

Optimisation multicritère de l'extension de réseaux

Soutenance de stage

Millian Poquet

25 septembre 2014



Plan

- ① Présentation du stage
- ② Méthode générale
- ③ Modélisation en problème de flot
- ④ Construction du graphe de l'espace de travail
- ⑤ Formulation en programme linéaire
- ⑥ Résultats
- ⑦ Conclusion

Géo-Hyd

Géo-Hyd

- Société d'études et de services à double compétence **Informatique** et **Environnement**,
- Un cercle vertueux dédié à l'environnement :
 - Acquisition et production de données,
 - Gestion et contrôle de la donnée,
 - Modélisation et représentation des données, **Développement d'outils d'aide à la décision**,
 - Interprétation des données.



Optipipe

Contexte

- Depuis 2013, Géo-Hyd fait partie du groupe Antea,
- Antea organise des concours d'innovation.

Optipipe

- Projet mis au point par Géo-Hyd,
- Vainqueur du prix de l'innovation,
- Optimisation d'un réseau de pipelines.

Aperçu du sujet initial

Aperçu du sujet initial

- Problème d'optimisation,
- Ajout de puits de pétrole et de gaz dans un champ pétrolifère dense et complexe,
- Extension des réseaux de surface existants :
 - Pipelines de pétrole et de gaz,
 - Routes,
 - Électricité,
 - Eau,
- Extension soumise à de nombreuses contraintes...

Aperçu du sujet initial

Contraintes

- Respecter les distances de sécurité entre les différents éléments des réseaux,
- Minimiser le coût et le temps de construction des extensions,
- Minimiser le temps de trajet des employés,
- Minimiser le temps d'intervention sur puits en cas d'urgence,
- Maximiser l'extensibilité du réseau généré,
- Maximiser la robustesse de ce réseau.

Certaines contraintes sont contradictoires...

→ Optimisation multicritère.

Sujet traité

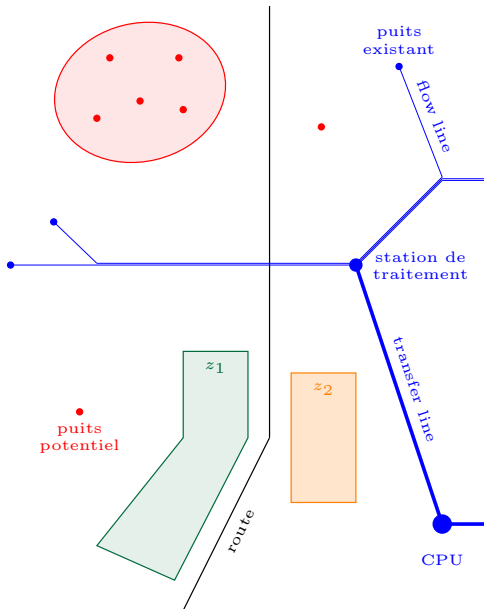
Un sujet difficile

- Plusieurs problèmes NP-difficiles interdépendants,
- Optimisation multicritère.

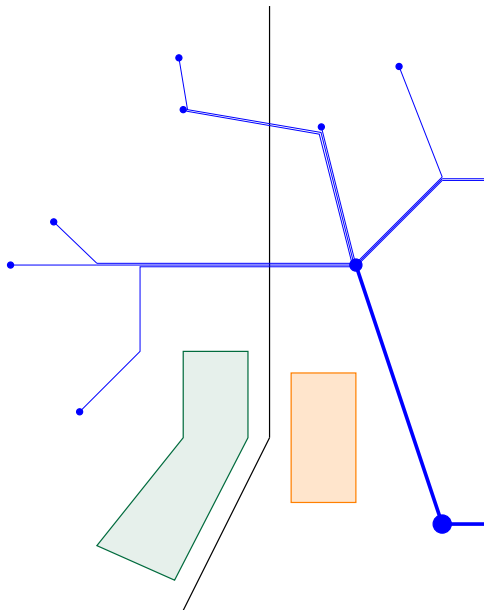
Sujet traité

- Le réseau des pipelines est le plus important,
- On s'est donc concentré dessus tout au long du stage.

Extension du réseau des pipelines



Extension du réseau des pipelines

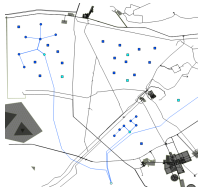


Plan

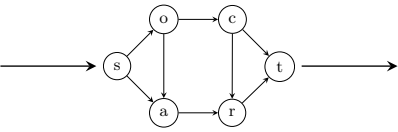
- 1 Présentation du stage
- 2 Méthode générale**
- 3 Modélisation en problème de flot
- 4 Construction du graphe de l'espace de travail
- 5 Formulation en programme linéaire
- 6 Résultats
- 7 Conclusion

Méthode générale

Problème initial



Problème de flots



Problème linéaire

$$\min \sum_{e \in E} v_e x_e + \sum_{e \in E} f_e y_e^0 + \sum_{\substack{e \in E_{it} \\ \text{cap}(e) > 0}} \gamma_e y_e^1$$

$$\text{s.t.} \quad \sum_{i: i \in v \in E} x_{iv} = \sum_{o: o \in v \in E} x_{vo}$$

$$\sum_{o: o \in t \in E} x_{so} = \sum_{i: i \in t \in E} x_{it}$$

$$l_e \leq x_e \leq u_e y_e^0$$

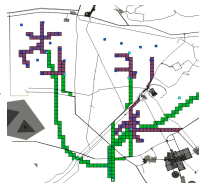
$$x_e = x_e^1 + x_e^2$$

$$l_e \leq x_e$$

$$b_e y_e^1 \leq x_e^1 \leq b_e y_e^0$$



Chemins réels



Solution exploitable

x_t4154_t3098	3
x_t3103_t3130	3
x_t3130_t3173	3
x_t3218_t5557	18
x_t4902_t3218	12
x_t6341_t3442	6
x_t3442_t5605	6
x_t3574_t3456	4

Solution au problème linéaire

Plan

- 1 Présentation du stage
- 2 Méthode générale
- 3 Modélisation en problème de flot**
 - Qu'est-ce qu'un problème de flot ?
 - Le type de problème de flot retenu
 - Construction du graphe de flot
- 4 Construction du graphe de l'espace de travail
- 5 Formulation en programme linéaire
- 6 Résultats

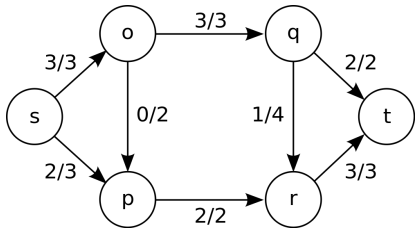
Qu'est-ce qu'un problème de flot ?

Problème de flot

- Problème sur un graphe de flot...

Graphe de flot

- Graphe orienté,
- Un **flot** passe sur chaque arête,
- Ce flot est borné par une **capacité**,
- Conservation du flot,
- **source** : émet un flot,
- **puits** : reçoit un flot.



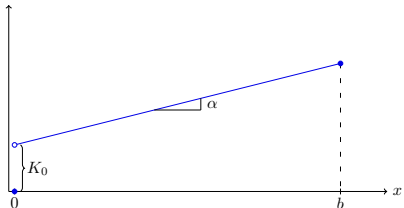
Problème de flot maximum

Quel problème de flot nous intéresse ?

Problème de flot de coût minimum

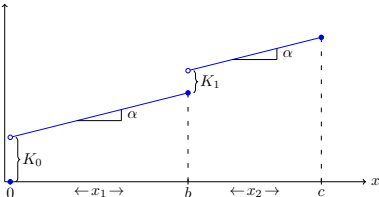
- Un coût est attribué à chaque arête,
- Minimisation du coût total,
- Présence forcée de certains flots.

$$c(x, \alpha, K_0, b)$$



Arête non initialement existante

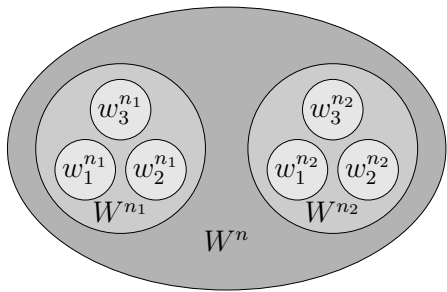
$$c(x, \alpha, K_0, K_1, b, c)$$



Arête initialement existante

Modèle

- Modèle**
- Créer un graphe de flot $G = (V, E)$,
 - Une supersource s , un superpuits t ,
 - L'ensemble des puits/stations existants W et S ,
 - Des puits/stations potentiels W^n et S^n ,
 - Un graphe G_M .



Les puits potentiels W^n regroupés dans les sous-ensembles W^{n_1} et W^{n_2} . Chaque sous-ensemble a une contrainte de flot minimum.

Deux types d'extension

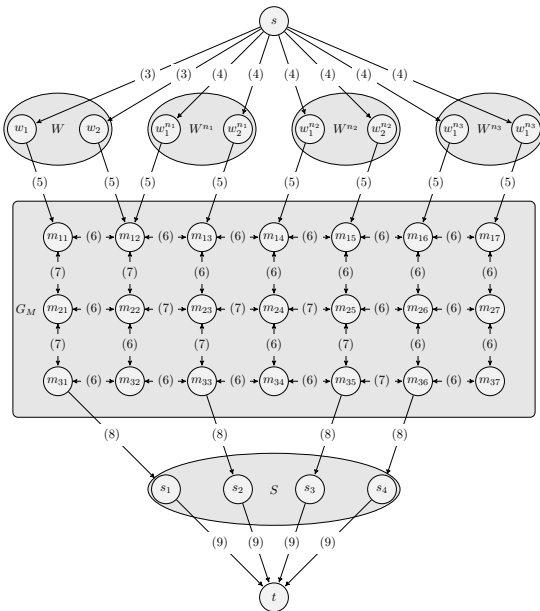
Extension simple

- Ajout de puits de pétrole/gaz,
- Création de flow lines (entre les puits et les stations intermédiaires).

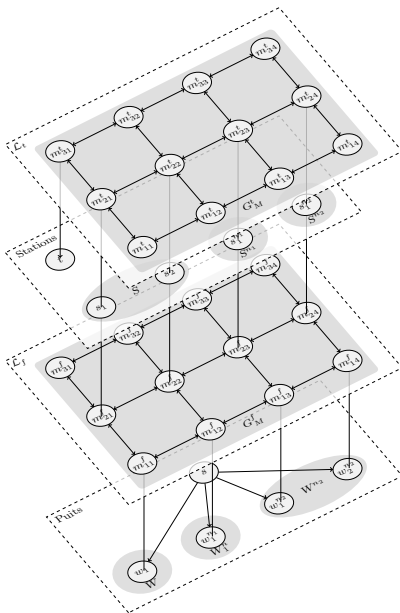
Extension avancée

- Ajout de stations intermédiaires,
- Ajout de transfer lines (entre les stations intermédiaires et le CPU).

Extension simple



Extension avancée

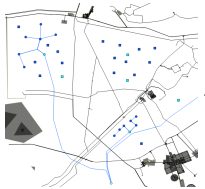


Plan

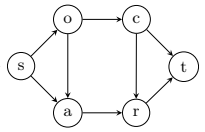
- 1 Présentation du stage
- 2 Méthode générale
- 3 Modélisation en problème de flot
- 4 Construction du graphe de l'espace de travail**
- 5 Formulation en programme linéaire
- 6 Résultats
- 7 Conclusion

Méthode générale

Problème initial



Problème de flots

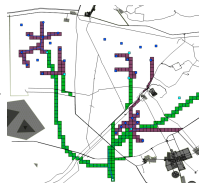


Problème linéaire

$$\begin{aligned}
 \min \quad & \sum_{e \in E} v_e x_e + \sum_{e \in E} f_e y_e^0 + \sum_{\substack{e \in E_{it} \\ \text{cap}(e) > 0}} \gamma_e y_e^1 \\
 \text{s.t.} \quad & \sum_{i: i \in v(e)} x_{ie} = \sum_{o: o \in e} x_{eo} \\
 & \sum_{o: o \in E} x_{so} = \sum_{i: i \in E} x_{it} \\
 & l_e \leq x_e \leq u_e y_e^0 \\
 & x_e = x_e^1 + x_e^2 \\
 & l_e \leq x_e \\
 & b_e y_e^1 \leq x_e^1 \leq b_e y_e^0
 \end{aligned}$$



Chemins réels



Solution exploitable

x_t4154_t3098	3
x_t3103_t3130	3
x_t3130_t3173	3
x_t3218_t5557	18
x_t4902_t3218	12
x_t6341_t3442	6
x_t3442_t5605	6
x_t3574_t3456	4

Solution au problème linéaire

Petit détail sur les coûts

Petit détail sur les coûts

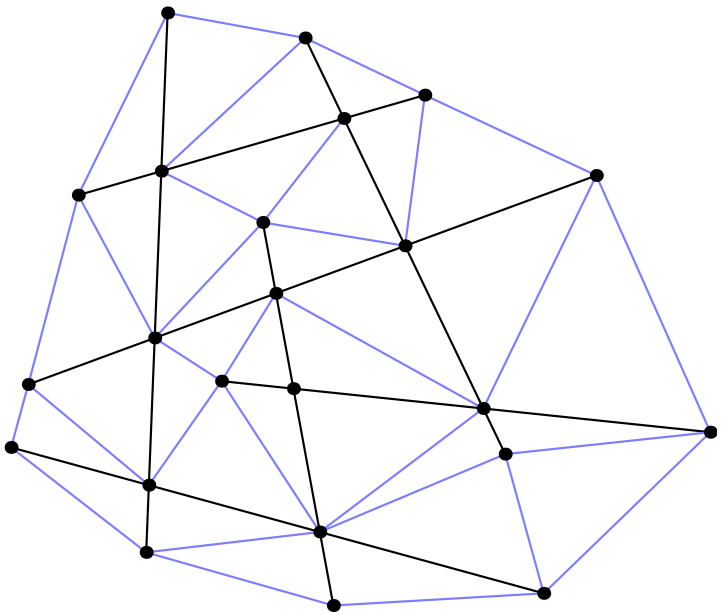
- Croisement pipeline/route très coûteux,
- Routes représentées par des segments.

Partitionnement du plan

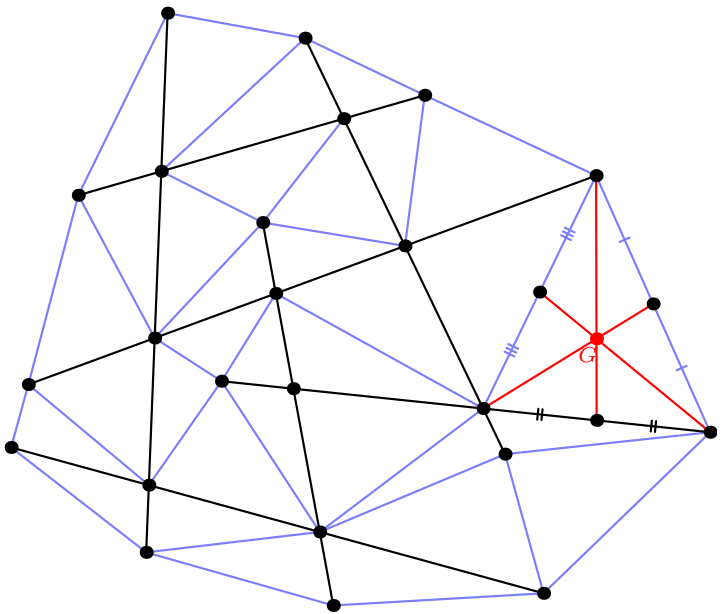
Partitionnement du plan

- Partitionnement du plan en faces,
- Le graphe dual à cette partition forme un graphe de parcours.

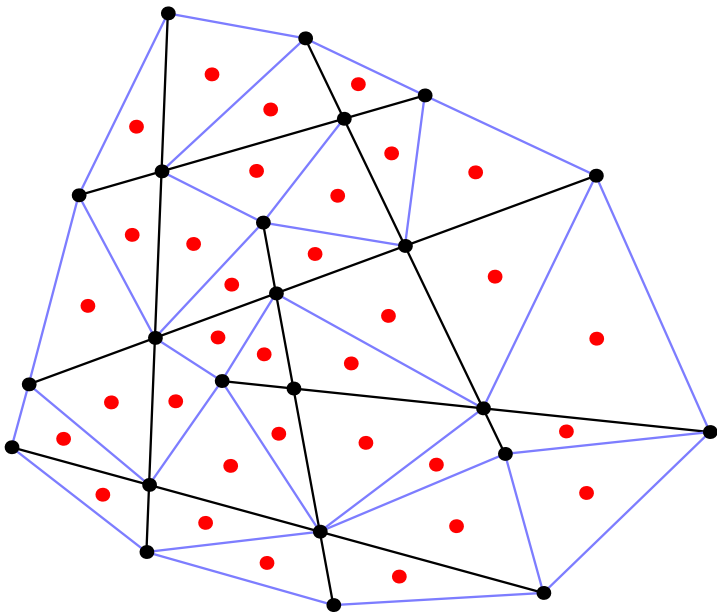
Partitionnement du plan



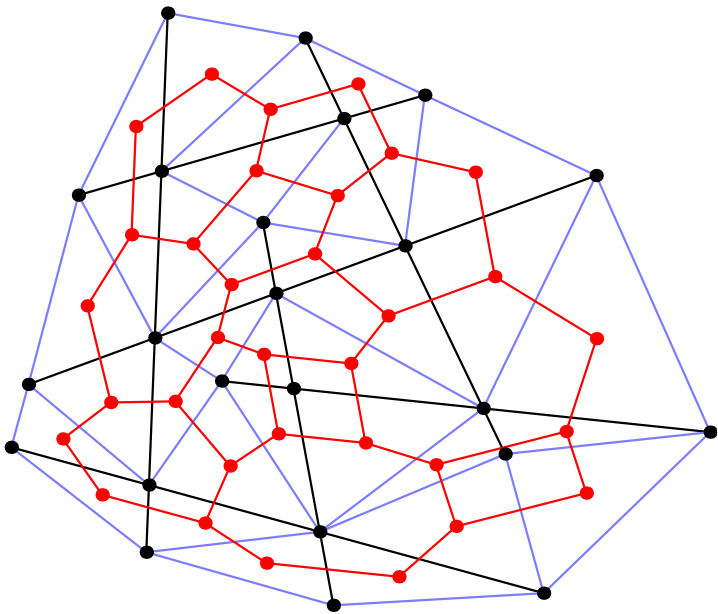
Partitionnement du plan



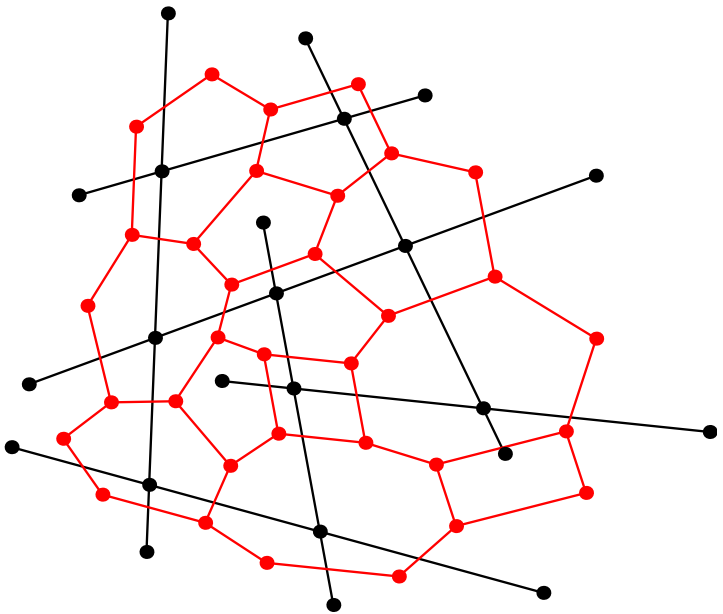
Partitionnement du plan



Partitionnement du plan



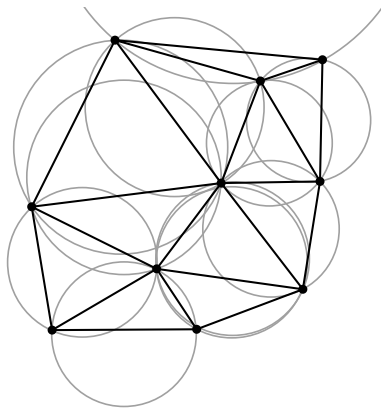
Partitionnement du plan



Triangulation de Delaunay

Triangulation de Delaunay

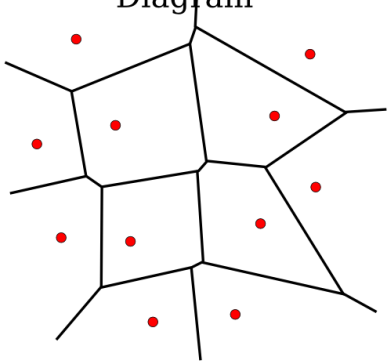
- Triangulation d'un ensemble de points P ,
- Aucun point de P n'est à l'intérieur du cercle circonscrit d'un des triangles,
- Évite les triangles allongés,
- Graphe dual au diagramme de Voronoi de P .



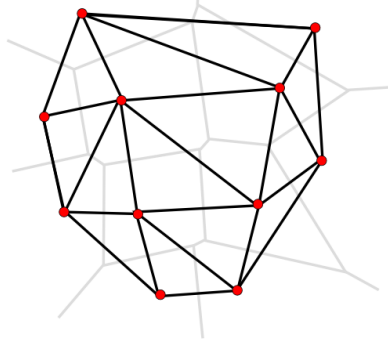
Triangulation de Delaunay et cercles circonscrits (gris)

Triangulation de Delaunay et diagramme de Voronoi

Voronoi
Diagram



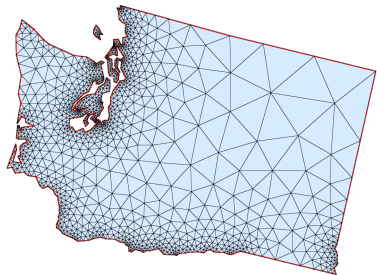
Delaunay
Triangulation



Triangulation de Delaunay contrainte

Triangulation de Delaunay contrainte

- Certaines arêtes ont leur présence forcée dans la solution,
- On essaye de trianguler le reste des points de manière la *plus Delaunay possible*,
- → La triangulation résultante n'est pas forcément une triangulation de Delaunay.

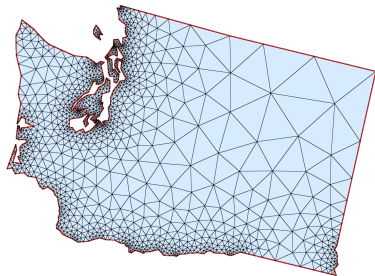


Triangulation de Delaunay contrainte

Triangulation de Delaunay contrainte

Solution non retenue

- L'aire des triangles varie beaucoup,
- Trop grands triangles → peu d'information sur l'occupation du sol par les pipelines,
- De trop nombreux petits triangles → complexité en temps et en mémoire...

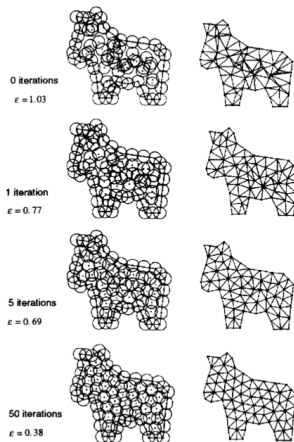


Triangulation de Delaunay contrainte

Bubble mesh

Bubble mesh[SG95]

- Simulation physique de bulles,
- Bulles initiales le long des arêtes (arbre) et à l'intérieur des faces (quadtree),
- Les bulles sont modifiées et déplacées (elles peuvent apparaître ou disparaître),
- Une triangulation de Delaunay contrainte permet d'obtenir une solution.

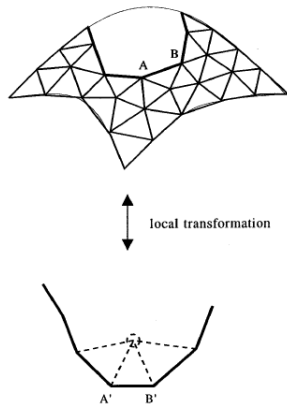


Nombre d'itérations et qualité du mesh généré

NETGEN

NETGEN[Sch97]

- Méthode constructive,
- Un front est construit sur la bordure de chaque domaine,
- Le front ferme progressivement chaque domaine en le triangulant,
- Utilise des règles abstraites de construction.



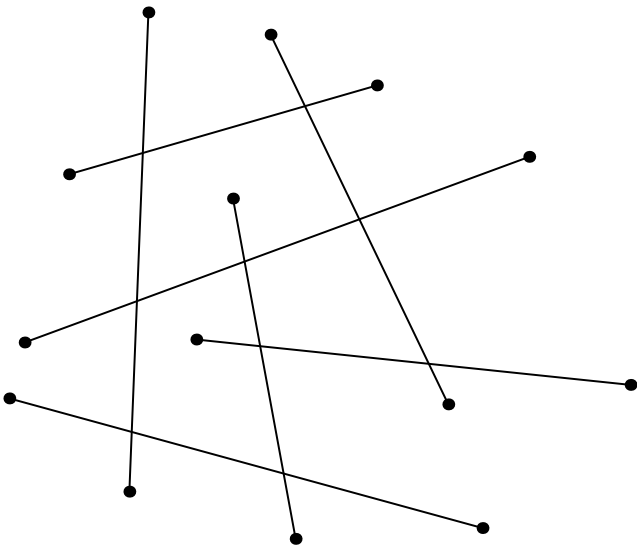
Front de NETGEN

Arrangement de segments

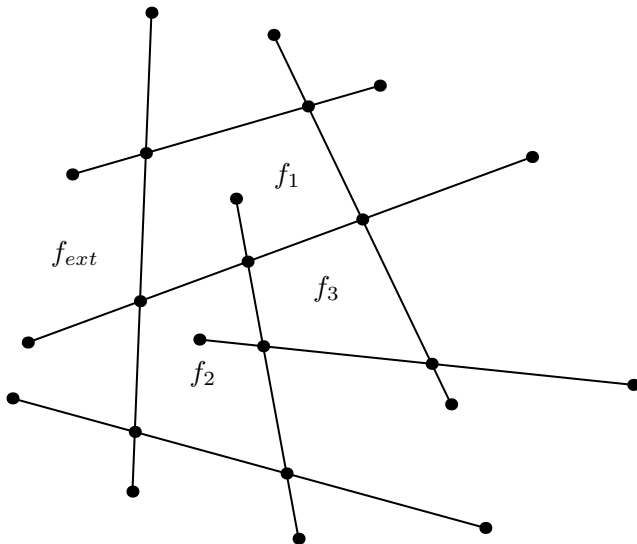
Arrangement de segments

- Entrée
 - Un ensemble de segments.
- Sortie
 - Un ensemble de sommets, de segments et de faces,
 - Des relations d'incidence entre les cellules.

Arrangement de segments



Arrangement de segments



Arrangement de segments

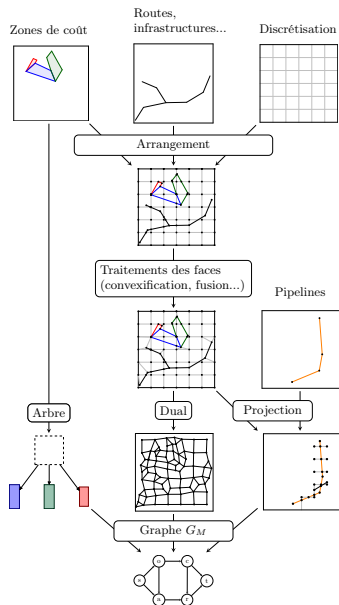
Algorithmes existants

- Bentley-Ottmann[BO79] (intersection de segments) : sweep line,
- Clarkson[CS89] et Mulmuley[Mul90] : composante aléatoire (de type quicksort), complexité optimale en moyenne,
- Chazelle-Edelsbrunner[CE92] : complexité optimale en pire cas mais difficile à implémenter[Hob99].

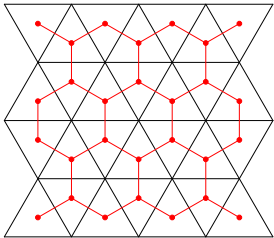
Cas dégénérés

- Ne gèrent pas les cas dégénérés...
 - Deux points sur la même abscisse,
 - Intersection de 3 segments ou plus au même point,
 - ...
- Problème relativement simple par rapport au problème du stage → implémentation naïve.

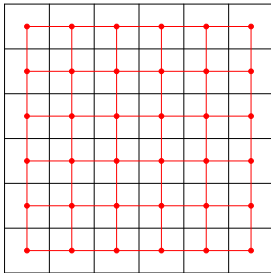
Solution retenue



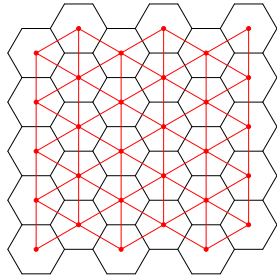
Comment discrétiser l'espace de travail ?



Pavage triangulaire



Pavage carré



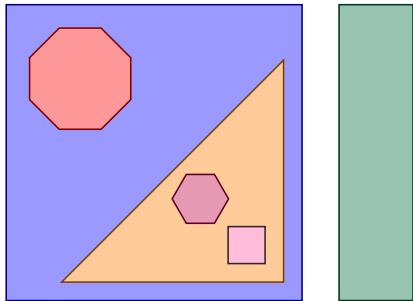
Pavage hexagonal.

Zones de coûts

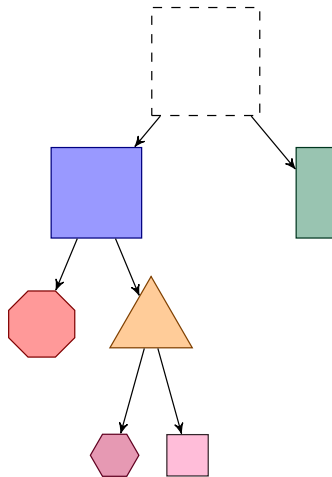
Zones de coûts

- Une zone est délimitée par un polygone,
- Une zone peut être dans une autre mais elles ne peuvent se chevaucher,
- On peut ainsi les organiser en arbre,
- Plusieurs types de zones
 - **Absolue** : les coûts de la zone z sont définis indépendamment de la zone parente de z ,
 - **Additive** : les coûts de la zone z sont ajoutés à ceux de sa zone parente,
 - **Multiplicative** : les coûts de la zone z sont des facteurs de ceux de sa zone parente.

Zones de coût



Représentation des zones dans l'espace de travail



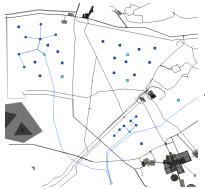
Représentation sous forme d'arbre

Plan

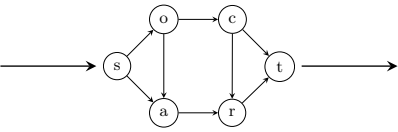
- 1 Présentation du stage
- 2 Méthode générale
- 3 Modélisation en problème de flot
- 4 Construction du graphe de l'espace de travail
- 5 Formulation en programme linéaire**
- 6 Résultats
- 7 Conclusion

Méthode générale

Problème initial



Problème de flots



Problème linéaire

$$\min \sum_{e \in E} v_e x_e + \sum_{e \in E} f_e y_e^0 + \sum_{\substack{e \in E_{it} \\ \text{cap}(e) > 0}} \gamma_e y_e^1$$

$$\text{s.t.} \quad \sum_{i: i \in v \in E} x_{iv} = \sum_{o: o \in v \in E} x_{vo}$$

$$\sum_{o: o \in t \in E} x_{so} = \sum_{i: i \in t \in E} x_{it}$$

$$l_e \leq x_e \leq u_e y_e^0$$

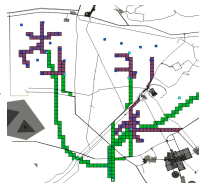
$$x_e = x_e^1 + x_e^2$$

$$l_e \leq x_e$$

$$b_e y_e^1 \leq x_e^1 \leq b_e y_e^0$$



Chemins réels



Solution exploitable

x_t4154_t3098	3
x_t3103_t3130	3
x_t3130_t3173	3
x_t3218_t5557	18
x_t4902_t3218	12
x_t6341_t3442	6
x_t3442_t5605	6
x_t3574_t3456	4

Solution au problème linéaire

Programme linéaire

$$\min \sum_{e \in E} v_e x_e + \sum_{e \in E} f_e y_e^0 + \sum_{\substack{e \in E_M \\ \text{cap}(e) > 0}} \gamma_e y_e^1 \quad (1)$$

$$\text{s. t.} \quad \sum_{i:iv \in E} x_{iv} = \sum_{o:vo \in E} x_{vo} \quad \forall v \in V \setminus \{s, t\} \quad (2)$$

$$\sum_{o:so \in E} x_{so} = \sum_{i:it \in E} x_{it} \quad (3)$$

$$l_e \leq x_e \leq u_e y_e^0 \quad \forall e \in E, \neg(e \in E_M \wedge \text{cap}(e) > 0) \quad (4)$$

$$x_e = x_e^1 + x_e^2 \quad \forall e \in E_M, \text{cap}(e) > 0 \quad (5)$$

$$l_e \leq x_e \quad \forall e \in E_M, \text{cap}(e) > 0 \quad (6)$$

$$b_e y_e^1 \leq x_e^1 \leq b_e y_e^0 \quad \forall e \in E_M, \text{cap}(e) > 0 \quad (7)$$

$$0 \leq x_e^2 \leq (u_e - b_e) y_e^1 \quad \forall e \in E_M, \text{cap}(e) > 0 \quad (8)$$

$$y_e^0 \in \{0, 1\} \quad \forall e \in E \quad (9)$$

$$y_e^1 \in \{0, 1\} \quad \forall e \in E_M, \text{cap}(e) > 0 \quad (10)$$

$$\sum_{w_i^{nk} \in W^{nk}} x_{sw_i^{nk}} \geq L_W^{nk} \quad \forall W^{nk} \in W^n \quad (11)$$

$$\sum_{\sigma_i^{nk} \in S^{nk}} \sigma_{\sigma_i^{nk}} \geq L_S^{nk} \quad \forall S^{nk} \in S^n \quad (12)$$

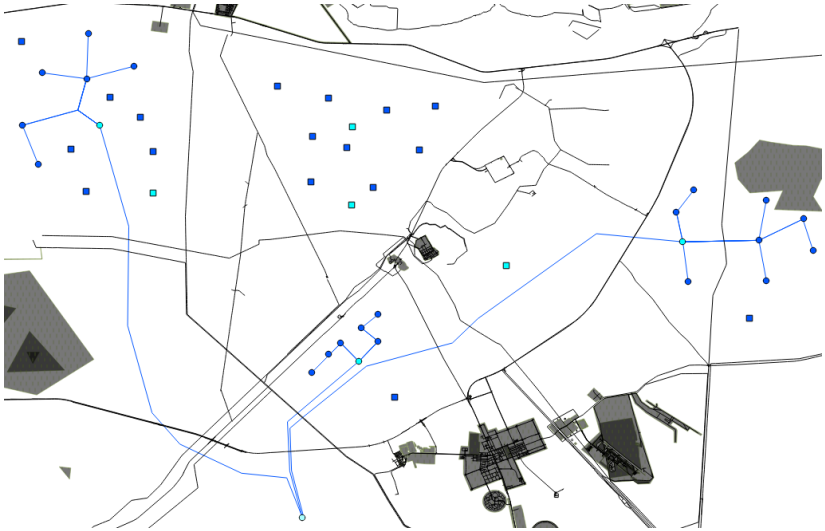
$$x_{ij} + x_{ji} \leq \min(u_{ij}, u_{ji}) \quad \forall ij \in E_M \quad (13)$$

$$x_{e^f} + x_{e^t} \leq \max(\text{size}_m(e^f), \text{size}_m(e^t)) \quad \forall (e^f, e^t) \in E_M^f \times E_M^t, e^f \parallel e^t \quad (14)$$

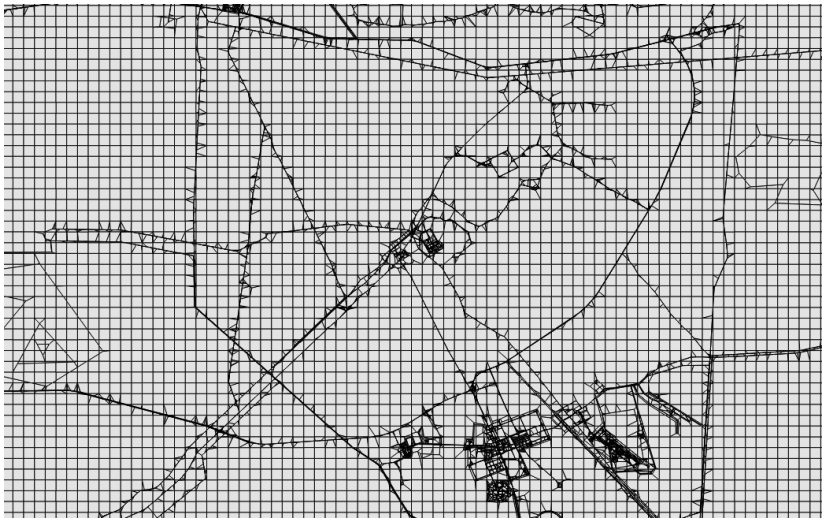
Plan

- 1 Présentation du stage
- 2 Méthode générale
- 3 Modélisation en problème de flot
- 4 Construction du graphe de l'espace de travail
- 5 Formulation en programme linéaire
- 6 Résultats**
- 7 Conclusion

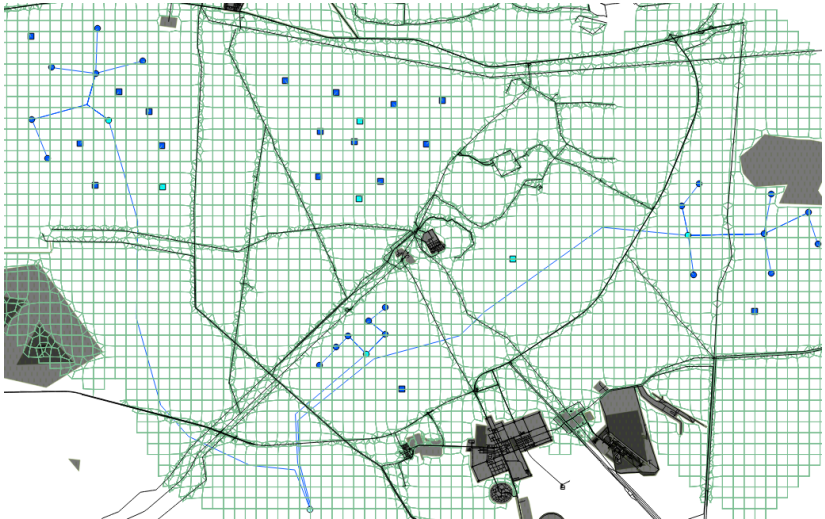
Cas d'étude



Partitionnement du plan



Graphe de flot



Coûts n°1

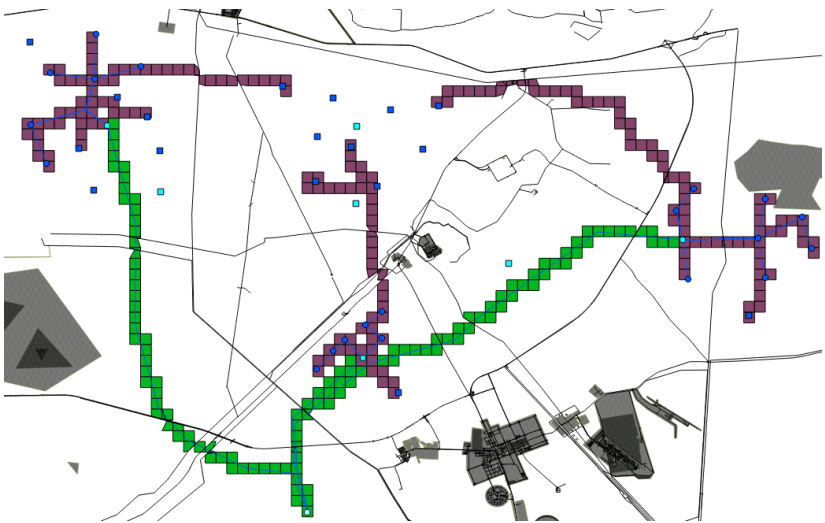
Coûts choisis

- Coût variable faible (1),
- Vide : coût de construction faible (1),
- Vide : coût d'extension nul (0),
- Route : coût de construction moyen (10)
- Route : coût d'extension nul (0)
- Stations : coût de construction fort (50)

Génération de la solution

- Solveur linéaire mixte SCIP
- Lancé sur 2 threads, sur laptop peu puissant
- Solveur arrêté après 1h30

Solution aux coûts n°1



Coûts n°2

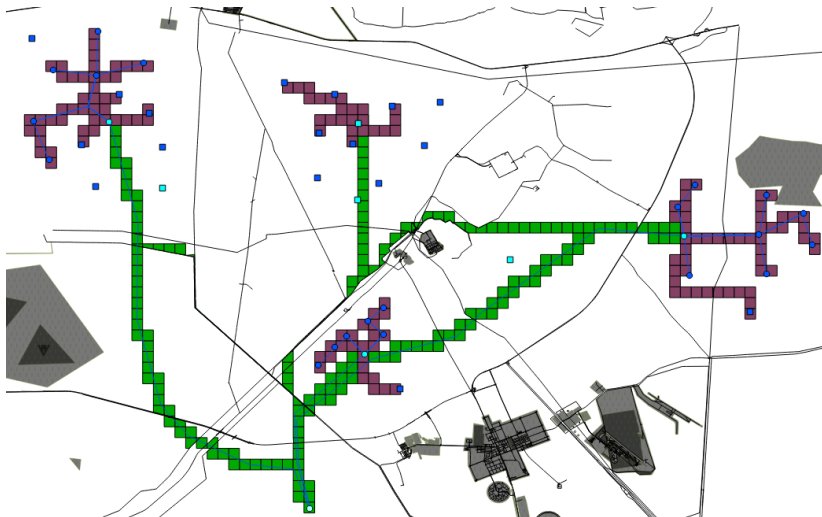
Coûts choisis

- Coût variable faible (1),
- Vide : coût de construction faible (2),
- Vide : coût d'extension faible (0.5),
- Route : coût de construction moyen (10),
- Route : coût d'extension = coût construction (10),
- Stations : coût de construction fort (50).

Génération de la solution

- Solveur linéaire mixte SCIP,
- Lancé sur 2 threads, sur laptop peu puissant,
- Solveur arrêté après 1h30.

Solution aux coûts n°2

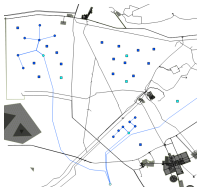


Plan

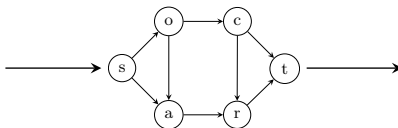
- 1 Présentation du stage
- 2 Méthode générale
- 3 Modélisation en problème de flot
- 4 Construction du graphe de l'espace de travail
- 5 Formulation en programme linéaire
- 6 Résultats
- 7 Conclusion**

Méthode générale

Problème initial



Problème de flots



Problème linéaire

$$\min \sum_{e \in E} v_e x_e + \sum_{e \in E} f_e y_e^0 + \sum_{\substack{e \in E_{it} \\ \text{cap}(e) > 0}} \gamma_e y_e^1$$

$$\text{s.t.} \quad \sum_{i:iv \in E} x_{iv} = \sum_{o:ov \in E} x_{vo}$$

$$\sum_{o:so \in E} x_{so} = \sum_{i:it \in E} x_{it}$$

$$l_e \leq x_e \leq u_e y_e^0$$

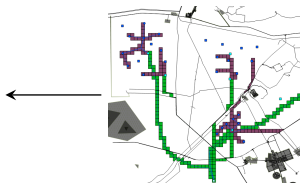
$$x_e = x_e^1 + x_e^2$$

$$l_e \leq x_e$$

$$b_e y_e^1 \leq x_e^1 \leq b_e y_e^0$$



Chemins réels

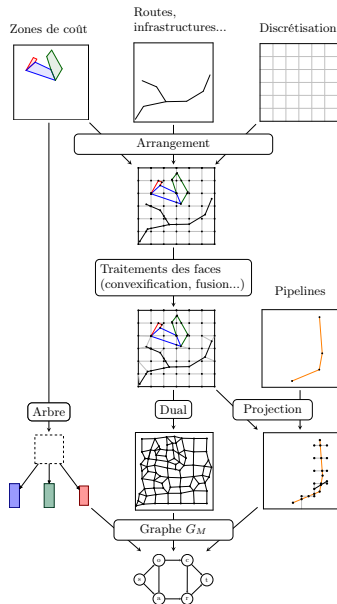


Solution exploitable

x_t4154_t3098	3
x_t3103_t3130	3
x_t3130_t3173	3
x_t3218_t5557	18
x_t4902_t3218	12
x_t6341_t3442	6
x_t3442_t5605	6
x_t3574_t3456	4

Solution au problème linéaire

Construction du graphe de l'espace de travail



Analyse de la méthode

Points positifs

- Générique,
- Génère plusieurs solutions → analyse multicritère,
- La qualité de la solution tend vers l'optimum (pour une discrétisation donnée) avec le temps,
- Peut être étendu
 - Coûts plus complexes (dénivelé, longueur des croisements),
 - Routes en tant que polygones,
 - Applicable à d'autres réseaux (eau, électricité).

Analyse de la méthode

Points négatifs

- Coûteux en temps (solveurs commerciaux 10x plus rapides sur 1 thread + parallèle + distribué),
- Coûteux en mémoire si on veut une discrétisation fine,
- Ne gère pas directement le côté multicritère mais minimise un coût.

Références I



Boris Aronov, Kevin Buchin, Maike Buchin, Bart Jansen, Tom de Jong, Marc van Kreveld, Maarten Löffler, Jun Luo, Rodrigo I Silveira, and Bettina Speckmann, *Connect the dot : Computing feed-links for network extension*, Journal of Spatial Information Science (2014), 3–31.



Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin, *Network flows : Theory, algorithms, and applications*, Prentice Hall, 1993.



Aydin Buluç and Kamesh Madduri, *Parallel breadth-first search on distributed memory systems*, Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, ACM, 2011, p. 65.



Ulrike Bartuschka, Kurt Mehlhorn, and Stefan Näher, *A robust and efficient implementation of a sweep line algorithm for the straight line segment intersection problem*, in proc. workshop on algorithm engineering, 1997, pp. 124–135.



Christoph Burnikel, Kurt Mehlhorn, and Stefan Schirra, *On degeneracy in geometric computations*, Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics, 1994, pp. 16–23.



Adi Botea, Martin Müller, and Jonathan Schaeffer, *Near optimal hierarchical path-finding*, Journal of game development 1 (2004), no. 1, 7–28.



Jon L Bentley and Thomas A Ottmann, *Algorithms for reporting and counting geometric intersections*, Computers, IEEE Transactions on 100 (1979), no. 9, 643–647.



Glenn O Brown, *The history of the darcy-weisbach equation for pipe flow resistance*, Environmental and Water Resources History 38 (2002), no. 7, 34–43.

Références II



Natthaporn Buaphut and Nanthi Suthikarnnarunai, *Effects of pipeline extension and network robustness evaluation : the case study of oil distribution to the northern region of thailand*, Proceedings of the International MultiConference of Engineers and Computer Scientists, vol. 2, 2014.



Bernard Chazelle and Herbert Edelsbrunner, *An optimal algorithm for intersecting line segments in the plane*, Journal of the ACM (JACM) 39 (1992), no. 1, 1–54.



Timothy M Chan, *A simple trapezoid sweep algorithm for reporting red/blue segment intersections.*, CCCG, Citeseer, 1994, pp. 263–268.



Salem Chakhar, *Cartographie décisionnelle multicritère : formalisation et implémentation informatique*, Ph.D. thesis, Université Paris Dauphine-Paris IX, 2006.



Kenneth L Clarkson and Peter W Shor, *Applications of random sampling in computational geometry, ii*, Discrete & Computational Geometry 4 (1989), no. 1, 387–421.



Roland Geraerts, *Planning short paths with clearance using explicit corridors*, Robotics and Automation (ICRA), 2010 IEEE International Conference on, IEEE, 2010, pp. 1997–2004.



Daniel Harabor and Adi Botea, *Hierarchical path planning for multi-size agents in heterogeneous environments*, Computational Intelligence and Games, 2008. CIG'08. IEEE Symposium On, IEEE, 2008, pp. 258–265.



Kenneth E Hoff III, John Keyser, Ming Lin, Dinesh Manocha, and Tim Culver, *Fast computation of generalized voronoi diagrams using graphics hardware*, Proceedings of the 26th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co., 1999, pp. 277–286.

Références III



Peter E Hart, Nils J Nilsson, and Bertram Raphael, *A formal basis for the heuristic determination of minimum cost paths*, Systems Science and Cybernetics, IEEE Transactions on 4 (1968), no. 2, 100–107.



Mike Hewitt, George L Nemhauser, and Martin WP Savelsbergh, *Combining exact and heuristic approaches for the capacitated fixed-charge network flow problem*, INFORMS Journal on Computing 22 (2010), no. 2, 314–325.



John D Hobby, *Practical segment intersection with finite precision output*, Computational Geometry 13 (1999), no. 4, 199–214.



Rafia Inam, *A* algorithm for multicore graphics processors*, Master's thesis, Chalmers University of Technology, 2010.



Tom de Jong and T Tillema, *Transport network extensions for accessibility analysis in geographic information systems*, Proceedings of AfricaGIS, 2005.



Hyoungshick Kim and Ross Anderson, *An experimental evaluation of robustness of networks*, IEEE Systems Journal 7 (2013), no. 2, 179–188.



Hak-Jin Kim and John N Hooker, *Solving fixed-charge network flow problems with a hybrid optimization and constraint programming approach*, Annals of Operations Research 115 (2002), no. 1-4, 95–124.



Fernando Kuipers, Turgay Korkmaz, Marwan Krunz, and Piet Van Mieghem, *Performance evaluation of constraint-based path selection algorithms*, Network, IEEE 18 (2004), no. 5, 16–23.



Hua-Yang Lin, Ping-Yu Hsu, and Gwo-Ji Sheen, *A fuzzy-based decision-making procedure for data warehouse system selection*, Expert systems with applications 32 (2007), no. 3, 939–953.

Références IV



Jean-Luc Marichal, *Fonctions d'agrégation pour la décision.*



Marta SR Monteiro, Dalila BMM Fontes, and Fernando ACC Fontes, *An ant colony optimization algorithm to solve the minimum cost network flow problem with concave cost functions*, Proceedings of the 13th annual conference on Genetic and evolutionary computation, ACM, 2011, pp. 139–146.



Ketan Mulmuley, *A fast planar partition algorithm, i*, Journal of Symbolic Computation 10 (1990), no. 3, 253–280.



Alex Nash, Kenny Daniel, Sven Koenig, and Ariel Felner, *Theta* : Any-angle path planning on grids.*, Proceedings of the National Conference on Artificial Intelligence, vol. 22, Menlo Park, CA ; Cambridge, MA ; London ; AAAI Press ; MIT Press ; 1999, 2007, p. 1177.



Steven J Owen, *A survey of unstructured mesh generation technology.*



Joachim Schöberl, *Netgen an advancing front 2d/3d-mesh generator based on abstract rules*, Computing and visualization in science 1 (1997), no. 1, 41–52.



Mehdi Samadi, Ariel Felner, and Jonathan Schaeffer, *Learning from multiple heuristics.*, AAAI, 2008, pp. 357–362.



Kenji Shimada and David C Gossard, *Bubble mesh : automated triangular meshing of non-manifold geometry by sphere packing*, Proceedings of the third ACM symposium on Solid modeling and applications, ACM, 1995, pp. 409–419.



Jonathan Richard Shewchuk, *Delaunay refinement algorithms for triangular mesh generation*, Computational geometry 22 (2002), no. 1, 21–74.

Références V



Ali Sydney, Caterina Scoglio, Phillip Schumm, and Robert E Kooij, *Elasticity : topological characterization of robustness in complex networks*, Proceedings of the 3rd International Conference on Bio-Inspired Models of Network, Information and Computing Systems, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, p. 19.