Présentation du problème

Réduction de SET-COVER

Algorithmes dans le cas général

Programme Linéair

Preprocessing

ripproximition

Coomátrio

Conclusion

# Dynamic Facility Location: Minimizing Sum of Radii

### N.Blanchard

Stage au Laboratoire d'Informatique Algorithmique, Fondements et Applications Sous la direction de Nicolas Schabanel

3 septembre 2015

Présentation du problème

Réduction de SET-COVER

Algorithmes dans le cas général (non-métriq

Programme Linéair

log(nT)-Approximation

log n-Approximatio

Cus meniqu

Conclusio

# Plan de la présentation

1 Présentation du problème Facility Location

Réduction de SET-COVER

2 Algorithmes dans le cas général (non-métrique)

Programme Linéaire Preprocessing log(*nT*)-Approximation log *n*-Approximation

- 3 Cas métrique
- 4 Conclusion

N.Blanchard

# Présentation du problème

Réduction de SET-COVER

Algorithmes dans le cas général

(non-metrique

Programme Linéair

rieprocessing

log(nT)-

Approximation

Cas métrique

Conclusion

# Présentation du problème

N.Blanchard

Présentation du

Facility Location
Réduction de

Algorithmes dans le cas

(non-métrique

Drogramma I inágir

Preprocessing

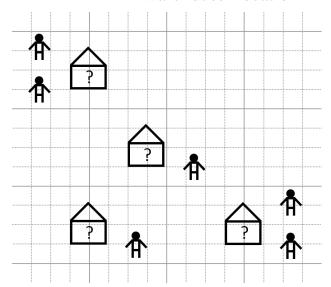
log(nT)-

log n. Approximati

Cas métrique

Conclusio

# Warehouse Location



Introduit par Hamburger et Kuehn en 1963

#### N.Blanchard

Présentation de problème

Facility Location Réduction de SET-COVER

Algorithmes dans le cas général

(non-métrique

Programme Linéair

Preprocessing

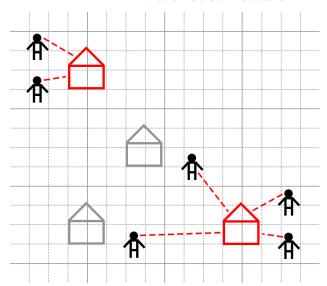
Approximation

log n-Approximation

Cas métrique

Conclusion

# Warehouse Location



Minimiser les coûts d'ouvertures plus les coûts de connection

Présentation du

Facility Location
Réduction de

Algorithme dans le cas général

B 11.7.1

1 Togramme Linea

Preprocessing

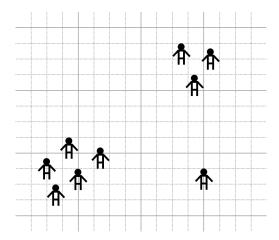
log(nT)-

1--- A-----

Cas métriqu

Conclusion

# Application au Clustering



Présentation du

Réduction de

Algorithme dans le cas général

Programme Linéai

Drangoossing

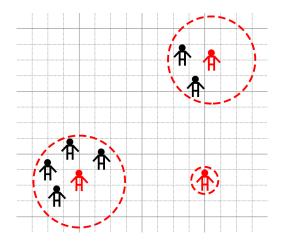
1 reprocessing

Approximation

10B111FF

Cas meurqu

# Application au Clustering



Présentation du

Réduction de

Algorithme dans le cas général

Programma I inási

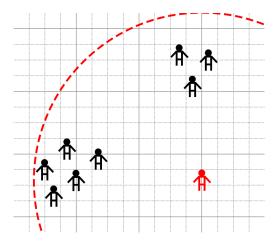
1 1051mmile Emem

Preprocessing

log(nT)-

.

# Application au Clustering



Présentation d problème

Réduction de

Algorithmes dans le cas général

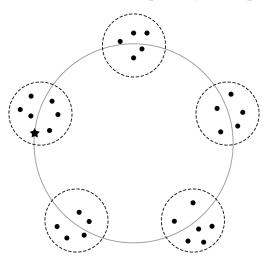
(Hon-metrique

Preprocessing

Approximation

Conclusion

# Aspect dynamique



Présentation d problème

Réduction de

Algorithmes dans le cas général

(non-metrique

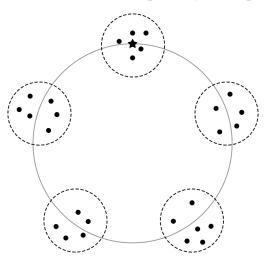
Preprocessing

 $\log(nT)$ -

log n-Approximation

Cas metriqu

# Aspect dynamique



Présentation de problème

Réduction de

Algorithmes dans le cas général

(non-meurque

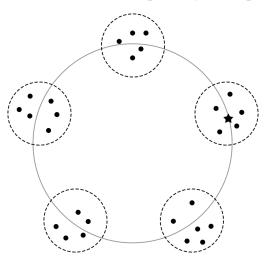
Preprocessing

Approximation

Cas métrion

Conclusion

# Aspect dynamique



Présentation d problème

Réduction de

Algorithmes dans le cas général

(non-metrique

Preprocessing

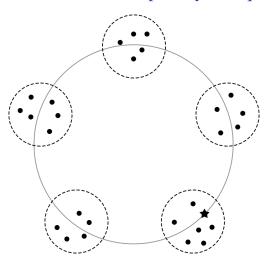
log(nT)-

log n-Approximation

Cas métriqu

Conclusion

# Aspect dynamique



Présentation d

Facility Location Réduction de

Algorithmes dans le cas général

n ....

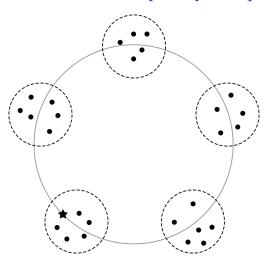
Preprocessing

Approximation

Coc mátrios

Conclusion

# Aspect dynamique



Présentation d

Réduction de

Algorithmes dans le cas général

- ....

Preprocessing

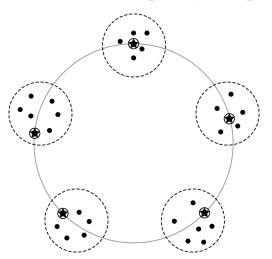
Approximation

log n-Approximation

Cas metriqu

Conclusion

# Aspect dynamique



Algorithmes dans le cas général (non-métrique

Preprocessing
log(nT)Approximation

log n-Approximatio

Cas métrique

Conclusion

# Définition du problème

### On a en entrée :

- Un ensemble  $\mathscr{C}$  de n clients
- Un ensemble  $\mathscr{F}$  de m services (facilities)
- Un ensemble  $\mathcal{T}$  de t pas de temps
- Un coût d'ouverture f et un coût de changement g
- Des distances  $d^t(i,j)$  pour chaque paire (service, client) et chaque pas de temps

On veut connecter tous les clients en minimisant les coûts.

#### N.Blanchard

Présentation d problème

Réduction de

Algorithmes dans le cas général

Programme Linés

i rogramme Emean

Preprocessing

Approximation

log n-Approximatio

Cas métriqu

Conclusion

# Problème avec l'objectif Somme des distances







Présentation du

Réduction de

Algorithmes dans le cas général

Programme I inési

1 Togramme Emea

Preprocessing

Approximation

Conclusion

# Problème avec l'objectif Somme des distances



#### N.Blanchard

Présentation de problème

Réduction de SET-COVER

Algorithmes dans le cas général

Programme I inési

Preprocessing

log(nT)-

log n-Approximatio

Cas métriqu

Conclusio

# Nouvelle fonction objectif: Somme des rayons





Modèle étudié par Charikar et Panigrahy en 2001 (en statique)

#### N.Blanchard

Présentation de problème

Réduction de

Algorithme dans le cas général

Programma I inánim

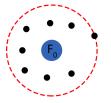
log(nT)-

1---- 1-----

Cas métriqu

Conclusion

# Nouvelle fonction objectif: Somme des rayons





Modèle étudié par Charikar et Panigrahy en 2001 (en statique)

### N.Blanchard

Présentation du problème

Réduction de SET-COVER

Algorithmes dans le cas général

B 11.7.1

Prenrocessing

log(nT)-

Approximation

Cas métrique

Conclusion

# Réduction de SET-COVER

Présentation du problème

Réduction de SET-COVER

Algorithmes dans le cas général (non-métrique

Programme Linéai

log(nT)-Approximation

Cas métrique

Conclusi

## Réduction de SET-COVER

### Théorème

[Dinur - Steurer, 2013] SET-COVER n'admet pas de  $(1 - \varepsilon) \log n$  approximation à moins que  $\mathbf{P} = \mathbf{NP}$ .

Ce qui permet de montrer :

### Théorème

Dynamic Facility Location Minimisant la Somme des Rayons n'admet pas de  $(1-\varepsilon)\log n$  approximation dans le cas non-métrique à moins que  $\mathbf{P}=\mathbf{NP}$ .

Dans le cas métrique ce résultat ne tient pas.

### Algorithmes dans le cas général (non-métrique)

# Algorithmes dans le cas général (non-métrique)

Réduction de SET-COVER

dans le cas général (non-métrique

### Programme Linéaire

log (nT)-Approximation

Cas métriqu

----

# Formulation en Programme Linéaire

Toutes les variables valent 0 ou 1, avec :

- $y_{ir}^t$  vaut 1 si i est ouverte avec rayon r au temps t
- $x_{ij}^t$  vaut 1 si j est connecté à i au temps t
- $z_{ij}^t$  vaut 1 si j s'est connecté à i entre les temps t-1 et t

### On minimise

$$\sum_{i,t,r} y_{ir}^t \cdot (f+r) + g \sum_{i,j,t} z_{ij}^t$$

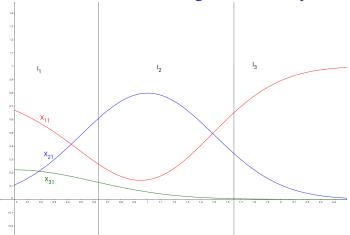
Avec les contraintes :

- $\forall j, t \; \sum_i x_{ij}^t \geq 1$
- $\forall i, j, t \ x_{ij}^t \leq \sum_{r \geq d^t(i,j)} y_{ir}^t$
- $\forall i, j, t \geq 1 \ z_{ii}^t \geq x_{ii}^t x_{ii}^{t-1}$

#### N.Blanchard

Preprocessing

# Preprocessing: décider quand les clients changent de facility



### Lemme

(EMS) Le preprocessing double au plus le coût de la solution



Présentation du problème

Réduction de SET-COVER

Algorithmes dans le cas général

Programme Linéaire

Preprocessing log(nT)-

Approximation

Cae mátrias

1

### Ouverture des facilities

# Algorithme 1:

- On répète ce qui suit log(Z) + 1 fois
  - Pour chaque facility i, on tire une variable a<sub>i</sub> uniformément dans [0, 1]
  - Pour chaque pas de temps t, on ouvre chaque facility i avec le plus grand rayon R tel que

$$a_i \leq \sum_{r \geq R} y_{ir}^t$$

Cela revient à ouvrir i avec un rayon suivant la distribution des  $y_{ir}^t$ , en maintenant une cohérence dans le temps

• On ouvre chaque facility avec le plus grand rayon parmi les log(Z) + 1 solutions partielles

Programme Linéair

 $\log(nT)$ Approximation

Co o mo étai ou

Cas meniqu

# Couverture d'un client

Un intervalle I d'un client j n'est pas couvert par une facility i s'il existe  $t \in I$ 

$$a_i \geq \sum_{r \geq d^t(i,j)} y_{ir}^t$$

Or, par la deuxième contrainte du LP, pour  $t \in I$ 

$$\sum_{r \geq d^t(i,j)} y^t_{ir} \geq \min_{t \in I} x^t_{ij} \stackrel{\text{def}}{=} x^I_{ij}$$

La probabilité d'être couvert par une solution partielle est au moins

$$1 - \left(\prod_{i} \left(1 - x_{ij}^{I}\right)\right) \ge \frac{1}{2}$$

En répetant log(Z) + 1 fois tous les clients sont couverts sur tous leurs intervalles avec probabilité  $\frac{1}{2}$ 

Réduction de SET-COVER

Algorithme dans le cas général

Programme Linéaire Preprocessing

log(nT)-Approximation

log n-Approximatio

---------

Conclusi

# Coût total

- Coût de changement au plus  $Z \cdot g$ , par le préprocessing au plus égal à  $2 \cdot OPT$
- Espérance de coût par facility égale à  $\sum y_{ir}^t(r+f)$
- Espérance des coûts des solutions partielles égale à OPT
- Recombiner coûte moins que la somme des coûts partiels

### Théorème

L'algorithme 1 est une  $O(\log(Z))$  approximation

### Corollaire

L'algorithme 1 est une  $O(\log(nT))$  approximation

Présentation du problème

Réduction de SET-COVER

Algorithmes dans le cas général (non-métrique

Programme Linéair

Preprocessing

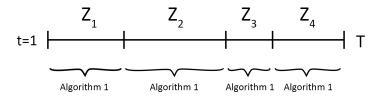
log n-Approximation

Cas métriqu

Conclusion

# log *n*-Approximation

- Si  $Z \le n^2$  lancer l'Algorithme 1
- Sinon découper en séquences telles que  $n \le Z_i < 2n$  et lancer l'Algorithme 1 sur chaque séquence



### Théorème

L'Algorithme 2 est une  $O(\log n)$  approximation.

#### N.Blanchard

Présentation du problème

Réduction de SET-COVER

Algorithmes dans le cas général

(non-metrique)

Programme Line

Preprocessing

Approximation

Cas métrique

Conclusion

# Cas métrique

Présentation du problème

Réduction de SET-COVER

Algorithmes dans le cas général

Programme Linéair

Preprocessing log(nT)-

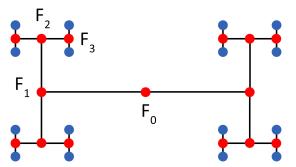
log n-Approximation

### Cas métrique

Conclusion

# Algorithme ANS

- Il existe une O(1) approximation pour la somme des distances [An, Norouzi-Fard, Svensson, 2014]
- On a proposé une adaptation naturelle de cet algorithme
- C'est hélas au mieux une  $\Omega(\log \log n)$  approximation
- Contre-exemple :



### N.Blanchard

Présentation du

Réduction de

Algorithme dans le cas général

(non-métriqu

Programme Linéa

Preprocessing

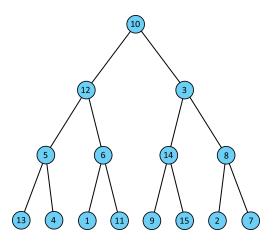
log(nT)-

1--- A----i---

### Cas métrique

Conclusion

# Lemme combinatoire



#### N.Blanchard

Présentation di

Réduction de

Algorithme dans le cas général

(non-metrique

Programme Liné

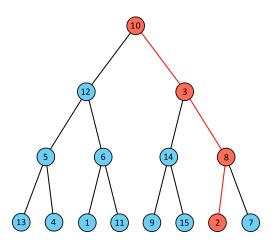
 $\log(nT)$ -

log n-Approximati

### Cas métrique

Conclusion

# Lemme combinatoire



### Lemme

La probabilité de trouver une telle branche est au moins  $\frac{1}{3}$ .



Présentation du problème

Réduction de

Algorithmes dans le cas général

(mon mounda

Programme Linéair

Danasan annian

 $log(nT)_{-}$ 

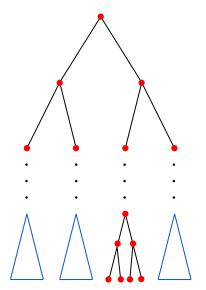
Approximation

logn-Approxima

### Cas métrique

0 1 1

# Lemme combinatoire (2)



#### N.Blanchard

Présentation du problème

Réduction de SET-COVER

Algorithmes dans le cas général

(non-métrique)

Programme Lin

Preprocessing

Approximation

Conclusion

# Conclusion

Facility Loca Réduction de SET-COVER

dans le cas général (non-métrique

Programme Lineau
Preprocessing

Approximation log n-Approximatio

Cas métriqu

Conclusion

# Conclusion

Nos résultats sur le Dynamic Facility Location :

|           | Non-Métrique (algorithme/difficulté) |                                   |  |
|-----------|--------------------------------------|-----------------------------------|--|
|           | ∑ Distances                          | ∑ Rayons                          |  |
| Statique  | $O(\log n) / \Omega(\log n)$         | $2\log n / (1-\varepsilon)\log n$ |  |
| Dynamique | $O(\log n)$ / $\Omega(\log n)$       | $4\log n / (1-\varepsilon)\log n$ |  |

De plus, notre ANS modifié ne peut pas donner une approximation meilleure que  $\Omega(\log \log n)$  pour la somme des rayons.

Réduction de SET-COVER

dans le cas général (non-métrique

Preprocessing

log(nT)Approximation

Cas métrique

Conclusion

## Travail futur

### Questions restantes:

- Peut-on prouver une borne supérieure pour le facteur d'approximation d'ANS modifié?
- Peut-on trouver un algorithme en approximation constante pour le cas métrique ?

Y-a-t'il des questions?

Présentation du problème

Réduction de SET-COVER

dans le cas général (non-métrique)

Programme Linéair

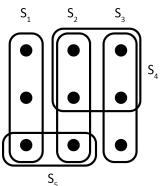
log(nT)-Approximation

\_\_\_\_\_

Conclusion

# Problème SET-COVER

- Un ensemble  $\mathscr{A}$  à *n* éléments
- Une collection  $\mathscr{B}$  de sous-ensembles de  $\mathscr{A}$
- Comment couvrir  $\mathscr{A}$  avec un sous-ensemble de  $\mathscr{B}$  de cardinalité minimale ?



Présentation d problème

Réduction de SET-COVER

Algorithmes dans le cas général

Programme Linéai

Preprocessing log(nT)-

log n-Approximation

Cas métriqu

Conclusion

## Réduction de SET-COVER

- On prend un client j pour chaque élément  $a_i \in \mathcal{A}$
- On prend une facility i pour chaque élément  $b_j \in \mathcal{B}$
- On fixe d(i,j) = 1 si  $a_j \in b_i$  et  $d(i,j) = \infty$  sinon



Réduction de SET-COVER

Algorithmes dans le cas général

Programme Linéai

log(nT)-

log n-Approximatio

Cas métrique

Conclusion

# Preprocessing

- On fixe  $t_0 = 1$
- On cherche de manière gloutonne le plus grand  $t_{k+1}$  possible tel que

$$\sum_{i} \min_{t \in [t_k, t_{k+1}]} x_{ij}^t \ge \frac{1}{2}$$

• Enfin on fixe pour tout intervalle  $[t_k, t_{k+1}]$ 

$$\widehat{x_{ij}^t} \stackrel{\text{def}}{=} 2 \cdot \min_{t \in [t_k, t_{k+1}]} x_{ij}^t$$

• Et

$$\hat{y_{ir}^t} \stackrel{\text{def}}{=} 2 \cdot y_{ir}^t$$

Présentation du problème

Réduction de SET-COVER

dans le cas général (non-métrique

Programme Linéaire Preprocessing

log(nT)Approximation log n-Approximation

Cas métrique

Conclusion

## Preuve du lemme combinatoire

- La racine a une clé dans le 1/3 supérieur avec probabilité 1/3
- Si on regarde une proportion  $\varepsilon$  des noeuds :
  - Chaque noeud a donc probabilité  $\frac{2}{3} \varepsilon$  d'être inférieur à la racine
  - On peut trouver un chemin à travers un arbre de hauteur <sup>n</sup>/<sub>2</sub> avec probabilité constante
  - À partir de ce noeud à hauteur <sup>n</sup>/<sub>2</sub> on peut trouver un deuxième chemin de longueur <sup>n</sup>/<sub>2</sub> et donc une branche satisfaisant la propriété
- On gère la dépendance et donc la valeur de  $\varepsilon$  en ne regardant qu'une petite proportion des noeuds.

### Lemme

Une bonne branche existe avec probabilité au moins

$$\frac{2+27\cdot tanh^{-1}\left(\frac{1}{2}\right)}{48}\approx 0.3506$$

Présentation du problème

Réduction de SET-COVER

Algorithmes dans le cas général (non-métrio

Preprocessing
log(nT)Approximation

Cas métriqu

Conclusion

# Algorithme ANS modifié

- On prend une solution au LP
- On utilise le preprocessing
- On tire une variable  $a_i \in \mathbb{R}$  par facility distribuée selon  $e^{-a_i x}$
- On tire une permutation uniforme des clients
- On met une arête de chaque client à la facility de  $a_i$  minimal tel que  $x_{ii}^t > 0$
- On met une arête de chaque facility au client d'ordre minimal tel que  $x_{ii}^t > 0$
- On assigne chaque client à la facility dans la boucle de longueur 2 qu'on obtient en suivant les arêtes

Présentation du problème

Réduction de SET-COVER

Algorithme dans le cas général

Programme Linéaire

log(nT)-

Approximation log n-Approximation

Cas métrique

Conclusion

# Résultats expérimentaux

Résultats d'une série de 10 groupes de 10<sup>7</sup> instances pour le lemme combinatoire :

| Hauteur               | 5      | 6      | 7      |
|-----------------------|--------|--------|--------|
| Probabilité empirique | 0.473  | 0.453  | 0.441  |
| Écart parmi 10 tests  | 0.0008 | 0.0018 | 0.0015 |

| Hauteur               | 8      | 9      | 10     |
|-----------------------|--------|--------|--------|
| Probabilité empirique | 0.431  | 0.423  | 0.418  |
| Écart parmi 10 tests  | 0.0013 | 0.0021 | 0.0017 |