Optimisation en nombres entiers

Daniel Porumbel

Contributions aux slides:
Cédric Bentz
Lê Nguyên Hoang

Plan

1. Difficulté des problèmes de nature discrète et explosion combinatoire
2. Méthodes de résolution en optimisation discrète
3. Exemples problèmes résolus avec `glpsol` (avec correction)

Un problème de nature discrète : les n dames

Comment placer n dames sur un échiquier de $n \times n$ cases sans s'attaquer mutuellement ? **Considérons $n = 4$**

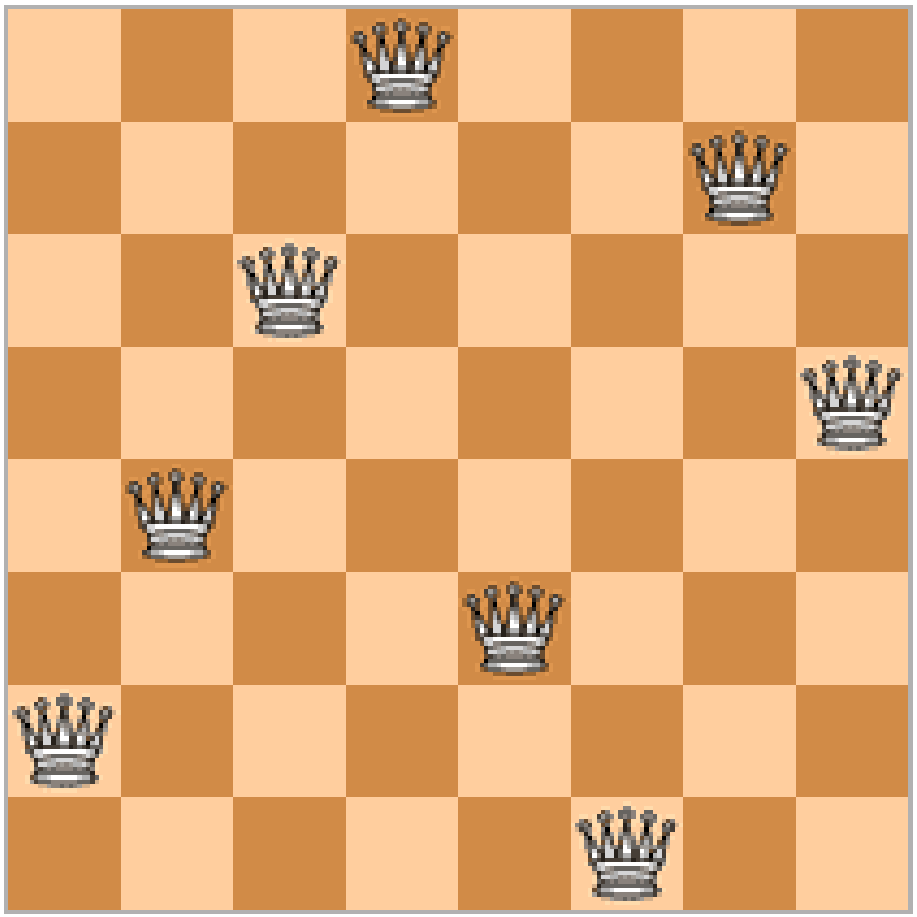
- Une solution pour $n = 8 \longrightarrow$
- Idée 2 (résolution complète mais naïve) : vérifier n^n solutions candidates ! ! !
  - Même $n = 4$ demande 256 solutions
    - voir vidéos pour $n = 8$

Explosion combinatoire du nombre exponentiel de solutions (possibilités) envisageables

Un problème de nature discrète : les n dames

Comment placer n dames sur un échiquier de $n \times n$ cases sans s'attaquer mutuellement ? Considérons $n = 4$

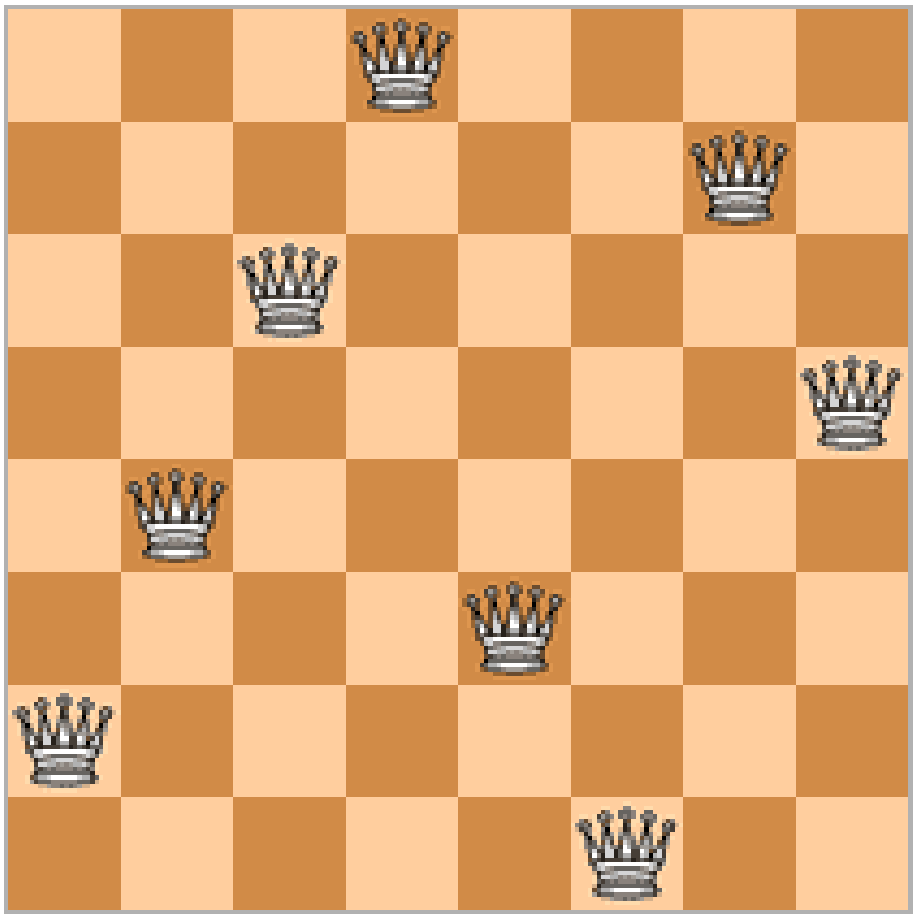
- Une solution pour $n = 8 \longrightarrow$
  - Idée 1 : placer n dames n'importe où et essayer de réparer étape par étape. Cela peut mener à une solution ou pas...
- Idée 2 (résolution complète mais naïve) : vérifier n^n solutions candidates ! ! !
  - Même $n = 4$ demande 256 solutions
    - voir vidéos pour $n = 8$



Un problème de nature discrète : les n dames

Comment placer n dames sur un échiquier de $n \times n$ cases sans s'attaquer mutuellement ? Considérons $n = 4$

- Une solution pour $n = 8 \longrightarrow$
- Idée 2 (résolution complète mais naïve) : vérifier n^n solutions candidates ! ! !
  - Même $n = 4$ demande 256 solutions
    - voir vidéos pour $n = 8$



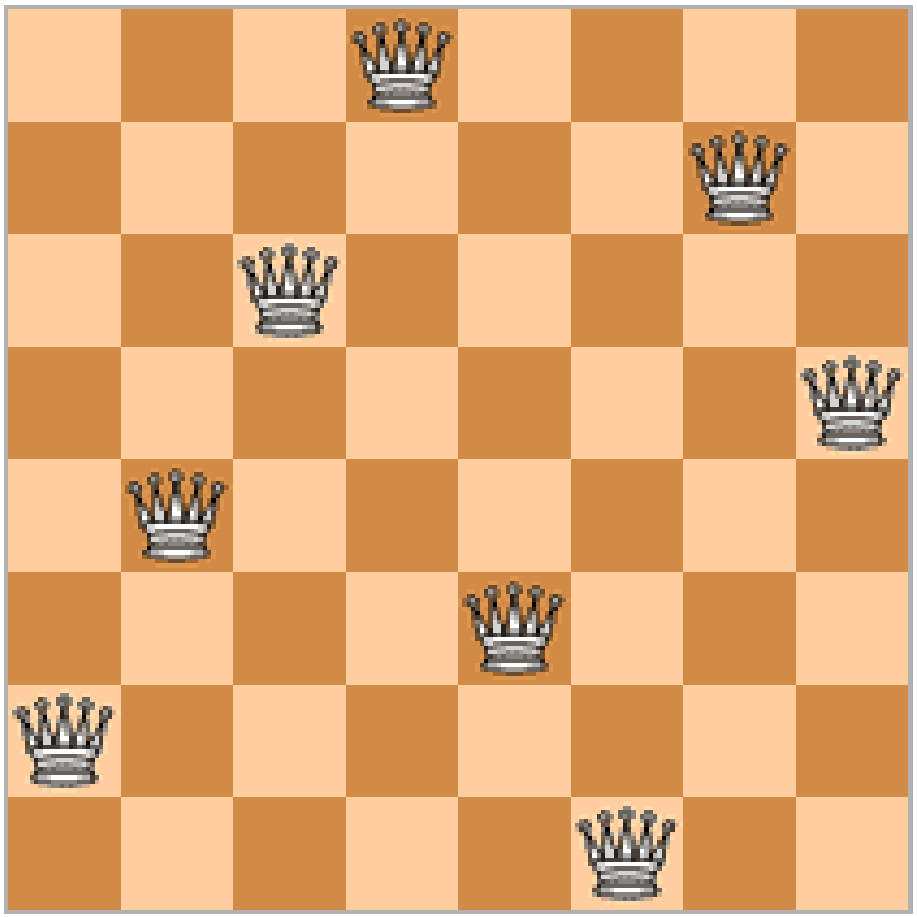
Explosion combinatoire

nombre exponentiel de solutions (possibilités) envisageables.

Un problème de nature discrète : les n dames

Comment placer n dames sur un échiquier de $n \times n$ cases sans s'attaquer mutuellement ? $\boxed{\text{Considérons } n = 4}$

- Une solution pour $n = 8 \longrightarrow$
- Idée 2 (résolution complète mais naïve) : vérifier n^n solutions candidates !!!
  - Même $\boxed{n = 4}$ demande 256 solutions
    - voir vidéos pour $n = 8$

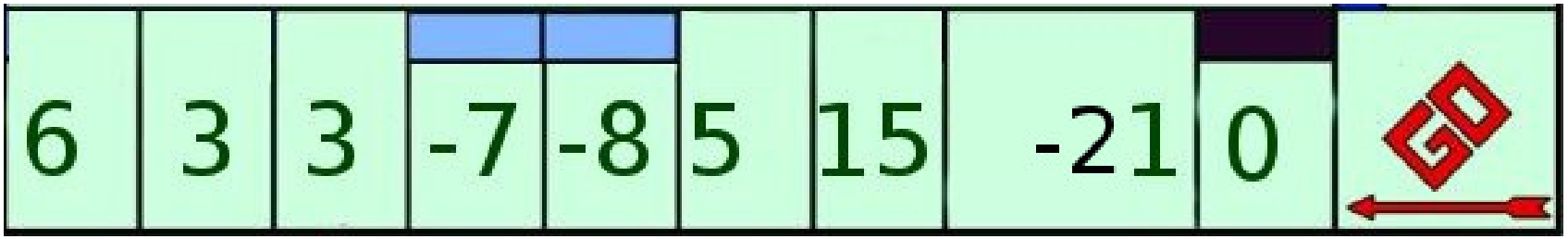


Explosion combinatoire

nombre exponentiel de solutions (possibilités) envisageables.

Un Jeu : Décision Optimale de Sauts

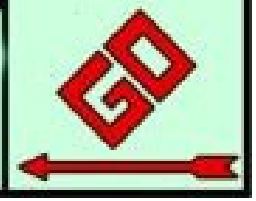
On considère le début d'un jeu de type *Monopoly* :



On avance vers la gauche à l'aide des opérations suivantes :

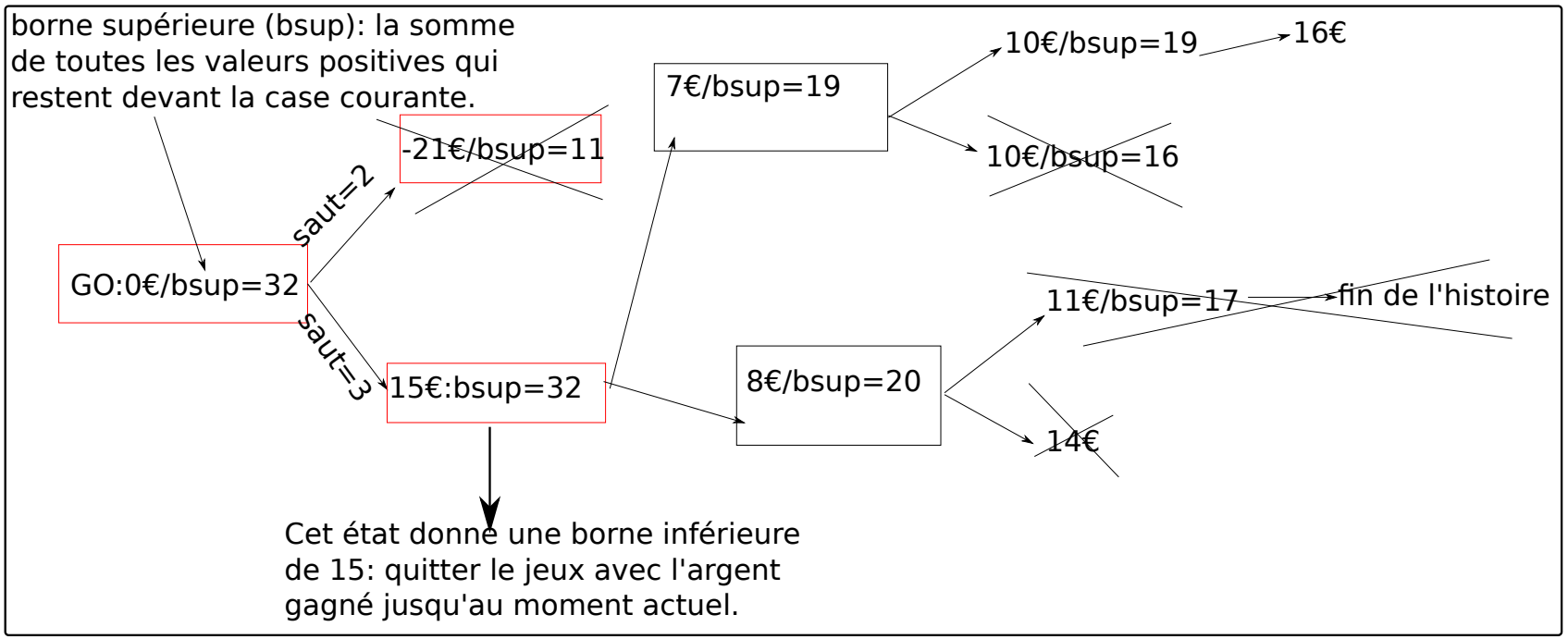
- avancer 2 cases
- avancer 3 cases
- quitter avec l'argent déjà récupéré

Comment calculer le profit maximal (noté OPT) ?

6	3	3	-7	-8	5	15	-21	0	

Solution jeu précédent

Voici l'arbre de branchement (flèche vers le haut=saut de 2, flèche vers la bas=saut de 3)



Comment dérober le maximum ?

- Un cambrioleur arrive à s'introduire dans une banque

Il peut voler

- des barres d'or et
- des liasses de billets

Problème : 2 limitations de son sac-à-dos

- charge max : 20kg
- volume max : 32 litres

une barre d'or : 300000$, 8kg, 6litres

un paquet de billets : 100000$, 3kg, 6litres

Comment dérober le maximum ?

- Un cambrioleur arrive à s'introduire dans une banque

Il peut voler

- des barres d'or et
- des liasses de billets

Problème : 2 limitations de son sac-à-dos

- charge max : 20kg
- volume max : 32 litres

une barre d'or : 300000$, 8kg, 6litres

un paquet de billets : 100000$, 3kg, 6litres

Comment dérober le maximum ?

- Un cambrioleur arrive à s'introduire dans une banque

Il peut voler

- des barres d'or et
- des liasses de billets

Problème : 2 limitations de son sac-à-dos

- charge max : 20kg
- volume max : 32 litres

une barre d'or : 300000$, 8kg, 6litres

un paquet de billets : 100000$, 3kg, 6litres

Comment dérober le maximum ?

- Un cambrioleur arrive à s'introduire dans une banque

Il peut voler

- des barres d'or et
- des liasses de billets

Problème : 2 limitations de son sac-à-dos

- charge max : 20kg
- volume max : 32 litres

une barre d'or : 300000$, 8kg, 6litres

un paquet de billets : 100000$, 3kg, 6litres

Et si on faisait l'effort de porter 1kg de plus ? Ou 2kg ?

Un programme linéaire en nombres entiers

variable x_1 nombre de barres d'or (300000\$, 8kg, 6l)

variable x_2 nombre de paquets de billets (100000\$, 3kg, 6litres)

$$\max 3x_1 + x_2 \quad \leftarrow \text{maximiser le profit}$$

$$\left.\begin{array}{r} 8x_1 + 3x_2 \leq 20 \\ 6x_1 + 6x_2 \leq 32 \end{array}\right\} \quad \leftarrow \text{Contraintes de poids et volume}$$

$$x_1, x_2 \in \mathbb{Z}_+ \quad \leftarrow \text{Contraintes d'intégralité}$$

Un simple exemple `glpk`

```
var x1>=0;
var x2>=0;
maximize obj:3*x1+x2;
subject to c1: 8*x1+3*x2<=20;
subject to c2: 6*x1+6*x2<=32;
solve;
display x1,x2;
end;
```

Vous pouvez l'exécuter à

`cocoto.github.io/glpk-online/`

Pour passer en nombres entiers, on déclare :

```
var x1>=0,integer;
```

Un simple exemple `glpk`

```
var x1>=0;
var x2>=0;
maximize obj:3*x1+x2;
subject to c1: 8*x1+3*x2<=20;
subject to c2: 6*x1+6*x2<=32;
solve;
display x1,x2;
end;
```

Vous pouvez l'exécuter à
`cocoto.github.io/glpk-online/`
Pour passer en nombres entiers, on déclare :
`var x1>=0,integer;`

Le même modele en julia 1

```
using JuMP;
using GLPK;                          #ou "using Cbc"
m = Model(GLPK.Optimizer); #ou "Cbc.Optimizer"

# Define the variables
@variable(m, x>=0, Int);
@variable(m, y>=0, Int);

# La fonction objectif
@objective(m, Max, 3x+y);

# Les contraintes
@constraint(m, 8x+3y<=20);
@constraint(m, 6x+6y<=32);
```

. . . , à suivre slide suivant, . . .

Le même modele en julia 2

Une fois un modèle m défini, on l'optimise ainsi :

```
# Faire tourner le solveur
optimize!(m);

# Output
println(objective_value(m))    # val opt obj
println("x = ", value.(x),     # val opt x
        "\n",                  # saut de ligne
        "y = ", value.(y))     # val opt y
```

Que pensez vous du problème suivant ?

- Vous êtes un restarateur de luxe, mais vous n'avez que 20 chaises et 5 tables. Vous pouvez offrir des tables de 8 personnes ou de 3 personnes.
- Une table de 8 vous apporte 3000 euros
- Une table de 3 vous apporte 1000 euros

$$\max 3 \cdot x_1 + x_2$$
$$8x_1 + 3x_2 \leq 20$$
$$x_1 + x_2 \leq 5$$
$$x_1, x_2 \in \mathbb{Z}_+$$

Peut-on gagner plus d'argent si on apporte encore une table ?

Que pensez vous du problème suivant ?

- Vous êtes un restarateur de luxe, mais vous n'avez que 20 chaises et 5 tables. Vous pouvez offrir des tables de 8 personnes ou de 3 personnes.
- Une table de 8 vous apporte 3000 euros
- Une table de 3 vous apporte 1000 euros

$$\max 3 \cdot x_1 + x_2$$
$$8x_1 + 3x_2 \leq 20$$
$$x_1 + x_2 \leq 5$$
$$x_1, x_2 \in \mathbb{Z}_+$$

Peut-on gagner plus d'argent si on apporte encore une table ?

Que pensez vous du problème suivant ?

- Vous êtes un restarateur de luxe, mais vous n'avez que 20 chaises et 5 tables. Vous pouvez offrir des tables de 8 personnes ou de 3 personnes.
- Une table de 8 vous apporte 3000 euros
- Une table de 3 vous apporte 1000 euros

$$\max 3 \cdot x_1 + x_2$$
$$8x_1 + 3x_2 \leq 20$$
$$x_1 + x_2 \leq 5$$
$$x_1, x_2 \in \mathbb{Z}_+$$

Peut-on gagner plus d'argent si on apporte encore une table ?

Que pensez vous du problème suivant ?

- Vous êtes un restarateur de luxe, mais vous n'avez que 20 chaises et 5 tables. Vous pouvez offrir des tables de 8 personnes ou de 3 personnes.
- Une table de 8 vous apporte 3000 euros
- Une table de 3 vous apporte 1000 euros

$$\max 3 \cdot x_1 + x_2$$
$$8x_1 + 3x_2 \leq 20$$
$$x_1 + x_2 \leq 5$$
$$x_1, x_2 \in \mathbb{Z}_+$$

Peut-on gagner plus d'argent si on apporte encore une table ?

Vers l'optimisation à grande échelle

- Le problème du cambrioleur : 2 variables
- Mais : souvent besoin d'un tableau de n ou n^2 variables
- Ex : Soit n articles et un tableau de prix, ex $a = [1, 10, 11, 3]$
  - pas le droit de prendre deux articles consecutifs ;
  - quels articles choisir pour maximiser le prix total ?

```
1 param n, >0, integer;
2 param a{1..n}, integer;
3 var x{1..n}, >=0, binary;
4 s.t. c_consec{i in 1..n-1}: x[i]+x[i+1]<=1;
5 maximize obj: sum{i in 1..n} a[i]*x[i];
6 solve;
7 for {i in 1..n}{ printf "%d ", x[i]; }
8 data;
9 param n:=4;
10 param a:= 1 1
11          2 10
12          3 11
13          4 3 ;
14 end;
```

Vers l'optimisation à grande échelle

- Le problème du cambrioleur : 2 variables
- Mais : souvent besoin d'un tableau de n ou n^2 variables
- Ex : Soit n articles et un tableau de prix, ex $a = [1, 10, 11, 3]$
  - pas le droit de prendre deux articles consecutifs ;
  - quels articles choisir pour maximiser le prix total ?

```
1 param n, >0, integer;
2 param a{1..n}, integer;
3 var x{1..n}, >=0, binary;
4 s.t. c_consec{i in 1..n-1}: x[i]+x[i+1]<=1;
5 maximize obj: sum{i in 1..n} a[i]*x[i];
6 solve;
7 for {i in 1..n}{ printf "%d ", x[i]; }
8 data;
9 param n:=4;
10 param a:= 1 1
11           2 10
12           3 11
13           4 3 ;
14 end;
```

Les tableaux en Julia

```
using JuMP;
using GLPK;
m = Model(GLPK.Optimizer);

n = 4;                          #on a 4 articles
a = [1, 10, 11, 3];             #avec 4 profits

@variable(m, x[1:n], Bin);#n variables binaires

@objective(m, Max, sum(a[i]*x[i] for i in 1:n))

@constraint(m, [i in 1:n-1], x[i]+x[i+1] <= 1)

optimize!(m);                       # on optimize!
println(objective_value(m)) # opt obj val
println("x = ", value.(x))  # opt x
```