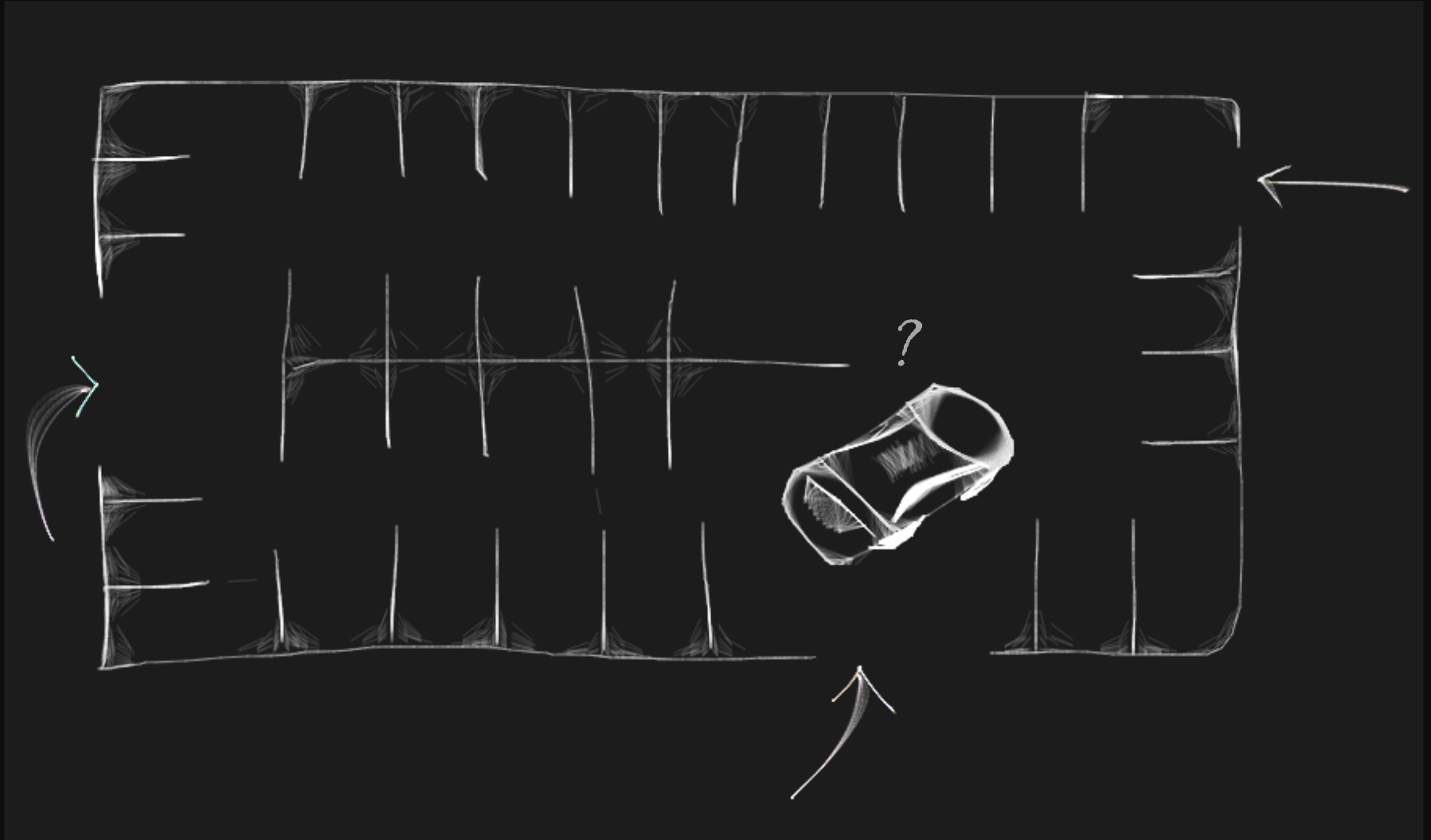

OPTIMISATION DU TRAFIC ROUTIER DANS LES PARKINGS

Victor Barilly - 10328

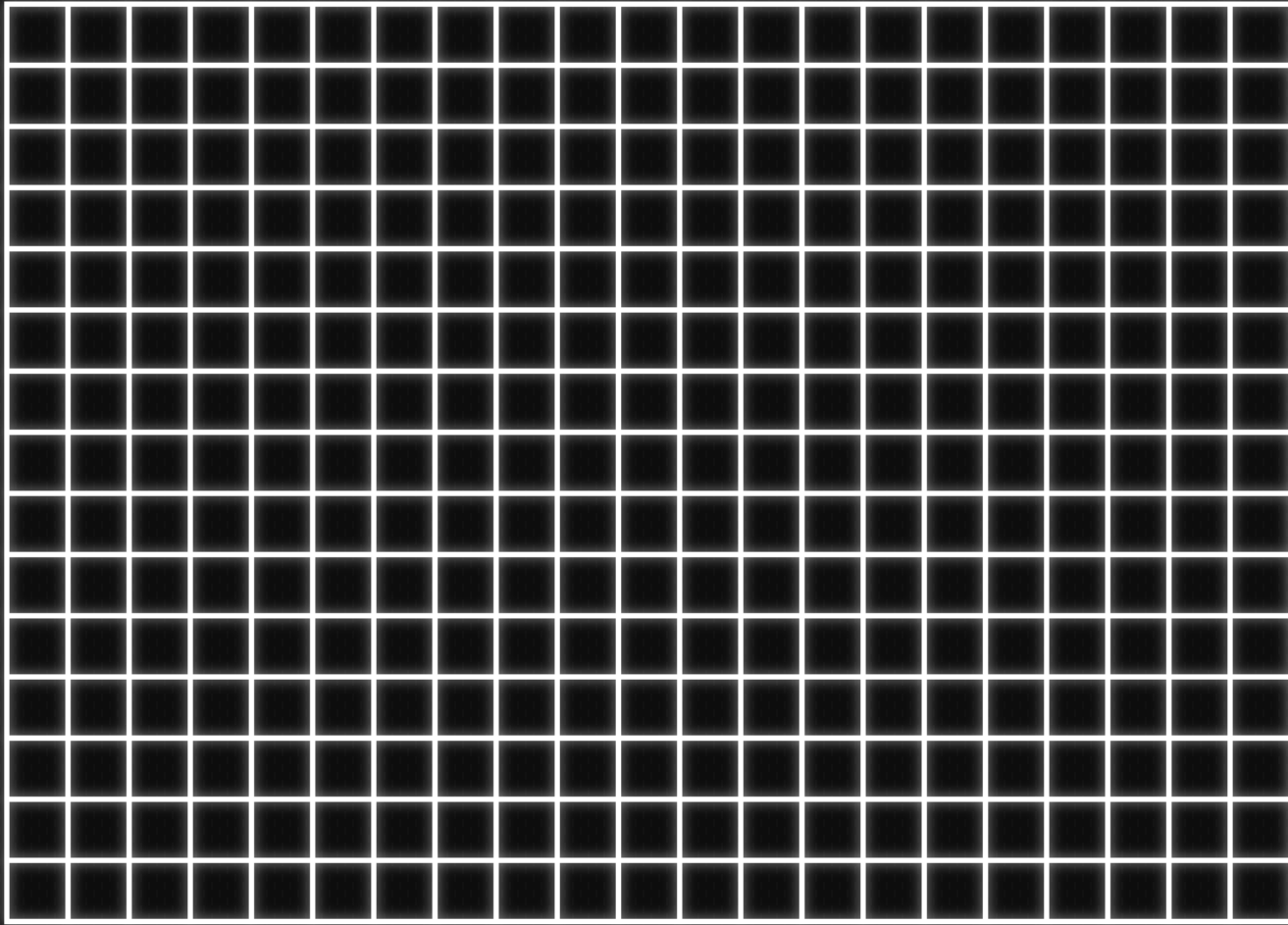
SOMMAIRE

- ❖ Introduction du problème
 - ❖ Modélisation d'un parking
 - ❖ Algorithmes génétiques
 - ❖ Aboutissement à un résultat approché
-

- ❖ Introduction du problème (1/1)
 - ❖ Modélisation d'un parking
 - ❖ Algorithmes génétiques
 - ❖ Aboutissement à un résultat approché
-




- ❖ Introduction du problème
 - ❖ Modélisation d'un parking (1/3)
 - ❖ Algorithmes génétiques
 - ❖ Aboutissement à un résultat approché
-

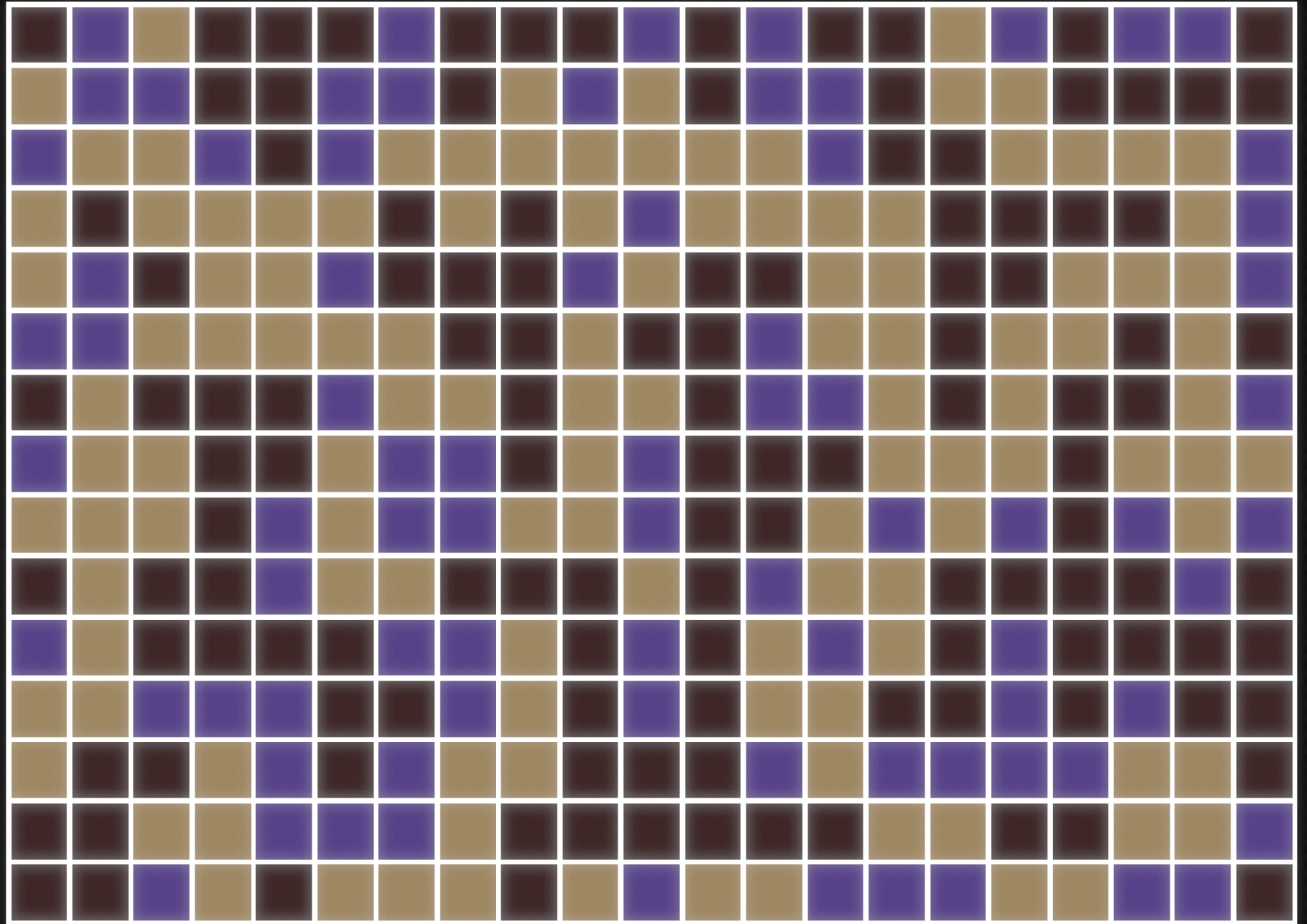


- ❖ Introduction du problème
 - ❖ Modélisation d'un parking (2/3)
 - ❖ Algorithmes génétiques
 - ❖ Aboutissement à un résultat approché
-

 : MUR


 : PLACE

 : ROUTE

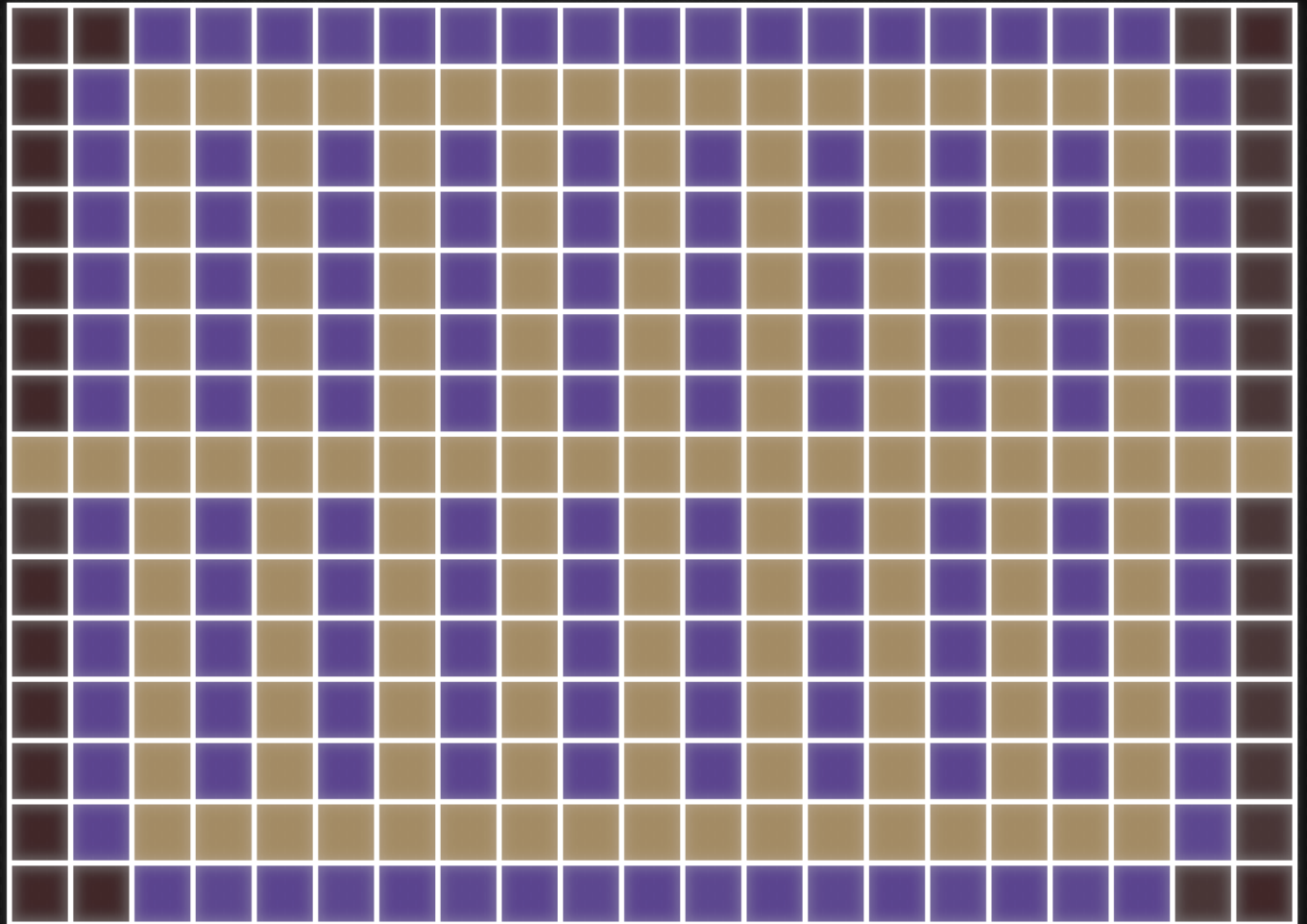


- ❖ Introduction du problème
 - ❖ Modélisation d'un parking (3/3)
 - ❖ Algorithmes génétiques
 - ❖ Aboutissement à un résultat approché
-

 : MUR

 : PLACE

 : ROUTE



- ❖ Introduction du problème
 - ❖ Modélisation d'un parking
 - ❖ Algorithmes génétiques (1/17)
 - ❖ Aboutissement à un résultat approché
-

Peut-on trouver la meilleure infrastructure de parking
à l'aide d'un algorithme génétique ?

$[P_1, P_2, P_3, P_4, P_5, \dots, P_{N-4}, P_{N-3}, P_{N-2}, P_{N-1}, P_N]$

↓ Sélection

$[P_1, P_2, P_3, P_4, P_5, \dots, P_{N-4}, P_{N-3}, P_{N-2}, P_{N-1}, P_N]$

↓ Sélection

$[P_2, P_4, P_5, \dots, P_{N-3}, P_{N-1}, P_N]$

↓ Croisement

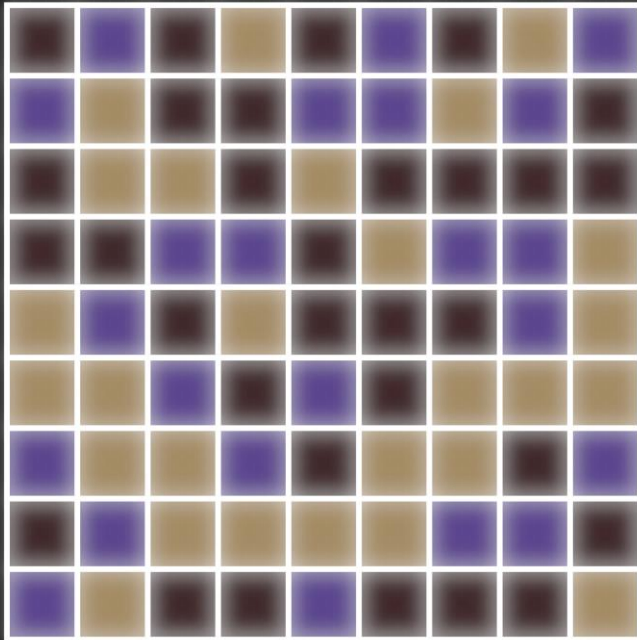
$[P'_1, P'_2, P'_3, P'_4, \dots, P'_{N-3}, P'_{N-2}, P'_{N-1}, P'_N]$

↓ Mutation

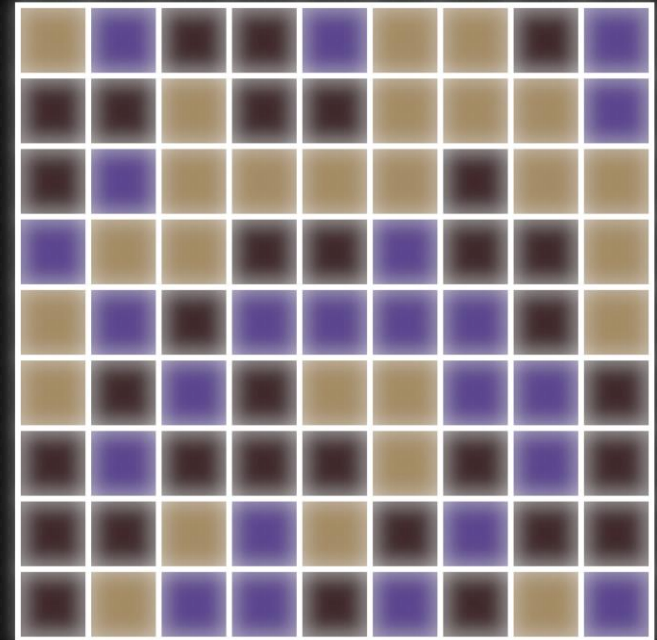
$[P_1, P_2, P_3, P_4, P_5, \dots, P_{N-4}, P_{N-3}, P_{N-2}, P_{N-1}, P_N]$

- ❖ Introduction du problème
 - ❖ Modélisation d'un parking
 - ❖ Algorithmes génétiques (3/17)
 - ❖ Aboutissement à un résultat approché
-

1. Sélection: $[P_1, P_2, \dots, P_{N-1}, P_N] \rightarrow [P_2, P_4, \dots, P_{N-1}, P_N]$



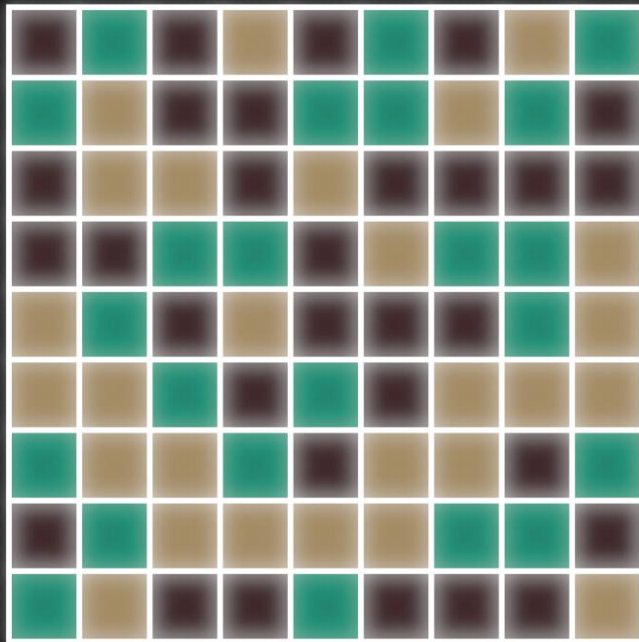
\neq



- ❖ Introduction du problème
 - ❖ Modélisation d'un parking
 - ❖ Algorithmes génétiques (4/17)
 - ❖ Aboutissement à un résultat approché
-

1. Sélection: $[P_1, P_2, \dots, P_{N-1}, P_N] \rightarrow [P_2, P_4, \dots, P_{N-1}, P_N]$

└→ 1^{re} idée de score: nombre de places du parking



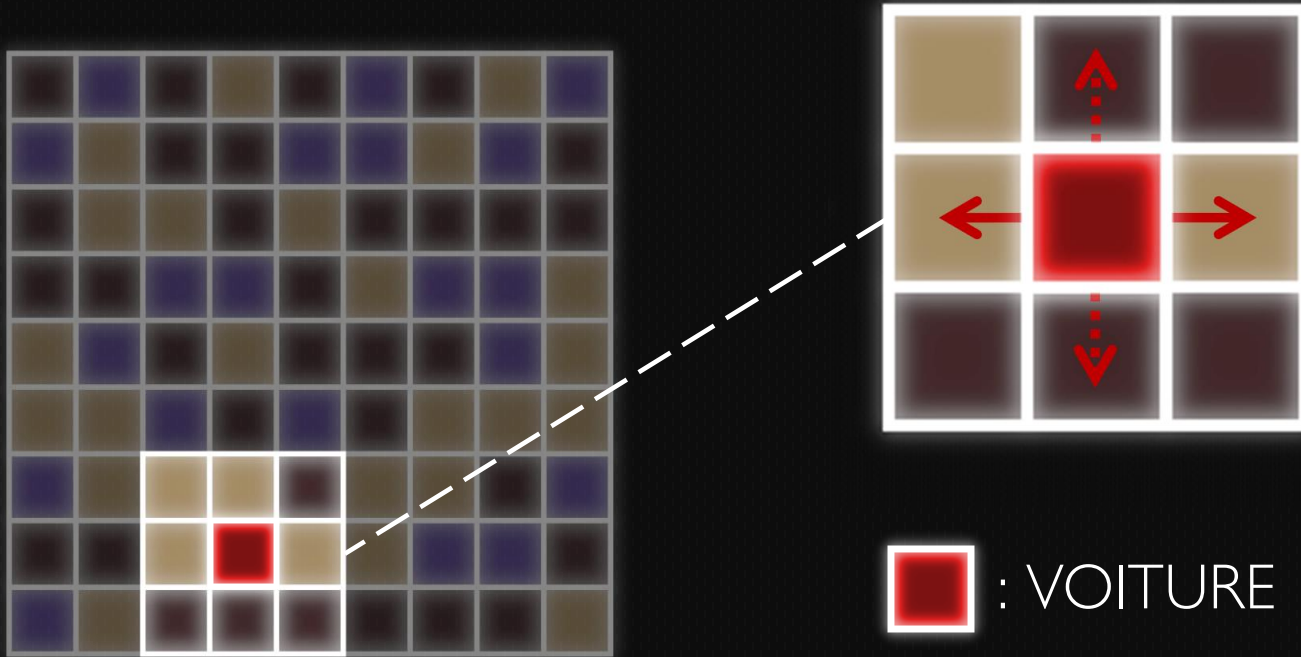
>> problème

- ❖ Introduction du problème
- ❖ Modélisation d'un parking
- ❖ Algorithmes génétiques (5/17)
- ❖ Aboutissement à un résultat approché

1. Sélection: $[P_1, P_2, \dots, P_{N-1}, P_N] \rightarrow [P_2, P_4, \dots, P_{N-1}, P_N]$

• 1^{re} idée de score: nombre de places du parking

→ 2^e idée de score: **simulation** par automate cellulaire

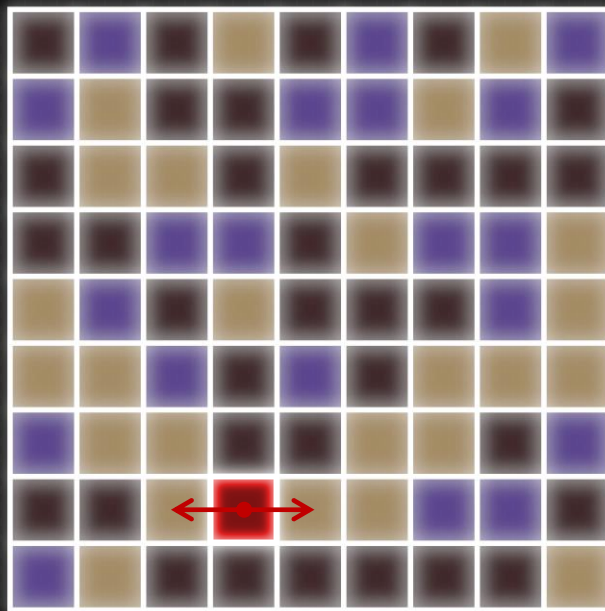


- ❖ Introduction du problème
- ❖ Modélisation d'un parking
- ❖ Algorithmes génétiques (6/17)
- ❖ Aboutissement à un résultat approché

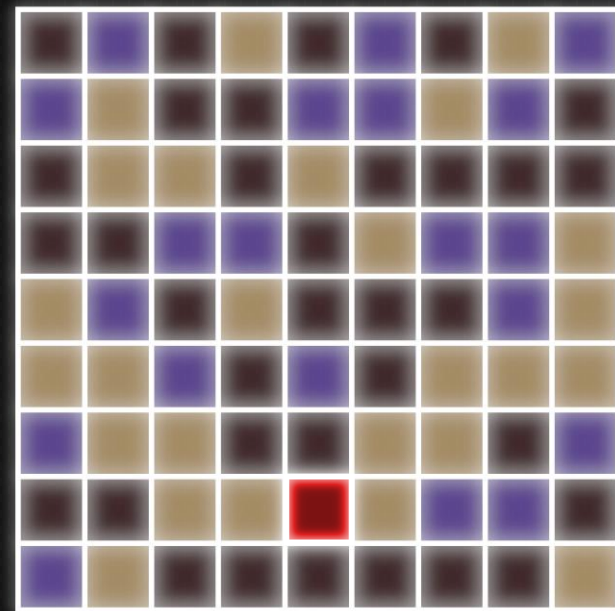
1. Sélection: $[P_1, P_2, \dots, P_{N-1}, P_N] \rightarrow [P_2, P_4, \dots, P_{N-1}, P_N]$

→ 2^e idée de score: **simulation** par automate cellulaire

→ Mouvement aléatoire



Instant t



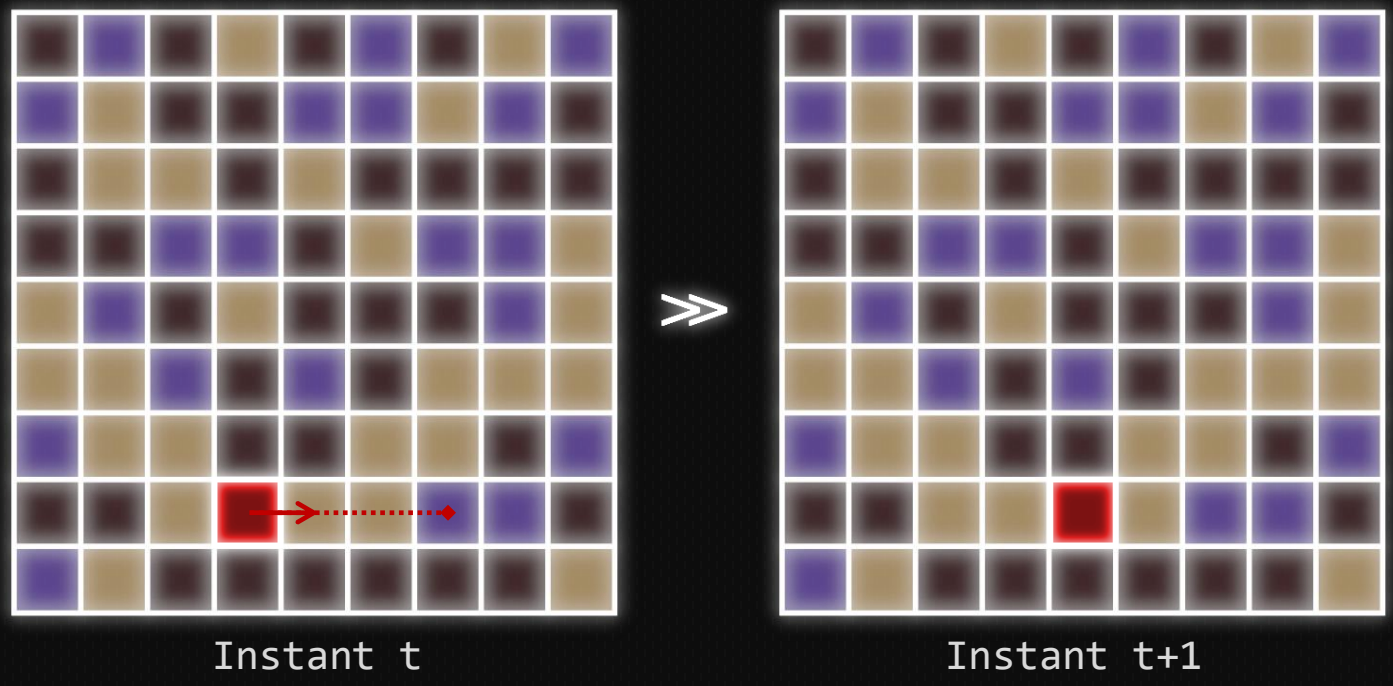
Instant t+1

- ❖ Introduction du problème
- ❖ Modélisation d'un parking
- ❖ Algorithmes génétiques (7/17)
- ❖ Aboutissement à un résultat approché

1. Sélection: $[P_1, P_2, \dots, P_{N-1}, P_N] \rightarrow [P_2, P_4, \dots, P_{N-1}, P_N]$

→ 2^e idée de score: **simulation** par automate cellulaire

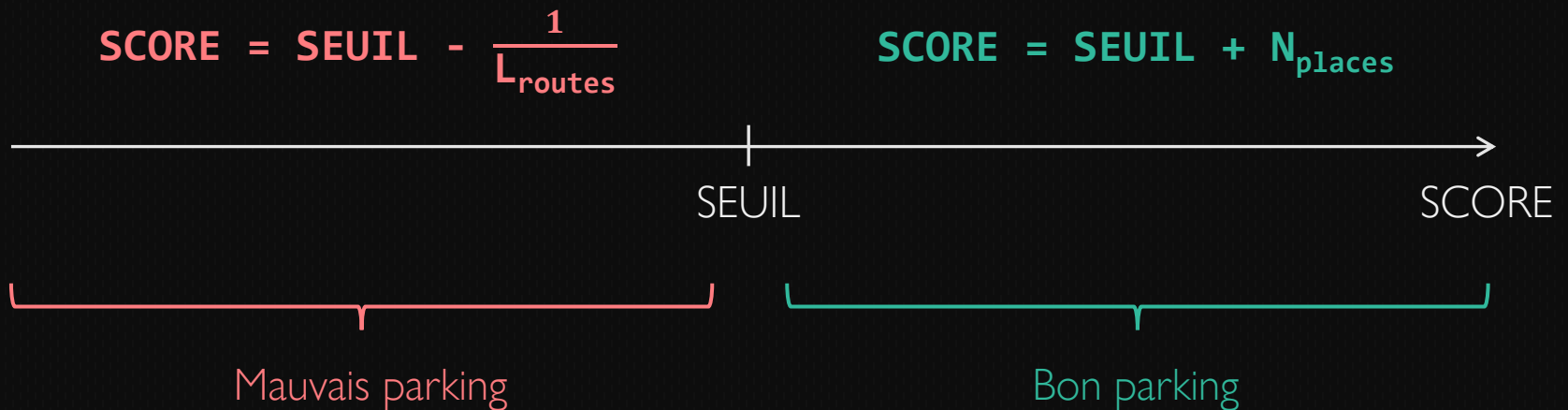
→ Mouvement déterminé



- ❖ Introduction du problème
- ❖ Modélisation d'un parking
- ❖ Algorithmes génétiques (8/17)
- ❖ Aboutissement à un résultat approché

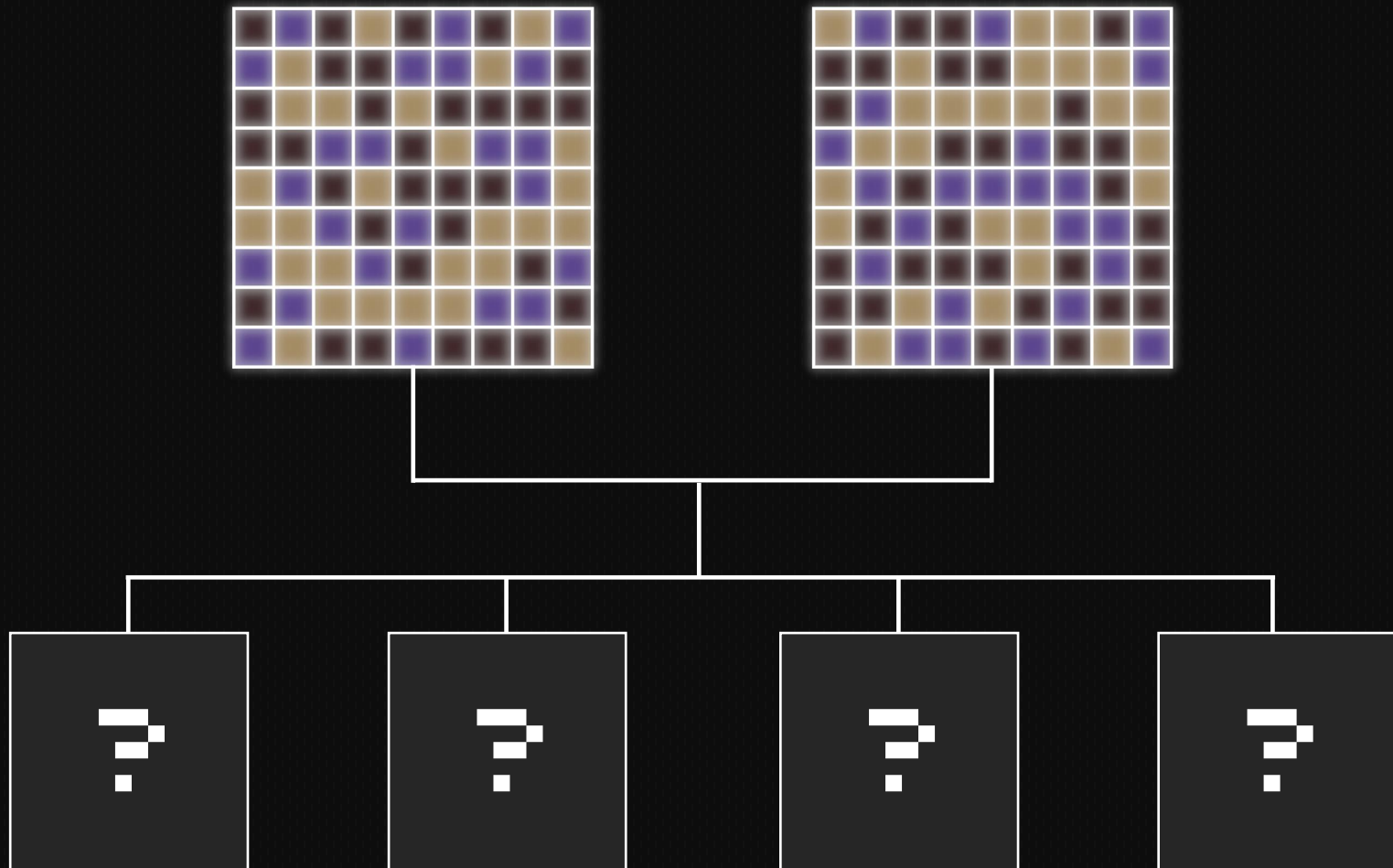
1. Sélection: $[P_1, P_2, \dots, P_{N-1}, P_N] \rightarrow [P_2, P_4, \dots, P_{N-1}, P_N]$

- 1^{re} idée de score: nombre de places du parking
- 2^e idée de score: simulation par automate cellulaire
- Notation retenue: disjonction selon la “linéarité” du parking



- ❖ Introduction du problème
 - ❖ Modélisation d'un parking
 - ❖ Algorithmes génétiques (9/17)
 - ❖ Aboutissement à un résultat approché
-

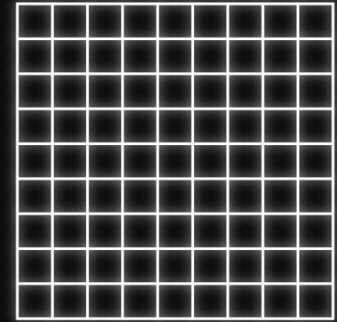
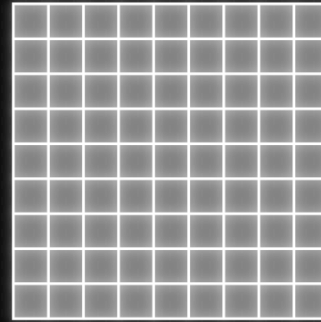
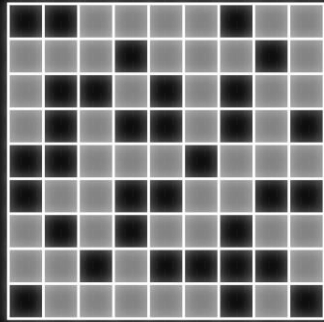
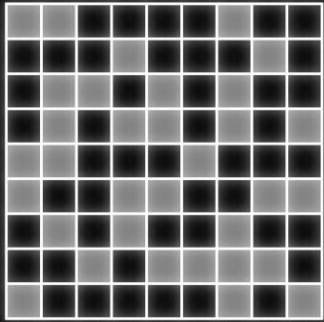
2. Croisement: $[P_2, P_4, \dots, P_{N-1}, P_N] \rightarrow [P'_1, P'_2, \dots, P'_{N-1}, P'_N]$



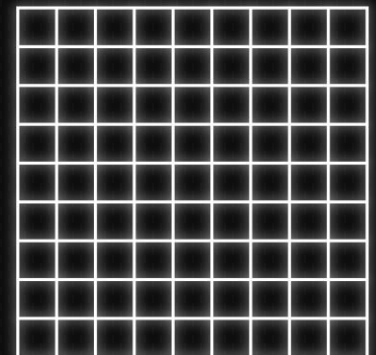
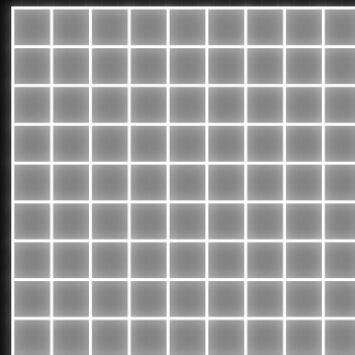
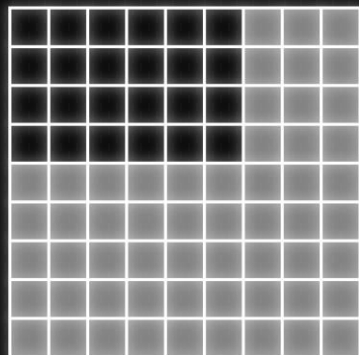
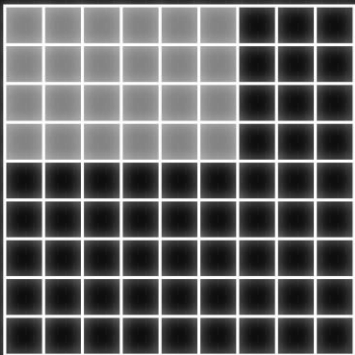
- ❖ Introduction du problème
 - ❖ Modélisation d'un parking
 - ❖ Algorithmes génétiques (10/17)
 - ❖ Aboutissement à un résultat approché
-

2. Croisement: $[P_2, P_4, \dots, P_{N-1}, P_N] \rightarrow [P'_1, P'_2, \dots, P'_{N-1}, P'_N]$

• UNIFORME



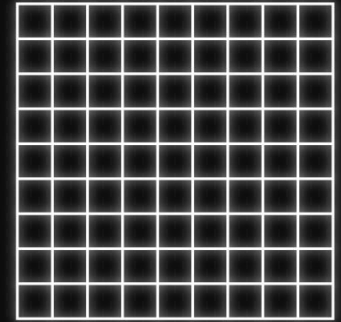
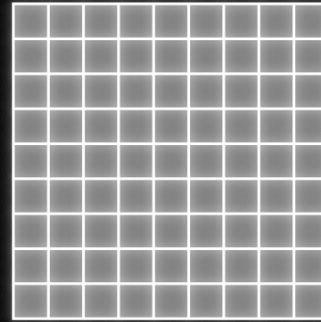
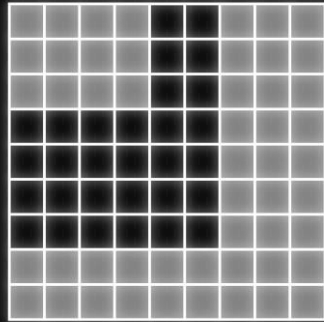
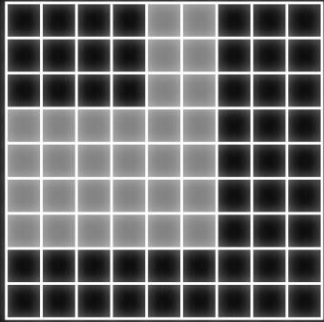
• À UN POINT DE COUPURE



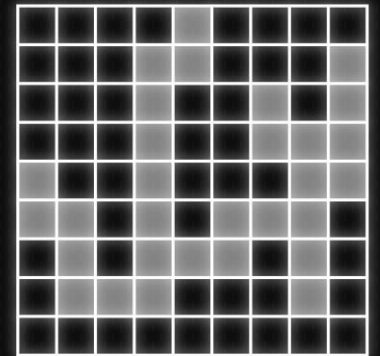
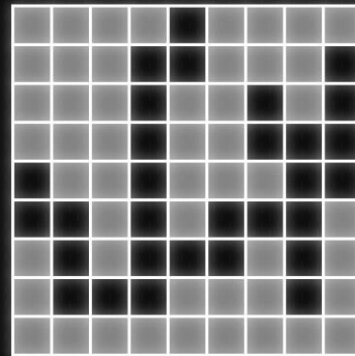
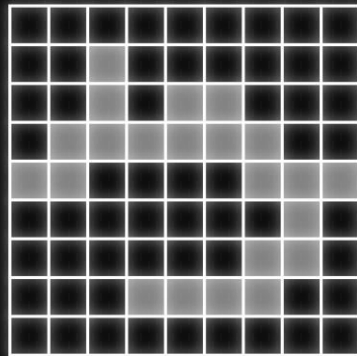
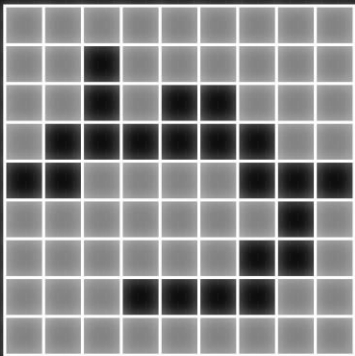
- ❖ Introduction du problème
- ❖ Modélisation d'un parking
- ❖ Algorithmes génétiques (11/17)
- ❖ Aboutissement à un résultat approché

2. Croisement: $[P_2, P_4, \dots, P_{N-1}, P_N] \rightarrow [P'_1, P'_2, \dots, P'_{N-1}, P'_N]$

• À PLUSIEURS POINTS DE COUPURES



• À PARTAGE DE ROUTES

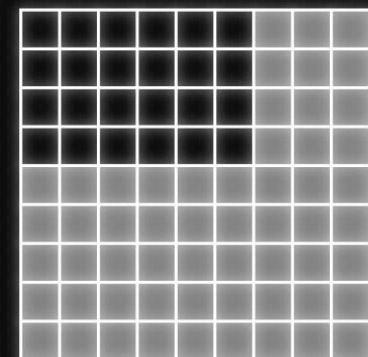
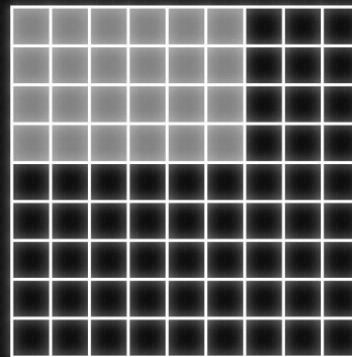


- ❖ Introduction du problème
 - ❖ Modélisation d'un parking
 - ❖ Algorithmes génétiques (12/17)
 - ❖ Aboutissement à un résultat approché
-

2. Croisement: $[P_2, P_4, \dots, P_{N-1}, P_N] \rightarrow [P'_1, P'_2, \dots, P'_{N-1}, P'_N]$

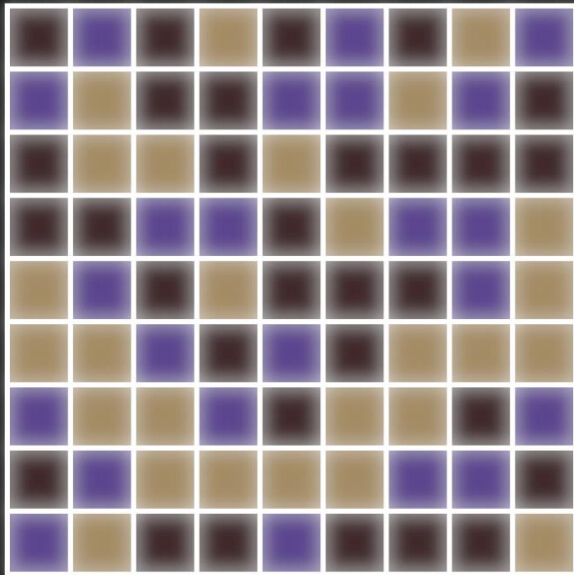
└→ Croisement retenu: 1 point de coupure

Aucune difference donc j'ai utilisé celui à 1 pt de coupure

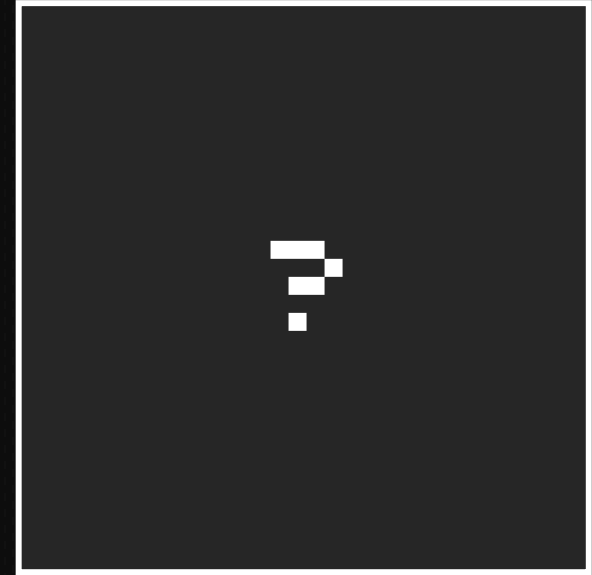


- ❖ Introduction du problème
 - ❖ Modélisation d'un parking
 - ❖ Algorithmes génétiques (13/17)
 - ❖ Aboutissement à un résultat approché
-

3. Mutation: $[P'_1, P'_2, \dots, P'_{N-1}, P'_N] \rightarrow [P_1, P_2, \dots, P_{N-1}, P_N]$



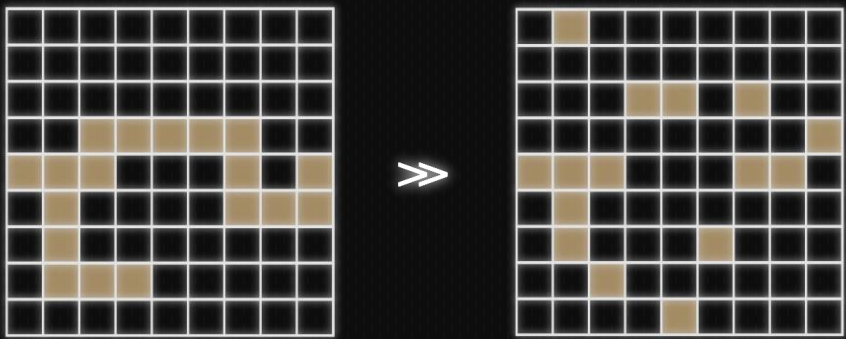
\Rightarrow



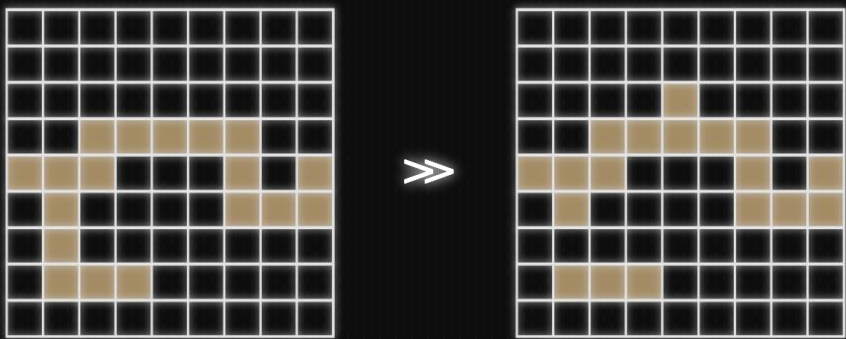
- ❖ Introduction du problème
 - ❖ Modélisation d'un parking
 - ❖ Algorithmes génétiques (14/17)
 - ❖ Aboutissement à un résultat approché
-

3. Mutation: $[P'_1, P'_2, \dots, P'_{N-1}, P'_N] \rightarrow [P_1, P_2, \dots, P_{N-1}, P_N]$

• ALEATOIRE



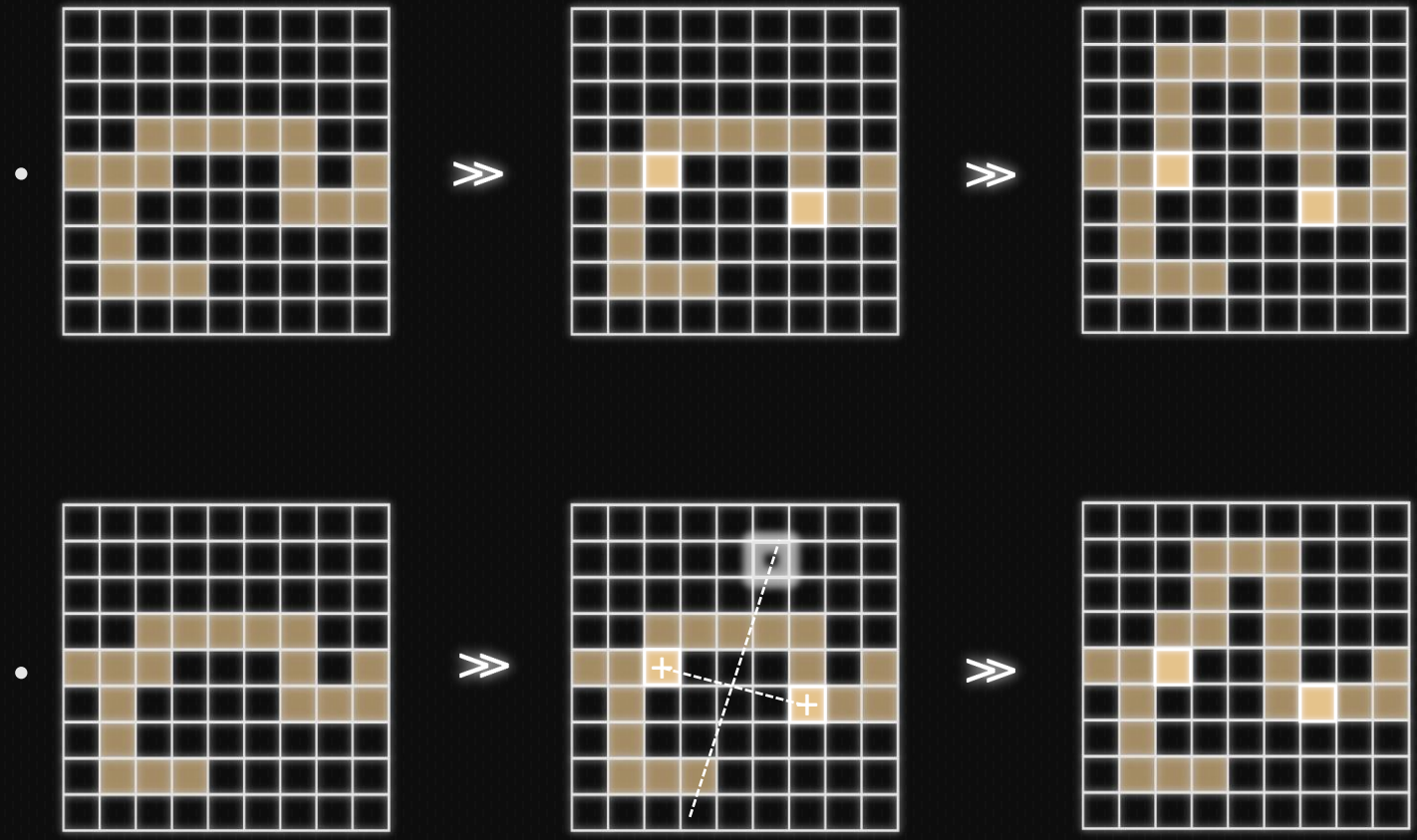
• ALLONGEMENT DE LA ROUTE



- ❖ Introduction du problème
- ❖ Modélisation d'un parking
- ❖ Algorithmes génétiques (15/17)
- ❖ Aboutissement à un résultat approché

3. Mutation: $[P'_1, P'_2, \dots, P'_{N-1}, P'_N] \rightarrow [P_1, P_2, \dots, P_{N-1}, P_N]$

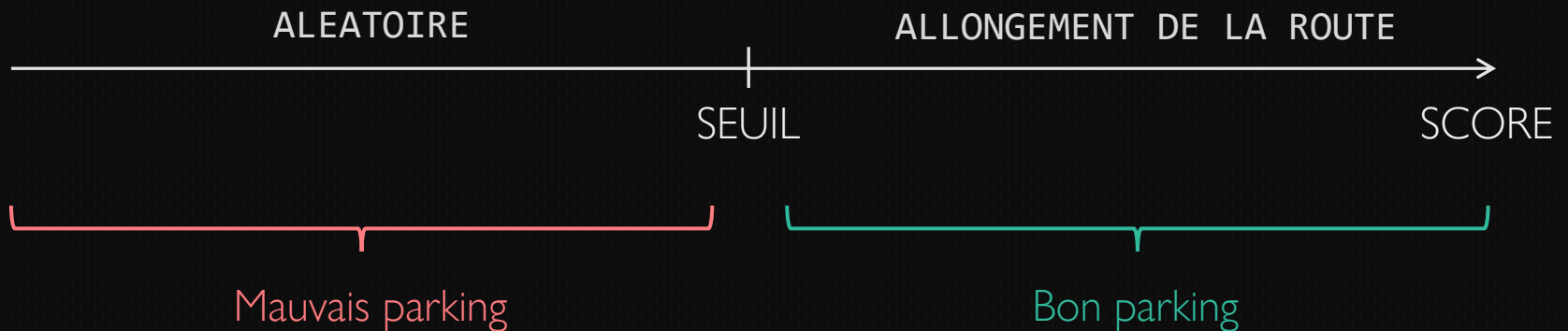
• DEVIATION DE LA ROUTE



- ❖ Introduction du problème
 - ❖ Modélisation d'un parking
 - ❖ Algorithmes génétiques (16/17)
 - ❖ Aboutissement à un résultat approché
-

3. Mutation: $[P'_1, P'_2, \dots, P'_{N-1}, P'_N] \rightarrow [P_1, P_2, \dots, P_{N-1}, P_N]$

↳ Mutation retenue: disjonction selon la “linéarité” du parking



$[P_1^1, P_2^1, P_3^1, \dots, P_{N-1}^1, P_N^1]$



$[P_1^2, P_2^2, P_3^2, \dots, P_{N-1}^2, P_N^2]$



⋮

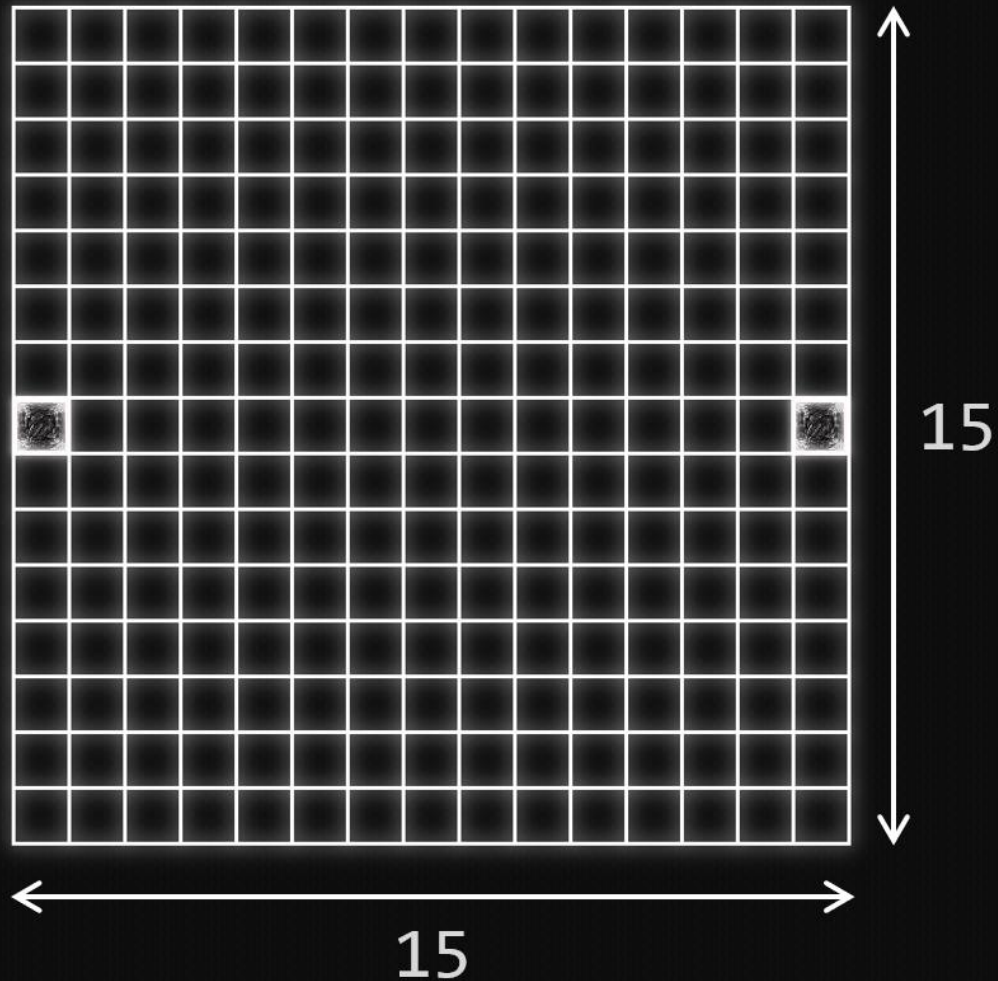


$[P_1^N, P_2^N, P_3^N, \dots, P_{N-1}^N, P_N^N]$

- ❖ Introduction du problème
 - ❖ Modélisation d'un parking
 - ❖ Algorithmes génétiques
 - ❖ Aboutissement à un résultat approché
-

N_PARKINGS = 500

N_GENERATIONS = 15



- ❖ Introduction du problème
- ❖ Modélisation d'un parking
- ❖ Algorithmes génétiques
- ❖ Aboutissement à un résultat approché

$$score = seuil + \frac{1}{L_{routes}}$$

$$score = seuil + N_{places}$$

SEUIL

SCORE

Mauvais parking

Bon parking

résultat pas
totalement
satisfaisant, essai
avec random puis
BINGO

0

250

250

GENERATIONS

- ❖ Introduction du problème
- ❖ Modélisation d'un parking
- ❖ Algorithmes génétiques
- ❖ Aboutissement à un résultat approché

$$\text{score} = \text{seuil} + \frac{1}{L_{\text{routes}}}$$

$$\text{score} = \text{seuil} + \frac{1}{\Delta t_{\text{garage}}}$$

SEUIL

SCORE

Mauvais parking

Bon parking



0

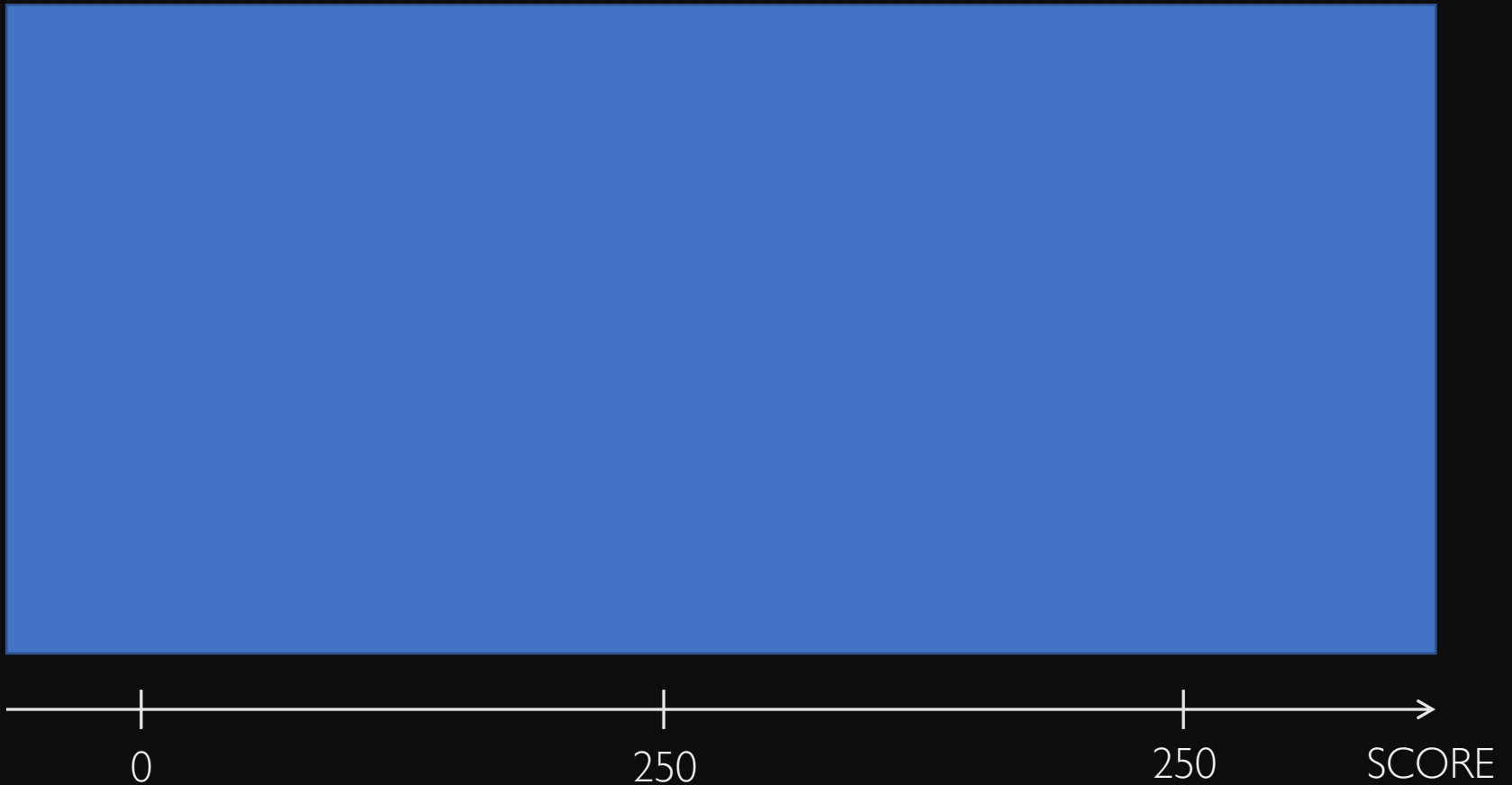
250

250

GENERATIONS

- ❖ Introduction du problème
 - ❖ Modélisation d'un parking
 - ❖ Algorithmes génétiques
 - ❖ Aboutissement à un résultat approché
-

- Dispersion des scores :



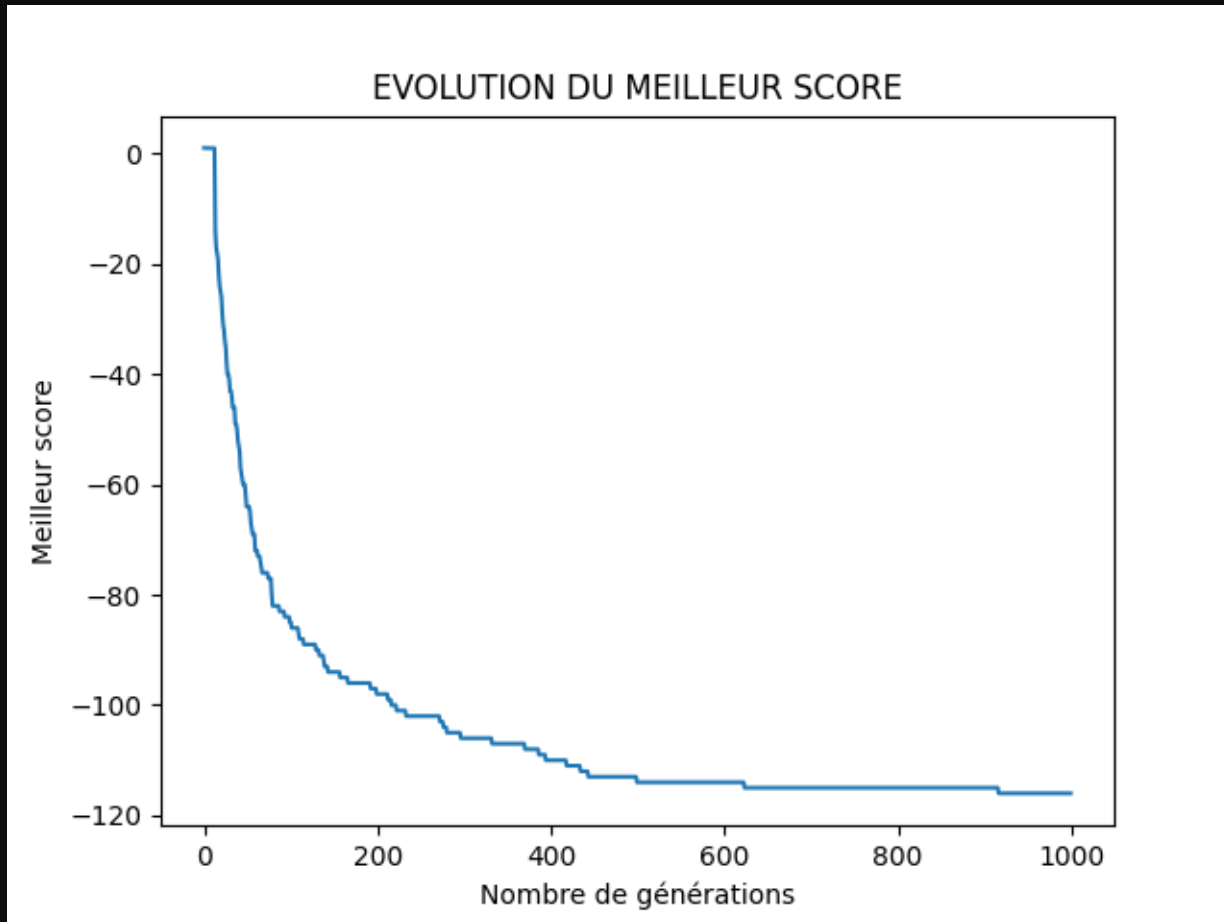
- ❖ Introduction du problème
 - ❖ Modélisation d'un parking
 - ❖ Algorithmes génétiques
 - ❖ Aboutissement à un résultat approché
-

- Dispersion des allures :

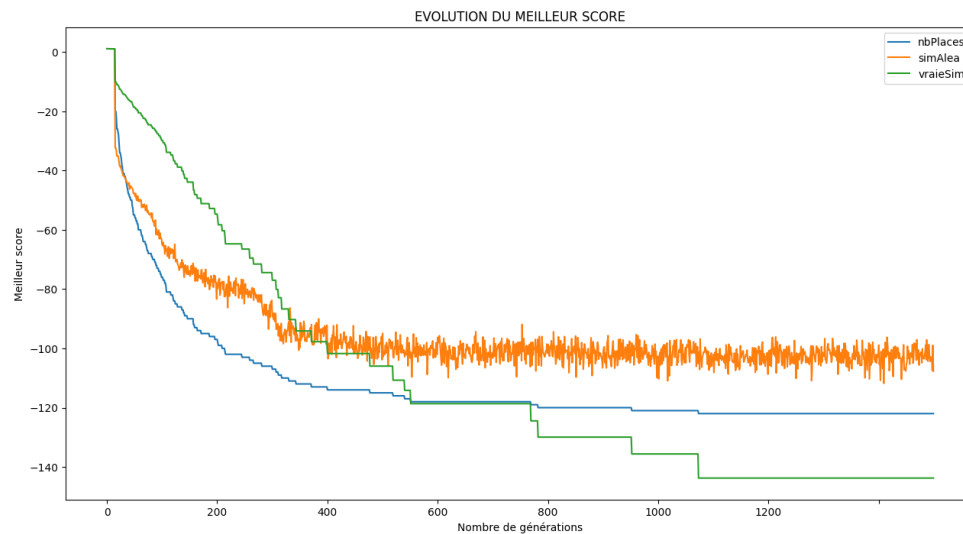
Tracer “écart type”
selon nb generation

- ❖ Introduction du problème
 - ❖ Modélisation d'un parking
 - ❖ Algorithmes génétiques
 - ❖ Aboutissement à un résultat approché
-

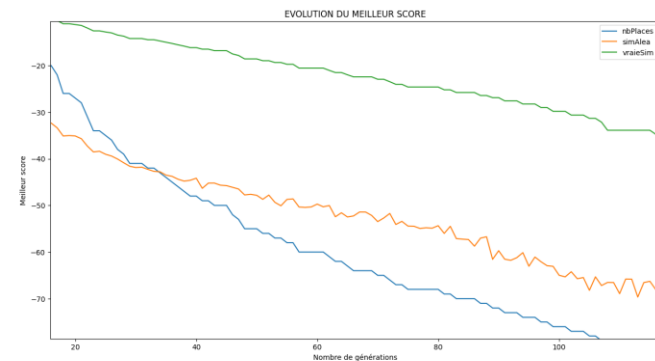
- Évolution du meilleur score :



- Cohérence avec d'autres simulations :



Étudier
impact
random



- Données “réelles” du parking final obtenu :



Flux de voitures	Duree Moyenn e g	(durée Moyenn e sortie?)			
Zdhd	Qeffds	Kjfqses			
edsnfk	Dsfsdf	Zdez			
efnskfns	Dsfvv	Dzgtyeh			
slkdfnlld	bgthfgh	uktui			

CONCLUSIONS ET LIMITES DE L'ÉTUDE

Comment faire plusieurs sens de circulation

Au début j'ai essayé de créer des règles pour recréer code de la route mais pbs pour parkings moches...

Donc on peut modéliser chaque case par un quintuplet (type, bool,..., bool) avec système de cords mais il faut algo + long