

# L'évacuation avant un tsunami grâce à pgrouting et postgis\_topology



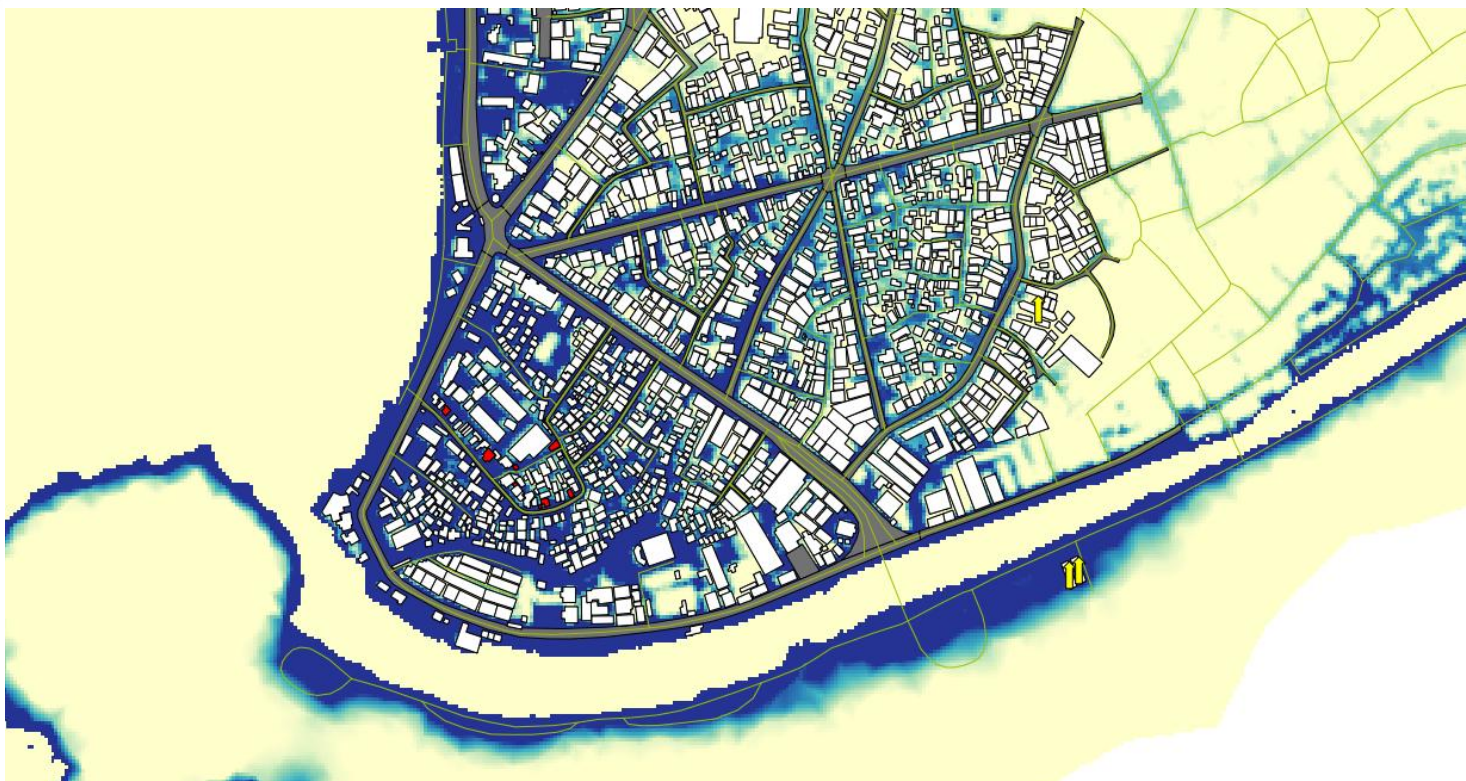
Christine Plumejeaud-Perreau

UMR 7266 Littoral Environnement et Sociétés, LIENSs  
2 rue Olympe de Gouges, La Rochelle  
[Christine.plumejeaud-perreau@univ-lr.fr](mailto:Christine.plumejeaud-perreau@univ-lr.fr)



# Le risque de submersion par Tsunami

- Padang est une ville côtière d'Indonésie qui vit avec un fort aléa de submersion marine. **Thèse Henky Mayaguezz**

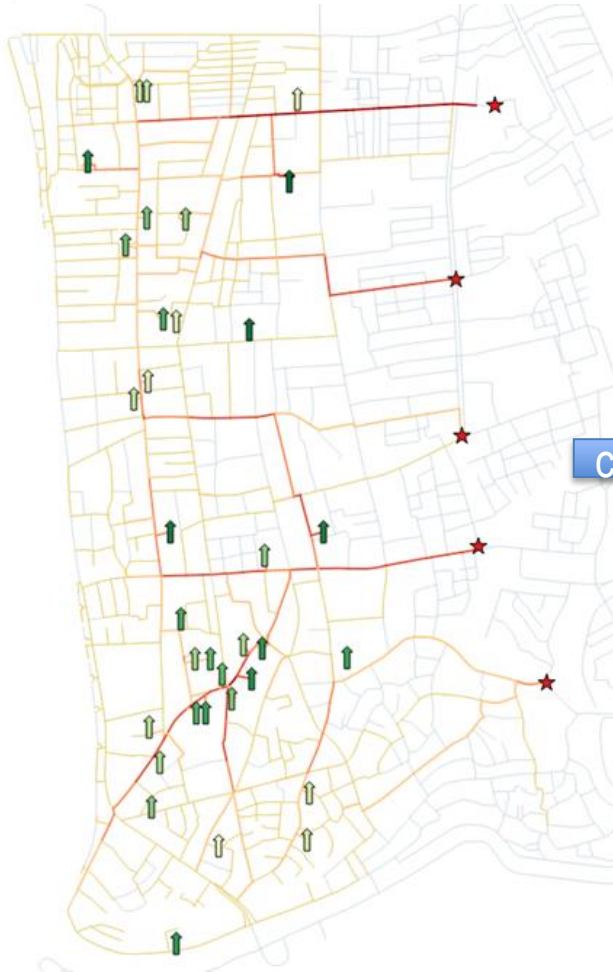


Henky Mayaguezz, Christine Plumejeaud-Perreau, Frédéric Pouget, Frédéric Leone. Evolution spatio-temporelle de l'exposition humaine face au tsunami à Padang: diagnostique de la vulnérabilité et des capacités d'évacuations à l'échelle infra-urbaine. *Revue Internationale de Géomatique*, Lavoisier, 2016, 26 (3), pp.273-306. [hal-01432902](https://hal.archives-ouvertes.fr/hal-01432902)

# Objectif : les refuges verticaux



## Utilisation

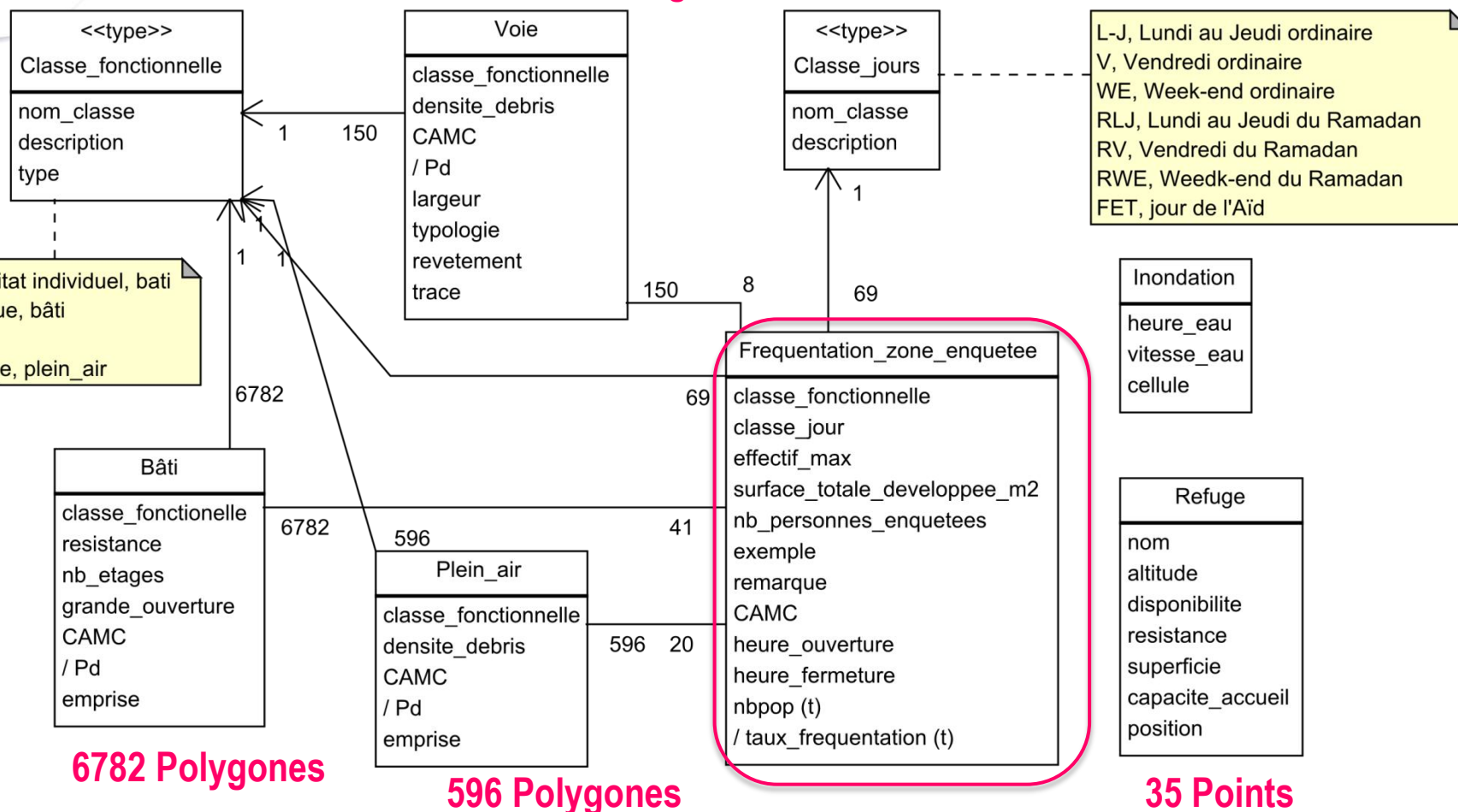


calculer





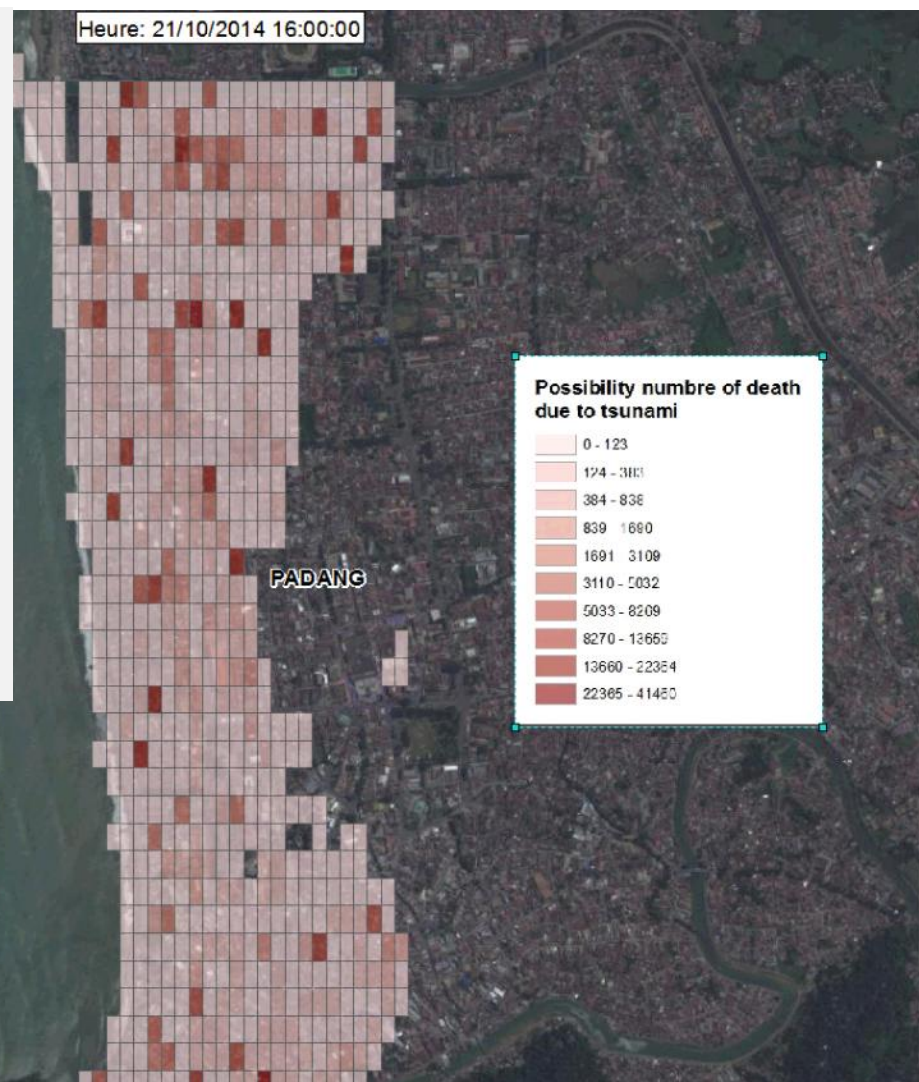
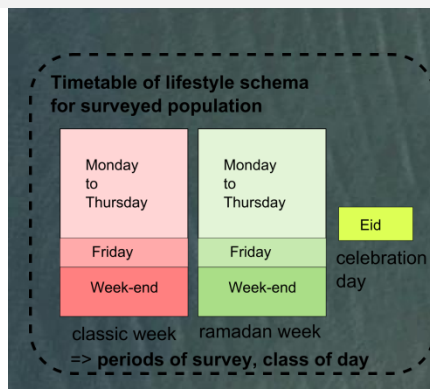
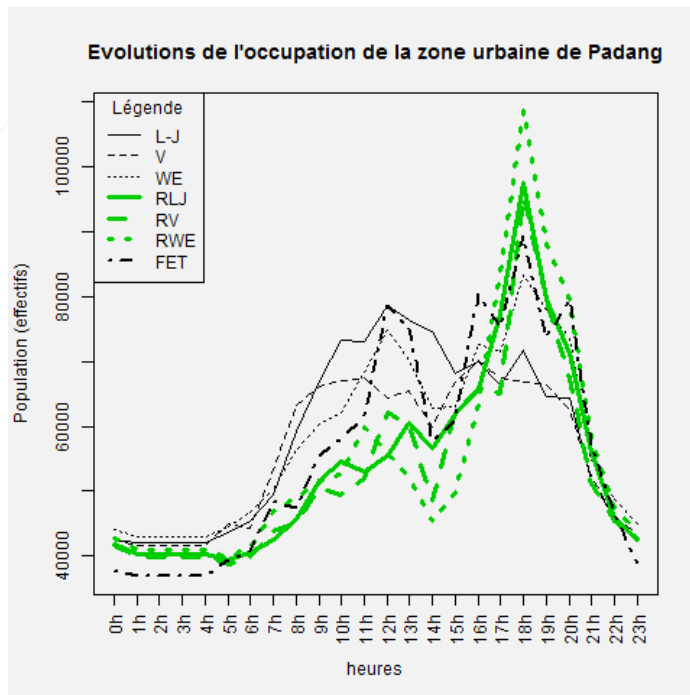
150 Linestrings



# Analyse de la densité de population heure par heure

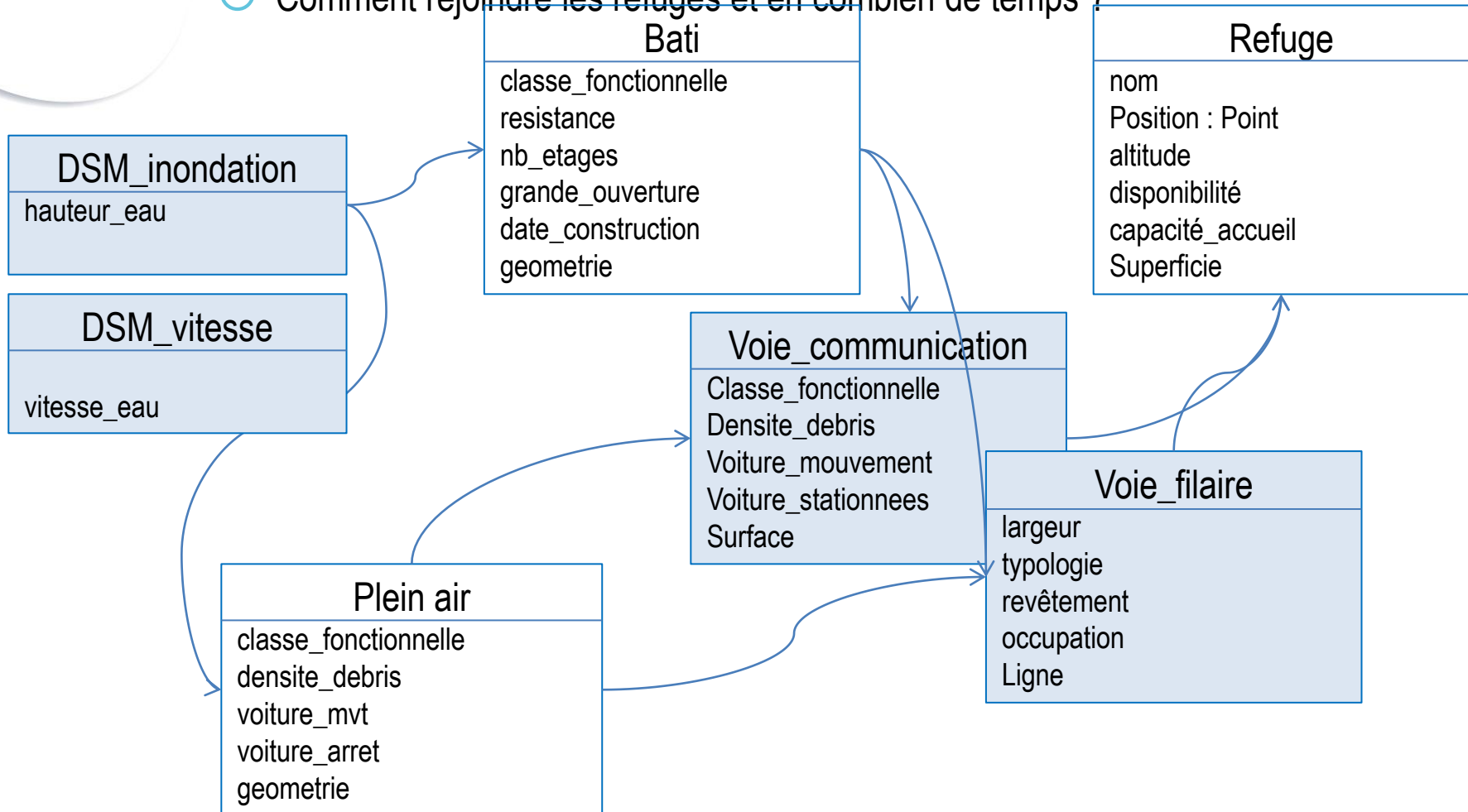
<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Christine.plumejeaud-perreau@univ-lr.fr



# Calculer l'accessibilité des refuges

- Comment rejoindre les refuges et en combien de temps ?



# Postgis\_topology : lignes → graphes

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

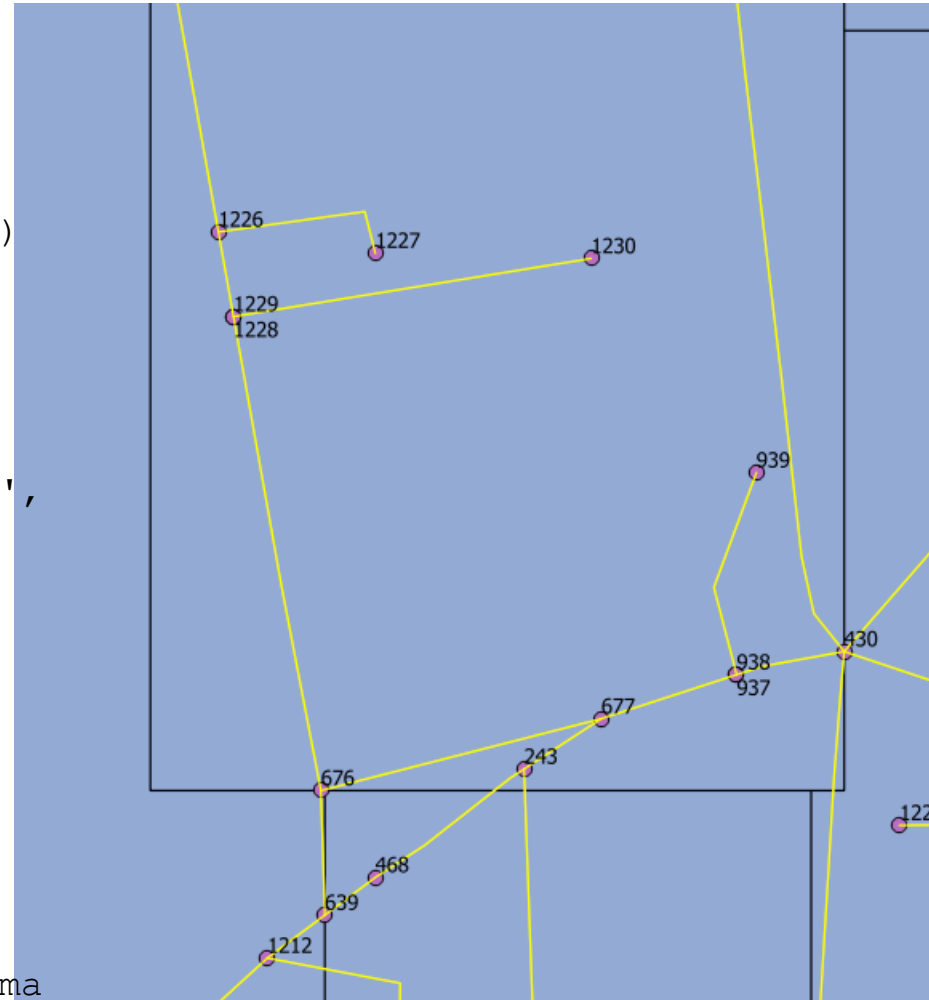
Christine.plumejeaud-perreau@univ-lr.fr

```
select
topology.CreateTopology('voies',32747)

create table voies_line (id integer);

select
topology.addTopoGeometryColumn('voies',
'public', 'voies_line', 'topogeom',
'MULTILINESTRING')
-- retourne 1

insert into voies_line (id, topogeom)
select ogc_fid,
topology.toTopoGeom(wkb_geometry,
'voies', 1) from
raw_data.voie_access_32747;
-- Ce qui a été créé est dans le schéma
voies
```

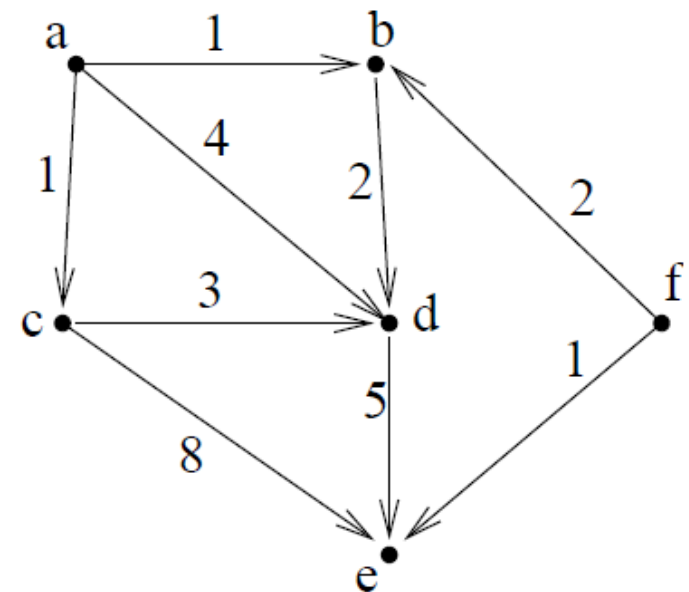




# Pg\_routing

- Projet pgRouting : <http://pgrouting.org>.
  - pgRouting est une extension de la base de données géospatiale [PostGIS/PostgreSQL](#) afin de proposer des fonctionnalités de routage géospatial et d'autres **analyses de réseaux**.
  - <http://docs.pgrouting.org/2.0/fr/doc/index.html> - Le Manuel pgRouting est distribué sous [Creative Commons Attribution-Share Alike 3.0 License](#).

Les algorithmes implémentés (Dijkstra, etc.) reposent sur la **théorie des graphes** et de la recherche opérationnelle. Cette théorie formalise en effet le problème du commis voyageur (recherche de plus court chemin) comme un problème de minimisation du coût total de parcours d'un ensemble d'arcs.





# Application à Padang

## Créer la table graph\_routier pour pgrouting

```
create table graph_routier as
(SELECT v.edge_id as id,
       v.start_node as source,
       v.end_node as target,
       st_length(v.geom)/r.vit_act_cr as cost,
       st_x(n1.geom) as x1, st_y(n1.geom) as y1,
       st_x(n2.geom) as x2, st_y(n2.geom) as y2 ,
       st_length(v.geom)/r.vit_act_pl as reverse_cost,
       v.geom as the_geom
FROM voies.edge v,
      voies.relation vr,
      raw_data.voie_access_32747 r,
      voies.node n1,
      voies.node n2
where vr.element_id=v.edge_id and vr.topogeo_id=r.ogc_fid
      and v.start_node=n1.node_id and v.end_node=n2.node_id
)
```



# Recherche de plus court chemin : Dijkstra



- ⦿ L'algorithme **A\*** (prononcé A étoile) est basé sur l'algorithme de Dijkstra avec une heuristique qui autorise de résoudre la plupart des problèmes de plus court chemin par l'évaluation de seulement un sous-ensemble du graphe général. Retourne un ensemble de ligne **pgr\_costResult(seq, id1, id2, cost)** qui fabriquent un chemin.

```
Pgr_costResult[] pgr_astar(sql text, source integer, target integer, directed boolean, has_rcost boolean)
```

- ⦿ Sur Padang

```
select seq, id1 as node, id2 as edge, cost
from
pgr_astar('select id, source, target, cost, x1, y1,
x2, y2 from graph_routier', 677, 639, false, false)
```

seq	node	edge	cost
0	677	586	73.7932023144824
1	468	588	18.6512287677995
2	639	-1	0

# Illustration

Extrait



# Algorithme simple

- ⊙ Construire le **graphe routier**, avec deux paramètres supplémentaires : la vitesse de déplacement en heure creuse ou pleine sur chaque segment.
- ⊙ Initialiser une relation **chemin** (*identifiant*, *objet*, *nœud\_depart*, *vide*, *refuge*, *nœud\_destination*, *disponible*, *coût\_réseau*, *coût\_hors\_réseau*, *coût\_total*, *séquence de déplacement*) avec les coûts et les séquences vides.
- ⊙ A chaque pas de temps,
  - Mettre à jour les coûts de déplacement sur les arcs du graphe routier.
  - Calculer les chemins les plus courts entre les couples (nœuds départ, nœuds arrivée) pour chaque objet non vide vers chaque refuge disponible. Mettre à jour la relation chemin.
  - Pour chaque objet (bati, voie, zone de plein-air)
    - Sélectionner le refuge dont le cout de transport est minimum pour cet objet.
    - Enregistrer la relation **évacuation** (*pas de temps*, *ce refuge*, *cet objet*, *part du population de l'objet évacuée*, *chemin*).
  - Faire le bilan pour chaque refuge afin de détecter une saturation en population : si la somme des populations recueillies dans le refuge atteint 95% ou plus de la CAMC du refuge, alors le refuge est mis **indisponible**.
  - Faire le bilan pour chaque objet afin de détecter s'il est complètement **vidé** (la somme des parts des populations évacuées est égale à 100% de la population initiale de l'objet).



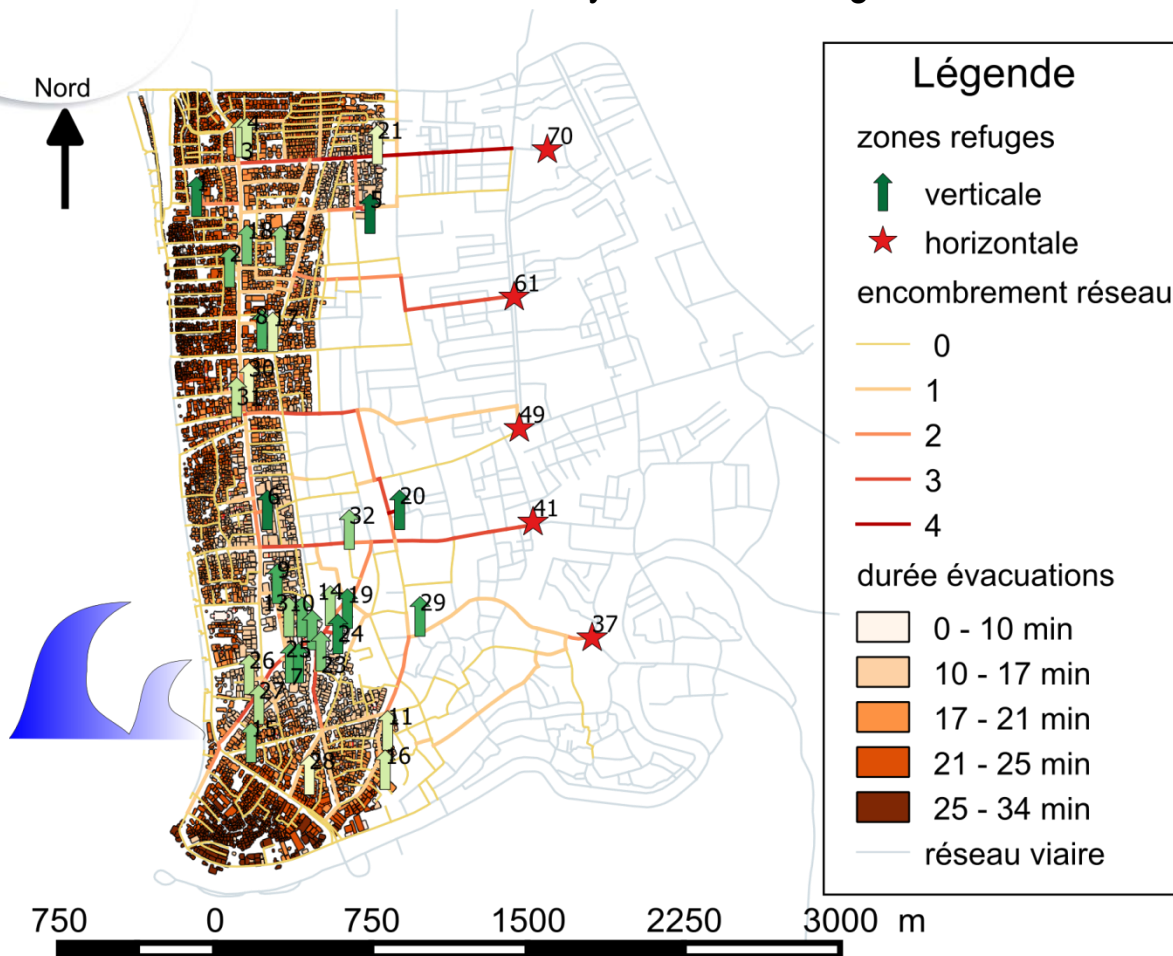
# Paramètres

- ⊙ Les paramètres d'entrée de l'algorithme sont donc :
  1. la classe de jour d'arrivée du tsunami
  2. l'heure d'arrivée du tsunami
  3. le temps d'évacuation maximum
  4. le pas de temps (la minute)
  5. la taille du buffer de sélection des nœuds proches des cibles (100m)
  6. le niveau maximal d'occupation toléré pour les refuges (95%)
  7. la part de personnes évacuées d'une zone à chaque pas de temps : 25% par exemple
  8. *les vitesses de déplacement en heures creuses ou pleines sur les arcs*
  9. *la population présente dans chaque objet à l'heure de déclenchement de l'évacuation*
  10. *le maximum d'encombrement observé sur chaque arc habituellement. Lorsque l'arc sera encombré à hauteur de 50% de cet encombrement, on passe alors en heures pleines.*
- ⊙ Les paramètres 8, 9, 10 sont lus ou calculés dans le schéma de padang :
  - 8 dans les tables `voie_access`,
  - 9 dans `occupation_bati_all`, `occupation_voie_all`, `occupation_plein_air`
  - 10 dans `occupation_voie_all`

# Résultats

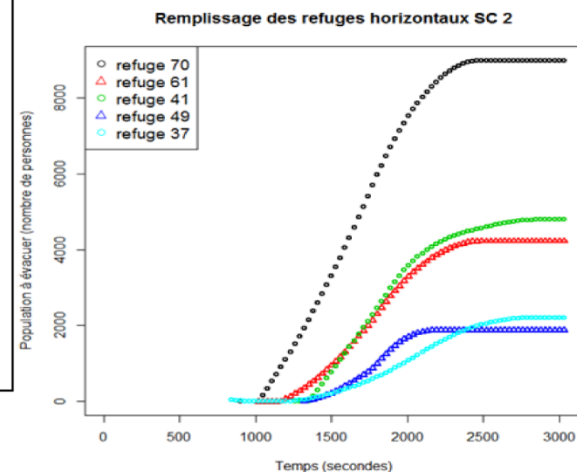


## Accessibilité moyenne des refuges

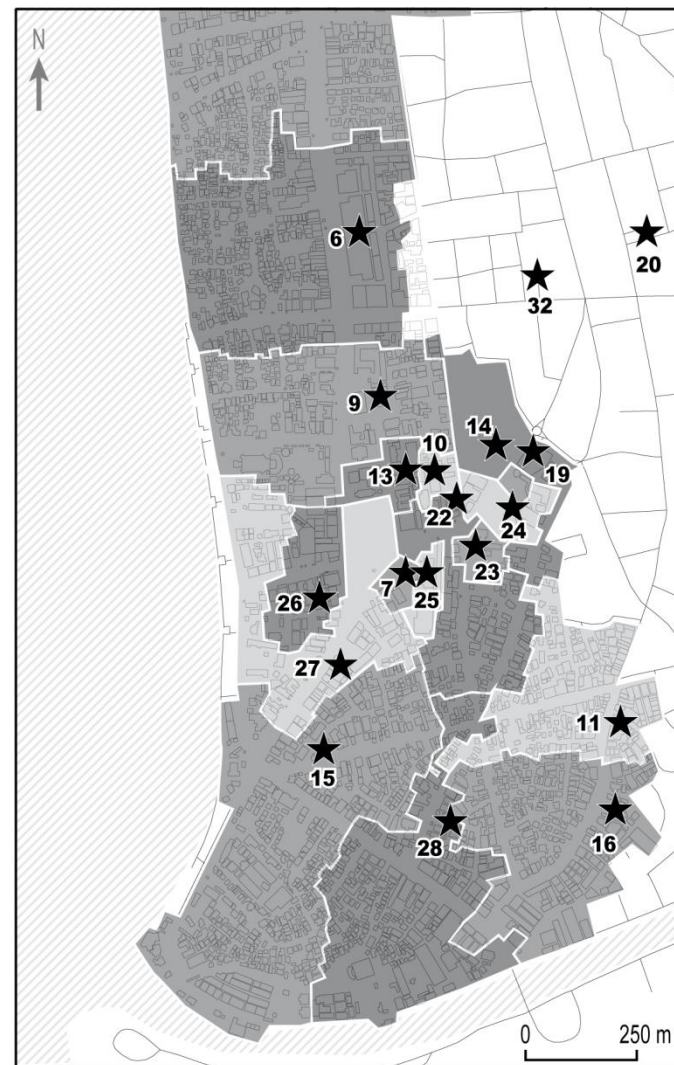
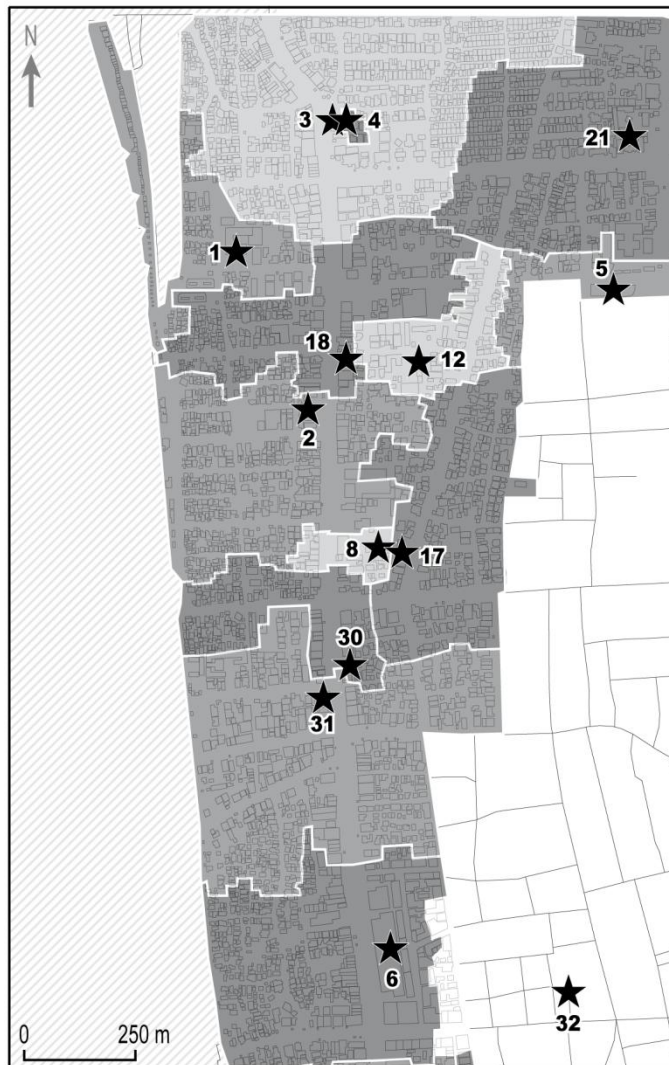


**Simulation d'une évacuation  
à Padang (Indonésie - 75 000 hab.),  
un jour de semaine à 10h.**

**Thèse Henky Mayaguezz**

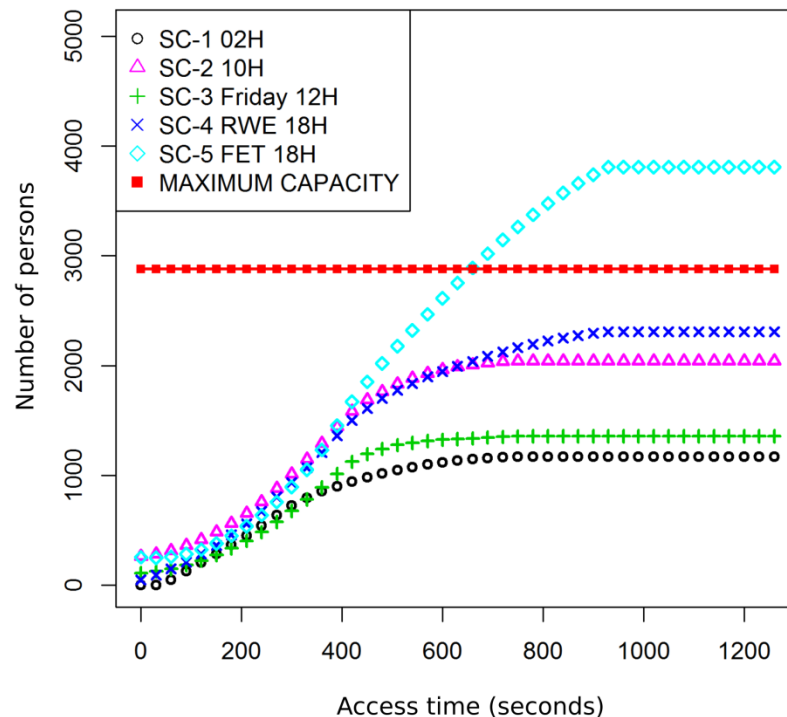


# Couverture spatiale des refuges

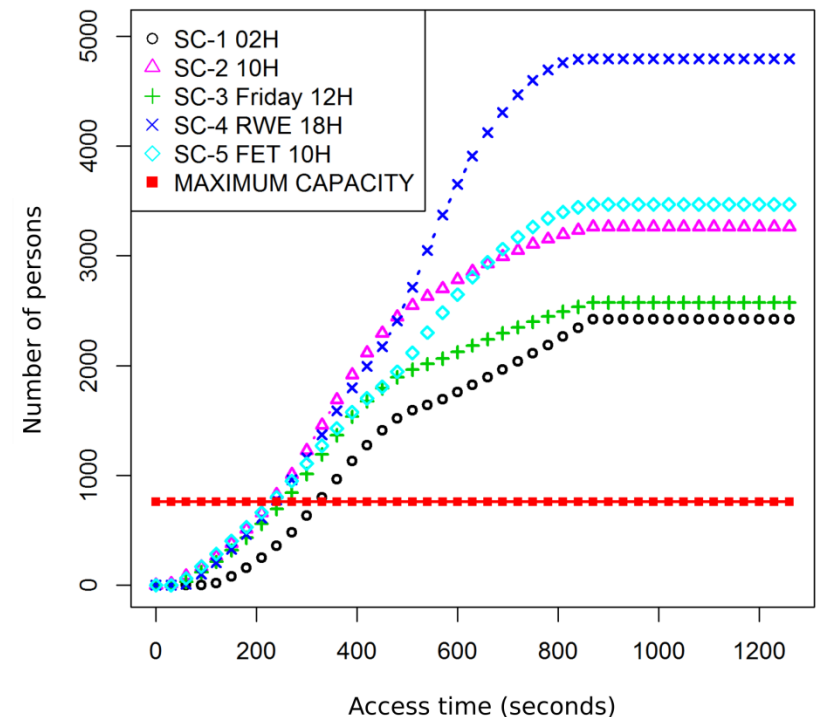


## Diagnostique des refuges en sous-capacité

Accessibility of refuge 1



Accessibility of refuge 2

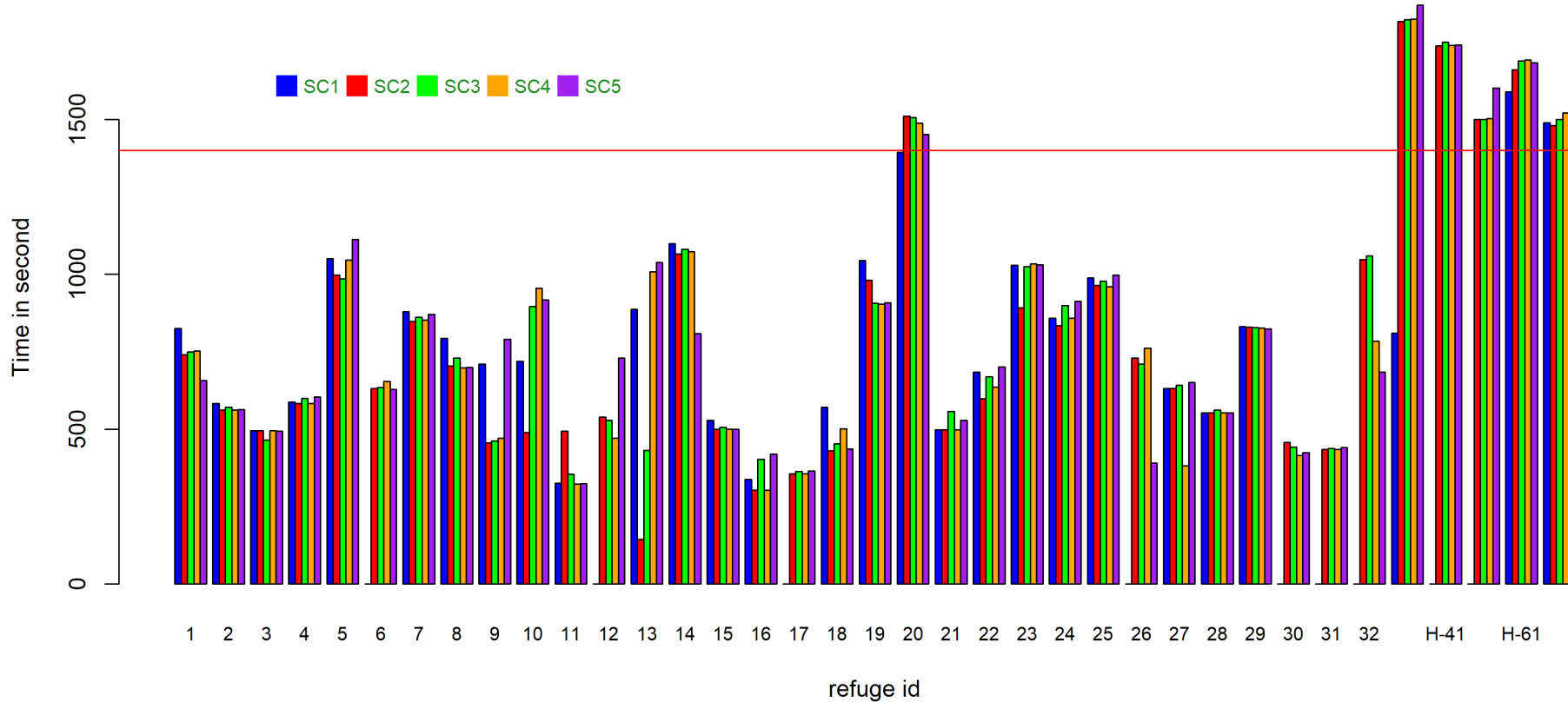




# Accessibilité des refuges

- Temps moyen d'accès au refuges en secondes dans les différents scénarios

Average access time per refuge



# Conclusion

- ⦿ Postgis\_topology : transformation en **graphe** des objets linéaires ou surfaciques
  - Economie de place
  - Vérification de cohérence
  
- ⦿ **PgRouting : analyse de réseaux**
  - Recherche du plus court chemin
  - Et bien d'autres : lire la doc !!
  
- ⦿ <http://www.postgis.fr/chrome/site/docs/workshop-routing-foss4g/docs/pgRoutingWorkshop.pdf>