The Traveling Postman Problem with Linear Barriers

JustoPuerto-and-CarlosValverdeMartin

1. Introduction

Routing problems are considered classical problems in the core of combinatorial optimization.Among them, the traveling salesman problem (TSP) is one of its more important representatives and has been studied extensively in many different forms. The problem is well-known to be NP-hard. Its analysis has led in the last decades to methodological advances and new optimization techniques that have allowed solving real life instances that few years ago were beyond solvable limits. The TSP has been studied in its geometric snd pure combinatorial forms because of its algorithmic and practical implications. One very interesting extension arises when we require to tour to visit a set of regions rather than points. In this version, the problem is APX-hard to approximate even for regions that are (intersecting) line segments. Therefore, making the problem very challenging although its applications in the delivery industry are important to model uncertainty in the positioning of the entities to be visited. On the other hand, the graphic TSP, where distances are measures by geodesics (shortest paths in the respective metric space), has also attracted the attention of many researchers since it models, in a natural way, TSP routes among forbidden barriers. One can easily find actual situations that fit naturally within the framework of combining both elements ...

Our goal in this paper is to deal with the TSP with neighbors and barriers. As commented above each one of these two elements makes the problem hard. Thus, mixing both together results in a new problem that, as far as we know, has not been addressed before but that has a lot of applications in the delivery industry. Moreover, it also has implications from the methodological point of view because introduces uncertainty in network representation and the issue of forbidden regions (barriers) in the solution space.

2. Description of the Problems

This section describes the two problems that are considered in this paper: the Hampered Shortest Path Problem with Neighborhoods H-SPPN and the Hampered Traveling Salesman Problem with Neighborhoods H-TSPN. In both models, we state the following assumptions:

A1 The line segments of $\mathcal{B}$ are located in general position, i.e., the endpoints of these segments are not aligned. Although it is possible to model the most general case, one can always to slightly modify one of the endpoints so that the segments are in general position.

A2 The line segments of $\mathcal{B}$ are opened, that is, it is possible that the drone visits endpoints of segments, but entering in the interior points of them is not allowed. Observe that without loss of generality, we can always slightly enlarge these segments to make them opened.

A3 If there are two barriers that have a common portion of them, it is only considered the smallest line segment that contains both barriers.

A4 There is no a rectilinear path to go from N_S to N_T. Otherwise, the problem becomes straightforward and the solution is the minimum distance between both neighborhoods.

2.1. *Description of the Hampered Shortest Path Problem with Neighborhoods*

In H-SPPN, we have a source neighborhood $N_S \subset \mathbb{R}^2$ and a target neighborhood $N_T \subset \mathbb{R}^2$, that we assume to be second order cone representable sets and a set $\mathcal{B}$ of line segments that play the role of barriers that the drone cannot cross.

The aim of the H-SPPN is to find the best pair of points $(P_S, P_T) \in N_S \times N_T$ in the source and target neighborhoods that minimize the length of the path that joins both points without crossing any barrier of $\mathcal{B}$ and assuming **A1-A4**. To state the model, we define the following sets:

- $V_S = \{P_S\}$: set composed by the point selected in the source neighborhood N_S.
- $V_{\mathcal{B}} = \{P_B^1, P_B^2 : B = \overline{P_B^1 P_B^2} \in \mathcal{B}\}$: set of vertices that come from the endpoints of barriers in the problem.
- $V_T = \{P_T\}$: set composed by the point selected in the target neighborhood N_T.
- $E_S = \{(P_S, P_B^i) : P_B^i \in V_{\mathcal{B}} \text{ and } \overline{P_S P_B^i} \cap B'' = \emptyset, \forall B'' \in \mathcal{B},\ i = 1, 2\}$: set of edges formed by the line segments that join the point selected in the source neighborhood and every endpoint in the barriers that do not cross any barrier in $\mathcal{B}$.
- $E_{\mathcal{B}} = \{(P_B^i, P_{B'}^j) : P_B^i, P_{B'}^j \in V_{\mathcal{B}} \text{ and } \overline{P_B^i P_{B'}^j} \cap B'' = \emptyset,\ \forall B'' \in \mathcal{B},\ i, j = 1, 2\}$: set of edges formed by the line segments that join two vertices of $V_{\mathcal{B}}$ and do not cross any barrier in $\mathcal{B}$.
- $E_T = \{(P_B^i, P_T) : P_B^i \in V_{\mathcal{B}} \text{ and } \overline{P_B^i P_T} \cap B'' = \emptyset, \forall B'' \in \mathcal{B},\ i = 1, 2\}$: set of edges formed by the line segments that join the point selected in the target neighborhood and every endpoint in the barriers that do not cross any barrier in $\mathcal{B}$.

At this point, we can define the graph $G = (V, E)$ induced by the barriers and neighborhoods, where $V = V_S \cup V_{\mathcal{B}} \cup V_T$ and $E = E_S \cup E_{\mathcal{B}} \cup E_T$. It is interesting to note that this graph can be split into two parts: a fixed graph $G_{\mathcal{B}} = (V_{\mathcal{B}}, E_{\mathcal{B}})$ whose edges can be computed by using the Remark 1 and the sets V_S, E_S, V_T and E_T that depend on where the points P_S and P_T are located as shown in Figure 1. The figures show how the graph G is generated. The blue dashed line segments represent the edges of E_S, the green ones, the edges of E_T and the red dashed lines, the edges of $E_{\mathcal{B}}$. A special case that can be remarked occurs when the neighborhoods are points. In that case, the induced graph is completely fixed and it is only necessary to find which edges are included by keeping in mind that there can not have crossings. This idea is exploited in Subsection 3.3.1.

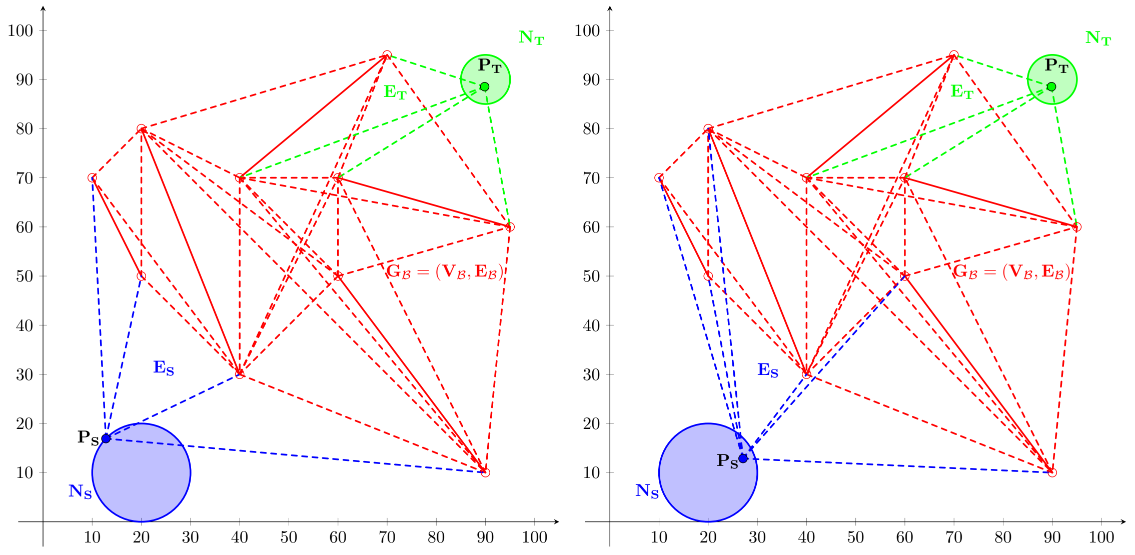


Figure 1: The construction of the graph $G = (N, V)$

2.2. Description of the Hampered Traveling Salesman Problem with Neighborhoods

The H-TSPN is an extension of the H-SPPN where neighborhood set $\mathcal{N}$ is considered to play the role of source and target in the H-SPPN and moreover, a set of given targets must be visited. The aim of the H-TSPN is to seek the shortest tour that visits each neighborhood $N \in \mathcal{N}$ exactly once without crossing any barrier $B \in \mathcal{B}$ and assuming again **A1-A4**.

To present our formulation for the H-TSPN, the graph induced by the endpoints of the barriers and the neighborhoods is different from the previous one for the H-SPPN. For its description, we introduce the following sets:

Discutir si abrir otra nueva sección con el modelo en el que las bolas se ven o añadir un comentario en esta subsección modificando solo la descripción de $E_{\mathcal{N}}$

- $V_{\mathcal{N}} = \{P_N : N \in \mathcal{N}\}$: set of the points selected in the neighborhoods $\mathcal{N}$ to be visited.
- $V_{\mathcal{B}} = \{P_B^1, P_B^2 : B = \overline{P_B^1 P_B^2} \in \mathcal{B}\}$: set of vertices that form the barriers of the problem.
- $E_{\mathcal{N}} = \{(P_N, P_B^i) : P_B^i \in V_{\mathcal{B}} \text{ and } \overline{P_N P_B^i} \cap B'' = \emptyset, \forall B'' \in \mathcal{B},\ i = 1, 2\}$: set of edges formed by the line segments that join the point selected in the neighborhoods of $\mathcal{N}$ and every endpoint in the barriers and do not cross any barrier in $\mathcal{B}$.
- $E_{\mathcal{B}} = \{(P_B^i, P_{B'}^j) : P_B^i, P_{B'}^j \in V_{\mathcal{B}} \text{ and } \overline{P_B^i P_{B'}^j} \cap B'' = \emptyset,\ \forall B'' \in \mathcal{B},\ i, j = 1, 2\}$: set of edges formed by the line segments that join two vertices of $V_{\mathcal{B}}$ and do not cross any barrier in $\mathcal{B}$.

Following the same idea as before, we set $G = (V, E)$ induced by the barriers and the neighborhoods, where $V = V_{\mathcal{N}} \cup V_{\mathcal{B}}$ and $E = E_{\mathcal{N}} \cup E_{\mathcal{B}}$.

Note that, in this case, it may be interesting to consider this problem without taking into account the assumption **A4**. This more general version is called the Hampered Traveling Salesman Problem with Visible Neighborhoods H-TSPVN. In this case, we do not require that the barriers separate neighborhoods completely, i.e., when going from one neighborhood to another is possible without crossing any barrier. The main difference lies in the description of edges for the graph induced by the neighborhoods and endpoints of barriers.

By taking the same approach, the sets that describe the graph in that case are:

- $V_{\mathcal{N}} = \{P_N : N \in \mathcal{N}\}$: set of points in the neighborhoods $\mathcal{N}$ that must be visited.
- $V_{\mathcal{B}} = \{P_B^1, P_B^2 : B = \overline{P_B^1 P_B^2} \in \mathcal{B}\}$: set of vertices that form the barriers of the problem.
- $V = V_{\mathcal{N}} \cup V_{\mathcal{B}}$.
- $E = \{(P, P') : P, P' \in V \text{ and } \overline{PP'} \cap B'' = \emptyset, \forall B'' \in \mathcal{B}\}$: set of edges formed by the line segments that join every pair of points in V that do not cross any barrier.

Figure 4 shows an example of the problems that are being considered. In the left picture, the blue neighborhood represents the source, the green the target and the red line segments show the barriers that the drone cannot cross. In the center picture, an instance of the H-TSPN is shown, where the neighborhood are balls and the barriers are, again, the red line segments. Finally, the picture in the right represents an instance of the H-TSPVN where the orange and blue balls can be joined by a rectilinear path.

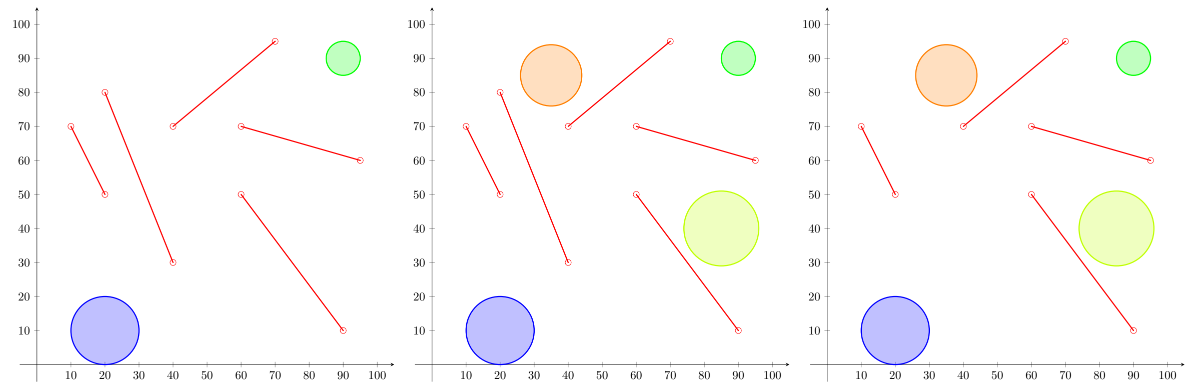


Figure 2: Problem data of the H-SPPN, H-TSPN and H-TSPVN

3. MINLP Formulations

This section introduces a Mixed Integer Non-Linear Programming formulation for the problems described in Section 2. First of all, we present the conic programming representation of the neighborhoods and distance. Then, we set the constraints that check if a segment is included in the set of edges E. Finally, the formulations for the H-SPPN, H-TSPN and H-TSPVN are described.

The would like to remark that having mathematical programming representation for the H-SPPN is very important even though we will prove in the following that this problem can be solved with a combinatorial algorithm in polynomial time. Indeed, we will need that type of representation to address some NP-hard problems that require computing shortest paths as building blocks, as for instance the location problem with barriers and neighbours and the H-TSPN. Therefore, we will start providing different mathematical programming formulations for the H-SPPN able to be embedded in more complex problems.

First of all, we introduce the decision variables that represent the problem. These are summarized in Table 1.

Table 1: Summary of decision variables used in the mathematical programming model

Binary Decision Variables	
Name	**Description**
$\alpha(P\|QQ')$	1, if the determinant $\det(P\|QQ')$ is positive, 0, otherwise.
$\beta(PP'\|QQ')$	1, if the determinants $\det(P\|QQ')$ and $\det(P'\|QQ')$ have the same sign, 0, otherwise.
$\gamma(PP'\|QQ')$	1, if the determinants $\det(P\|QQ')$ and $\det(P'\|QQ')$ are both positive, 0, otherwise.
$\delta(PP'\|QQ')$	1, if the line segments $\overline{PP'}$ and $\overline{QQ'}$ intersect, 0, otherwise.
$\varepsilon(PP')$	1, if the line segment $\overline{PP'}$ does not cross any barrier, 0, otherwise.
$y(PQ)$	1, if the edge (P, Q) is selected in the solution of the model, 0, otherwise.
Continuous Decision Variables	
Name	**Description**
P_N	Coordinates representing the point selected in the neighborhood N.
$d(PQ)$	Euclidean distance between the points P and Q.
$g(PQ)$	Amount of commodity passing through the edge (P, Q).

3.1. Conic programming constraints in the models

In both considered problems, namely H-SPPN and H-TSPN, there exist two second-order cone constraints that model the distance between pair of points P and Q, as well the representation of neighborhoods where points can be selected.

For the former case, we introduce the non-negative continuous variable $d(PQ)$ that represents the distance between P and Q:

$$\|P - Q\| \leq d(PQ), \quad \forall (P, Q) \in E. \tag{d-C}$$

For the latter case, since we are assuming that the neighborhoods are second-order cone (SOC) representable, they can be expressed by means of the constraints:

$$P_N \in N \iff \|A_N^i P_N + b_N^i\| \leq (c_N^i)^T P_N + d_N^i, \quad i = 1, \ldots, nc_N, \tag{N-C}$$

where A_N^i, b_N^i, c_N^i and d_N^i are parameters of the constraint i and nc_N denotes the number of constraints that appear in the block associated to the neighborhood N.

It is remarkable that these inequalities can model the special case of linear constraints (for $A_N^i, b_N^i \equiv 0$), ellipsoids and hyperbolic constraints (see [?] for more information).

Puede ser muy interesante el caso 3D pero quizás en otro trabajo, no?

3.2. Checking whether a segment is an edge of the induced graph

Let $P, Q \in V$ two vertices of the graph defined in the previous section such that $(P, Q) \notin E_{\mathcal{B}}$. The goal is to check if the edge (P, Q) is in E, i.e., if the line segment $\overline{PQ}$ does not intersect with any barrier of $\mathcal{B}$. The following well-known computational geometry result can be used to check if two line segments intersect:

Remark 1. *Let $\overline{PQ}$ and $B = \overline{P_B^1 P_B^2} \in \mathcal{B}$ be two different line segments. Let also denote*

$$\det(P|P_B^1P_B^2) = \det\left(\overrightarrow{PP_B^1} \;\middle|\; \overrightarrow{PP_B^2} \right) := \det\begin{pmatrix} P^1_{B_x} - P_x & P^2_{B_x} - P_x \\ P^1_{B_y} - P_y & P^2_{B_y} - P_y \end{pmatrix}$$

the determinant whose arguments are $P = (P_x, P_y)$, $P_B^1 = (P^1_{B_x}, P^1_{B_y})$ and $P_B^2 = (P^2_{B_x}, P^2_{B_y})$. If

$$\operatorname{sign}\left(\det(P|P_B^1P_B^2)\right) = \operatorname{sign}\left(\det(Q|P_B^1P_B^2)\right) \quad \text{or} \quad \operatorname{sign}\left(\det(P_B^1|PQ)\right) = \operatorname{sign}\left(\det(P_B^2|PQ)\right),$$

then $\overline{PQ}$ and B do not intersect.

Since E is not fixed, the determinants in Remark 1 also depend on the location of P and Q. Hence, it is essential to model the previous constraint by using binary variables that check the sign of determinants, the equality of signs, and the disjunctive condition.

To model the sign of each determinant in Remark 1. We introduce the binary variable α, that assumes the value one if the determinant is positive and zero, otherwise. Note that the case when determinants are null does not need to be considered, because the segments are located in general position.

It is possible to represent the sign condition by including the following constraints:

$$\begin{aligned}
\left[1 - \alpha(P|P_B^1P_B^2)\right] L(P|P_B^1P_B^2) \le \det(P|P_B^1P_B^2) \le U(P|P_B^1P_B^2)\, \alpha(P|P_B^1P_B^2), \\
\left[1 - \alpha(Q|P_B^1P_B^2)\right] L(Q|P_B^1P_B^2) \le \det(Q|P_B^1P_B^2) \le U(Q|P_B^1P_B^2)\, \alpha(Q|P_B^1P_B^2), \\
\left[1 - \alpha(P_B^1|PQ)\right] L(P_B^1|PQ) \le \det(P_B^1|PQ) \le U(P_B^1|PQ)\, \alpha(P_B^1|PQ), \\
\left[1 - \alpha(P_B^2|PQ)\right] L(P_B^2|PQ) \le \det(P_B^2|PQ) \le U(P_B^2|PQ)\, \alpha(P_B^2|PQ),
\end{aligned} \tag{α-C}$$

where L and U are a lower and an upper bound for the value of determinants, respectively. If a determinant is positive, then α must be one to make the second inequality feasible. The analogous case happens if the determinant is not positive.

Now, to check if the pairs of determinants

$$\det(P|P_B^1P_B^2),\ \det(Q|P_B^1P_B^2) \quad \text{and} \quad \det(P_B^1|PQ),\ \det(P_B^2|PQ)$$

have the same sign, we introduce the binary variable β, that is one if the pair has the same sign, and zero otherwise.

Hence, the β variable can be represented by the equivalence constraint of the α variables

$$\begin{aligned}
\beta(PQ|P_B^1P_B^2) &= \alpha(P|P_B^1P_B^2)\alpha(Q|P_B^1P_B^2) + \left[1 - \alpha(P|P_B^1P_B^2)\right]\left[1 - \alpha(Q|P_B^1P_B^2)\right], \\
\beta(P_B^1P_B^2|PQ) &= \alpha(P_B^1|PQ)\alpha(P_B^2|PQ) + \left[1 - \alpha(P_B^1|PQ)\right]\left[1 - \alpha(P_B^2|PQ)\right].
\end{aligned}$$

This condition can be equivalently written using the auxiliary binary variable γ is that models the product of the α variables:

$$\begin{aligned}
\beta(PQ|P_B^1P_B^2) &= 2\gamma(PQ|P_B^1P_B^2) - \alpha(P|P_B^1P_B^2) - \alpha(Q|P_B^1P_B^2) + 1, \\
\beta(P_B^1P_B^2|PQ) &= 2\gamma(P_B^1P_B^2|PQ) - \alpha(P_B^1|PQ) - \alpha(P_B^2|PQ) + 1,
\end{aligned} \tag{β-C}$$

We observe that γ can be linearized by using the following constraints:

$$\begin{aligned}
&\gamma(PQ|P_B^1P_B^2) \le \alpha(P|P_B^1P_B^2), && \gamma(P_B^1P_B^2|PQ) \le \alpha(P_B^1|PQ), \\
&\gamma(PQ|P_B^1P_B^2) \le \alpha(Q|P_B^1P_B^2), && \gamma(P_B^1P_B^2|PQ) \le \alpha(P_B^2|PQ), \\
&\gamma(PQ|P_B^1P_B^2) \ge \alpha(P|P_B^1P_B^2) + \alpha(Q|P_B^1P_B^2) - 1, && \gamma(P_B^1P_B^2|PQ) \ge \alpha(P_B^1|PQ) + \alpha(P_B^2|PQ) - 1.
\end{aligned} \tag{γ-C}$$

Later, we need to check whether there exists any coincidence in the sign of determinants, so we define the binary variable δ that is one if segments do not intersect and zero, otherwise. This condition can be modeled by using these disjunctive constraints:

$$\frac{1}{2}\left[\beta(PQ|P_B^1P_B^2) + \beta(P_B^1P_B^2|PQ)\right] \le \delta(PQ|P_B^1P_B^2) \le 2\left[\beta(PQ|P_B^1P_B^2) + \beta(P_B^1P_B^2|PQ)\right]. \tag{δ-C}$$

Indeed, the above constraint states that if there exists a sign coincidence, then δ is one to satisfy the first constraint, and the second is always fulfilled. However, if the sign of the determinants is not the same, then the second constraint is active and δ is null.

Finally, it is required that

$$\overline{PQ}\cap B''=\emptyset,\quad \forall B''\in\mathcal{B},\quad\iff\quad \delta(PQ|P^1_{B''}P^2_{B''})=1,\quad \forall B''\in\mathcal{B}.$$

Hence, if we denote by $\varepsilon(PQ)$ the binary variable that is one when the previous constraint is satisfied and zero otherwise, this variable can be represented by means of the following inequalities:

$$\left[\sum_{B''\in\mathcal{B}}\delta(PQ|P^1_{B''}P^2_{B''})-|\mathcal{B}|\right]+1\leq\varepsilon(PQ)\leq\frac{1}{|\mathcal{B}|}\sum_{B''\in\mathcal{B}}\delta(PQ|P^1_{B''}P^2_{B''}).\qquad(\varepsilon\text{-C})$$

If there is a barrier $B''\in\mathcal{B}$ that intersects the segment $\overline{PQ}$, then $\delta(PQ|P^1_{B''}P^2_{B''})$ is zero and the second inequality enforces ε to be zero because the right hand side is fractional and the first inequality is non-positive. Nonetheless, if there is no barrier that intersects the segment, then ε is equals to one, because the left hand side of the first inequality is one and the right hand side of the second inequality too.

Note that, it is possible to represent the set of edges by means of ε variables as follows:

$$E=\{P,Q\in V:\varepsilon(PQ)=1\}.$$

This representation of E will be utilized for the formulations that are detailed in the following subsections.

3.3. *A formulation for the* H-SPPN

The idea of the formulation of the H-SPPN is to extend the classical formulation of the Shortest Path Problem by taking into account the description of the graph G in Subsection 2.1.

The path that the drone can follow by taking into account the edges of the induced graph is described. Let $P,Q\in V$ and let $y(PQ)$ be the binary variable that is one if the drone goes from P to Q. Then, the inequalities

$$y(PQ)\leq\varepsilon(PQ),\qquad(\text{y-C})$$

assure that the drone can go from P to Q only if the segment $\overline{PQ}$ does not cross any barrier.

By taking into account the constraints explained in subsections above, the following MINLP formulation for the H-SPPN is presented:

$$\begin{aligned}
\text{minimize}\quad & \sum_{(P,Q)\in E} d(PQ)y(PQ) && (\text{H-SPPN})\\
\text{subject to}\quad & \sum_{\{Q:(P,Q)\in E\}} y(PQ)-\sum_{\{Q:(Q,P)\in E\}} y(QP)=\begin{cases}1, & \text{if } P\in V_S,\\ 0, & \text{if } P\in V_{\mathcal{B}},\\ -1, & \text{if } P\in V_T.\end{cases},\\
& (\alpha\text{-C}),(\beta\text{-C}),(\delta\text{-C}),(\gamma\text{-C}),(\varepsilon\text{-C}),(\text{y-C}),\\
& (\text{d-C}),(\text{N-C}).
\end{aligned}$$

The objective function minimizes the length of the path followed by the drone on the edges of the induced graph G. The first constraints are the flow conservation constraints, the second constraints represent the sets E_S and E_T and the third ones state that the points selected must be in their respective neighborhoods.

3.3.1. *Reformulating the* H-SPPN

The formulation for the H-SPPN presented above can be simplified by taking into account the following observation.

Proposition 1. *There exists two finite dominant sets, N_S^* and N_T^*, of possible candidates to be in N_S and N_T, respectively. Moreover,*

$$N_S^*=\{P_S(P_B^i)=\arg\min_{P_S\in N_S}\|P_S-P_B^i\|:(P_S,P_B^i)\in E_S\},$$
$$N_T^*=\{P_T(P_B^i)=\arg\min_{P_T\in N_T}\|P_B^i-P_T\|:(P_B^i,P_T)\in E_T\}.$$

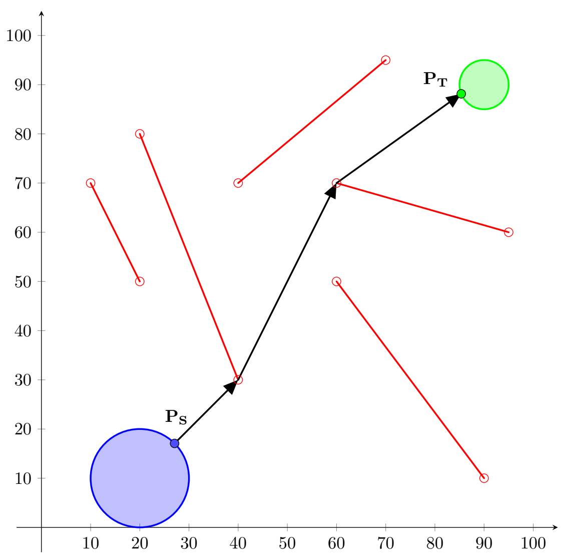


Figure 3: Solution for the instance of the H-SPPN

Proof. Note that the points selected in N_S and N_T in an optimal solution for H-SPPN must be those that give the minimum distances to the points of the first and last visited barriers in the optimal solution, respectively. Therefore, N_S^* and N_T^* must be composed, at most, by points in sets

$$\{P_S(P_B^i) = \arg\min_{P_S \in N_S} \|P_S - P_B^i\| : (P_S, P_B^i) \in E_S\},$$
$$\{P_T(P_B^i) = \arg\min_{P_T \in N_T} \|P_B^i - P_T\| : (P_B^i, P_T) \in E_T\},$$

as claimed. □

Therefore, we can compute sets N_S^* and N_T^* by solving a convex problem for each endpoint of the barriers:

$$N_S^* = \{P_S(P_B^i) = \arg\min_{P_S \in N_S} \|P_S - P_B^i\| : \varepsilon(P_S P_B^i) = 1\},$$
$$N_T^* = \{P_T(P_B^i) = \arg\min_{P_T \in N_T} \|P_B^i - P_T\| : \varepsilon(P_B^i P_T) = 1\}.$$

Moreover, the points chosen in the solution, P_S and P_T, can be represented by the points in N^* as shown below:

$$P_S = \sum_{P_B^i \in V_{\mathcal{B}}} \mu_S(P_B^i) P_S(P_B^i), \qquad 1 = \sum_{P_B^i \in V_{\mathcal{B}}} \mu_S(P_B^i), \tag{N$_S^*$-C}$$

$$P_T = \sum_{P_B^i \in V_{\mathcal{B}}} \mu_T(P_B^i) P_T(P_B^i), \qquad 1 = \sum_{P_B^i \in V_{\mathcal{B}}} \mu_T(P_B^i), \tag{N$_T^*$-C}$$

where μ_S and μ_T are indicator variables that select one of the points in N_S^* and N_T^* in the solution.

The major advantage of this approach is that, once sets N_S^* and N_T^* are computed beforehand, the whole graph G is fixed because the incident edges can be computed for each point $P_S(P_B^i)$ and $P_T(P_B^i)$, $P_B^i \in V_{\mathcal{B}}$, separately. Defining again the variable y for the edges in E, the new formulation for the H-SPPN can be expressed as the following simplified program:

$$\begin{aligned} \text{minimize} \quad & \sum_{(P,Q) \in E} d(PQ) y(PQ) && \text{(H-SPPN}^*\text{)} \\ \text{subject to} \quad & \sum_{\{Q:(P,Q)\in E\}} y(PQ) - \sum_{\{Q:(Q,P)\in E\}} y(QP) = \begin{cases} 1, & \text{if } P \in V_S, \\ 0, & \text{if } P \in V_{\mathcal{B}}, \\ -1, & \text{if } P \in V_T. \end{cases}, \\ & \text{(d-C)}, (\text{N}_S^*\text{-C}), (\text{N}_T^*\text{-C}). \end{aligned}$$

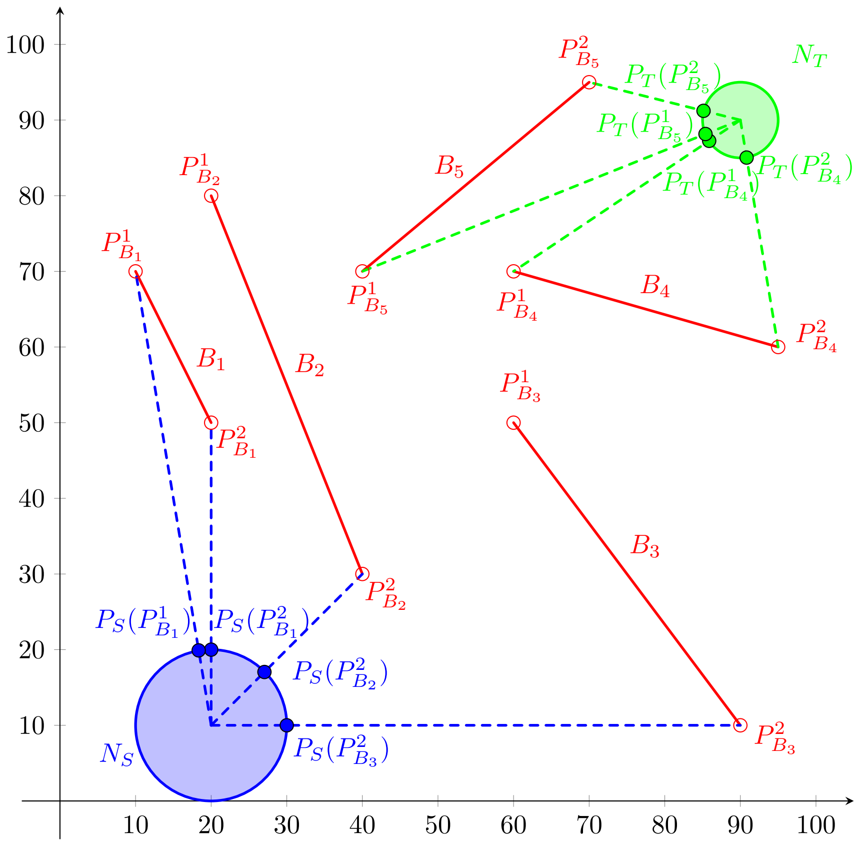


Figure 4: Problem data of the H-SPPN, H-TSPN and H-TSPVN

Proposition 2. *Let $n = |\mathcal{B}|$. The* H-SPPN *can be solved in polynomial time $O(n^3)$*

Proof. The proof follows using the finite dominant ingsets N^*_S and N^*_T. First of all, it is clear that the cardinality of these finite dominating sets is $O(n)$ since there is one point in each set for each vertex of a segment in $\mathcal{B}$. Next, we recall that given a set of polygonal obstacles with $O(n)$ vertices, the linear map from any given point can be computed in $O(n \log n)$ time. Once one has that map finding the shortest path to any other point in the region is done in $O(n)$ time. Therefore, the H-SPPN can be solved as follows.

For each point $P_S(P^i_B) \in N^*_S$ we compute its linear map and solve the shortest path problem with respect to all the $O(n)$ points in N^*_T. Overall, this operations takes $O(n \log n + n^2)$. The same operation has to be repeated for all the $O(n)$ points in N^*_S. Hence, the overall complexity for solving the H-SPPN is $O(n^3)$.

□

3.4. *A formulation for the* H-TSPN

The rationale of the formulation for the H-TSPN is to consider the variant called Steiner TSP (STSP) (reference to this problem), where some nodes in $V_{\mathcal{B}}$ do not have to be visited, but if necessary they can be visited more than once.

It is well known that it is possible to convert any instance of the STSP into an instance of the standard TSP, by computing the shortest paths between every pair of required nodes, when these nodes are fixed. However, in our problem, since the points in the neighborhoods are not fixed, this simplification cannot be applied to obtain the optimal solution for the H-TSPN, although it may produce an approximation to generate lower bounds for the H-TSPN.

Our formulation rest on adjusting single-commodity flow formulation to ensure connectivity. We can assume wlog that the neighborhood N_1 is required and the drone departs from that depot (assuming to be N_1) with $|\mathcal{N}| - 1$ units of commodity. The idea is that the model must deliver one unit of commodity to each of the required neighborhoods. Then, for each edge $(P, Q) \in E$, we define the following variables:

- $y(PQ)$, binary variable that is equals to one if the drone goes from P to Q.

- $g(PQ)$, non-negative continuous variable that represents the amount of the commodity passing through the edge (P, Q).

Hence, we can adjust the single-commodity flow formulation to the induced graph G as follows:

$$\begin{aligned}
\text{minimize} \quad & \sum_{(P,Q)\in E} d(PQ)y(PQ) && \text{(H-TSPN)}\\
\text{subject to} \quad & \sum_{\{Q:(P_N,Q)\in E_{\mathcal{N}}\}} y(P_NQ) \geq 1, && \forall P_N \in V_{\mathcal{N}},\\
& \sum_{\{Q:(P,Q)\in E\}} y(PQ) = \sum_{\{Q:(Q,P)\in E\}} y(QP), && \forall P \in V,\\
& \sum_{\{Q:(Q,P_N)\in E_{\mathcal{N}}\}} g(QP_N) - \sum_{\{Q:(P_N,Q)\in E_{\mathcal{N}}\}} g(P_NQ) = 1, && \forall P_N \in V_{\mathcal{N}} \setminus \{P_{N_1}\},\\
& \sum_{\{Q:(Q,P_B^i)\in E\}} g(QP_B^i) - \sum_{\{Q:(P_B^i,Q)\in E\}} g(P_B^iQ) = 0, && \forall P_B^i \in V_{\mathcal{B}},\\
& g(PQ) \leq (|\mathcal{N}|-1)y(PQ), && \forall (P,Q) \in E,\\
& (\alpha\text{-C}), (\beta\text{-C}), (\delta\text{-C}), (\gamma\text{-C}), (\varepsilon\text{-C}), (\text{y-C}),\\
& (\text{d-C}), (\text{N-C}).
\end{aligned}$$

The first constraints impose that the drone departs from each neighborhood. The second constraints are the flow conservation constraints. The third inequalities ensure that one unit of commodity is delivered to each required neighborhood. The fourth ones ensure that the amount of commodity passing through the points in the barriers is not consumed. Finally, the last inequalities enforce that if some commodity passes along an edge only if this edge is used in the tour.

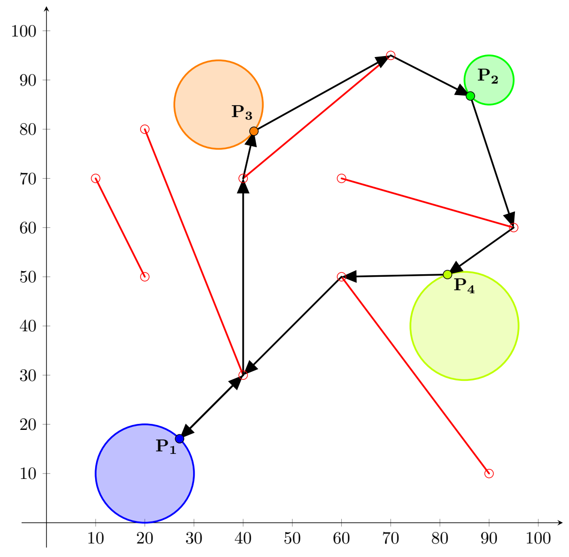


Figure 5: Solution for the instance of the H-TSPN

Proposition 3. *The* H-TSPN *is NP-complete.*

Note that, once a point is fixed in each neighborhood, the problem obtained in the induced graph G is the Steiner TSP (STSP), that is NP-complete.

3.4.1. Reformulating the H-TSPN

The idea of computing a dominant set that represent each of the neighborhoods in the H-SPPN can be adopted to reformulate the H-TSPN in the same way. The dominant sets are stated in the next proposition:

Proposition 4. *Given a neighborhood $N \in \mathcal{N}$, there exists a finite dominant set, N^* of possible candidates to be in N. Moreover,*

$$N^* = \{P_N(P_B^i, P_{B'}^j) = \arg\min_{P_N \in N} \|P_B^i - P_N\| + \|P_N - P_{B'}^j\| : (P_B^i, P_N), (P_N, P_{B'}^j) \in E_{\mathcal{N}}\}.$$

Proof. The way a drone visits a neighborhood N is

$$P_B^i \longrightarrow P_N \longrightarrow P_{B'}^j,$$

for some points P_B^i, $P_{B'}^j \in V_{\mathcal{B}}$, since there do not exist a rectilinear path that join any pair of neighborhoods. Hence, the points chosen in an optimal solution for H-TSPN must be those that produce the minimum distances to the points of the previously and next visited barriers in the optimal solution. Therefore, N^* must be composed, at most, by points in set

$$\{P_N(P_B^i, P_{B'}^j) = \arg\min_{P_N \in N} \|P_B^i - P_N\| + \|P_N - P_{B'}^j\| : (P_B^i, P_N), (P_N, P_{B'}^j) \in E_{\mathcal{N}}\}.$$

what was to be shown. $\square$

Again, we can compute the respective dominant set N^* of a neighborhood $N \in \mathcal{N}$ by solving a convex problem for each pair of barrier endpoints more complex:

$$N^* = \{P_N(P_B^i, P_{B'}^j) = \arg\min_{P_N \in N} \|P_B^i - P_N\| + \|P_N - P_{B'}^j\| : \varepsilon(P_B^i P_N) = 1 \wedge \varepsilon(P_N P_{B'}^j) = 1\}.$$

Moreover, the points P_N chosen in the solution can be represented by the points in N^* as shown below:

$$\begin{aligned} P_N &= \sum_{P_B^i \in V_{\mathcal{B}}} \sum_{P_{B'}^j \in V_{\mathcal{B}}} \mu_N(P_B^i P_{B'}^j) P_N(P_B^i P_{B'}^j), \\ 1 &= \sum_{P_B^i \in V_{\mathcal{B}}} \sum_{P_{B'}^j \in V_{\mathcal{B}}} \mu_N(P_B^i P_{B'}^j), \\ 2\mu_N(P_B^i P_{B'}^j) &= y(P_B^i P_N) + y(P_N P_{B'}^j), \end{aligned} \tag{N*-C}$$

where μ_N are indicator variables that select one of the points in N^* in the solution. Last equality ensures that the path $P_B^i \longrightarrow P_N \longrightarrow P_{B'}^j$ is followed by the drone if and only if $P_N(P_B^i P_{B'}^j)$ is selected.

Again sets N^* can be computed in advance so that the whole graph G will be fixed. The new formulation for the H-TSPN can be represented as the next simplified program:

$$\begin{aligned} \text{minimize} \quad & \sum_{(P,Q) \in E} d(PQ) y(PQ) && \text{(H-TSPN}^*\text{)} \\ \text{subject to} \quad & \sum_{\{Q:(P_N,Q) \in E_{\mathcal{N}}\}} y(P_N Q) \geq 1, && \forall P_N \in V_{\mathcal{N}}, \\ & \sum_{\{Q:(P,Q) \in E\}} y(PQ) = \sum_{\{Q:(Q,P) \in E\}} y(QP), && \forall P \in V, \\ & \sum_{\{Q:(Q,P_N) \in E_{\mathcal{N}}\}} g(QP_N) - \sum_{\{Q:(P_N,Q) \in E_{\mathcal{N}}\}} g(P_N Q) = 1, && \forall P_N \in V_{\mathcal{N}} \setminus \{P_{N_1}\}, \\ & \sum_{\{Q:(Q,P_B^i) \in E\}} g(QP_B^i) - \sum_{\{Q:(P_B^i,Q) \in E\}} g(P_B^i Q) = 0, && \forall P_B^i \in V_{\mathcal{B}}, \\ & g(PQ) \leq (|\mathcal{N}| - 1) y(PQ), && \forall (P,Q) \in E, \\ & \text{(d-C), (N}^*\text{-C)}. \end{aligned}$$

The reader may note that, although this formulation is equivalent to (H-TSPN), the cost of computing sets.

3.5. Relaxing the assumptions of the problem: The H-TSPVN

In this subsection, we expose the differences that appear when, in the model of the H-TSPN, we do not require that the barriers separate the neighborhoods completely, i.e., when going from one neighborhood to another one is possible without crossing any barrier. The main difference lies in the description of the edges of the graph induced by the neighborhoods and the endpoints of the barriers.

By taking the same approach, the sets that describe the graph in that case are:

- $V_{\mathcal{N}} = \{P_N : N \in \mathcal{N}\}$: set of points in the neighborhoods $\mathcal{N}$ that must be visited.
- $V_{\mathcal{B}} = \{P_B^1, P_B^2 : B = \overline{P_B^1 P_B^2} \in \mathcal{B}\}$: set of vertices that form the barriers of the problem.
- $V = V_{\mathcal{N}} \cup V_{\mathcal{B}}$.
- $E = \{(P, P') : P, P' \in V \text{ and } \overline{PP'} \cap B'' = \emptyset, \forall B'' \in \mathcal{B}\}$: set of edges formed by the line segments that join every pair of points in V that do not cross any barrier.

The difference between the set of edges in the H-TSPVNwith respect to the graph in H-TSPNis that, in the former case, the edges that join each pair of neighborhoods are considered. This fact leads to include product of continuous variables in the α constraints of the model that represent the determinants that determine if the two variable points in the neighborhoods cross any barrier or not and the problem becomes non-convex.

4. Strengthening the formulations

Proposition 5. *Under the hypothesis that balls are visible among them, the maximum number of balls that can be considered in* H-TSPN *is* $O(n^2)$.

Proof. The proof is based on the properties of the visibility graph $VG(\mathcal{B})$ of the configuration of barriers in $\mathcal{B}$: It is clear that the maximum number of balls coincides with the number of faces, $f_{VG(\mathcal{B})}$, of $VG(\mathcal{B})$. Recall that the number of vertices in $\mathcal{B}$ is n. Next, by the Euler formula the number of faces $f_{VG(\mathcal{B})}$ is $2 + E_{VG(\mathcal{B})} - n$. Since $E_{VG(\mathcal{B})} = O(n^2)$, it follows that $f_{VG(\mathcal{B})} = O(n^2)$. □

4.1. Preprocessing

In this subsection, a preprocessing result that allows to fix some variables is shown by taking into account the relative position between the neighborhoods and the barriers. In particular, we are going to present an outcome that ensures that there are some barriers whose endpoints can not be incident in the edges of $E_{\mathcal{N}}$ and it is not necessary to include it in $E_{\mathcal{N}}$.

Let denote

$$\begin{aligned}
\text{cone}(P, Q, R) &:= \{\mu_1 \overrightarrow{PQ} + \mu_2 \overrightarrow{PR} : \mu_1, \mu_2 \geq 0\},\\
\text{cone}(P, Q, R)^- &:= \{\mu_1 \overrightarrow{PQ} + \mu_2 \overrightarrow{PR} : \mu_1, \mu_2 \geq 0,\ \mu_1 + \mu_2 \leq 1\},\\
\text{cone}(P, Q, R)^+ &:= \{\mu_1 \overrightarrow{PQ} + \mu_2 \overrightarrow{PR} : \mu_1, \mu_2 \geq 0,\ \mu_1 + \mu_2 \geq 1\}.
\end{aligned}$$

Note that $cone(P, Q, R)$ is the union of $\text{cone}(P, Q, R)^-$ and $\text{cone}(P, Q, R)^+$. It is also important to remark that $\text{cone}(P, Q, R)^+$ is the part of cone that crosses the barrier $\overline{QR}$ when we consider a segment whose endpoints are P and another point of this set, i.e.

$$\text{cone}(P, Q, R)^+ = \{P' : \overline{PP'} \cap \overline{QR} \neq \emptyset\}.$$

Let $B = \overline{P_B^1 P_B^2} \in \mathcal{B}$ a barrier. In the following proposition, we give a sufficient condition to not include the edge (P_N, P_B^i) in $E_{\mathcal{N}}$:

Proposition 6. *Let* $B' = \overline{P_{B'}^1 P_{B'}^2} \in \mathcal{B}$ *and* $\text{cone}(P_B^i, P_{B'}^1, P_{B'}^2)^+$ *the conical hull generated by these points. If*

$$N \subset \bigcup_{B' \in \mathcal{B}} \text{cone}(P_B^i, P_{B'}^1, P_{B'}^2)^+,$$

then $(P_N, P_B^i) \notin E_{\mathcal{N}}$.

Proof. If $P_N \in N$, then there exists a $B' \in \mathcal{B}$ such that $P_N \in \mathrm{cone}(P_B^i, P_{B'}^1, P_{B'}^2)^+$. Therefore, $\overline{P_B^i P_N} \cap B' \neq \emptyset$ and $(P_{\mathcal{N}}, P_B^i) \notin E_{\mathcal{N}}$, as we claimed. $\square$

We can check computationally the condition of the previous proposition by using the following procedure. The first issue that is solved is the one when the neighborhoods are segments. Let $N = \overline{P_N^1 P_N^2}$ be a line segment and r_N the straight line that contains the line segment N that is represented as:

$$r_N : P_N^1 + \lambda \overrightarrow{P_N^1 P_N^2}, \qquad \lambda \in \mathbb{R}.$$

Algorithm 1: Checking computationally if $(P_{\mathcal{N}}, P_B^i) \notin E_{\mathcal{N}}$ when N is a segment.

```
Initialization: Let $P_B^i$ be the point whose edge $(P_{\mathcal{N}}, P_B^i)$ is going to check if $(P_{\mathcal{N}}, P_B^i) \notin E_{\mathcal{N}}$.
                Set $points = \{P_N^1, P_N^2\}$, $lambdas = \{0, 1\}$.
1  for $B'' \in \mathcal{B}$ do
2      for $j \in \{1, 2\}$ do
3          Compute the straight line
                   $r(P_B^i, P_{B''}^j) = P_B^i + \mu_{B''}^j \overrightarrow{P_B^i P_{B''}^j}$,
           that contains the points $P_B^i$ and $P_{B''}^j$.
4          Intersect $r(P_B^i, P_{B''}^j)$ and $r_N$ in the point $Q_{B''}^j$ and compute $\overline{\mu}_{B''}^j$ such that
                   $Q_{B''}^j = P_B^i + \overline{\mu}_{B''}^j \overrightarrow{P_B^i P_{B''}^j}$.
5          if $|\overline{\mu}_{B''}^j| \geq 1$ then
6              Compute $\lambda_{B''}^j$ such that
                   $Q_{B''}^j = P_N^1 + \lambda_{B''}^j \overrightarrow{P_N^1 P_N^2}$.
7              if $\overline{\mu}_{B''}^j \geq 1$ then
8                  Include $\lambda_{B''}^j$ in $lambdas$.
9              else
10                 if $\lambda_{B''}^j \geq 0$ then
11                     Set $\lambda_{B''}^j = M << 0$ and include it in $lambdas$.
12                 else
13                     Set $\lambda_{B''}^j = M >> 0$ and include it in $lambdas$.
14 Order in non-decreasing order the set $lambdas$.
15 If it is verified that
```

$$\begin{aligned}
\min\{\lambda_{B'}^1, \lambda_{B'}^2\} \leq 0 \leq \max\{\lambda_{B'}^1, \lambda_{B'}^2\}, &\quad \text{for some } B' \in \mathcal{B},\\
\min\{\lambda_{B'}^1, \lambda_{B'}^2\} \leq 1 \leq \max\{\lambda_{B'}^1, \lambda_{B'}^2\}, &\quad \text{for some } B' \in \mathcal{B},\\
\min\{\lambda_{B'}^1, \lambda_{B'}^2\} \leq \lambda_{B''}^j \leq \max\{\lambda_{B'}^1, \lambda_{B'}^2\}, &\quad \text{for some } B' \in \mathcal{B} \setminus \{B''\}, \quad \forall \lambda_{B''}^j \in lambdas \setminus \{M\},
\end{aligned}$$

or

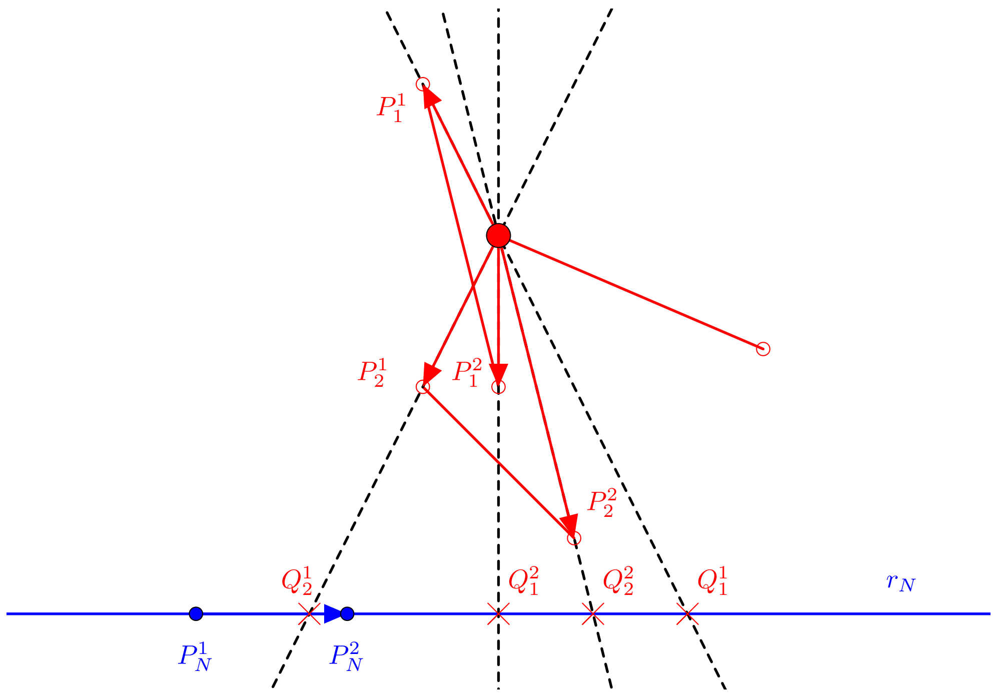
$$\min\{\lambda_{B'}^1, \lambda_{B'}^2\} \leq 0, 1 \leq \max\{\lambda_{B'}^1, \lambda_{B'}^2\}, \qquad \text{for some } B' \in \mathcal{B},$$

then $(P_{\mathcal{N}}, P_B^i) \notin E_{\mathcal{N}}$.

Note that this algorithm also allows us to decide if the drone can access to a point of a barrier from any point of the neighborhood N. It is enough to check in (15) that

$$0 \notin \left[\min\{\lambda_{B'}^1, \lambda_{B'}^2\}, \max\{\lambda_{B'}^1, \lambda_{B'}^2\}\right] \quad \text{and} \quad 1 \notin \left[\min\{\lambda_{B'}^1, \lambda_{B'}^2\}, \max\{\lambda_{B'}^1, \lambda_{B'}^2\}\right], \quad \forall B' \in \mathcal{B}.$$

For the case when N is an ellipse, the same rationale can be followed. The idea is to generate the largest line segment contained in the ellipse and repeat the procedure exposed in the Algorithm 1. Let F_1 and F_2 the focal points of N.



	Q_1^1	P_N^1	Q_2^1	P_N^2	Q_1^2	Q_2^2
μ	$-\frac{5}{2}$	$\times$	$\frac{5}{2}$	$\times$	$\frac{5}{2}$	$\frac{5}{4}$
λ	$M << 0$	0	$\frac{3}{4}$	1	2	$\frac{9}{4}$

Figure 6: Example of the Algorithm 1

Algorithm 2: Checking computationally if $(P_N, P_B^i) \notin E_{\mathcal{N}}$ when N is an ellipse.

```
Initialization: Let $P_B^i$ be the point whose edge $(P_B^i, P_N)$ is going to check if $(P_N, P_B^i) \notin E_{\mathcal{N}}$.
                Set $points = \{\}$, $lambdas = \{\}$.
1 Compute the straight line $r(F^1, F^2)$.
2 Intersect $r(F^1, F^2)$ and $\partial N$ in the points $P_N^1$ and $P_N^2$.
3 Include $P_N^1$ and $P_N^2$ in $points$.
4 Apply the Algorithm 1.
```

Estudiar resultados que eliminen algunas de las posibles aristas de E_S y E_T Hablar que las aristas $E_{\mathcal{B}}$ se pueden preprocesar porque los puntos están fijados

Es posible encontrar un caso en el que utilizar la barrera como arista sea lo mejor?

4.2. Valid inequalities

This subsection is devoted to show some results that adjust the bigM constants that appear in the previous formulation, specifically, in the (α-C), where the modelling of the sign requires to compute the lower and upper bounds L and U, respectively. We are going to determine these bounds explicitly for the cases when the neighborhoods are ellipses and segments.

Let $\overline{P_{B'}^1 P_{B'}^2} = B' \in \mathcal{B}$ be a barrier and $P_N \in N$. Let $\det(P_N | P_{B'}^1, P_{B'}^2)$ also be the determinant whose value must be bounded. Hence, the solution of the following problem gives the lower bound of the determinant:

$$\overline{L} = \min_{P_N=(x,y)\in N} F(x,y) := \det(P_N|P^1_{B'}P^2_{B'}) = \begin{vmatrix} P^1_{B'_x} - x & P^2_{B'_x} - x \\ P^1_{B'_y} - y & P^2_{B'_y} - y \end{vmatrix}. \qquad \text{(L-Problem)}$$

4.2.1. Lower and upper bounds when the neighborhoods are line segments

In this case, the segment whose endpoints are P^1_N and P^2_N can be expressed as the following convex set:

$$N = \{(x,y) \in \mathbb{R}^2 : (x,y) = \mu P^1_N + (1-\mu)P^2_N,\ 0 \leq \mu \leq 1\}.$$

Since we are optimizing a linear function in a compact set we can conclude that the objective function in (L-Problem) achieves its minimum and its maximum in the extreme points of N, that is, in P^1_N and P^2_N.

4.2.2. Lower and upper bounds when the neighborhoods are ellipsoids

The first case that is considered is the one when N is an ellipse, that is, N is represented by the following inequality:

$$N = \{(x,y) \in \mathbb{R}^2 : ax^2 + by^2 + cxy + dx + ey + f \leq 0\} =$$

where a, b, c, d, e, f are coefficients of the ellipse. In an extended form, we need to find:

$$\text{minimize} \quad F(x,y) = \begin{vmatrix} P^1_{B'_x} - x & P^2_{B'_x} - x \\ P^1_{B'_y} - y & P^2_{B'_y} - y \end{vmatrix} = xP^1_{B'_y} - xP^2_{B'_y} + yP^2_{B'_x} - yP^1_{B'_x} + P^1_{B'_x}P^2_{B'_y} - P^1_{B'_y}P^2_{B'_x}, \qquad \text{(L-Ellipse)}$$

$$\text{subject to} \quad ax^2 + by^2 + cxy + dx + ey + f \leq 0.$$

Since we are minimizing a linear function in a convex set, we can conclude that the extreme points are located in the frontier, so we can use the Lagrangian function to compute these points.

$$F(x,y;\lambda) = xP^1_{B'_y} - xP^2_{B'_y} + yP^2_{B'_x} - yP^1_{B'_x} + P^1_{B'_x}P^2_{B'_y} - P^1_{B'_y}P^2_{B'_x} + \lambda(ax^2 + by^2 + cxy + dx + ey + f).$$

$$\nabla F(x,y;\lambda) = 0 \iff \begin{cases} \frac{\partial F}{\partial x} &= P^1_{B'_y} - P^2_{B'_y} + 2ax\lambda + cy\lambda + d\lambda &= 0, \\ \frac{\partial F}{\partial y} &= P^2_{B'_x} - P^1_{B'_x} + 2by\lambda + cx\lambda + e\lambda &= 0, \\ \frac{\partial F}{\partial \lambda} &= ax^2 + by^2 + cxy + dx + ey + f &= 0. \end{cases}$$

From the first two equations we can obtain:

$$\lambda = \frac{P^2_{B'_y} - P^1_{B'_y}}{2ax + cy + d} = \frac{P^1_{B'_x} - P^2_{B'_x}}{2by + cx + e}.$$

From this equality, we obtain the following general equation of the straight line:

$$(P^2_{B'_y} - P^1_{B'_y})(2by + cx + e) - (P^1_{B'_x} - P^2_{B'_x})(2ax + cy + d) = 0,$$

$$\left[c(P^2_{B'_y} - P^1_{B'_y}) - 2a(P^1_{B'_x} - P^2_{B'_x})\right]x + \left[2b(P^2_{B'_y} - P^1_{B'_y}) - c(P^1_{B'_x} - P^2_{B'_x})\right]y + \left[e(P^2_{B'_y} - P^1_{B'_y}) - d(P^1_{B'_x} - P^2_{B'_x})\right] = 0,$$

$$\left[(2a,c)\cdot \overrightarrow{P^1_{B'}P^2_{B'}}\right]x + \left[(c,2b)\cdot \overrightarrow{P^1_{B'}P^2_{B'}}\right]y + \left[(d,e)\cdot \overrightarrow{P^1_{B'}P^2_{B'}}\right] = 0,$$

where $\cdot$ denotes the scalar product of two vectors. Solving the quadratic system:

$$\begin{cases} \left[(2a,c)\cdot \overrightarrow{P^1_{B'}P^2_{B'}}\right]x + \left[(c,2b)\cdot \overrightarrow{P^1_{B'}P^2_{B'}}\right]y + \left[(d,e)\cdot \overrightarrow{P^1_{B'}P^2_{B'}}\right] &= 0, \\ ax^2 + by^2 + cxy + dx + ey + f &= 0. \end{cases}$$

arises two solutions $x^{\pm}$ and $y^{\pm}$ that are evaluated in the objective function to obtain the lowest and highest value correspond to $L(P_N|P^1_{B'}P^2_{B'})$ and $U(P_N|P^1_{B'}P^2_{B'})$, respectively.

5. Computational experiments

The following section is devoted to study the behaviour of the formulations proposed in the Section 3. In the first subsection, the procedure of generating random instances is described. The second details the configuration of the experiments that have been executed. The third subsection reports the results obtained in these experiments.

5.1. Data generation

To generate the instances of our experiments, we will take into account the assumptions **A1-A4** stated in the Section 2. The idea is to generate line segments located in a general position without crossings and neighborhoods that do not intersect with these line segments. The following procedure describes how to construct the instances when the neighborhoods are balls.

Algorithm 3: Generation of instances when the neighborhoods are balls

```
Initialization: Let $|\mathcal{N}|$ be the number of neighborhoods to generate. Let $r_{\text{init}} = 10$ be the half
                of the initial length of the barriers. Set $\mathcal{N} = \{\}$; $points = \{\}$;
                $\mathcal{B} = \{\overline{(0,0)(100,0)}, \overline{(100,0)(100,100)}, \overline{(100,100)(0,100)}, \overline{(0,100)(0,0)}\}$.
1  Generate $|\mathcal{N}|$ points uniformly distributed in the square $[0,100]^2$ and include them in $points$.
2  for $P, P' \in points$ do
3      if $\overline{PP'} \cap B = \emptyset,\ \forall B \in \mathcal{B}$ then
4          Compute $\overrightarrow{d} = \overrightarrow{PP'}$.
5          Compute $M = P + \frac{1}{2}\overrightarrow{d}$.
6          Compute the unitary vector $\overrightarrow{n_u}$ perpendicular to $\overrightarrow{d}$.
7          Set $r = r_{\text{init}}$.
8          Generate the barrier $B(r) = \overline{P_B^+ P_B^-}$ where $P_B^\pm = M \pm r\overrightarrow{n_u}$.
9          while $B(r) \cap B' \neq \emptyset$ for some $B' \in \mathcal{B}$ do
10             Set $r := r/2$.
11             Generate the barrier $B(r)$.
12         Include $B(r)$ in $\mathcal{B}$.
13 for $P \in points$ do
14     Set $r_{\min} = \min_{\{P_B \in B : B \in \mathcal{B}\}} d(P, P_B)$.
15     Generate a random $radii$ uniformly distributed in the interval $\left[\frac{1}{2}r_{\min}, r_{\min}\right]$.
16     Set the ball $N$ whose center is $P$ and radii is $radii$.
17     Include $N$ in $\mathcal{N}$.
```

To generate the instances when the neighborhoods are segments, we can take the balls built in the previous procedure and draw a random angle that determines which point and its diametrically opposite point are selected as the endpoints of the line segment.

5.2. Configuration of the experiments

Since no benchmark instances are available in the literature for this problem, we have generated ten instances with a number $|\mathcal{N}| \in \{5, 10, 20, 30, 50, 80\}$ of two typologies: balls and line segments. We have considered the cases with and without preprocessing the variables of the formulations as explained in Subsection 4.1 and we have reported the average results.

The formulations were coded in Python 3.9.2 and solved in Gurobi 9.1.2 [?] in a AMD® Epyc 7402p 24-core processor. A time limit of 1 hour was set in the experiments.

5.3. Results of the experiments

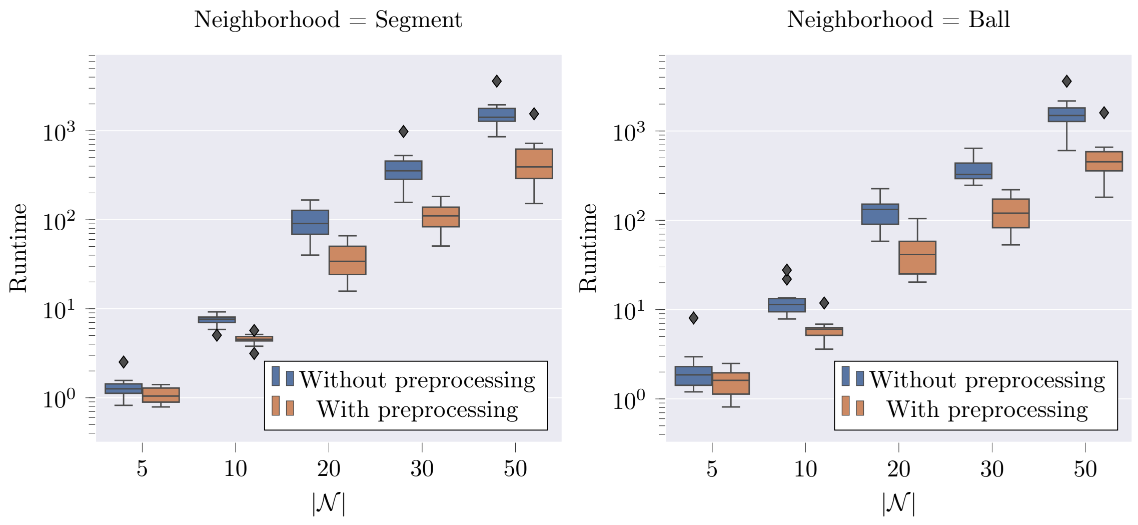


Figure 7: Runtime of the model H-TSPN without and with preprocessing when the neighborhoods are segments and balls.