TD3 - Algorithmes d'approximation

Optimisation et Recherche Operationnelle

M1 Info - semestre d'automne 2021-2022

Université Claude Bernard Lyon 1

Christophe Crespelle
`christophe.crespelle@univ-lyon1.fr`

Florian Ingels
`florian.ingels@ens-lyon.fr`

Exercice 1. *Au suivant !*

a. Appliquer l'algo *List Scheduling* vu en cours pour l'equilibrage de charge sur les listes suivantes :
— $L_1 = (5, 9, 4, 12, 4, 7, 5)$ et 3 machines
— $L_2 = (4, 4, 5, 5, 7, 9, 12)$ et 3 machines
— $L_3 = (3, 5, 2, 4)$ et 2 machines
— $L_4 = (1, 7, 8, 7, 2, 8)$ et 3 machines

b. Pour chacune des solutions obtenues, dites si elle est optimale ou non et prouvez le. Donnez le ratio d'approximation des solutions non optimales.

On note Π_1 le probleme donne par la liste de duree des taches $T = \{\!\{5, 9, 4, 12, 4, 7, 5\}\!\}$ et 3 machines ; on note Π_2 le probleme donne par $T = \{\!\{3, 5, 2, 4\}\!\}$ et 2 machines.

c. Pour chacun des deux problemes Π_1 et Π_2 donnez deux listes qui produisent respectivement le meilleur et le pire resultat pour l'algo *List Scheduling*. Donnez le ratio d'approximation de chacune de ces solutions.

d. Prouvez que vos reponses a la question precedente sont bien les meilleurs et les pires resultats pour l'algo *List Scheduling*.

e. Pour chacun des deux problemes Π_1 et Π_2 quelle est la solution obtenue en appliquant l'algo *List Scheduling* sur la liste triee par ordre decroissant de duree des taches ? Cette solution est elle optimale ?

f. Donnez un exemple de probleme sur deux machines tel que l'algo *List Scheduling* qui recoit en entree la liste triee par ordre decroissant de duree des taches ne fournit pas la solution optimale au probleme.

Exercice 2.

Le but de cet exercice est de montrer qu'il existe toujours une liste qui, fournie en entree a l'algo *List Scheduling*, resulte en une affectation des taches realisant le makespan

$L_1 = (5, 9, 4, 12, 4, 7, 5) \to 46/3 = 15,333\ldots$

M_1	M_2	M_3
7		
4	5	12
5	9	4
16	14	16

makespan = 16 OPT

$OPT \geq 15,33\ldots$ entiers $\Rightarrow OPT \geq 16$

$L_2 = (4, 4, 5, 5, 7, 9, 12)$

M_1	M_2	M_3
12		
5	7	9
4	4	5
21	11	14

makespan = 21 pas OPT (voir L_1)

mêmes durées et makespan = 16

ratio = $\frac{21}{16}$

$L_3 = (3, 5, 2, 4)$

M_1	M_2
2	4
3	5
5	9
7	7

makespan = 9

pas OPT ratio = $\frac{9}{7}$

pas bien équilibré.

$\to$ OPT par la borne de la moyenne.

OPT = 7

$L_4 = (1, 7, 8, 7, 2, 8)$

M_1	M_2	M_3
8		
7	2	1
1	7	8
14	10	9
16	9	8
15	9	9

$33/3 = 11$

$OPT \geq 11$

makespan = 16

= 15

= 14 OPT

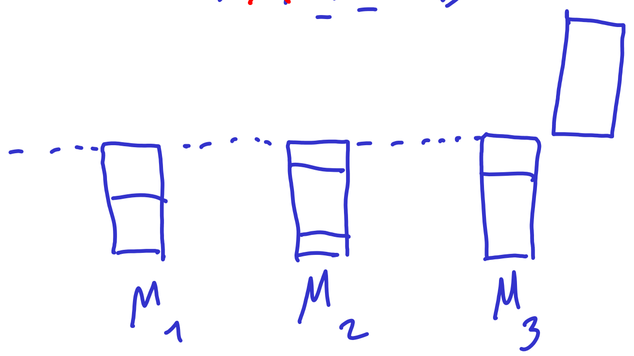
principe du pigeonnier:

au moins une machine M_i reçoit au moins 2 tâches parmi celles de durée : 7, 7, 8, 8.

charge$(M_i) \geq 7 + 7 = 14$

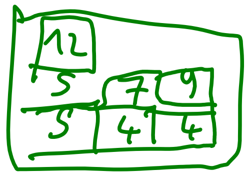
$OPT \geq 14$

$T = \{\{5, 9, 4, \cancel{12}, 4, 7, 5\}\}$

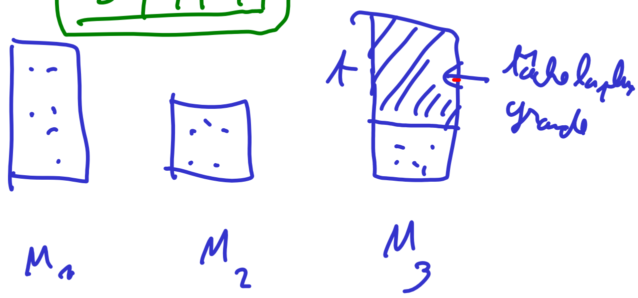


le plus petit est $= 11$ impossible

4
7



22



$L' = (5, 4, 4, 7, 9, 5, 12)$

S : somme des durées des tâches

12		
5	7	9
5	4	4
M_1	M_2	M_3
22	11	13

affectation A

M_i : machine la plus chargée dans A.

t : tâche la plus longue sur M_i

$$\text{charge}(M_i) - t \leq \frac{S-t}{m}$$

≥ 0

$$\text{makespan} \leq t + \frac{S-t}{m} = f(t) = \left(1 - \frac{1}{m}\right) t + \frac{S}{m} \qquad m \geq 2$$

$\boxed{S = 46}$

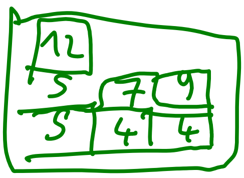
$f(t) \nearrow$ en t.

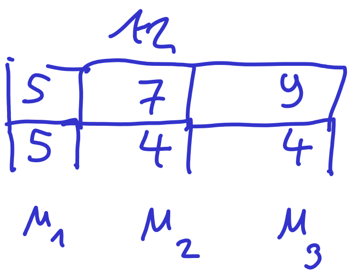
$$\text{makespan} \leq f(t)$$

makespan $\leq 23{,}33$

$$f(12) = 12 + \frac{34}{3} = 23{,}33\ldots \qquad \leq 23$$

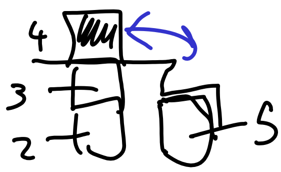
$$f(9) = 9 + \frac{37}{3} = 21{,}33\ldots \qquad \leq 21$$





$L' = (5, 4, 4, 7, 9, 5, 12$

$T = \{\{3, 5, 2, 4\}\}$



$L = (2, 5, 3, 4)$

makespan $= 9$

OPT $= 7$

$L' = (12, 9, 7, 5, 5, 4, 4)$

M_1	M_2	M_3	
12	9	7	
4	5	5	
		4	
16	14	16	OPT

$L'' = (5, 4, 3, 2)$

M_1	M_2	
5	4	
2	3	
7	7	OPT

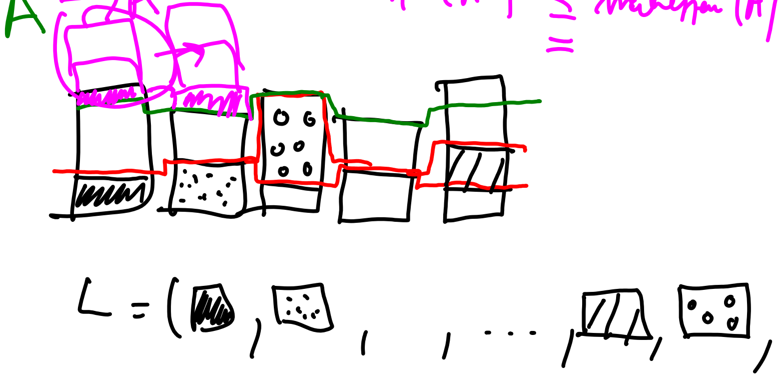
$6, 6, 5, 4, 3$

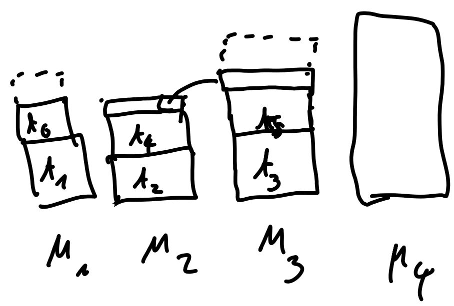
M_1	M_2
6	6
5	4
	3
11	13
12	12

$\boxed{\text{une affectation } A \text{ donnée}}$

<u>idée :</u> Construire une liste L des tâches pas à pas en choisissant à chaque fois pour la tâche suivante une des tâches sur une des machines ayant la charge minimum <u>jusqu'à lors</u>.

makespan$(A') \leq$ makespan(A)





minimum. On note $\mathcal{T}$ l'ensemble des n taches et $\mathcal{M}$ l'ensemble des m machines d'une instance du probleme d'equilibrage de charges. Soit A une affectation des taches de $\mathcal{T}$ aux machines de $\mathcal{M}$, c'est a dire une application de $\mathcal{T}$ dans $\mathcal{M}$, qui a chaque tache associe une et une seule machine. Pour une machine $M \in \mathcal{M}$, on note $\mathcal{T}_A(M)$ l'ensemble des taches affectees a M dans A.
De plus, on definit le profil d'une affectation A, note $p(A)$, comme la sequence decroissante des charges des m machines dans A. Un profil p est donc une sequence de m reels que l'on note $p = (p_1, p_2, \ldots, p_m)$ avec $p_1 \geq p_2 \geq \cdots \geq p_m$. L'ordre lexicographique defini un ordre total sur les profils. C'est a dire que pour deux profils $p = (p_1, p_2, \ldots, p_m)$ et $p' = (p'_1, p'_2, \ldots, p'_m)$, avec $p \neq p'$, en notant i le plus petit indice dans $[\![1, m]\!]$ tel que $p_i \neq p'_i$, on a $p < p'$ ssi $p_i < p'_i$.

a. Soit A une affectation dont le profil est minimum pour l'ordre lexicographique. Montrez que A realise le makespan minimum.

b. Soit A une affectation dont le profil est minimum pour l'ordre lexicographique. Soit $S \subsetneq \mathcal{T}$ un sous-ensemble strict des taches et soit $A[S]$ l'affectation obtenue en restreignant A aux taches de S. Montrez que l'ensemble $R_{min} \subseteq \mathcal{M}$ des machines dont la charge est minimum dans $A[S]$ contient au moins une machine M telle que $\mathcal{T}_{A[S]}(M) \neq \mathcal{T}_A(M)$.

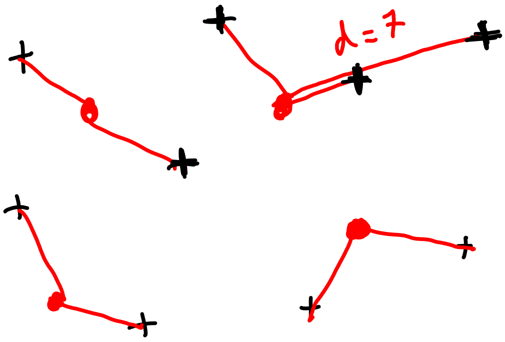
c. Montrez que pour toute instance du probleme d'equilibrage de charges, il existe toujours une liste qui, fournie en entree a l'algo *List Scheduling*, donne la solution optimale au probleme sur cette instance.

> **Indication.** Considerez une affectation A dont le profil est minimum pour l'ordre lexicographique et construisez, tache par tache, une liste L telle que lorsque L est donnee en entree a l'algo *List Scheduling*, il existe une execution de l'algorithme qui donne exactement l'affectation A.

d. Si les n taches ont des durees deux a deux distinctes, combien y a-t-il de listes distinctes sur les durees des taches ? L'algorithme qui consiste a essayer toutes les listes possibles en entree de l'algorithme de liste est-il un algorithme efficace ? Comparez sa complexite a celle de l'algorithme *brute force* vu en cours.

Exercice 3. *k-center selection*

Dans ce probleme, on donne un ensemble S de n sites dans le plan euclidien (mais en toute generalite, n'importe quelle distance symetrique et qui verifie l'inegalite triangulaire permettrait d'obtenir les memes resultats) et on veut selectionner au plus k centres (pas necessairement sur les sites donnes en entree) ou placer des casernes de pompiers pour proteger au mieux les n sites. Le critere que l'on veut minimiser pour l'ensemble des k centres $C = \{c_1, c_2, \ldots, c_k\}$ selectionnes est le rayon de couverture $r(C) = \max_{s \in S}\{d(s, C)\}$, ou $d(s, C) = \min_{c \in C}\{d(s, c)\}$. Le but de l'exercice est de concevoir un algorithme d'approximation du rayon de couverture minimum avec k centres, qui ait un ratio d'approximation de 2 sur le rayon de couverture et qui fournisse un ensemble de k centres realisant cet objectif.
Supposons pour commencer qu'on connaisse le rayon de couverture minimum r. On propose alors l'algorithme suivant.



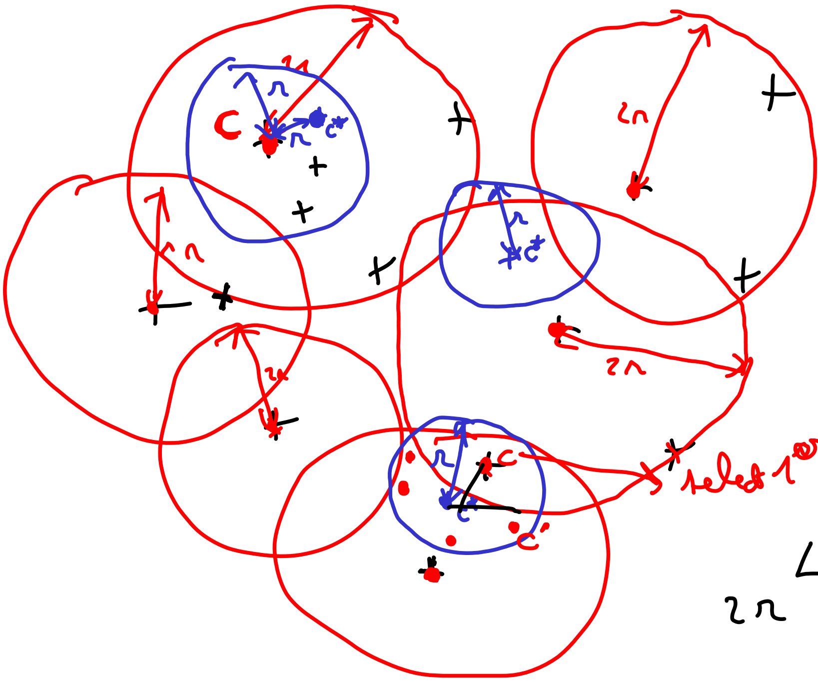
n sites

k centres

k fixé (=12)

minimiser le rayon de couverture

rayon de couverture = 7



select 1er par l'algo

$l = 5$

$$d(c,c') \le 2r$$

$$2r < d(c,c') \le \underbrace{\underbrace{d(c,c^*)}_{\le r} + \underbrace{d(c^*,c')}_{\le r}}_{\le 2r}$$

Contradiction.

supposons que dans $B(c^*, r)$ il y a ≥ 2 centres de l'algo.

j'appelle c le 1er parmi eux select par l'algo

et $c' \neq c$ alors $d(c,c') \le d(c,c^*) + d(c^*,c') \le r+r = 2r$ car $c, c' \in B(c^*,r)$

$d(c,c') \le 2r \Rightarrow$ c' a été retiré de S' lorsque ~~c a été sélectionné.~~

donc c' n'a jamais été select : absurde

Algorithme 1 : Algorithme de 2-approximation en connaissant le rayon de couverture minimum r.

1 $S' \leftarrow S$; sites $O(n)$
2 $C \leftarrow \varnothing$; $O(1)$ centres (à construire)
3 **tant que** $S' \neq \varnothing$ **faire** n fois
4   Selectionner un $s \in S'$ quelconque; $\rightarrow O(1)$
5   $C \leftarrow C \cup \{s\}$; $O(1)$
6   Retirer de S' tous les sites qui sont a distance au plus $2r$ de s; $O(n)$ $O(n^2)$
7 **si** $|C| \leq k$ **alors**
8   **retourner** C; $O(n)$
9 **sinon**
10   **retourner** "le rayon de couverture minimum est $\boxed{> r}$"; le rayon de couverture de C

a. Que retourne l'algorithme ? Qu'est-ce qui est approxime ici ? Que retourne l'algorithme si le rayon r qui lui est fourni n'est pas le rayon de couverture minimum (contrairement a ce que nous supposons ici) ?

b. Quel est la complexite de cet algorithme ?

Le but des questions suivantes est de montrer que l'algorithme 1 est correct et realise un rapport d'approximation de 2 sur le rayon de couverture qui lui est fourni en entree. C'est a dire que s'il existe un ensemble d'au plus k centres qui a un rayon de couverture r, alors l'algorithme 1 termine avec $|S| \leq k$ et S realise un rayon de couverture d'au plus $2r$. Soit C^* une solution de rayon de couverture r. Soit C la solution ALGO retournee par l'algorithme.

c. Montrez que pour tout $c \in C$ (algo), il existe $c^* \in C^*$ (OPT) tel que $d(c, c^*) \leq r$.

d. Soit $c^* \in C^*$ tel qu'il existe $c \in C$ tel que $d(c, c^*) \leq r$. Montrez que quel que soit $c' \in C \setminus \{c\}$, $d(c', c^*) > r$.

e. Montrez par l'absurde que si l'algorithme termine avec $|C| > k$ alors il n'existe pas d'ensemble d'au plus k centres qui ait un rayon de couverture r. $|C| \leq k$

f. En deduire que s'il existe un ensemble d'au plus k centres qui a un rayon de couverture r, alors l'algorithme trouve un ensemble d'au plus k centres qui a un rayon de couverture au plus $2r$.

Contraposée de e + Rq que le C de l'algo a un rayon $\leq 2r$

On veut maintenant faire un algorithme de 2-approximation sur le rayon de couverture sans connaitre le rayon de couverture optimal. Pour cela, on remarque qu'a la ligne 4 du precedent algo, on peut choisir n'importe quel site s qui ne soit pas encore couvert par les centres selectionnes auparavant avec un rayon de $2r$.

g. Lorsqu'il existe un site non couvert par les centres selectionnes auparavant dans l'algorithme avec un rayon $2r$, comment peut-on choisir un site s pour etre sur qu'il soit non couvert sans meme connaitre r ?

h. En utilisant le resultat de la question precedente, ecrivez un algorithme pour le probleme k-center selection qui retourne toujours un ensemble C d'exactement k centres (on supposera $n > k$ sinon le rayon de couverture minimum est 0) et qui garantisse un facteur d'approximation de 2 sur le rayon de couverture minimum, sans supposer cette fois aucune connaissance sur le rayon de couverture minimum.

Algorithme 1 : Algorithme de 2-approximation en connaissant le rayon de couverture minimum r.

1 ~~$S' \leftarrow S$;~~
2 $C \leftarrow \varnothing$; $|C| < k$
3 **tant que** ~~$S' \neq \varnothing$~~ **faire** s tq $d(s, C)$ est max
4   Selectionner un ~~$s \in S'$ quelconque~~;
5     $C \leftarrow C \cup \{s\}$;
6   ~~Retirer de S' tous les sites qui sont a distance au plus $2r$ de s;~~
7 ~~**si** $|C| \leq k$ **alors**~~   retourner C
8   ~~**retourner** C;~~
9 ~~**sinon**~~
10   ~~**retourner** "le rayon de couverture minimum est $> r$";~~

a. Que retourne l'algorithme ? Qu'est-ce qui est approxime ici ? Que retourne l'algorithme si le rayon r qui lui est fourni n'est pas le rayon de couverture minimum (contrairement a ce que nous supposons ici) ?

b. Quel est la complexite de cet algorithme ?

Le but des questions suivantes est de montrer que l'algorithme 1 est correct et realise un rapport d'approximation de 2 sur le rayon de couverture qui lui est fourni en entree. C'est a dire que s'il existe un ensemble d'au plus k centres qui a un rayon de couverture r, alors l'algorithme 1 termine avec $|S| \leq k$ et S realise un rayon de couverture d'au plus $2r$. Soit C^* une solution de rayon de couverture r. Soit C la solution retournee par l'algorithme.

c. Montrez que pour tout $c \in C$, il existe $c^* \in C^*$ tel que $d(c, c^*) \leq r$.

d. Soit $c^* \in C^*$ tel qu'il existe $c \in C$ tel que $d(c, c^*) \leq r$. Montrez que quel que soit $c' \in C \setminus \{c\}$, $d(c', c^*) > r$.

e. Montrez par l'absurde que si l'algorithme termine avec $|C| > k$ alors il n'existe pas d'ensemble d'au plus k centres qui ait un rayon de couverture r.

f. En deduire que s'il existe un ensemble d'au plus k centres qui a un rayon de couverture r, alors l'algorithme trouve un ensemble d'au plus k centres qui a un rayon de couverture au plus $2r$.

On veut maintenant faire un algorithme de 2-approximation sur le rayon de couverture sans connaitre le rayon de couverture optimal. Pour cela, on remarque qu'a la ligne 4 du precedent algo, on peut choisir n'importe quel site s qui ne soit pas encore couvert par les centres selectionnes auparavant avec un rayon de $2r$.

g. Lorsqu'il existe un site non couvert par les centres selectionnes auparavant dans l'algorithme avec un rayon $2r$, comment peut-on choisir un site s pour etre sur qu'il soit non couvert sans meme connaitre r ?

h. En utilisant le resultat de la question precedente, ecrivez un algorithme pour le probleme k-center selection qui retourne toujours un ensemble C d'exactement k centres (on supposera $n > k$ sinon le rayon de couverture minimum est 0) et qui garantisse un facteur d'approximation de 2 sur le rayon de couverture minimum, sans supposer cette fois aucune connaissance sur le rayon de couverture minimum.

Algorithme 1 : Algorithme de 2-approximation en connaissant le rayon de couverture minimum r.

1 $S' \leftarrow S$;
2 $C \leftarrow \varnothing$;
3 **tant que** $S' \neq \varnothing$ **faire**
4   Selectionner un $s \in S'$ quelconque;
5     $C \leftarrow C \cup \{s\}$;
6   Retirer de S' tous les sites qui sont a distance au plus $2r$ de s;
7 **si** $|C| \leq k$ **alors**
8   **retourner** C;
9 **sinon**
10   **retourner** "le rayon de couverture minimum est $> r$";

a. Que retourne l'algorithme ? Qu'est-ce qui est approxime ici ? Que retourne l'algorithme si le rayon r qui lui est fourni n'est pas le rayon de couverture minimum (contrairement a ce que nous supposons ici) ?

b. Quel est la complexite de cet algorithme ?

Le but des questions suivantes est de montrer que l'algorithme 1 est correct et realise un rapport d'approximation de 2 sur le rayon de couverture qui lui est fourni en entree. C'est a dire que s'il existe un ensemble d'au plus k centres qui a un rayon de couverture r, alors l'algorithme 1 termine avec $|S| \leq k$ et S realise un rayon de couverture d'au plus $2r$. Soit C^* une solution de rayon de couverture r. Soit C la solution retournee par l'algorithme.

c. Montrez que pour tout $c \in C$, il existe $c^* \in C^*$ tel que $d(c, c^*) \leq r$.

d. Soit $c^* \in C^*$ tel qu'il existe $c \in C$ tel que $d(c, c^*) \leq r$. Montrez que quel que soit $c' \in C \setminus \{c\}$, $d(c', c^*) > r$.

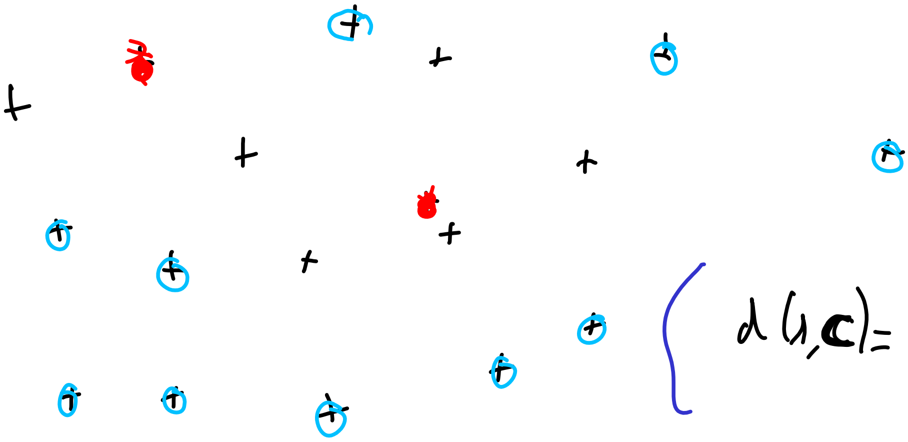
e. Montrez par l'absurde que si l'algorithme termine avec $|C| > k$ alors il n'existe pas d'ensemble d'au plus k centres qui ait un rayon de couverture r.

f. En deduire que s'il existe un ensemble d'au plus k centres qui a un rayon de couverture r, alors l'algorithme trouve un ensemble d'au plus k centres qui a un rayon de couverture au plus $2r$.

On veut maintenant faire un algorithme de 2-approximation sur le rayon de couverture sans connaitre le rayon de couverture optimal. Pour cela, on remarque qu'a la ligne 4 du precedent algo, on peut choisir n'importe quel site s qui ne soit pas encore couvert par les centres selectionnes auparavant avec un rayon de $2r$.

g. Lorsqu'il existe un site non couvert par les centres selectionnes auparavant dans l'algorithme avec un rayon $2r$, comment peut-on choisir un site s pour etre sur qu'il soit non couvert sans meme connaitre r ?

h. En utilisant le resultat de la question precedente, ecrivez un algorithme pour le probleme k-center selection qui retourne toujours un ensemble C d'exactement k centres (on supposera $n > k$ sinon le rayon de couverture minimum est 0) et qui garantisse un facteur d'approximation de 2 sur le rayon de couverture minimum, sans supposer cette fois aucune connaissance sur le rayon de couverture minimum.



$$d(s, C) = \min_{x \in C} \{ d(s, x) \}$$

pour le prochain je prends le site s le plus loin des centres déjà sélect.

$d(s, C)$ est max.

choisir s qcq $O(kn)$

$C \leftarrow \cancel{\emptyset}\ \{s\}$

tant que $|C| < k$ faire

- sélectionner s tq $d(s, C)$ est max
- $C \leftarrow C \cup \{s\}$

retourner C

i. Quel est la complexite de l'algorithme que vous avez propose a la question precedente ?

j. Prouvez que le rayon de couverture $\tilde{r}$ des k-centres retournes par votre algorithme est une 2-approximation du rayon de couverture minimum.

$$C \to C^*$$

$$c \mapsto c^* \in C^* \text{ tq } d(c, c^*) \leq r$$

- application bien définie par question c
- application injective par question d

conclusion $|C^*| \geq |C| > k$