Network design and game theory

16 - 17 January 2013

Federico Perea Rojas-Marcos

16-17/01/2013

Contents

Designing or extending railway networks is an important issue for may governments since trains:

1. reduce traffic congestion.
2. do not depend on petrol as much as road vehicles.
3. have lower risks of accident.
4. ...

A railway network must be attractive for passengers, otherwise nobody will use it and all the (huge) investment will be wasted!

Steps in railway network design

When designing a railway network one should (among others):

1. estimate potential trips (origin-destination matrix)
2. design the infrastructure: stations and tracks (this is the part of the problem we will focus on)
3. propose lines
4. line frequencies
5. schedules
6. crew assignment
7. ...

RND Problem: Input Data

Over an area, where there already exists a transportation mode (bus):

- A set $N = \{1, 2, \dots, n\}$ of nodes representing potential sites for stations is given.
- A set $E \subseteq \{(i, j) : i, j \in N,\ i < j\}$ of m feasible edges linking the elements of N is known. $A(E)$ denotes the set of arcs obtained from E.
- Every feasible edge $(i, j) \in E$ has an associated length d_{ij} (necessary time to traverse it by train).
- c_i is the cost of building a station at node i, c_e is the cost of building link e. $C_{\max}$ is the available budget.

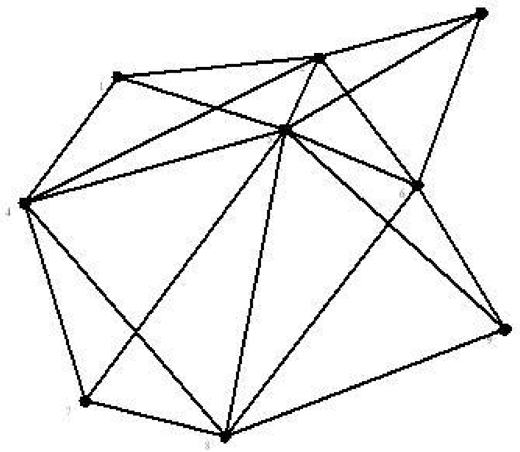
RND Problem: Input Data II

- The mobility pattern is given by a matrix $G = (g_{pq}) : (p, q) \in W$, where W is the ordered index pair set: $W = \{(p, q) : p, q \in N\}$, also referred to as the set of demands.
- The generalized cost of satisfying each demand (p, q) (e.g. travel time) by the complementary mode is v_{pq}.

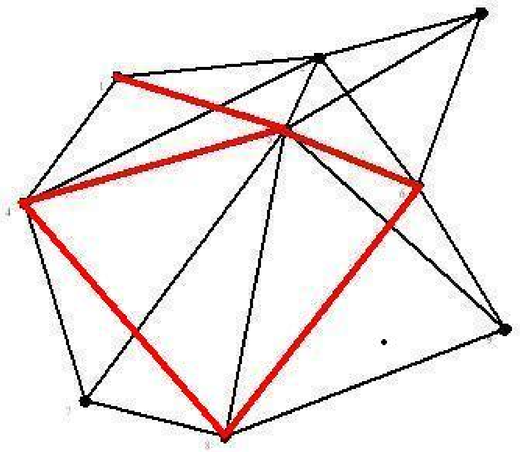
RND goal

Our goal is to choose a set edges satisfying the budget constraints, so that the resulting network covers the maximum number of travelers, meaning, so that as many travelers as possible find the railway network more attractive (faster?) than the already existing alternative transportation mode.

Example potential links



Example solution



RND problem: competition example

Laurier-Université de Montréal → Bus: 25 min. Metro: 20 min. → Metro!

RND Problem: Variables for ILP model

1. $y_i = 1$, if node i is a station of the network; 0 otherwise.
2. $x_{ij} = 1$, if $(i, j) \in E$ belongs to the RTN; 0 otherwise.
3. $f_{ij}^{pq} = 1$ if $(p, q) \in W$ is assigned to the RN and uses edge $(i, j) \in A(E)$.
4. $r_{pq} = 1$ if demand (p, q) is allocated to the RN.
5. u_{pq} is the generalized cost of satisfying demand (p, q). We will assume that the demand (p, q) will use the RN or the complementary mode depending on which one is faster.

In total: $O(mn^2)$ variables.

RND: ILP problem

$$
\begin{array}{ll}
\text{maximize} & \sum_{(p,q)\in W} g_{pq} r_{pq} \\
\text{subject to} & \text{budget constraints} \\
& \text{alignment location constraints} \\
& \text{routing demand conservation constraints} \\
& \text{location-allocation constraints} \\
& \text{splitting demand constraints} \\
& \text{binary conditions.}
\end{array} \quad (1)
$$

RND: Budget constraints

We cannot expend more than the available budget

$$\sum_{(i,j)\in E} c_{ij}\, x_{ij} + \sum_{i\in N} c_i y_i \leq C_{\max}$$

RND: Alignment location constraints

- If a station is not built, the links departing or finishing at such station may not be built.

$$\sum_{i:(i,k)\in E} x_{ik} + \sum_{j:(k,j)E} x_{kj} \leq M y_k,\ k \in N, \tag{2}$$

Better in a disaggregated way

$$x_{ik} \leq y_k,\ \forall\ k \in N, i:(i,k) \in E \tag{3}$$

$$x_{kj} \leq y_k,\ \forall\ k \in N, j:(k,j) \in E \tag{4}$$

- Links can be used in both directions.

$$x_{ij} = x_{ji},\ (i,j) \in E. \tag{5}$$

RND: Routing demand conservation constraints

The route of OD pairs should respect the flow conservation constraints.

$$\sum_{i:(i,p)\in A} f_{ip}^{pq} = 0,\ (p,q)\in W,$$
$$\sum_{j:(p,j)\in A} f_{pj}^{pq} = r_{pq},\ (p,q)\in W,$$
$$\sum_{i:(i,q)\in A} f_{iq}^{pq} = r_{pq},\ (p,q)\in W,$$
$$\sum_{j:(q,j)\in A} f_{qj}^{pq} = 0,\ (p,q)\in W,$$
$$\sum_{i:(i,k)\in A} f_{ik}^{pq} - \sum_{j:(k,j)\in A} f_{kj}^{pq} = 0,\ \forall\, k \notin \{p,q\},\ (p,q)\in W.$$

RND: Other constraints

- Location-allocation constraints: if a demand pair is allocated to the railway network and uses a link on its route, then such link must be built.

$$f_{ij}^{pq} + r_{pq} - 1 \leq x_{ij}, \ (i,j) \in A, \ (p,q) \in W.$$

Could we change this group by $f_{ij}^{pq} \leq x_{ij}$? Note that $f_{ij}^{pq} \leq r_{pq}$ is implied by the flow conservation constraints.

- The travel time for OD pairs needs to be calculated:

$$u_{pq} = \sum_{(i,j)\in A} d_{ij} f_{ij}^{pq} + v_{pq}(1 - r_{pq}), \ \forall (p,q) \in W$$

More constraints

- Splitting demand constraints: if the time spent by an OD pair using the railway is larger than the time spent using the alternative mode, such pair may not be assigned to the RN.

$$\varepsilon + u_{pq} - v_{pq} - M\left(1 - r_{pq}\right) \leq 0, \ (p, q) \in W,$$

- Binary constraints

$$x_{ij}, \ y_i, \ f_{ij}^{pq}, \ r_{pq} \in \{0, 1\}.$$

In total: $O(n^3 + mn^2)$ constraints.

RND: complexity

- The problem posed before is (strongly) NP-hard.
- Solving toy instances made of 15 stations and 30 edges takes around 1 week.
- Real instances (more than 100 possible stations and 200 possible edges) would be impossible to solve with the current technology.
- Heuristic algorithms are needed.
- More about these models in Laporte et al. 2010, Laporte et al. 2011 and García-Archilla et al. 2012.

Table of contents

- In the model presented before, it was assumed that all passengers of an OD pair use either the railway network or the alternative mode (modeled with binary variable r_{pq}).
- This is not a realistic assumption, as we may have the case in which a proportion of users prefer the train and others prefer the bus.
- As an example of a function that models this situation, we will use the *logit* function.
- $\varphi(x) = \frac{1}{1+\beta e^{-\alpha x}}$. Take for instance $\alpha = \beta = 1$.
- We will then maximize $\sum_{(p,q)\in W} \varphi(v_{pq} - u_{pq})$.

For a particular OD pair we have that:

- $\varphi(v_{pq} - u_{pq})$ is continuous on u_{pq}.
- If the time using the alternative mode is much higher than the time using the train, then the proportion of passengers using the train is (close to) one: $\lim_{x\to\infty} \varphi(x) = 1$.
- If the time using the alternative mode is much lower than the time using the train, then the proportion of passengers using the train is (close to) zero: $\lim_{x\to-\infty} \varphi(x) = 0$.
- For this particular choice of α and β, we have that if both travel times are equal, the proportion of passengers using the train is 0.5: $\varphi(0) = 0.5$.

New complexity

- The rest of the model is the same as before.
- We now maximize the number of travelers using the train in a more realistic way.
- This comes at a price: the new problem is a Mixed Integer Non Linear Programming Problem.
- Not convex, not concave. EXTREMELY DIFFICULT TO SOLVE!
- The application of heuristics gets even more important.

Relaxation of the logit function

- In Marín and García-Ródenas (2009), the logit function is approximated by means of a piecewise linear function (a polygonal).
- We are now working on a discretization of such function.
- In both cases, the objective function becomes linear but new variables and constraints need to be added to the problem.

Table of contents

- As pointed out before, the proposed problems are strongly complex.
- We have tried a GRASP algorithm to solve it.

GRASP in RND: notation

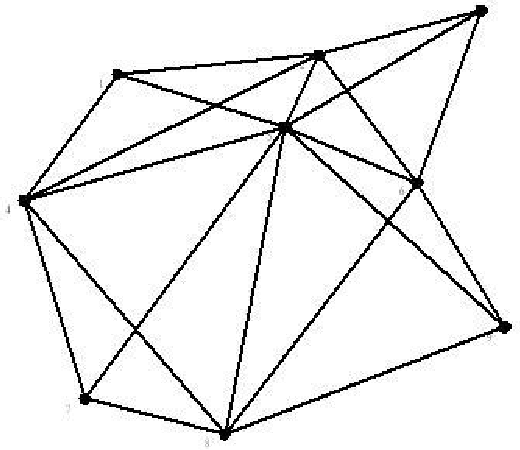
- Given a set of edges E, let $V(E)$ denote the set of vertices included in the edges of E.
- Let $k \in \mathbb{N}$ be the number of possible edges to be included at each iteration.
- Let RN_t be the set of edges that form the solution, not necessarily optimal, obtained after the t^{th} iteration.
- In a given iteration t, edge $e_t \in E$ is said to be feasible if $e_t \notin RN_{t-1}$, graph $(V(RN_{t-1} \cup \{e_t\}), RN_{t-1} \cup \{e_t\})$ is connected and the construction cost of such a graph is not larger than the maximum cost $C_{\max}$. FE_t denotes the set of feasible edges at iteration t.

GRASP in RND: construction phase

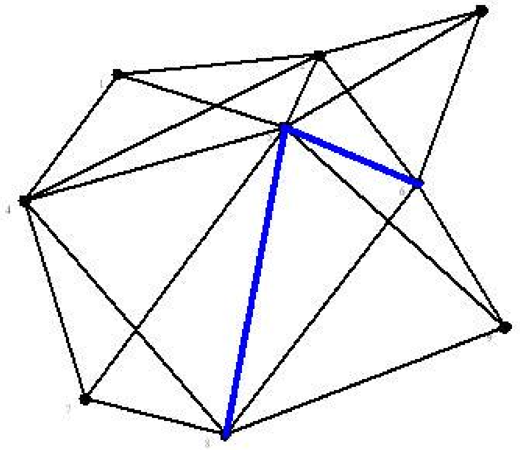
- (Step 0:) Initialize $RN_0 = \emptyset$, $t = 0$.
- (Step 1:) Set $t = t + 1$ and determine the set FE_t. If $|FE_t| = 0$, go to Step 4.
- (Step 2:) If $|FE_t| > k$, determine a set $SE_t^k = \{e_{i_1}, \ldots e_{i_k}\}$ consisting of k edges of FE_t that individually generate the k largest improvements of the objective function when they are added to RN_{t-1}. Otherwise, that is if $|FE_t| \leq k$, set $SE_t^k = FE_t$.
- (Step 3:) Randomly choose one edge $e_t \in SE_t^k$. Set $RN_t = RN_{t-1} \cup \{e_t\}$ and go to Step 1.
- (Step 4:) RN_{t-1} is the best solution found.

In order to obtain a variety of solutions, we repeat this $I_{\max}$ times, choosing the best solution as a final network in this phase.

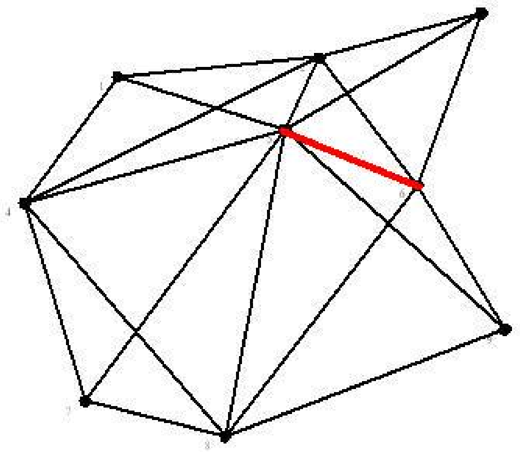
An example with $k = 2$



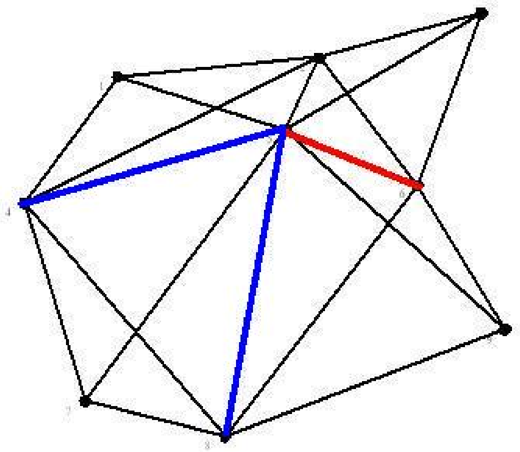
An example with $k = 2$



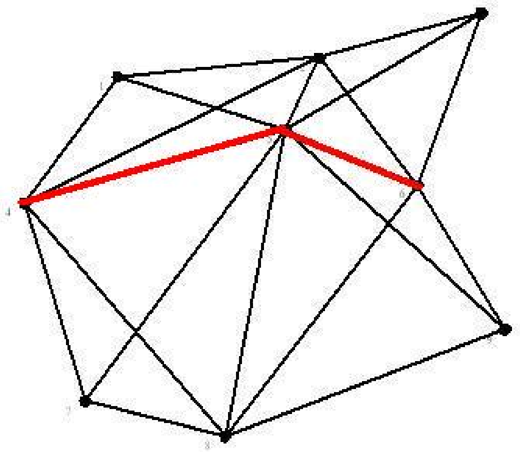
An example with $k = 2$



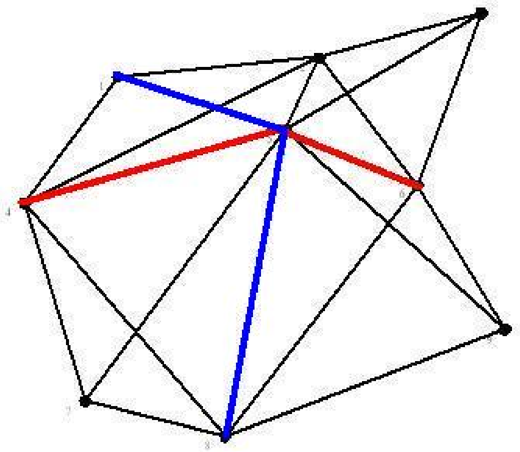
An example with $k = 2$



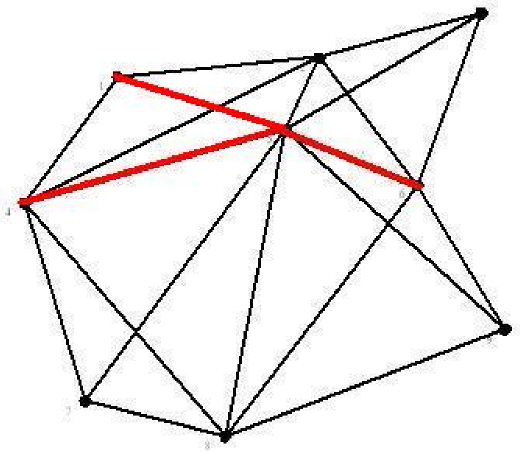
An example with $k = 2$



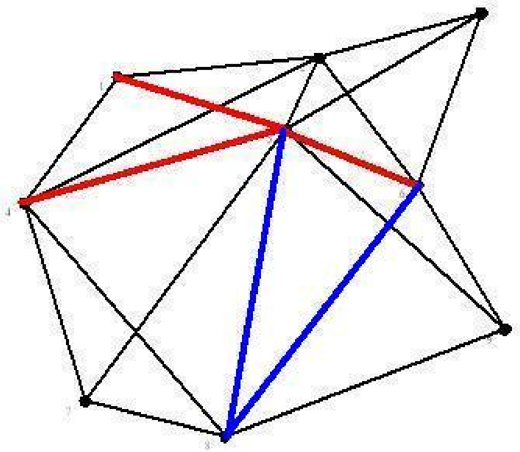
An example with $k = 2$



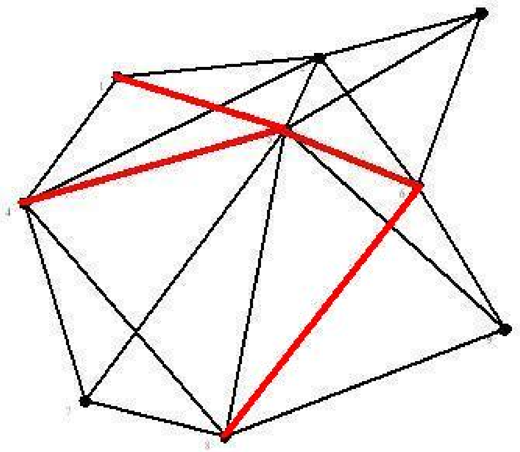
An example with $k = 2$



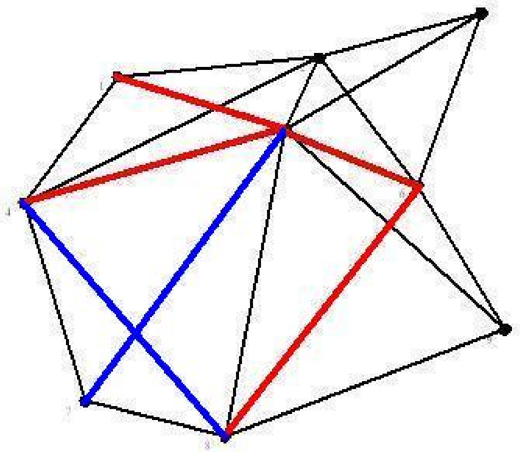
An example with $k = 2$



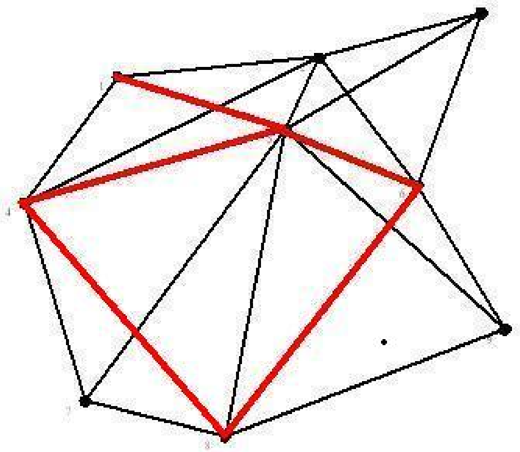
An example with $k = 2$



An example with $k = 2$



An example with $k = 2$



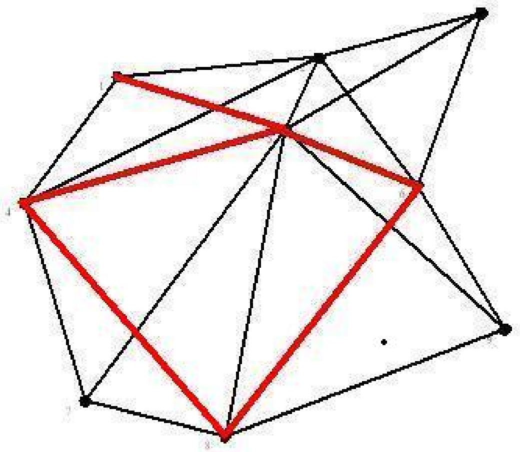
GRASP in RND: improvement phase

Let $RN = \{e_1, ..., e_n\}$ be the best solution found in the construction phase, and let $\overline{RN} = E \setminus RN$.
For i from 1 to n do:

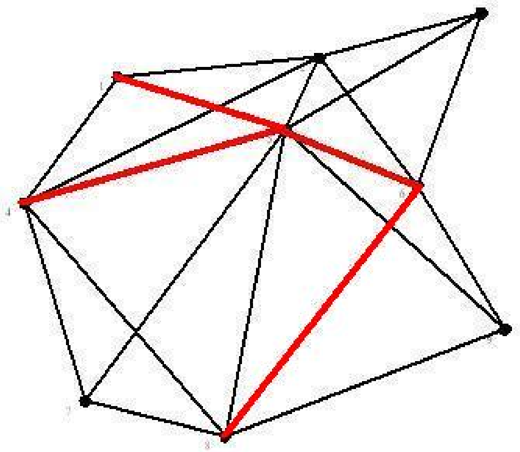
- (Step 1:) Let $RN^i = RN \setminus \{e_i\}$.
- (Step 2:) Let $e_j \in \overline{RN}$ such that it gives the largest improvement in the objective function when added to RN^i among all edges in $\overline{RN}$ that satisfy the cost constraints. Keep doing this while there is budget available.
- (Step 3:) $RN(i) = RN^i \cup \{e_j\}$.

Let i^* be such that $RN(i^*)$ is *not worse* than $RN(j)$, $\forall j \neq i^*$. If $RN(i^*)$ improves RN, then we choose as a final solution $RN(i^*)$. Otherwise we remain with RN.

GRASP in RND: improvement phase, an example



GRASP in RND: improvement phase, an example



GRASP in RND: improvement phase, an example

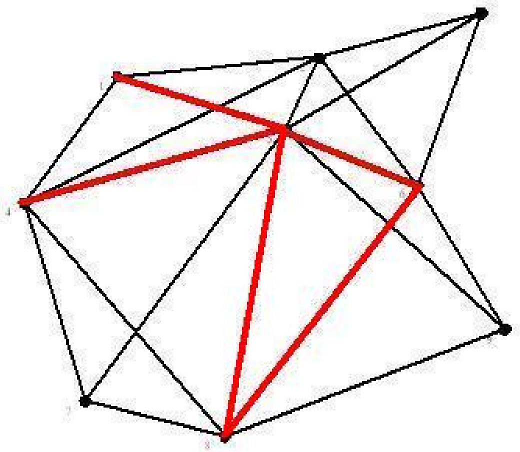
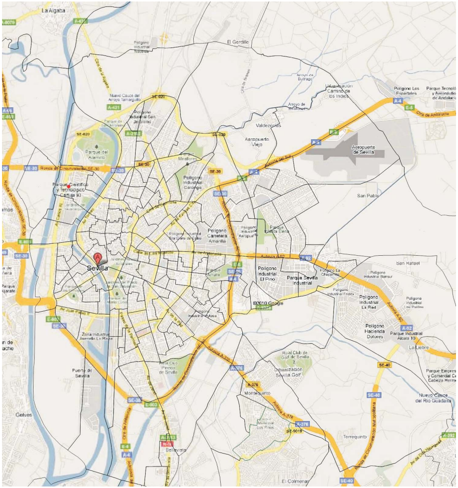
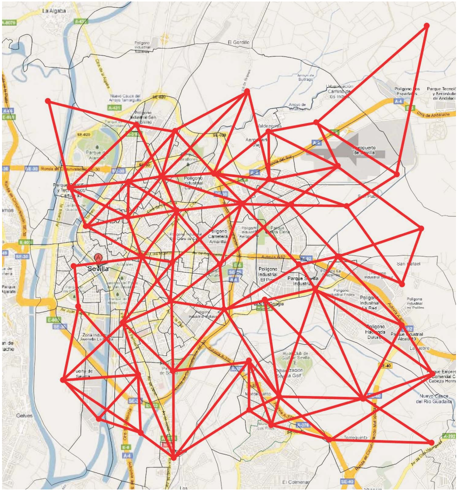


Table of contents

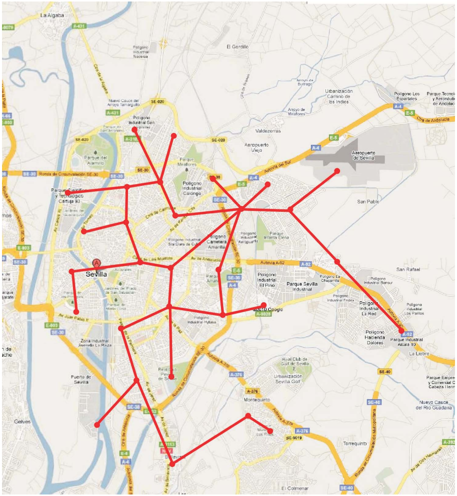
Sevilla divided into areas



Potential links and stations



Solution obtained by GRASP



Next steps...

- Design lines
- Frequencies
- Schedules
- Crew assignment
- So much to be done!

Robust RND problem: example

Now the link *Parc-Acadie* is out of service.

Laurier-Université de Montréal → Bus: 25 min. / Metro: 30 min. → Bus!

Robust approaches

- Now we take into account possible edge failures. Let $K(r,e)$ be the trip coverage of network r if edge e fails
- The new goal could be:
  1. to maximize (over all possible networks) the minimum trip coverage when one of the edges fails,

     $$\max_r \min_e K(r,e), \ (SRND).$$

     (robustness against intentional attacks)
  2. If the probability that edge e fails is known and equal to γ_e, to maximize the expected trip coverage,

     $$\max_r \{(1-\sum_e \gamma_e)K(r) + \sum_e \gamma_e K(r,e)\}, \ (PRND).$$

     (robustness against random failures)

- These two new problems are even more complex than RND. In the corresponding ILP problems, the number of variables and constraints in the previous ILP model is now multiplied times m. The search for heuristics is again a need.
- These models were introduced in Laporte et al. (2010).
- The second case could be considered as robustness against *natural disasters*.
- We will focus on the first case.
- Another possibility is studying robustness by building alternative routes for OD pairs (see Laporte et. al 2011).
- We are now working on robustness to changes in the OD matrix.

Table of contents

RRND as a game in normal form

- "Game Theory studies the behavior of decision-makers (*players*) whose decisions affect each other", Aumann and Hart 1999.
- The robust network design problem against intentional attacks has been modeled as a noncooperative two-person zero-sum game in normal form, where:
  1. Players: $N = \{OPERATOR(PlayerI), ATTACKER(PlayerII)\}$.
  2. Strategies: $S_{OPERATOR} = R$, $S_{ATTACKER} = E$.
  3. Payoffs: $v_{OPERATOR}(r, e) = K(r, e)$, $v_{ATTACKER} = -K(r, e)$.

Test Network

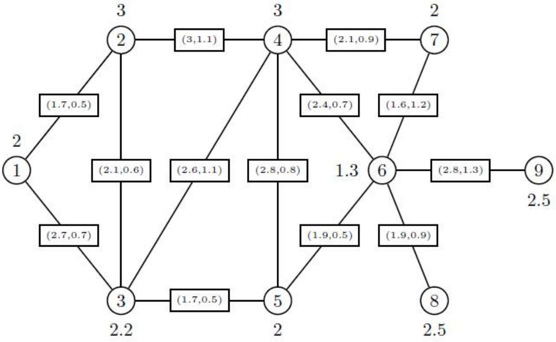


Figure: Test Network.

Example

Network	Edges	TC
r_1	(1,2), (2,3), (3,5), (4,6), (5,6), (6,7), (6,8), (6,9)	831
r_2	(1,2), (1,3), (3,5), (4,6), (4,7), (5,6), (6,8), (6,9)	825
r_3	(1,2), (2,3), (3,5), (4,6), (4,7), (5,6), (6,8), (6,9)	795
r_4	(1,3), (3,4), (3,5), (4,7), (5,6), (6,7), (6,8), (6,9)	792
r_5	(1,3), (2,3), (3,4), (3,5), (4,6), (5,6), (6,7), (6,8)	791
r_6	(1,3), (2,3), (3,4), (3,5), (5,6), (6,8), (6,9)	783

Table: Six first infrastructure networks in terms of trip coverage (TC).

Assuming the minimum Trip Coverage acceptable is 790, the operator only has 5 possible strategies!
In principle, the attacker can choose any of the 13 potential edges.

The following table gives us the payoff of player I (the operator) depending on the network chosen by Player I and the edge chosen by Player II.

	(1,2)	(1,3)	(2,3)	(3,4)	(3,5)	(4,6)	(4,7)	(5,6)	(6,7)	(6,8)	(6,9)
r_1	723	831	629	831	569	657	831	490	674	588	647
r_2	729	596	825	825	548	615	709	461	825	615	641
r_3	687	795	596	795	536	585	679	457	795	585	611
r_4	792	599	792	680	577	792	759	579	735	565	625
r_5	791	588	665	712	670	711	791	655	639	589	791

The payoffs of Player II are the opposite.

- A saddle point is a strategy (r^*, e^*) that satisfies

$$K(r^*, e^*) = \max_{r \in R} \min_{e \in E} K(r, e) = \min_{e \in E} \max_{r \in R} K(r, e)\,, \qquad (6)$$

  and (r^*, e^*) is a Nash equilibrium strategy, which means that no player can benefit by changing its strategy unilaterally.
- If no saddle point exists (which is our case) it is possible for players to enlarge the available set of strategies by considering probability vectors, and look for a saddle point in the enlarged game, in which players can choose a convex combination of their pure strategies, thus defining a *mixed strategy*.

Pure strategies

	(1,2)	(1,3)	(2,3)	(3,4)	(3,5)	(4,6)	(4,7)	(5,6)	(6,7)	(6,8)	(6,9)
r_1	723	831	629	831	569	657	831	490	674	588	647
r_2	729	596	825	825	548	615	709	461	825	**615**	641
r_3	687	795	596	795	536	585	679	457	795	585	611
r_4	792	599	792	680	577	792	759	579	735	565	625
r_5	791	**588**	665	712	670	711	791	655	639	589	791

- MaxMin strategy for the operator: build network r_5. (ensures 588, min with edge $(1,3)$, security level for PI)
- MinMax strategy for the attacker: attack edge $(6,8)$. (615, max with network r_2, security level for PII)
- MaxMin $\neq$ MinMax!!! $\Rightarrow$ no saddle point exists in pure strategies.

Mixed strategies

	(1,2)	(1,3)	(2,3)	(3,4)	(3,5)	(4,6)	(4,7)	(5,6)	(6,7)	(6,8)	(6,9)
r_1	723	831	629	831	569	657	831	490	674	588	647
r_2	729	596	825	825	548	615	709	461	825	**615**	641
r_3	687	795	596	795	536	585	679	457	795	585	611
r_4	792	599	792	680	577	792	759	579	735	565	625
r_5	791	**588**	665	712	670	711	791	655	639	589	791

In behavioral (mixed) strategies, a saddle-point strategy is given if

- player I builds r_1 with probability 0.025, r_2 with probability 0.281 and r_5 with probability 0.694,
- player II attacks edge e_2 with probability 0.079, edge e_{10} with probability 0.112 and edge e_{12} with probability 0.809.

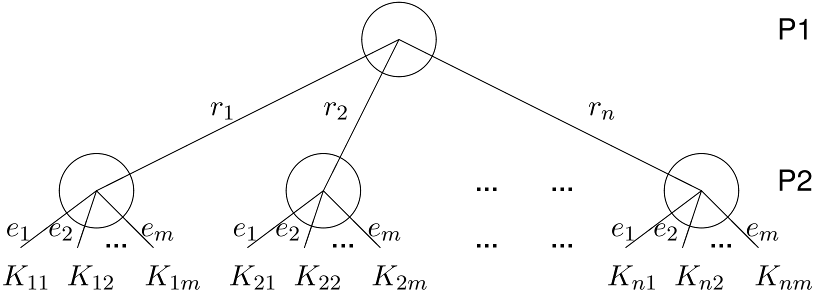
All this results in an expected utility (payoff for the operator) of 596.293 (better than MaxMin).

Table of contents

Dynamic version of the game

- A normal form (as before) may not provide the full picture of the decision process.
- The order in which players act may be important, as well as the information they have available at each moment.
- The *extensive form* of the two-person zero-sum game explicitly displays the dynamic character of the decision problem.
- In our robust transportation decision process, player I first designs the network and player II later attacks.

- Let $r_1, ..., r_n$ be the set of possible networks (strategies) for player I.
- Let $e_1, ..., e_m$ be the set of possible edges (strategies) for player II.
- Denote by $K_{ij} = K(r_i, e_j)$, $i = 1, ..., n, j = 1, ..., m$.
- For PI there are n possibilities (strategies), $\gamma^1 = r_i, \; i = 1, ..., n$.
- For PII, however, because he observes the action of PI, there exist m^n possible strategies. One such strategy is, for instance, $\gamma^2(r_i) = e_1$, for all $i = 1, ..., n$. Another could be $\gamma^2(r_i) = e_1$ if i is even and e_2 otherwise.



Representation of a one-stage game with its information sets.

If we denote by $J(\gamma^1, \gamma^2)$ the payoff of PI when PI and PII employ the strategies γ^1 and γ^2, respectively, we say that $\{\gamma^{1*}, \gamma^{2*}\}$ is in saddle-point equilibrium if

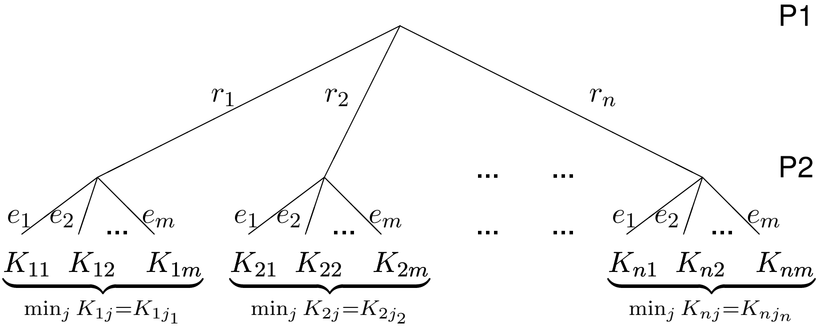
$$J(\gamma^1, \gamma^{2*}) \leq J(\gamma^{1*}, \gamma^{2*}) \leq J(\gamma^{1*}, \gamma^2),$$

and $J^* = J(\gamma^{1*}, \gamma^{2*})$ is known as the saddle-point value of the game.

One way to find a saddle point of the game in extensive form consists of first transforming the game into one in normal form, to then find a saddle-point strategy. Unfortunately, this may lead us to an enormous matrix game! (In our case, an $n \times m^n$ matrix!!). Instead, the method for obtaining the pure strategy of single-act zero-sum games in extensive form in Basar and Olsder (1999) will be adapted, resulting in:

1. For each feasible network r_i, let j_i such that $K_{i,j_i} = \min_j K_{ij}$.
2. Let i^* such that $K_{i^*,j_{i^*}} = \max_i K_{i,j_i}$.
3. $\gamma^{1*} = r_{i^*}$ is the saddle-point strategy of PI (the network operator).
4. $\gamma^{2*}(r_i) = e_{j_i}$ is the saddle-point strategy of PII (the attacker).

- Therefore, $\{\gamma^{1*}, \gamma^{2*}\}$ is the saddle-point strategy of the game, leading to the actions $u^1 = r_{i^*}, u^2 = e_{j_{i^*}}$. The value of the game is $K(r_{i^*}, e_{j_{i^*}})$.
- Note that $K(r_{i^*}, e_{j_{i^*}}) = \max_i K(r_i, e_{j_i}) = \max_i \min_j K(r_i, e_j)$, which is the MaxMin strategy defined before (security level for PI).



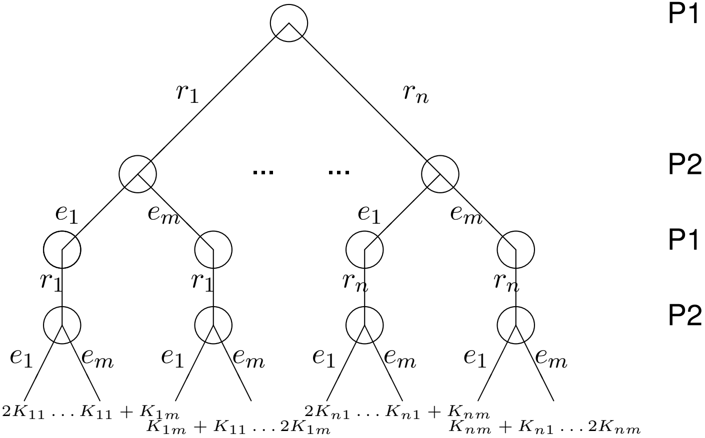
- $K_{i^* j_{i^*}} = \max_i K_{ij_i}$
- $\gamma^{1*} = r_{i^*}$, $\gamma^{2*} = e_{j_i}$ if $u^1 = r_i$.

- Although the previous approach reflects the actual situation better than the game in normal form, a more realistic picture could be modeled.
- The game can be played in several stages!!
- Zero-sum games in which at least one player is allowed to act more than once and with possibly different information sets at each level of play are known as *multi-act zero-sum games*.

- Within this class, our game belongs to the subclass of *feedback games*, which satisfy:
  1. at the time of his action, each player has perfect information concerning the current level of play.
  2. information sets of the first-acting player at every level of play are singletons, and the information sets of the second-acting player at every level of play are such that none of them include nodes corresponding to branches emanating from two or more different information sets of the other player.

- **The designer/operator cannot redesign the network at will!!**
  Player I's strategies are of the form $(\gamma_1^1, ..., \gamma_K^1)$ (γ_k^1 is the strategy of Player I at the k^{th} level of play.) Actually, $\gamma_k^1 = \gamma^1 \; \forall \; k$, since the operator cannot change the network every time Player II decides to attack.
- **The attacker can choose different edges at different stages of the game!!**
  Player II actually has the possibility to attack different edges, and therefore its strategy along the game is $(\gamma_1^2, ..., \gamma_K^2)$, where γ_k^2 for different $k's$ need not be the same, as opposed to PI.

A 2-stage game



- A strategy $(\gamma_1^{1*}, ..., \gamma_K^{1*}; \gamma_1^{2*}, ..., \gamma_K^{2*})$ is a saddle-point strategy if

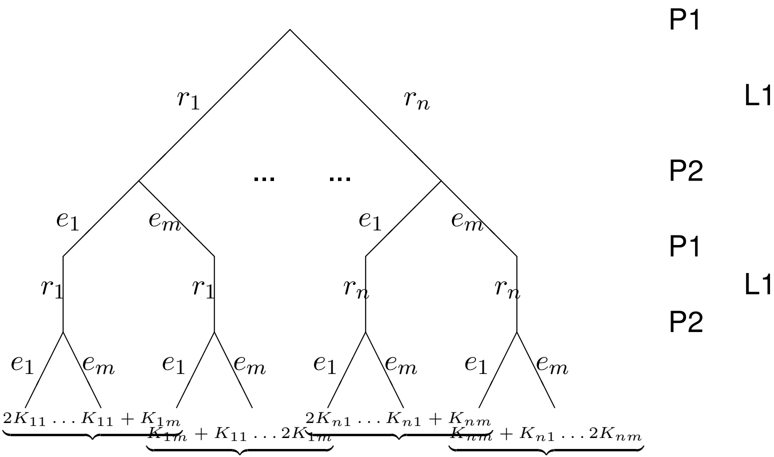
$$\begin{aligned} J(\gamma_1^{1}, ..., \gamma_K^{1}; \gamma_1^{2*}, ..., \gamma_K^{2*}) & \leq J(\gamma_1^{1*}, ..., \gamma_K^{1*}; \gamma_1^{2*}, ..., \gamma_K^{2*}) \\ & \leq J(\gamma_1^{1*}, ..., \gamma_K^{1*}; \gamma_1^{2}, ..., \gamma_K^{2}), \\ & \forall \ \gamma_k^i. \end{aligned}$$

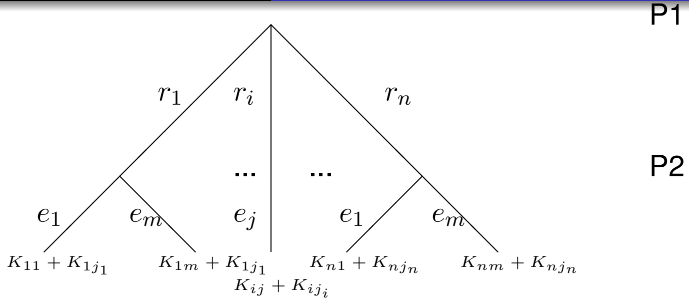
  $J^* = J(\gamma_1^{1*}, ..., \gamma_K^{1*}; \gamma_1^{2*}, ..., \gamma_K^{2*})$ is known as the saddle-point value of the game.
- There is a recursive procedure to determine the saddle-point strategies of a feedback game, Basar and Olsder (1999), which will be adapted to our case.

Finding saddle-point strategies

1. Starting at the last level of play, K, solve each single-act game corresponding to the information sets of the first-acting player at that level. Let $\{\gamma_K^{1*}, \gamma_K^{2*}\}$, and record the value of each of these games.
2. Cross out the K^{th} level of play, and consider the resulting $(K-1)$-level feedback game (with the values stored before). Denote $\{\gamma_{K-1}^{1*}, \gamma_{K-1}^{2*}\}$ the corresponding saddle point strategy, obtained as before.
3. Keep doing this until the fist level of play. The corresponding saddle-point strategy of the whole game and its value follow.

In our case, this results in the strategies $\gamma_k^{1*} = r_{i^*}$ and $\gamma_k^{2^*}(r_i) = e_{j_i} \; \forall \; k$, which leads to PI choosing network r_{i^*} and P2 attacking edge e_{i^*} at each stage of the game.


P1
L1
P2
P1
L1
P2
r_1
r_n
...
...
e_1
e_m
e_1
e_m
r_1
r_1
r_n
r_n
e_1
e_m
e_1
e_m
e_1
e_m
e_1
e_m
$2K_{11} \ldots K_{11} + K_{1m}$
$K_{1m} + K_{11} \ldots 2K_{1m}$
$2K_{n1} \ldots K_{n1} + K_{nm}$
$K_{nm} + K_{n1} \ldots 2K_{nm}$



- $\gamma_1^{1*} = \gamma_2^{1*} = r_{i^*}$
- $\gamma_1^{2*} = \gamma_2^{2*} = e_{j_i}$ if $u^1 = u^2 = r_i$.
- Therefore, the saddle-point actions are

$$u_1^1 = u_2^1 = r_{i^*},\ u_1^2 = u_2^2 = e_{j_{i^*}},$$

  with a value $V = 2K_{i^* j_{i^*}}$.
- Again: MaxMin strategies!!

Table of contents

A joint model for network design and security resource allocation

- In this section we propose a more general game in which player I, the operator, can distribute a certain number of security guards $X \in \mathbb{Z}_+$ over the edges.
- The set of strategies of player I is defined as (r_i, x), where r_i is the network to be built and $x \in \mathbb{Z}_+^m$ is the distribution of security guards, satisfying that $\sum_{j=1}^m x_j \leq X$.
- The probability for an attack made by player II over a certain edge to be successful is a decreasing function on the number of guards located on that edge.
- Therefore, let $\mathcal{K}((r_i, x), e_j)$ be the expected trip coverage of network r_i when edge e_j is attacked and player I distributes its security resources according to x.

As we have justified before, if player II has perfect information about the strategy followed by player I before an attack, the best player I can do is to build its security level strategy, which consists of finding a network $\bar{r}$ and a security guard vector $\bar{x}$ such that

$$\max_{(r_i,x)} \min_{e_j} \mathcal{K}((r_i,x),e_j) = \min_{e_j} \mathcal{K}((\bar{r},\bar{x}),e_j).$$

This problem can be modeled as:

$$\begin{array}{rl} \max & z_{min} + \alpha \sum_{i=1}^{n} z_i + \beta \sum_{i=1}^{n} \sum_{j=1}^{m} \mathcal{K}((r_i, x_i), e_j) \\ \text{s.t.:} & \mathcal{K}((r_i, x_i), e_j) \geq z_i, \; i = 1, ..., n \\ & z_{min} \leq z_i, \; i = 1, ..., n \\ & \sum_{j=1}^{m} x_{ij} \leq X, \; i = 1, ..., n \\ & x_{ij} \in \mathbb{Z}_+, \; i = 1, ..., n; j = 1, ..., m \end{array} \qquad (7)$$

where x_{ij} is the number of guards to be located on edge e_j if r_i is built, z_i is the minimum trip coverage of network r_i when one of the edges fails, and z_{min} is the minimum z_i.
Note the second term in the objective function, which makes the maximization of the average minimum trip coverage of a network when edges are attacked a second objective, and the third term, which makes the average security a third objective. Therefore α and β are small positive numbers with $\alpha \gg \beta$.

An instance

An instance of $\mathcal{K}((r_i, x_i), e_j)$ could be the following. Assume that, if no guards are located on an edge, then the probability of an attack on such edge to be successful is 1, whereas if there are u_j guards the probability of success is 0. Assuming that having a number of guards between 0 and u_j decreases the probability of success linearly, we end up with:

$$\mathcal{K}((r_i, x), e_j) = \begin{cases} K(r_i, e_j) & \text{if } x_{ij} = 0 \\ K(r_i, e_j) + \frac{x_{ij}}{u_j}(K(r_i) - K(r_i, e_j)) & \text{if } 0 < x_{ij} < u_j \\ K(r_i) & \text{if } x_{ij} \geq u_j. \end{cases}$$

Therefore, problem (7) can be written as a mixed integer linear programming problem as follows:

Modeling the instance

$$
\begin{array}{ll}
\max & z_{min} + \alpha \sum_{i=1}^{n} z_i + \beta \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{x_{ij}}{u_j}(K(r_i) - K(r_i, e_j)) \\
\text{s.t.:} & K(r_i, e_j) + \frac{x_{ij}}{u_j}(K(r_i) - K(r_i, e_j)) \geq z_i, \;\; i = 1, ..., n \\
& z_{min} \leq z_i, \;\; i = 1, ..., n \\
& \sum_{j=1}^{m} x_{ij} \leq X, \;\; i = 1, ..., n \\
& x_{ij} \leq u_j, \;\; i = 1, ..., n; j = 1, ..., m \\
& x_{ij} \in \mathbb{Z}_+, \;\; i = 1, ..., n; j = 1, ..., m
\end{array} \tag{8}
$$

Example

As an example of this situation, consider the same network as before, and assume that for any of the links, having 10 security guards guarantees total security and, therefore, no attack is to be successful. Consider as well that the number of guards available is $X = 50$. With this data, a solution to Problem (8) taking $\alpha = 10^{-4}$ and $\beta = 10^{-7}$ is given in Table 2, whereas the expected trip coverage of each network when each of the feasible links is attacked and guards are distributed according to Table 2 is shown in Table 3. We note that both tables are obtained from the solution to problem (8), which is solved in around 0.2 seconds.

Example

x	(1,2)	(1,3)	(2,3)	(3,4)	(3,5)	(4,6)	(4,7)	(5,6)	(6,7)	(6,8)	(6,9)
r_1	3	0	7	0	8	6	0	8	5	7	6
r_2	3	7	0	0	8	7	4	8	0	7	6
r_3	3	0	7	0	8	7	4	8	0	7	6
r_4	0	8	0	6	8	0	0	8	3	9	8
r_5	0	8	6	4	6	4	0	7	7	8	0

Table: Optimal values of variables x_{ij}.

$\mathcal{K}$	(1,2)	(1,3)	(2,3)	(3,4)	(3,5)	(4,6)	(4,7)	(5,6)	(6,7)	(6,8)	(6,9)
r_1	755.4	831.0	770.4	831.0	778.6	761.4	831.0	762.8	752.5	758.1	757.4
r_2	757.8	756.3	825.0	825.0	769.6	762.0	755.4	752.2	825.0	762.0	751.4
r_3	719.4	795.0	735.3	795.0	743.2	732.0	725.4	727.4	795.0	732.0	721.4
r_4	792.0	753.4	792.0	747.2	749.0	792.0	759.0	749.4	752.1	769.3	758.6
r_5	791.0	750.4	740.6	743.6	742.6	743.0	791.0	750.2	745.4	750.6	791.0

Table: Values of expected trip coverage $\mathcal{K}((r_i, x_i), e_j)$ assuming the values of x_{ij} as given before. The minimum expected coverage for each potential network when one of the edges fails is:
$z_1 = 752.5,\ z_2 = 751.4,\ z_3 = 719.4,\ z_4 = 747.2,\ z_5 = 740.6.$

Therefore the security strategy for player I is to build network r_1 with the security distribution showed in Table 2. This way the operator ensures an expected trip coverage of, at least, 752.5 (no matter which edge the attacker decides to attack). Note that in this case the attacker would prefer to attack edge (6,7), since an attack in this edge would produce the highest expected damage in the ridership. Therefore the actions $((r_1, (3, 0, 7, 0, 8, 6, 0, 8, 5, 7, 6)), (6, 7))$ derive a saddle-point strategy.

Table of contents

A model for security resource allocation

- We now assume that the operator has already built network r
- Still the competition game between the operator and the attacker continues.
- The operator can now install a security system over the network that is difficult to modify and known by the attacker.
- This situation is modeled as a Stackelberg game in which the operator is the leader and the attacker is the follower.

- The attacker wants to locate a bomb so that the maximum damage is caused to the network, and the operator wants to design a security system that allows interdicting the possible attacks.
- Let K_j be the cost incurred by the operator if a bomb is successfully detonated on edge e_j, and let $p_j \in [0, 1]$ be the probability that a bomb located on edge e_j is interdicted (both parameters are known by the players). The cost to keep this probability is $c(p_j) = \frac{d_j}{(1-p_j)^{\alpha_j}} - d_j$, ($c(p_j)$ can represent, for instance, the investment in a security system to interdict a bomb on edge e_j with probability p_j), and d_j is the length of edge e_j.

- This cost function has some nice properties $(c(0) = 0, c' > 0, c'' > 0, \lim_{p_j \to 1} c(p_j) = +\infty)$.
- Assuming that the attacker tries to locate a bomb where his expected payoff is maximized, that is, on edge $e_{j'} : j' = \arg\max_j \{(1-p_j)K_j\}$, the defender's objective is to minimize

$$\begin{array}{ll} \min & \sum_{j=1}^{m} c(p_j) + (1-p_{j'})K_{j'} \\ \text{s.t.:} & (1-p_j)K_j \leq (1-p_{j'})K_{j'} \ \forall\, j = 1, ..., m, j \neq j' \\ & p_j \in [0,1]. \end{array} \quad (9)$$

- Note that, in order to solve this problem, we first have to find out which edge $e_{j'}$ is.
- A first idea using brute force would be to solve problem (9) for any possible edge $e_{j'}$.
- The following theorem provides a more suitable way for finding an equilibrium of this game proving that, whenever the costs incurred by the operator when one bomb explodes are sufficiently large (which is a logical assumption) the equilibrium of this game is for the operator to choose its security system so that the expected cost of not interdicting a bomb is constant for every edge.

The theorem

Theorem

Consider an instance of the Stackelberg game defined in this section. If K_j is sufficiently large for all e_j, the equilibrium is for the operator to choose the interdiction probabilities p_j so that $(1 - p_j)K_j$ is constant for all j.

- In other words, this theorem says that if the costs provoked by the attack are large enough, then what the operator should do is to balance its expected loss at all edges, so that the maximum damage is minimized.
- If the costs K_j are small enough, not doing anything might be optimal (that is, make all $p_j = 0$).

Example

- Assume the operator has already designed network r_1
- Assume as well that the loss incurred by the operator if a bomb explodes in edge e_j is $K_j = 1000(K(r_1) - K(r_1, e_j)) = 831000 - 1000K(r_1, e_j)$
- The choice of d_j and α_j is constant and equal to 1 for every j.

As a conclusion to the theorem, the reader may note that an optimal solution to problem

$$\begin{array}{ll} \min & \sum_{j=1}^{m} c(p_j) + z \\ \text{s.t.:} & (1 - p_j)K_j \leq z \; \forall \; j = 1, ..., m \\ & p_j \in [0, 1] \end{array} \qquad (10)$$

coincides with the solution to Problem (9) for any j', and is:

$$p_1^* = 0.988, p_3^* = 0.994, p_5^* = 0.995, p_6^* = 0.993, p_8^* = 0.996,$$
$$p_9^* = 0.992, p_{10}^* = 0.995, p_{11}^* = 0.993, z^* = 1292.672 = K_j(1 - p_j) \; \forall \; j,$$

with an optimal value of 2577.343. This optimal value is the cost incurred by the operator if the attacker explodes the bomb at any edge plus the cost to keep probabilities p_j^*.

Bibliography

- Basar, T., Olsder, G.J., 1999. Dynamic noncoopoerative game theory. In: SIMÕs Classics in Applied Mathematics, 2nd ed. Academic Press, New York.
- Bosco García-Archilla, Antonio J. Lozano, Juan A. Mesa and FP. GRASP algorithms for the robust railway network design problem. JH (2011). Online version.
- Gilbert Laporte, Juan A. Mesa and FP. A game theoretic framework for the robust railway transit network design problem. TRB 44 (2010) 447– 459.

Bibliography

- Gilbert Laporte, Angel Marín, Juan A. Mesa and FP. Designing robust rapid transit networks with alternative routes. JAT (2011) 45:54–65.
- Angel Marín and Ricardo García-Ródenas. Location of infrastructure in urban railway networks. COR 36 (2009) 1461–1477.
- FP and Justo Puerto. Revisiting a game theoretic framework for the robust railway network design against intentional attacks. EJOR 226 (2013) 286–292.