Optimisation en nombres entiers

Recherche Opérationnelle

GC-SIE

Approximation
Heuristiques

Approximation

Idée : trouver une solution sous-optimale rapidement et mesurer sa qualité

Définition :

- Un algorithme H est une ε-approximation pour le problème de minimisation P avec coût optimal Z* si, pour toute instance de P, l'algorithme fonctionne en temps polynomial et identifie une solution admissible de coût Z_H telle que

$$Z_H \leq (1+\varepsilon)Z^*$$

Approximation

Exemple : couverture minimale

- Soit un graphe G=(V,E)
- V est l'ensemble des nœuds
- E est l'ensemble des arêtes
- Trouver le plus petit ensemble de nœuds $C \subseteq V$ tel que chaque arête du graphe ait au moins une extrémité dans C. On dit que C couvre les arêtes.

Problème très difficile (NP-complet)

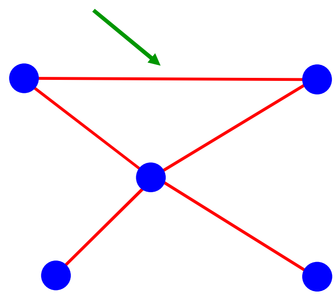
Application : installation de caméras de surveillance

Approximation

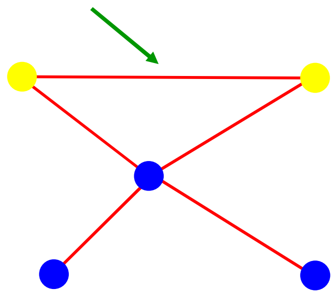
Algorithme simple

- $C = \emptyset$
- Tant que $E \neq \emptyset$
  - Choisir une arête (u,v) de E
  - Ajouter u et v à C
  - Supprimer u et v ainsi que les arêtes incidentes

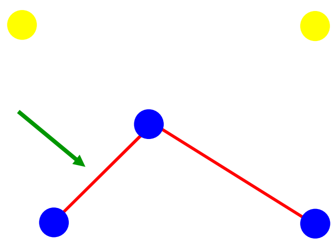
Approximation



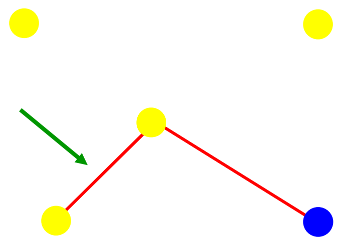
Approximation



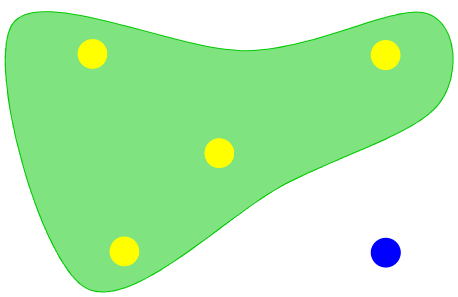
Approximation



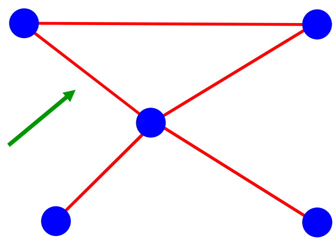
Approximation



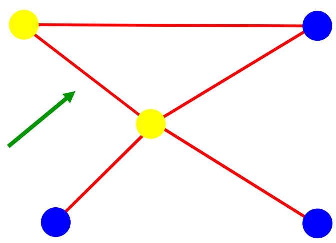
Approximation



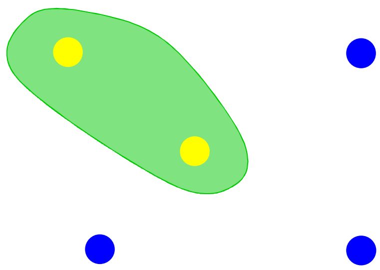
Approximation



Approximation



Approximation



Approximation

Analyse de l'algorithme

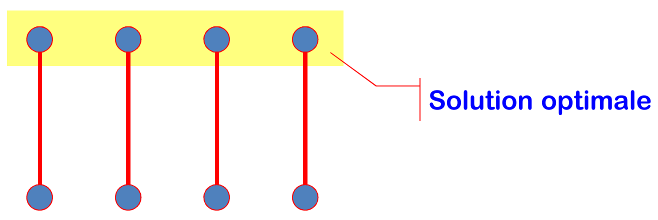
- Z_H = #nœuds(solution) = 2 #Arêtes sélectionnées
- Dans la solution optimale, chaque arêtes sélectionnées doit être couverte
- Donc, Z^*=#nœuds(optimum) $\geq$ #Arêtes sélectionnées
- On a donc $Z_H \leq 2\,Z^* = (1+1)Z^*$
- Il s'agit d'une 1-approximation

Approximation

- Cet algorithme va trouver un ensemble de nœuds couvrant toutes les arêtes
- Chaque arête choisie par l'algorithme sera couverte par un nœud différent de C
- Ces arêtes n'ont aucun nœud en commun
- Si on prend une couverture quelconque, elle devra contenir au moins un nœud de chaque arête choisie.
- La couverture optimale ne peut donc pas être plus petite que la moitié de C.
- Il s'agit d'une 1-approximation

Approximation

- Pire des cas :



- **Solution de l'algorithme : 8 nœuds (C=V)**
- **Solution optimale : 4 nœuds**

Heuristiques

Définition

- Une heuristique est une technique qui cherche une bonne solution (i.e. presque optimale) en un temps de calcul raisonnable sans être capable de garantir ni l'optimalité ni l'admissibilité.

Heuristiques

- La plupart des heuristiques sont basées sur la notion de voisinage.
- Un voisinage est l'ensemble des solutions obtenues à partir d'une solution donnée en effectuant un petit nombre de transformations simples.
- La définition de voisinage dépend du problème.

Heuristiques

Exemple

- Pour un programme linéaire en variables 0/1, le voisinage d'une solution $(x_1,\dots,x_n)$ est l'ensemble des $(y_1,\dots,y_n)$ telles que
  - $y_i = x_i$ si $i \neq k$
  - $y_k = 1-x_k$

  pour un k donné.

Heuristiques

Optimisation locale

- Soit une solution x_c
- Répéter
  - Choisir x voisin de x_c tel que $f(x) < f(x_c)$
  - Remplacer x_c par x
- Jusqu'à ce que $f(x) > f(x_c)$ pour tout x voisin de x_c.

Heuristiques

Notes :

- Facile à implémenter
- Potentiellement lent
- Trouve un minimum local

Heuristiques

Recuit simulé

- Motivation thermodynamique
- La structure d'un solide refroidi dépend de la vitesse de refroidissement
- A température T, la probabilité d'une augmentation d'énergie d'amplitude E est donnée par

$$\text{Prob}(E) = \exp(-E/kT)$$

où k est la constante de Boltzmann

$k=1{,}380\ 662\ 10^{-23}$ joules/degré absolu

Heuristiques

Algorithme de recuit simulé

- Soit une solution initiale x_c
- Soit une température initiale T_0
- Soit une fonction de réduction de température α
- Répéter
  - Répéter k fois
    - Choisir au hasard un voisin x de x_c
    - $\delta = f(x) - f(x_c)$
    - Si $\delta < 0$ alors $x_c = x$
    - Sinon
      - choisir r aléatoirement entre 0 et 1
      - Si $r < \exp(-\delta/T)$ alors $x_c = x$
  - $T = \alpha(T)$

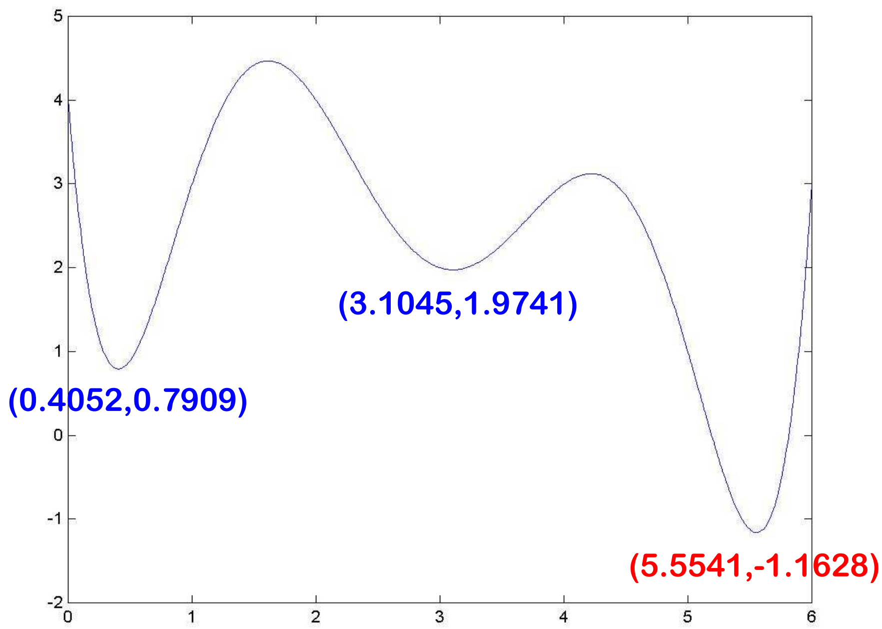
Heuristiques

Notes :

- Algorithme aléatoire
- Chaque exécution peut produire une solution différente

Exemple :

- $f(x)= 4 - 19.0167x + 36.39167x^2 - 25.2917x^3 + 8.041667x^4 - 1.19167x^5 + 0.066667x^6$

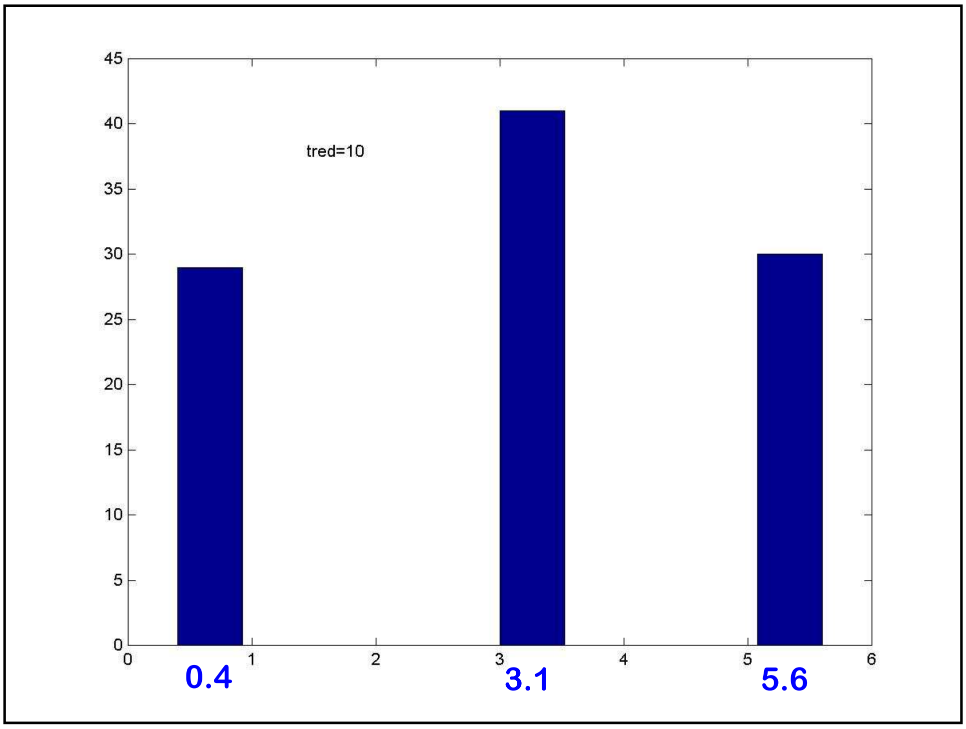


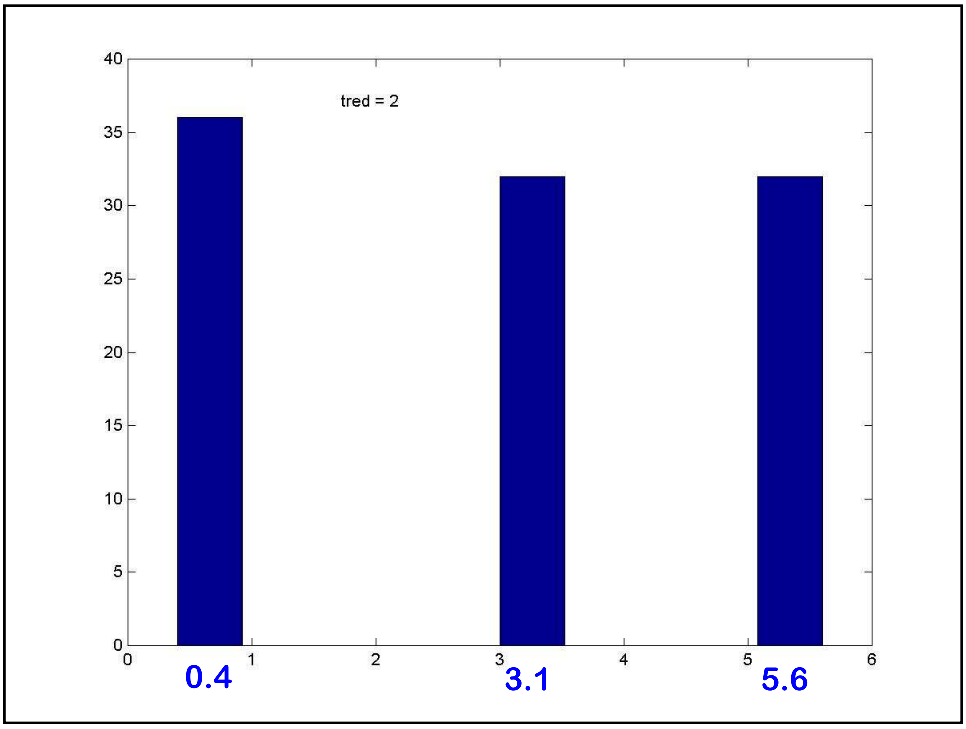
Heuristiques

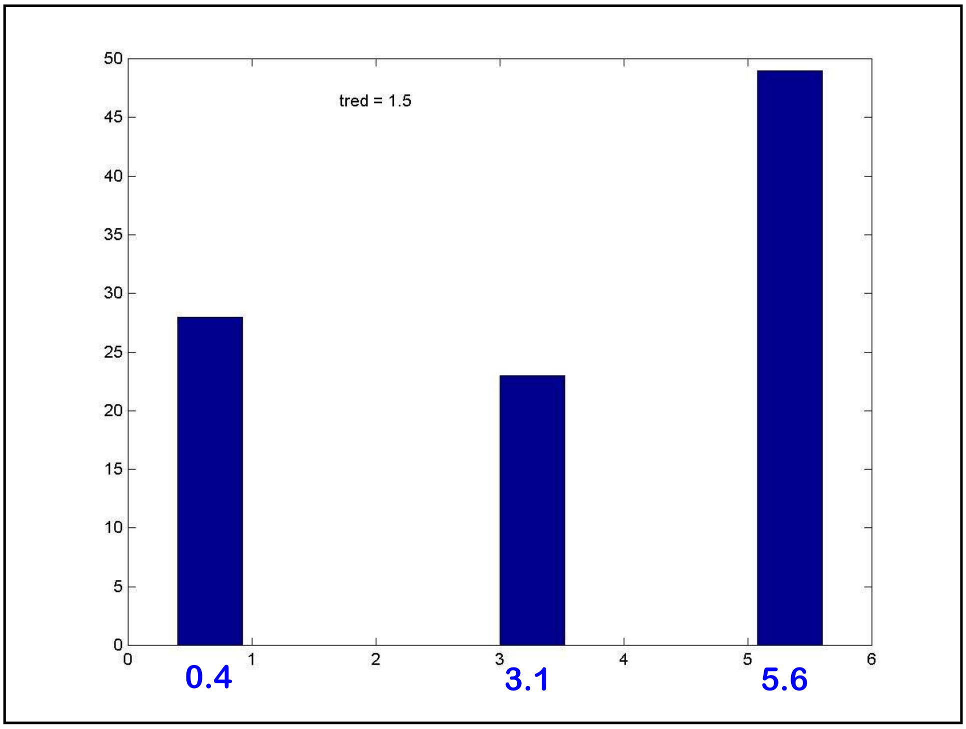
Paramètres utilisés :

- $x_c = 3$
- $T_0 = 10000$
- $\alpha(T) = T$ / tred
- k = 100
- Définition du voisinage de x :

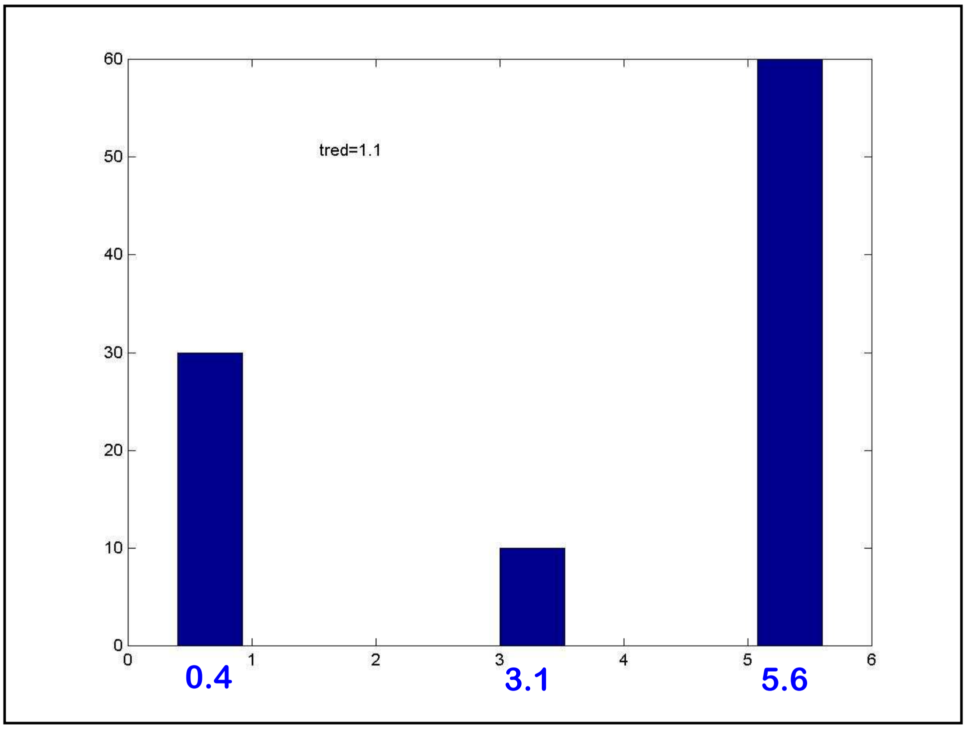
  {x-0.1,x+0.1}
- Résultats obtenus sur 100 exécutions


tred=10
0.4
3.1
5.6


tred = 2
0.4
3.1
5.6



tred = 1.5
50
45
40
35
30
25
20
15
10
5
0
0
1
2
3
4
5
6
0.4
3.1
5.6



tred=1.1
60
50
40
30
20
10
0
0
1
2
3
4
5
6
0.4
3.1
5.6

Heuristiques

Notes :

- Plus la température est diminuée lentement, plus la probabilité de trouver le minimum global augmente.
- Les paramètres présentés ici sont loin d'être optimaux.
- La méthode peut requérir énormément de temps calcul.

Heuristiques

Algorithme génétique

- Utilise, à chaque itération, un ensemble de solutions, appelé population.
- Chaque individu de cette population est codé en un ensemble de κ symboles, appelé chromosome.
- Exemple : notation binaire

  Individu = 19 Chromosome = 10011
- Pour chaque chromosome x, on peut calculer son « aptitude », son « niveau de qualité » f(x)

Heuristiques

Algorithme génétique :

- Initialisation : définir une population initiale P(0)
- A chaque itération k :
  - Sélection
  - Evolution
  - Mutation

Heuristiques

Sélection

- Pour chaque chromosome $x_i \in P(k)$, on définit
$$p_i = f(x_i)/\Sigma_{y\in P(k)} f(y)$$
- Cela définit une loi de probabilité sur la population.
- Choisir les individus reproducteurs aléatoirement selon cette loi de probabilité.

Heuristiques

Sélection (exemple)

maximiser $f(x)=x^3-60x^2+900x+100$

i	Chrom.	Individu	Aptitude	p_i	Cumulatif
1	10011	19	2399	0.206	0.206
2	00101	5	3225	0.277	0.484
3	11010	26	516	0.044	0.528
4	10101	21	1801	0.155	0.683
5	01110	14	3684	0.317	1.000
			11625		

Heuristiques

Sélection (exemple)

r(0,1)	Individus reproducteurs	Chromosomes
0.627	4	10101
0.140	1	10011
0.141	1	10011
0.934	5	01110
0.444	2	00101

Heuristiques

Evolution

- Deux parents sont choisis aléatoirement
- On effectue un croisement :
  - On sélectionne aléatoirement un emplacement i entre 1 et κ
  - On échanges les « gênes » des deux parents situés après l'emplacement i
- On remplace les parents par les « enfants »

Heuristiques

Evolution (exemple)

- Parents : 4 et 1

(21)	10101	10111 (23)
(19)	10011	10001 (17)

Heuristiques

Mutation

- On change les symboles de chaque chromosome avec une probabilité p_m (souvent très petite)

Heuristiques

Mutation (exemple)

p_m=0.1

r(0,1)	Symboles	Change	Mutant
0.934	1	non	1
0.048	0	oui	1
0.346	1	non	1
0.891	1	non	1
0.076	1	oui	0
	23		30

Heuristiques

Notes :

- Énormément de variantes possibles
- Il faut adapter les processus de sélection, d'évolution et de mutation à chaque problème spécifique