimize latency in distribution systems.
In *IE Annual Conference, Proceedings*, pages 235–240, 2017.

Sergio Mourelo Ferrandez, Timothy Harbison, Troy Weber, Robert Sturges, and Robert Rich.
Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm.
Journal of Industrial Engineering and Management; Vol 9, No 2 (2016)DO - 10.3926/jiem.1929, apr 2016.

Hyondong Oh, Seungkeun Kim, Antonios Tsourdos, and Brian A White.
Coordinated road-network search route planning by a team of UAVs.
International Journal of Systems Science, 45(5):825–840, 2014.

Bibliography VIII

Hyondong Oh, H S Shin, A Tsourdos, B A White, and P Silson.
Coordinated Road Network Search for Multiple UAVs Using Dubins Path BT - Advances in Aerospace Guidance, Navigation and Control.
pages 55–65, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch.
Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey.
Networks, Vol. 72:1–48, 2018.

S. Poikonen and J.F. Campbell.
Future directions in drone routing research.
Networks, pages 1–11, 2020.

Stefan Poikonen and Bruce Golden.
Multi-visit drone routing problem.
Computers and Operations Research, 113:104802, 2020.

Bibliography IX

Justo Puerto and Carlos Valverde.
Routing for unmanned aerial vehicles: Touring dimensional sets.
arXiv, 2020.

Pratap Tokekar, Joshua Vander Hook, David Mulla, and Volkan Isler.
Sensor Planning for a Symbiotic UAV and UGV System for Precision Agriculture.
IEEE Transactions on Robotics, 32(6):1498–1511, 2016.

Angelo Trotta, Fabio D Andreagiovanni, Marco Di Felice, Enrico Natalizio, and Kaushik Roy Chowdhury.
When UAVs Ride A Bus: Towards Energy-efficient City-scale Video Surveillance.
Proceedings - IEEE INFOCOM, 2018-April:1043–1051, 2018.

Bibliography X

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, and et al.
SciPy 1.0: fundamental algorithms for scientific computing in Python.
Nature Methods, 17(3):261–272, 2020.

Tingxi Wen, Zhongnan Zhang, and Kelvin K L Wong.
Multi-objective algorithm for blood supply via unmanned aerial vehicles to the wounded in an emergency situation.
PLoS ONE, 11(5), 2016.

Temporary page!

LaTeX was unable to guess the total number of pages correctly. As the
was some unprocessed data that should have been added to the final
page this extra page has been added to receive it.
If you rerun the document (without altering it) this surplus page will g
away, because LaTeX now knows how many pages to expect for this
document.