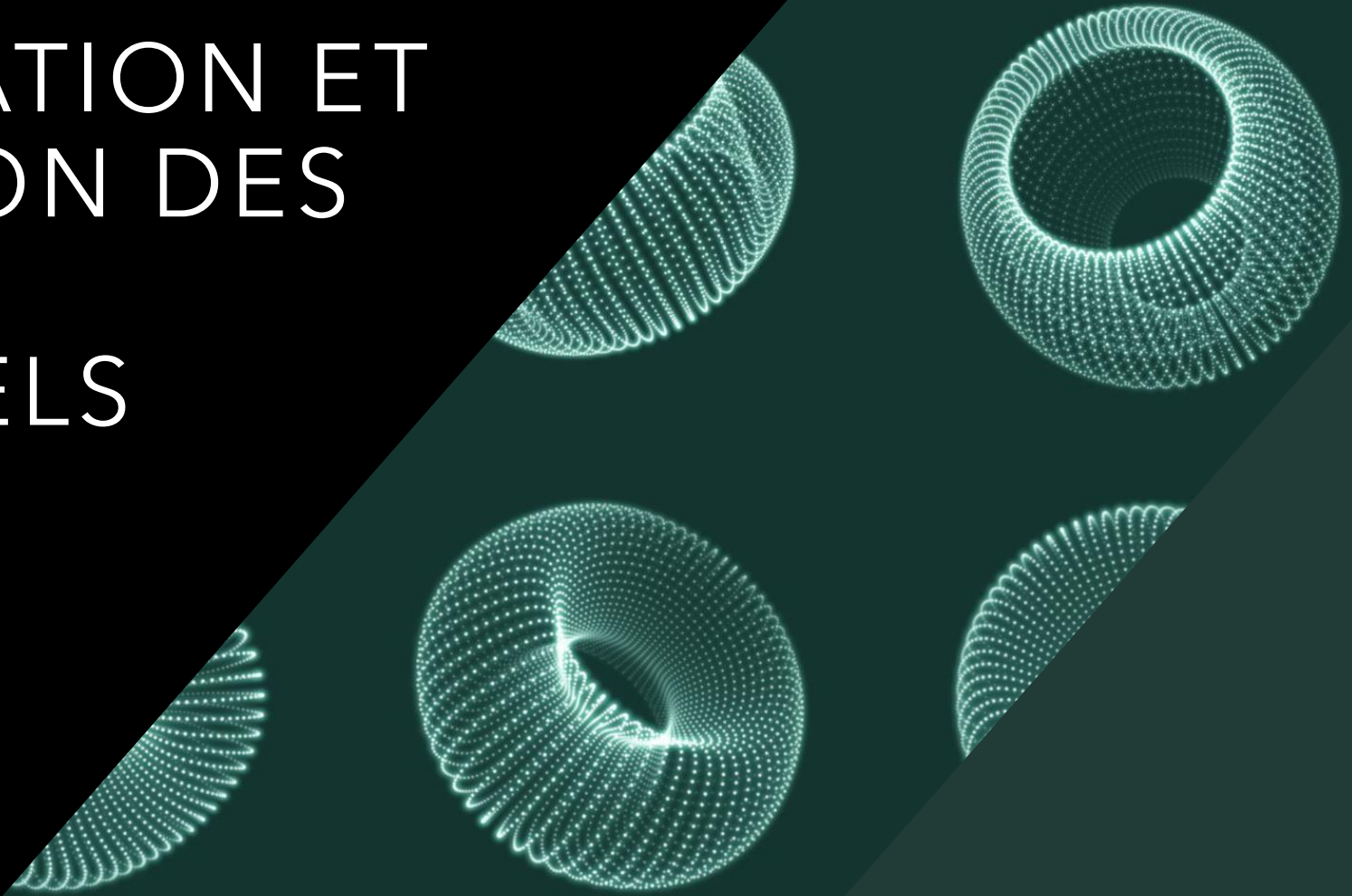


# MODÉLISATION ET SIMULATION DES SYSTÈMES INDUSTRIELS



# Sommaire

1. Objectifs
2. Représentation des données
3. Algorithme
4. Améliorations

# Sommaire

1. Objectifs
2. Représentation des données
3. Algorithme
4. Améliorations

# Problématique

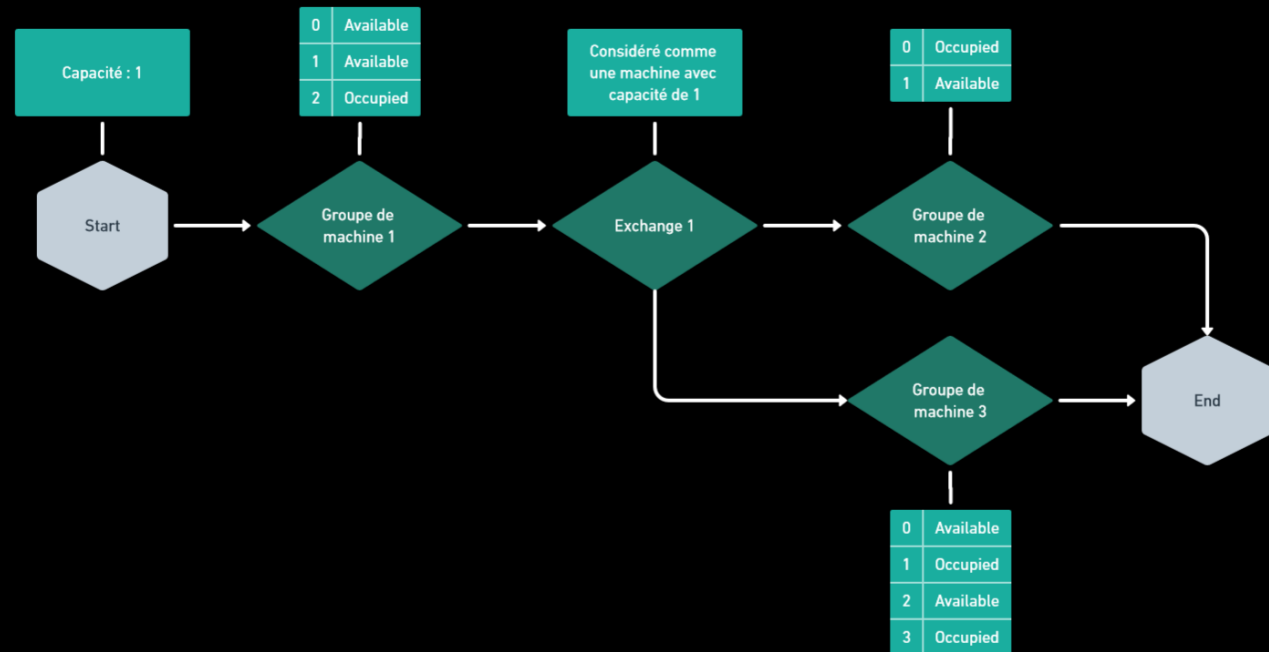
- Optimiser l'ordre de production pour réduire le temps de production
- Machines pouvant être dédoublées, plusieurs produits...

# Sommaire

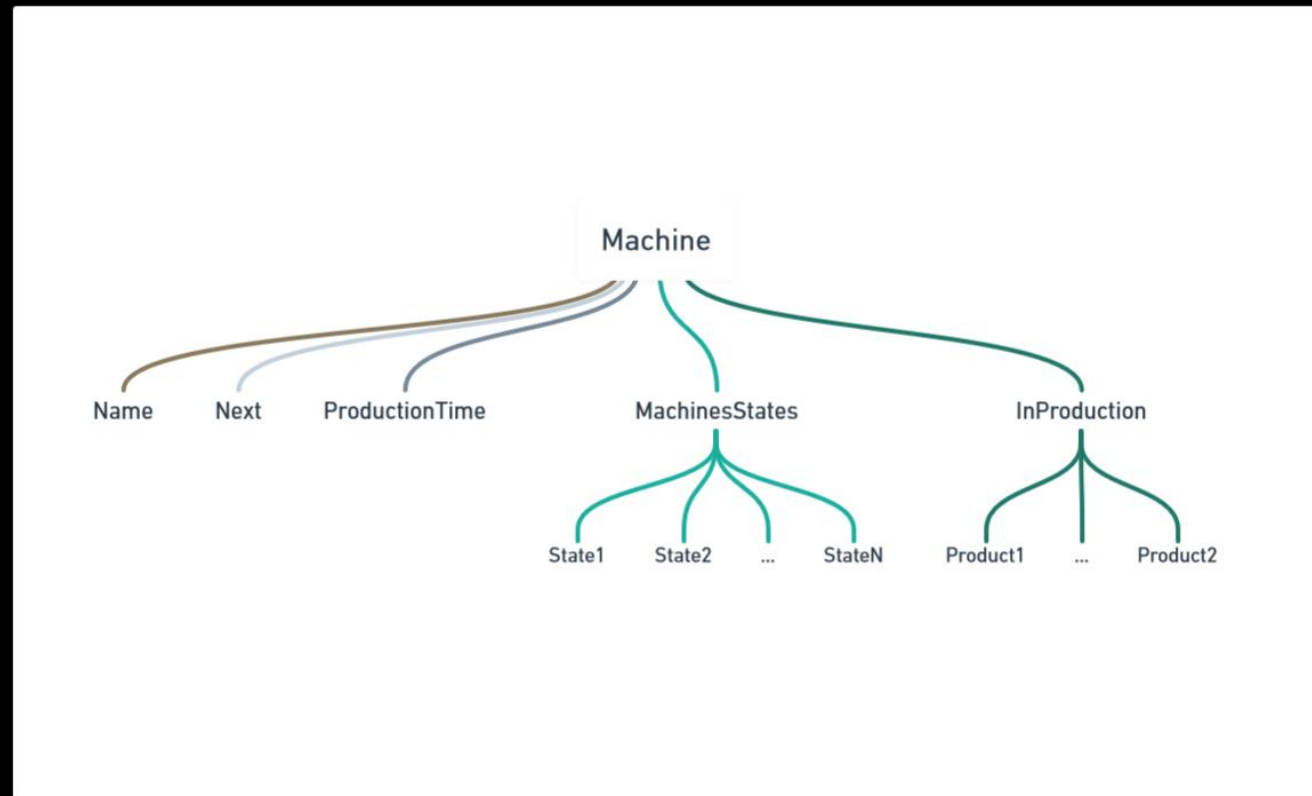
1. Objectifs
2. Représentation des données
3. Algorithme
4. Améliorations

# Représentation des données - Topology

- Représentation sous forme de graphe
- Groupe de machine représenté par une seule machine
- Chaque machine a une liste de "disponibilités"



# Représentation des données - Machines



# Sommaire

1. Objectifs
2. Représentation des données
3. Algorithme
4. Améliorations



# Algorithme - Parsing

- Pour le parsing des paramètres de lancement, utilisation du NuGet CommandLine
- Utilisation des bibliothèques Xml de C#, pour pouvoir utiliser LINQ

# Algorithme

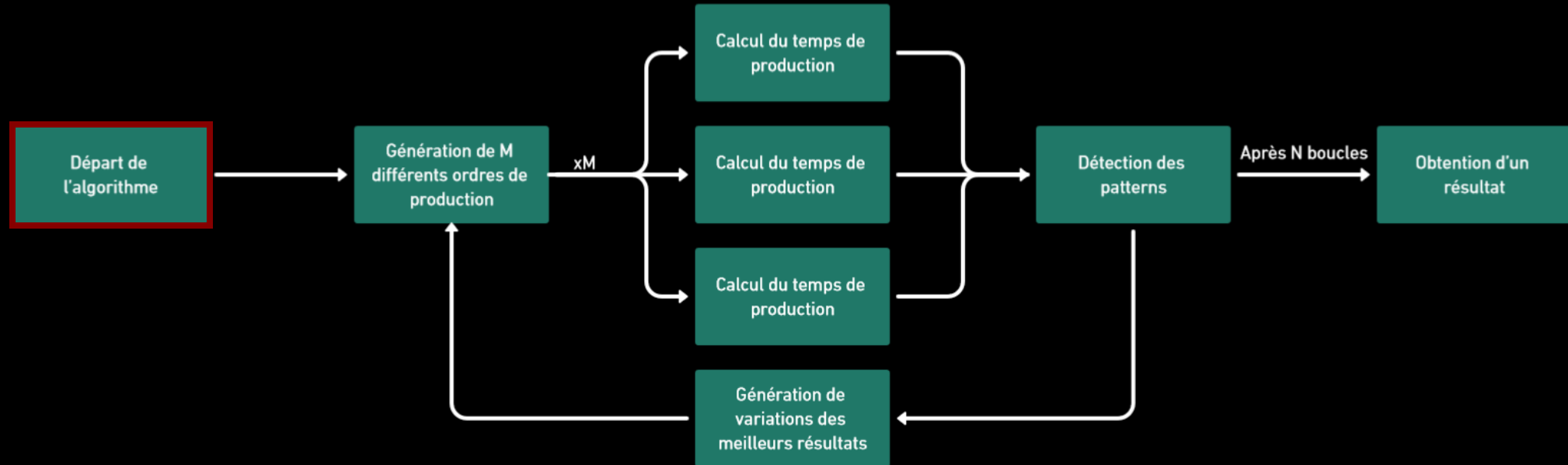
- Problème NP-Hard
- Très difficile d'obtenir toujours **LA** solution optimale
- Utilisation d'heuristiques

# Algorithme

## Heuristique :

En informatique, l'heuristique est un **algorithme qui permet de rapidement dénouer des difficultés complexes d'optimisation, sans concevoir une modélisation formelle**. Cela ne se résout pas forcément par une solution optimale.

# Algorithme



# Algorithme

## Génération des variations :

On crée des perturbations en échangeant des positions. On essaye d'échanger un produit avec son voisin, puis le voisin de son voisin...

Etape 1 : Voisin du produit

a1	a1	a1
a2	b1	a2
b1	a2	b1
b2	b2	c1
c1	c1	b2
c2	c2	c2

# Algorithme

## Génération des variations :

On crée des perturbations en échangeant des positions. On essaye d'échanger un produit avec son voisin, puis le voisin de son voisin...

Etape 2 : Voisin du voisin  
du produit

a1	b1	a1
a2	a2	a2
b1	a1	c1
b2	b2	b2
c1	c1	b1
c2	c2	c2

# Sommaire

1. Objectifs
2. Représentation des données
3. Algorithme
4. Améliorations

# Améliorations

Multi-threading :

On veut pouvoir tester plusieurs permutations en même temps.

Gain de temps, mais attention aux performances.



# Améliorations

D'autres heuristiques :

Utiliser d'autres heuristiques, comme par exemple le recuit simulé ou recherche tabou.

# Avec l'exemple

- On trouve un temps optimal de 603s
- Algorithme terminé en XXXs (PC de XXX)

# Merci