Génération automatique de code de simulation via l'information de localisation des produits

Andrés VÉJAR Patrick CHARPENTIER

{andres.vejar,patrick.charpentier}@cran.uhp-nancy.fr

Centre de Recherche en Automatique de Nancy (CRAN)

CNRS UMR 7039, Nancy-Université

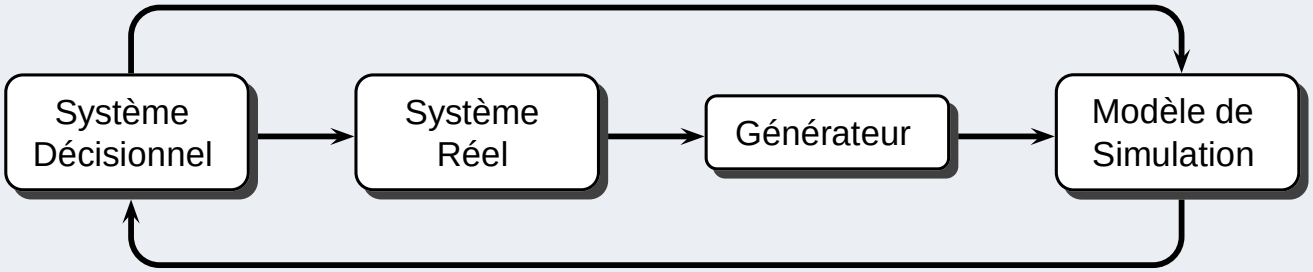
8ème Congrès International de Génie Industriel (CIGI'09).
10-11 et 12 juin 2009. Bagnères de Bigorre, France.

Problématique

La simulation de flux

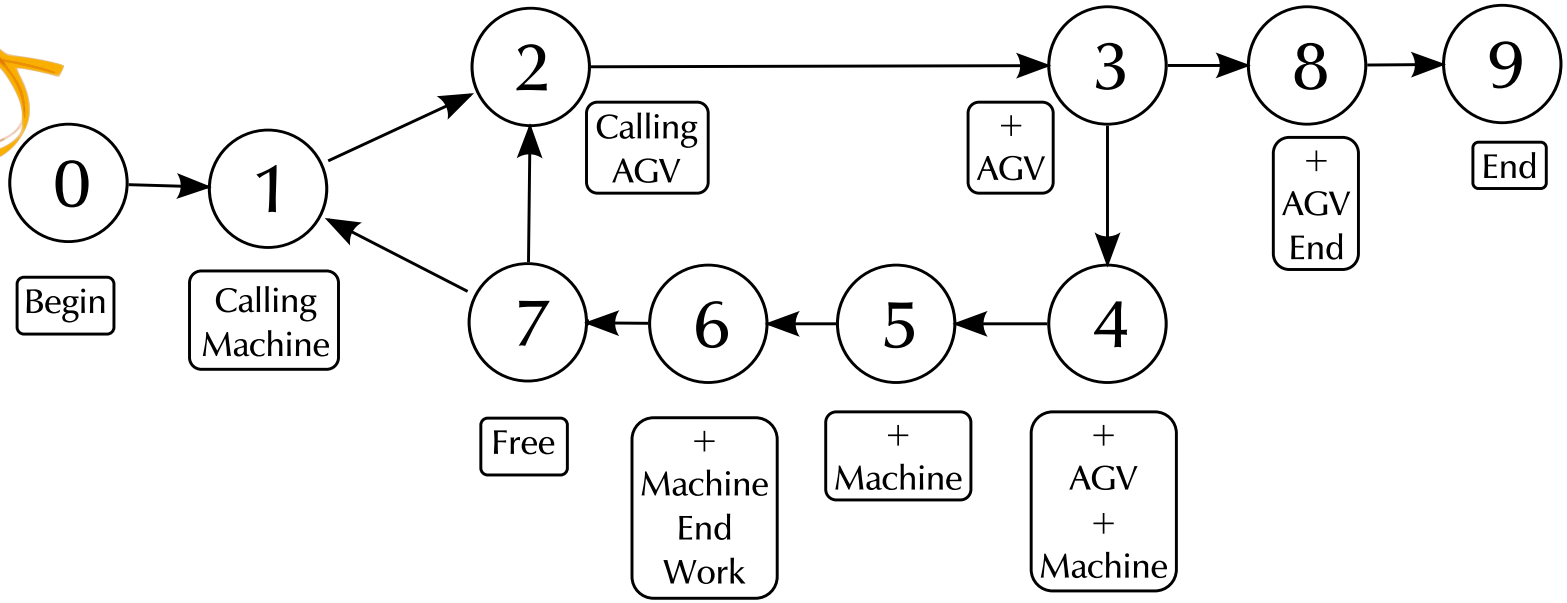
- incontournable pour l'évaluation de la dynamique des systèmes manufacturiers, utilisée dans les phases de :
  - *Design* (pour Concevoir),
  - *Re-engineering* (pour Améliorer), et
  - *Control*, (en Exploitation).
- les phases de modélisation, puis de maintenance restent des opérations délicates et chronophages
- les résultats sont dépendants de l'habileté des experts humains

Automatiser ces phases (en *re-engineering* ou en *explotation*)

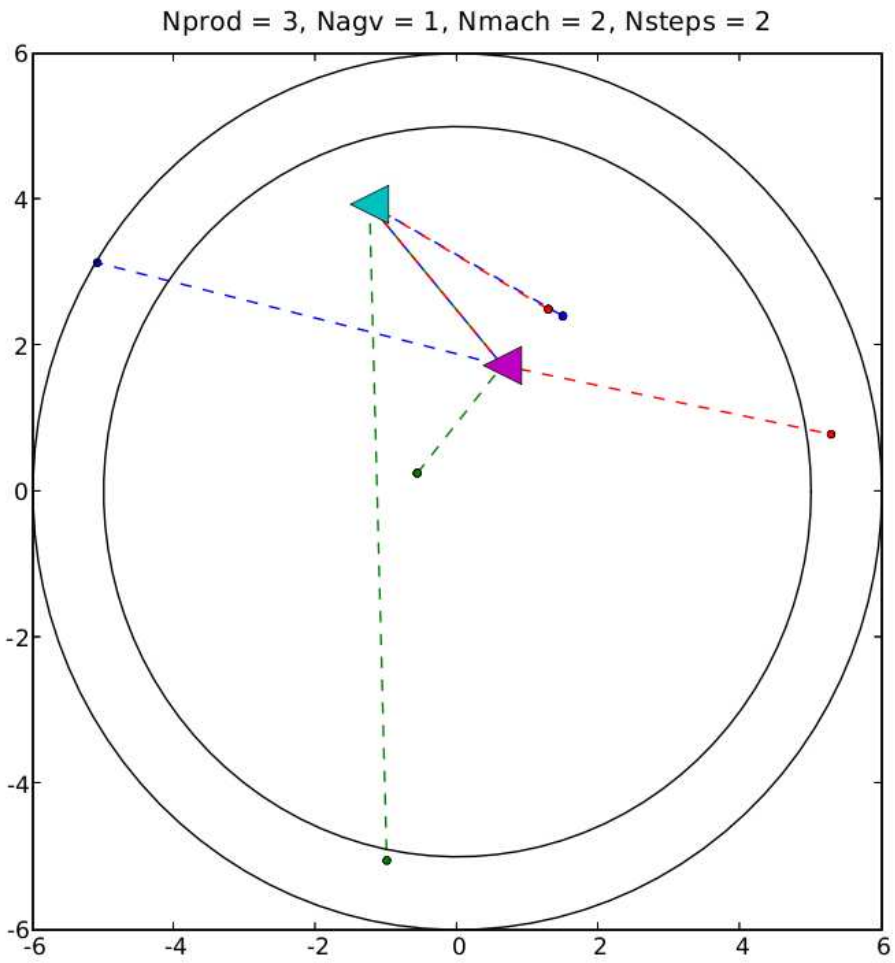


Vie du produit dans l'atelier

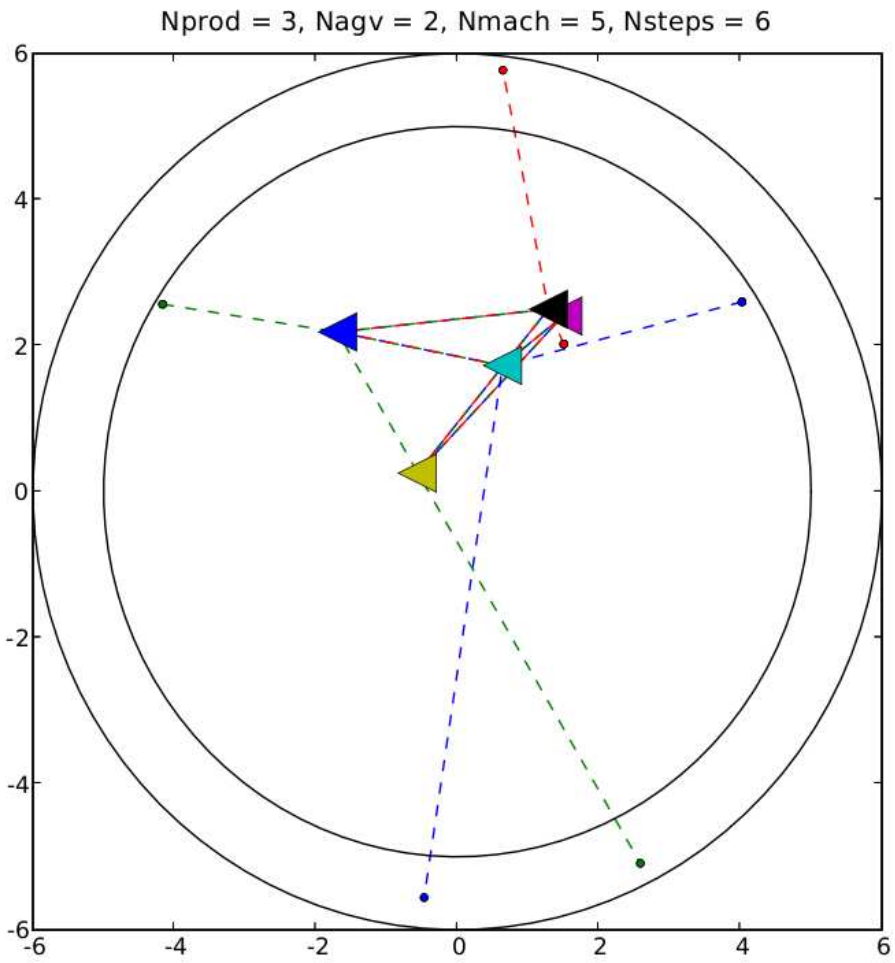
Diagramme d'états



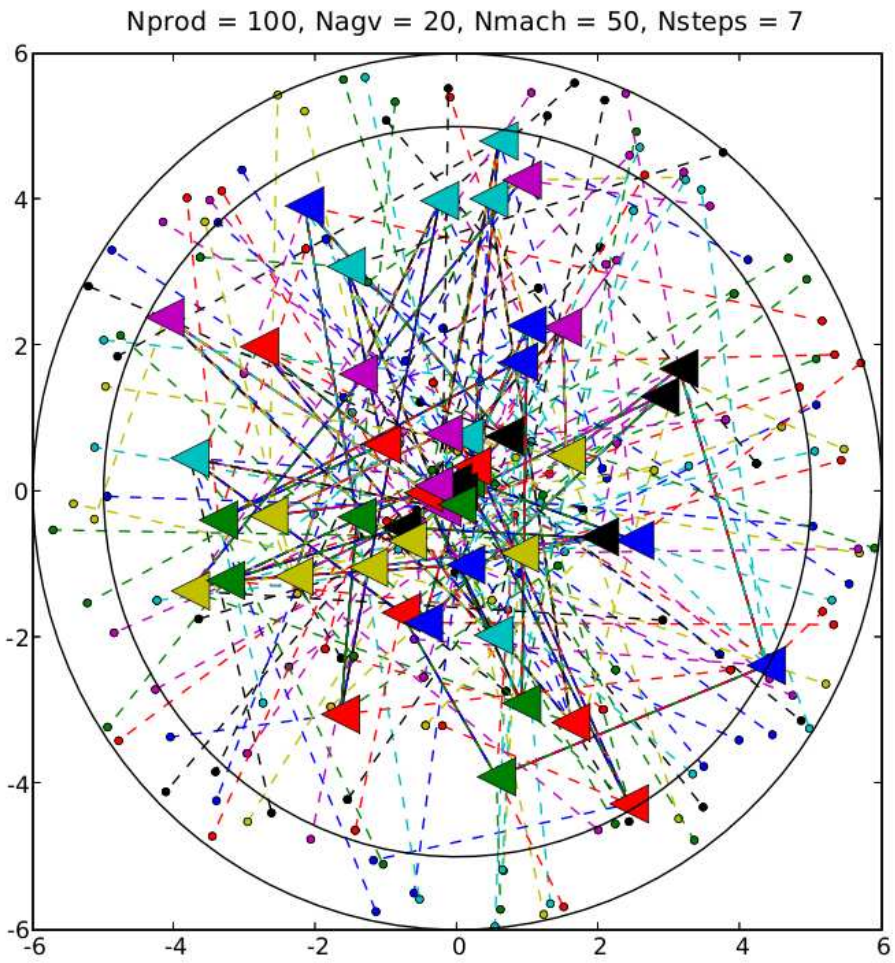
Simulation : Exemple 1



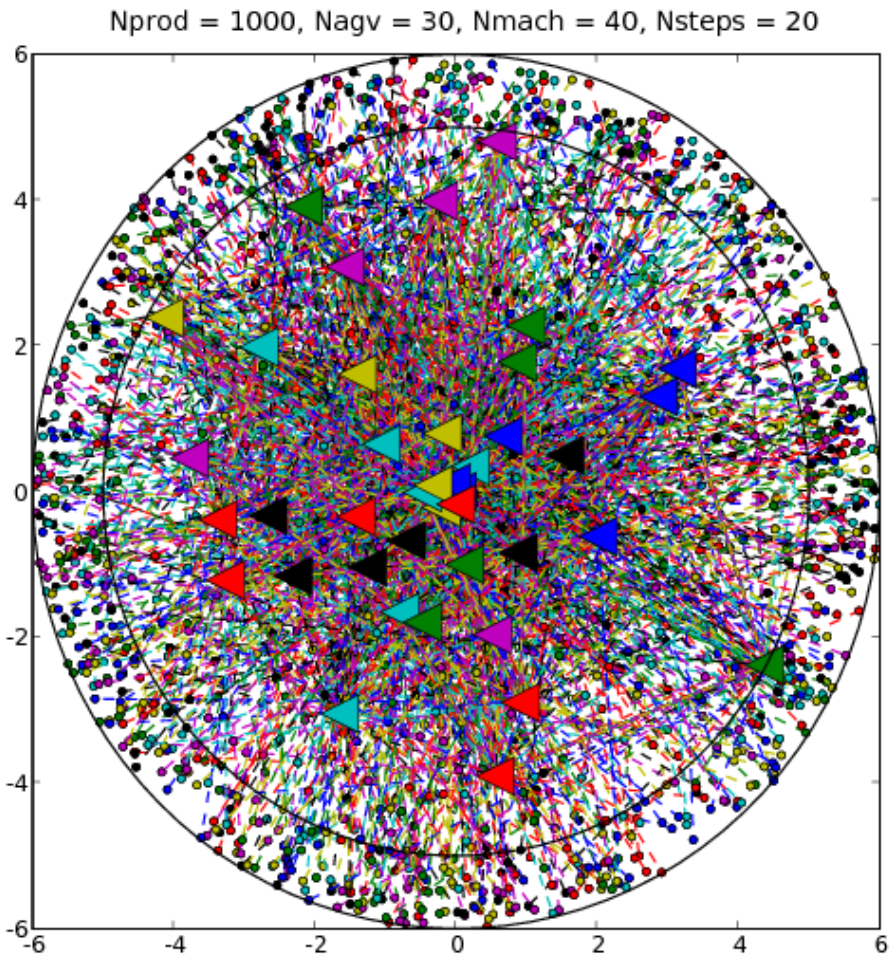
Simulation : Exemple 2



Simulation : Exemple 3

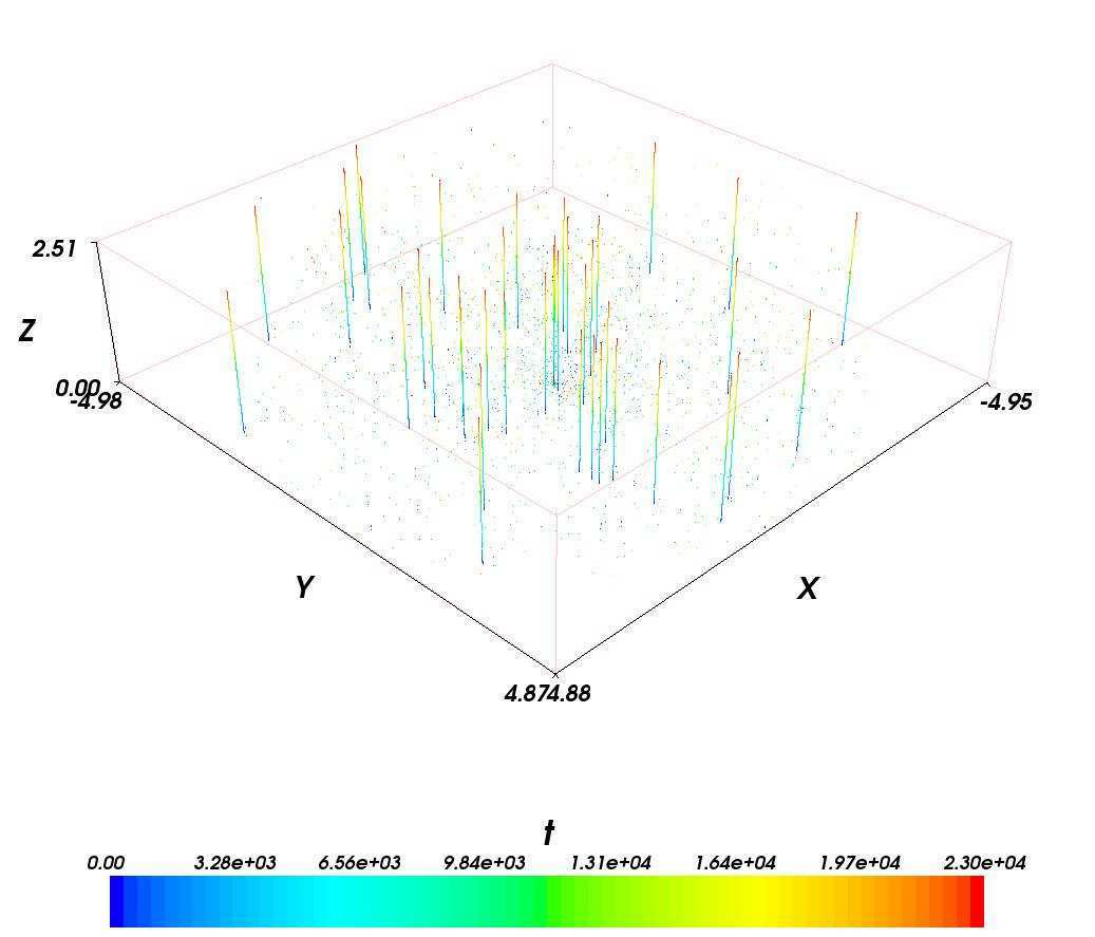


Simulation : Exemple 4



Atelier

40 machines, $z - axe$ est l'axe de temps



Indicateur Spatio-Temporal

Densité ρ

$(\epsilon, \tau) = (\text{voisinage espace}, \text{voisinage temps})$

$$\rho_{\epsilon,\tau}(\mathbf{x}_s, t_s) = \frac{1}{\pi\tau\epsilon^2} \sum_{\{(\mathbf{x},t)\in\mathcal{S}|t\in\mathcal{T}\}} \theta(\epsilon - \|\mathbf{x} - \mathbf{x}_s\|)\theta(\tau - |t - t_s|)$$

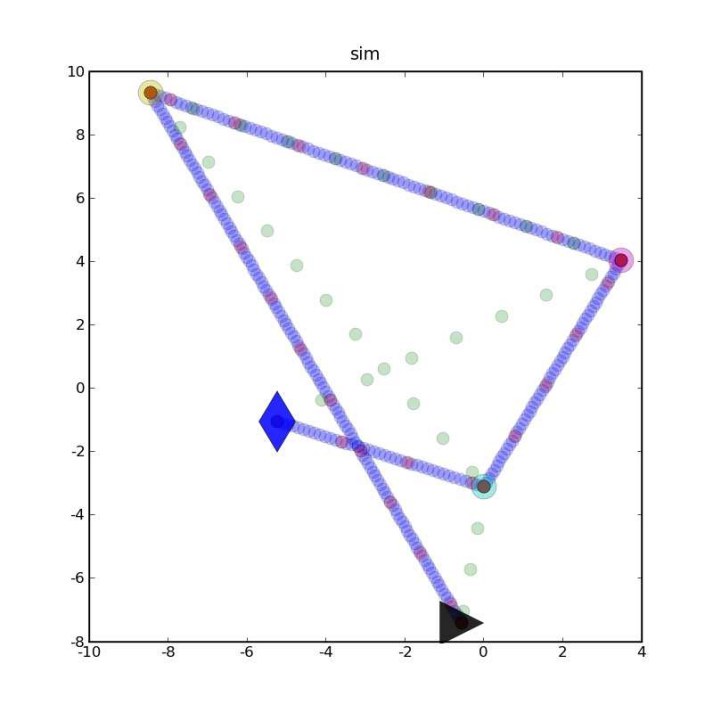
Combien de points existent dans une boule de radius ϵ (centrée en $\mathbf{x}_s$) et dans une fenetre temporel de longeur 2τ (avec centre en $t = t_s$)

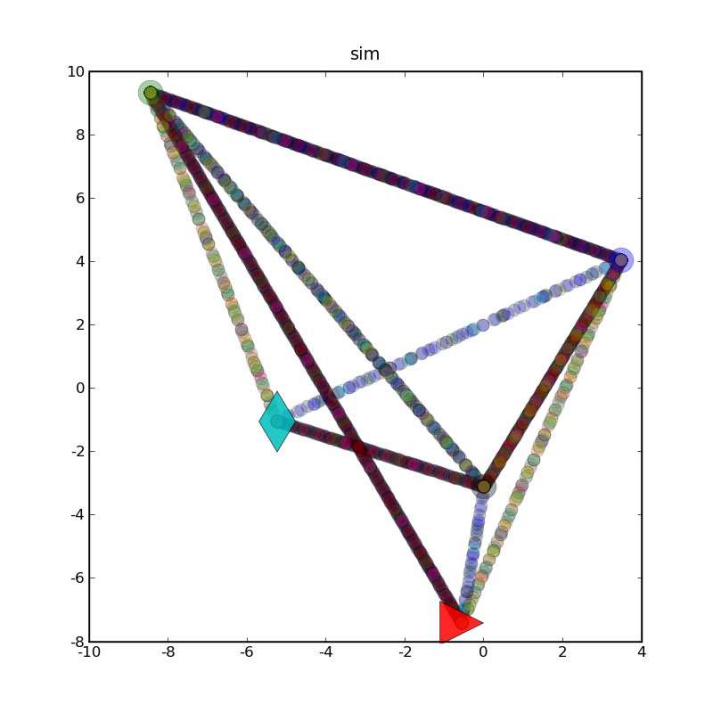
$$\mathbf{x}_s \in k\mathbb{Z}^2 \qquad t_s \in \tau_d\mathbb{Z} \qquad \mathcal{T} \in t_s + \tau/2[-1, 1]$$

$$\mathcal{S} = \{(\mathbf{x}_k, t_k)\}_{k=1}^N \qquad \theta(y) = \begin{cases} 1 & \text{if } y \geq 0 \\ 0 & \text{if } y < 0 \end{cases}$$

Simulation
3 types, 3 machines

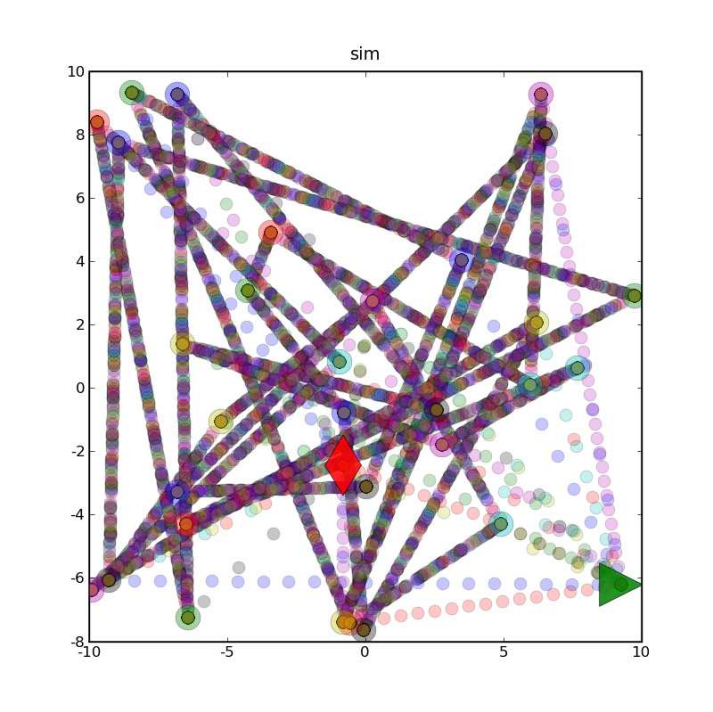
3 produits - 20 produits

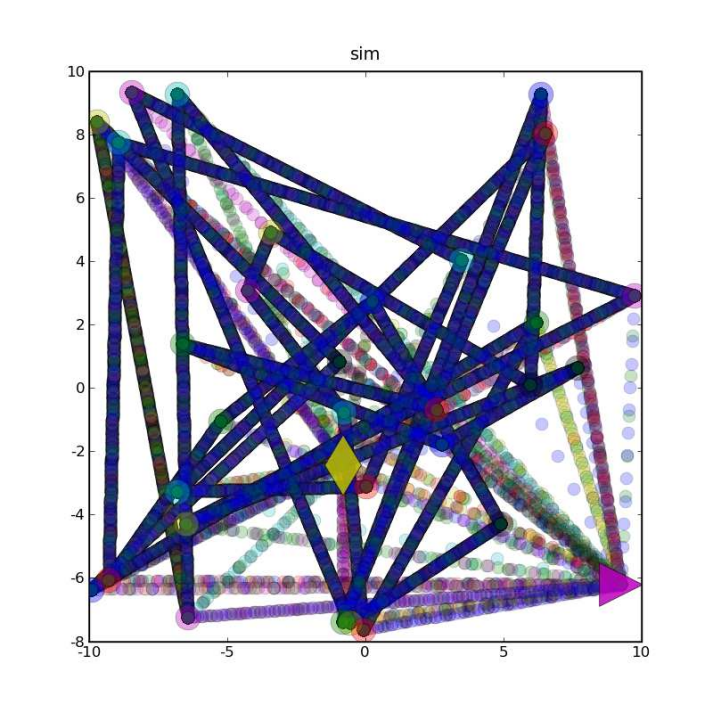




Simulation
30 types, 30 machines

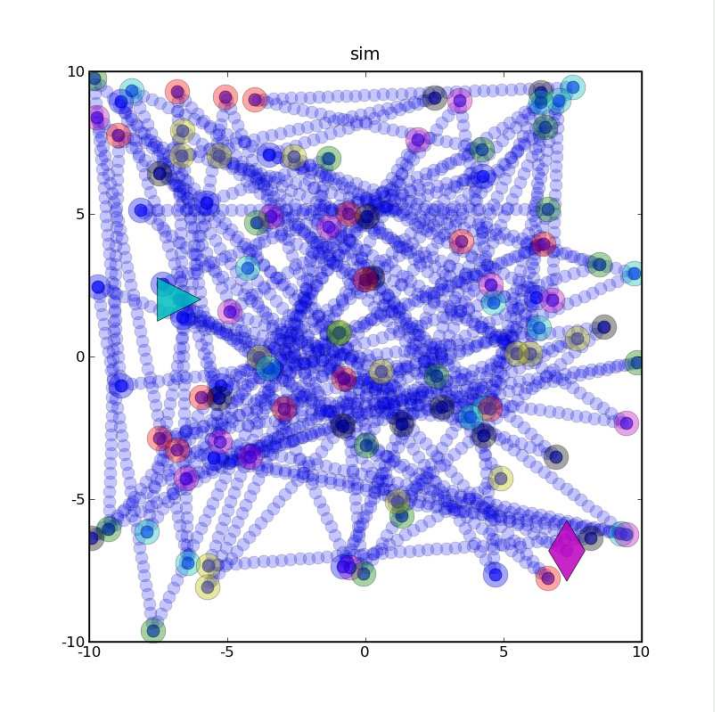
20 produits - 100 produits

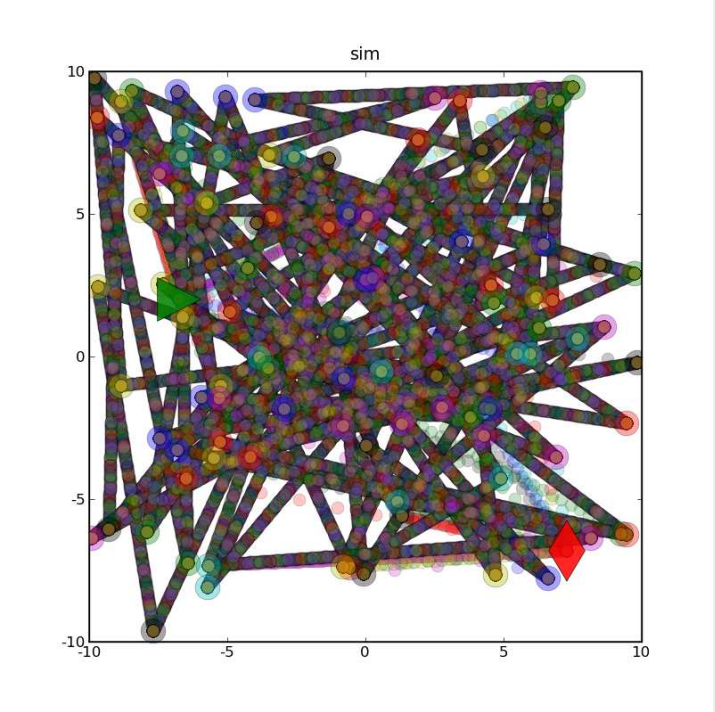




Simulation
100 types, 100 machines

1 produits - 100 produits

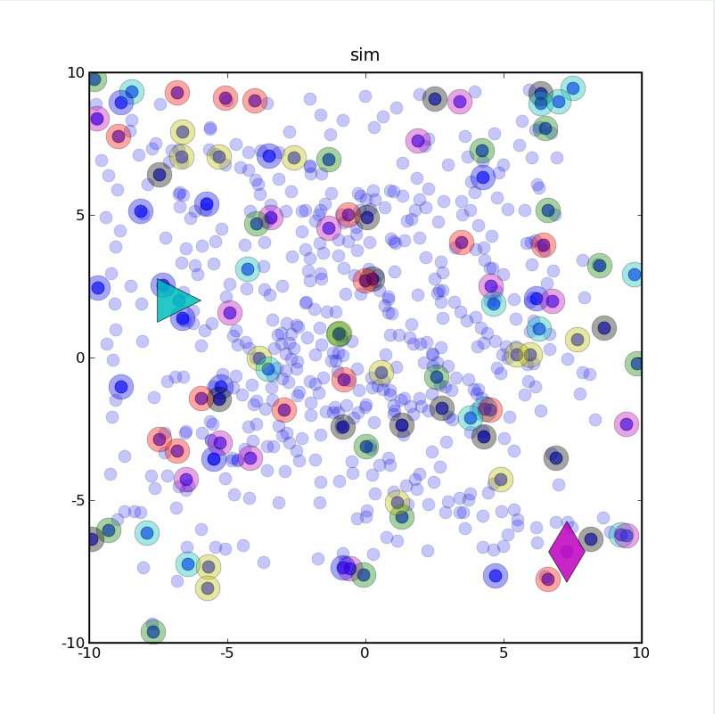


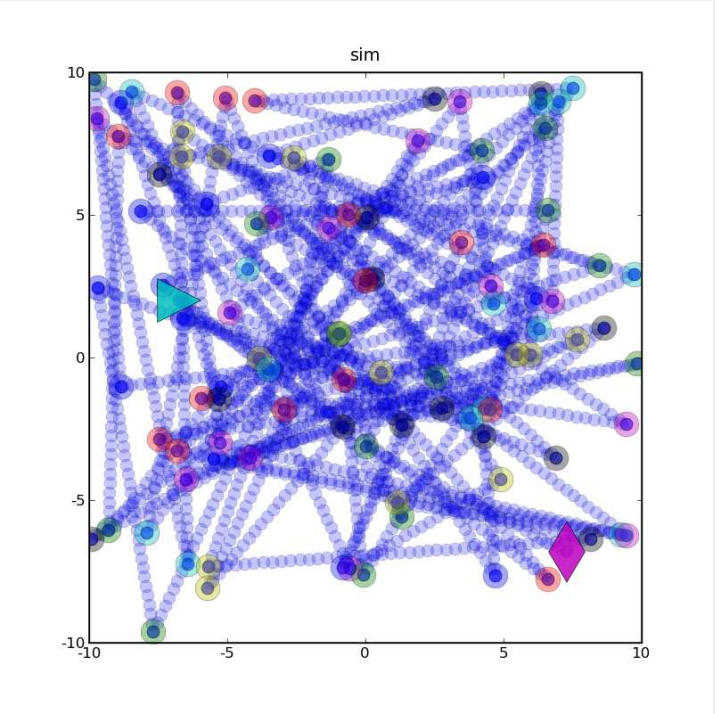


Simulation

100 types, 100 machines

1 produit, loc. basse fréquence - loc. haute fréquence





Etapes d'une étude de simulation

Développer un modèle de simulation

- Identifier le problème
- Formuler le problème
- Collecter et processer les données
- Formuler et développer le m.
- Vérifier et valider le modèle
- Documenter le modèle

Scenarii

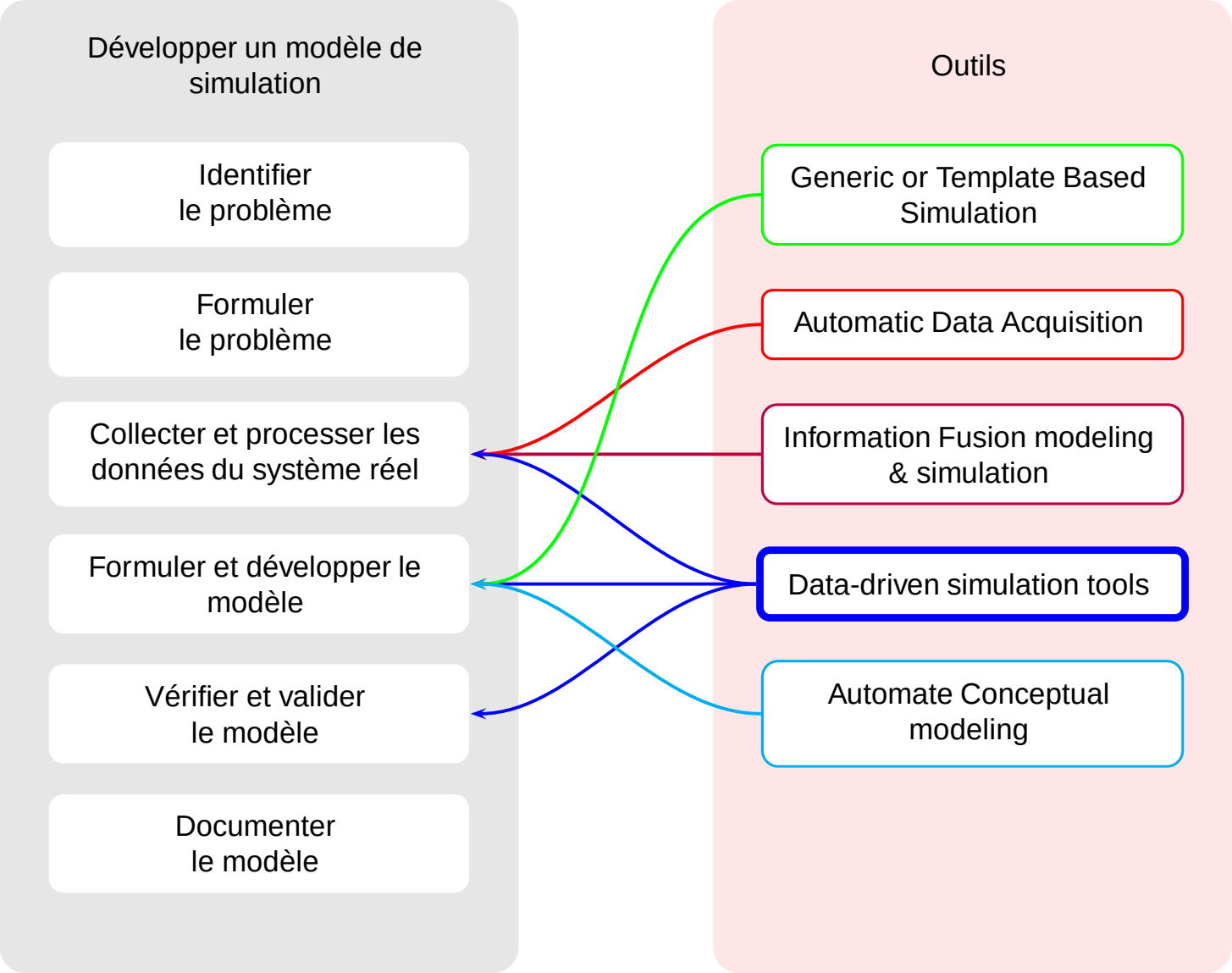
- Choisir les scenarii
- Établir le plan pour les exécutions
- Réaliser les exécutions de la simulation

Réaliser une analyse de la simulation

- Interpréter et présenter les résultats
- Suggérer les alternatives
- Implémentation
- Rapport

Processus de modélisation . . .

Outils supports



Data-Driven Simulation Tools. . .

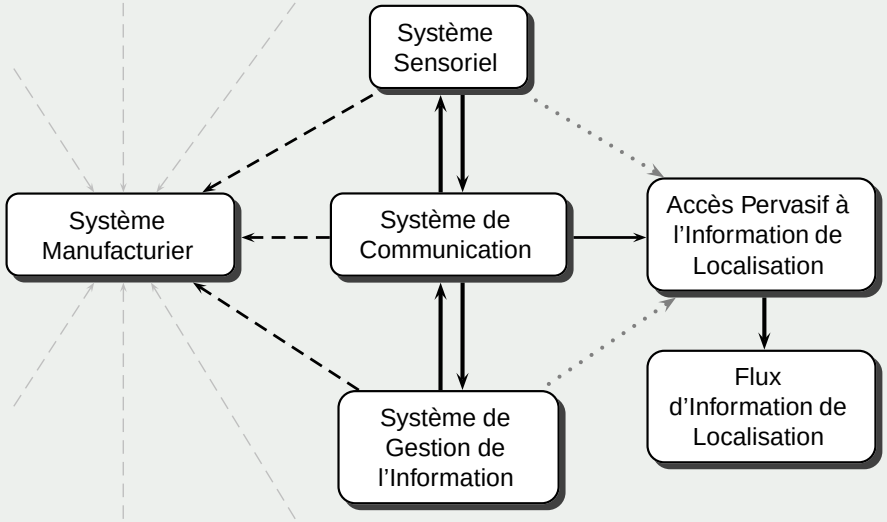
Outils supports

- *Data-driven simulation of networks with manufacturing blocking*.
  Clark, G. M. and Cash, C. R. (1993)
- *A data-driven modelling approach to product level decision support*.
  D'Souza, G. and Allaway, A. (1997)
- *Dynamic Data Driven Applications Systems : A New Paradigm for Application Simulations and Measurements*.
  Darema, F. (2004)
- *Towards Intelligent Data-Driven Simulation for Policy Decision Support in the Social Sciences*.
  Kennedy, C. and Theodoropoulos, G. (2005)
- *Data-driven Simulation of the Extended Enterprise*.
  Cao, B. et al. (2004)
- *Simbiotic Adaptive Multisimulation : An Autonomic Simulation Framework for Real-Time Decision Support*.
  B. Mitchell and L. Yilmaz. (2008)

Proposition
et sujet de recherche

Générateur de code de simulation à partir de l'information de localisation du produit

Montrer les apports, sur le contrôle et les performances d'un système manufacturier, d'un flux d'information de localisation de produits durant leur élaboration.

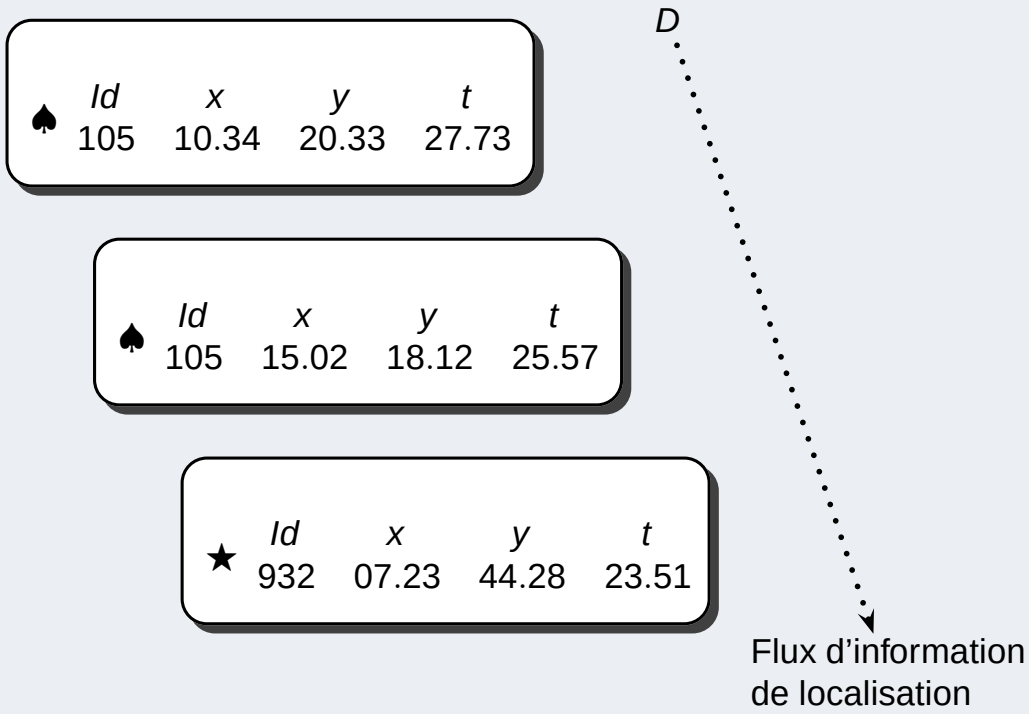


Flux d'information de localisation

- La technologie pour localiser des objets est devenue disponible et abordable
- Les données spatio-temporelles liées aux produits obéissent à des règles utiles et nécessaires lors de la construction de modèles
  (EXEMPLE : un objet ne peut être qu'à une et une seule position à un même instant, etc.)

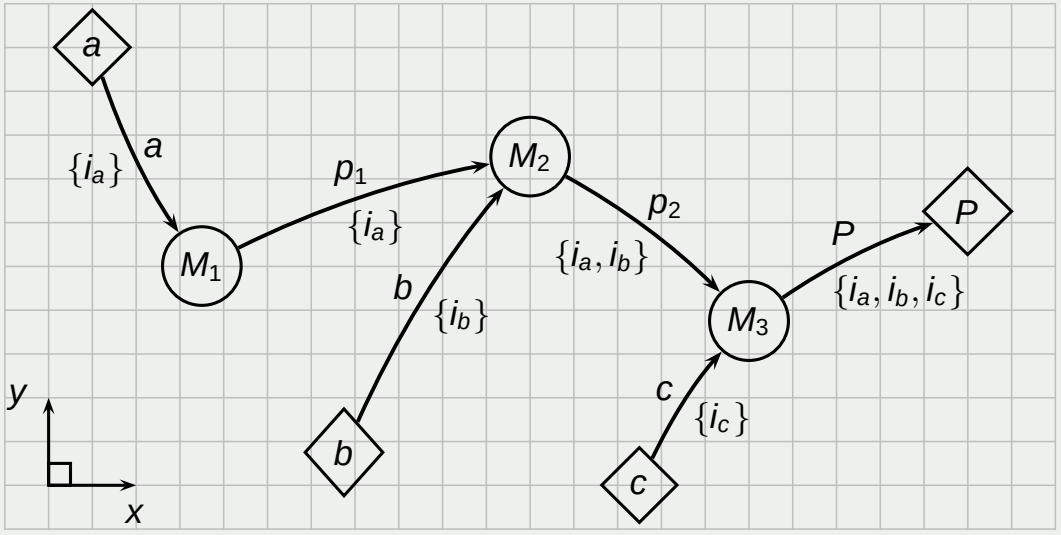
- Id Objet
- Position (x, y)
- Temps

$$d_k = (Id, x, y, t)_k$$
$$D = \{d_0, d_1, d_2, \ldots\}$$



Nomenclature du produit

Atelier, trois machines et un produit

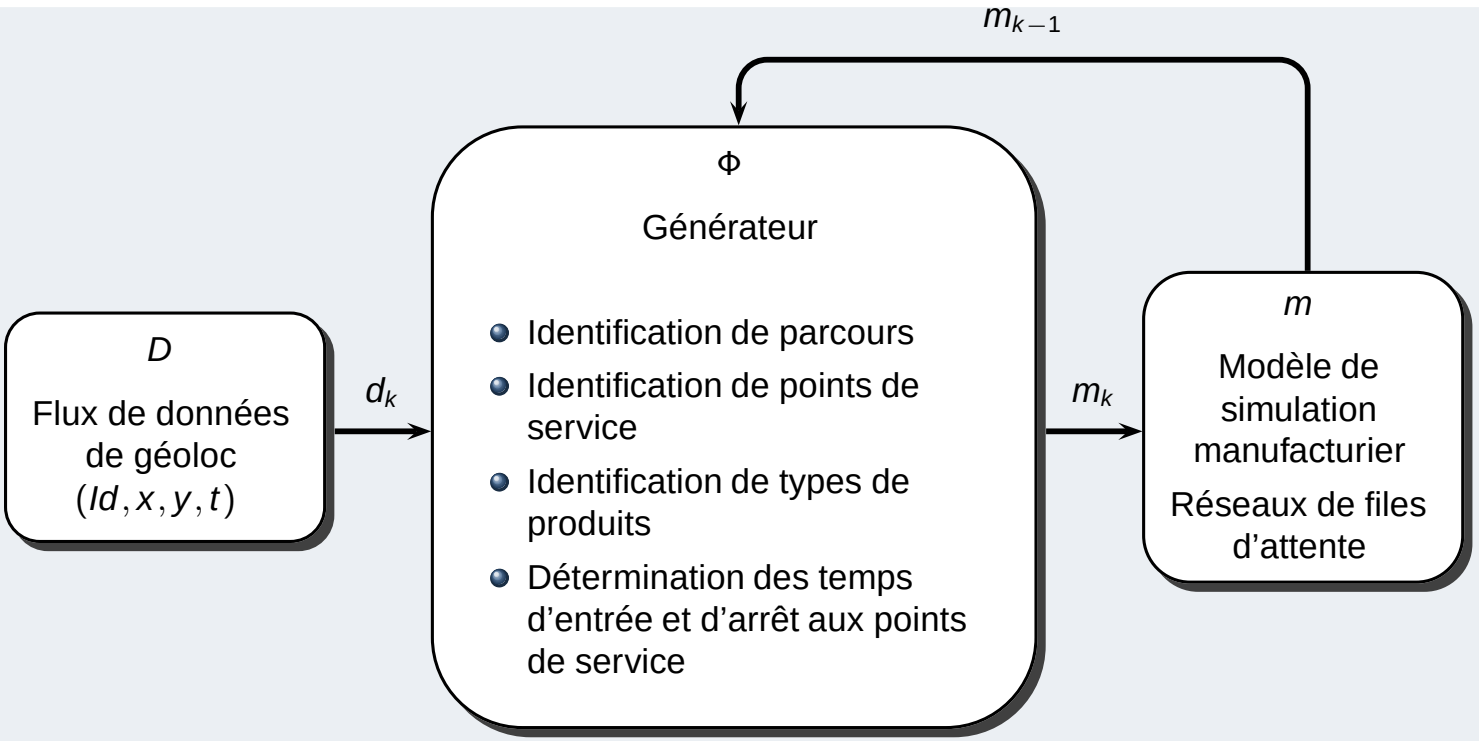


$$
\begin{array}{ccccc}
a & \mapsto & M_1(a) & = & p_1 \\
(p_1, b) & \mapsto & M_2(p_1, b) & = & p_2 \\
(p_2, c) & \mapsto & M_3(p_2, c) & = & P
\end{array}
$$

$$P = M_3\left(M_2\left(M_1(a), b\right), c\right)$$

Générateur

Les seules données utilisées sont les données de localisation des produits durant leur passage en production



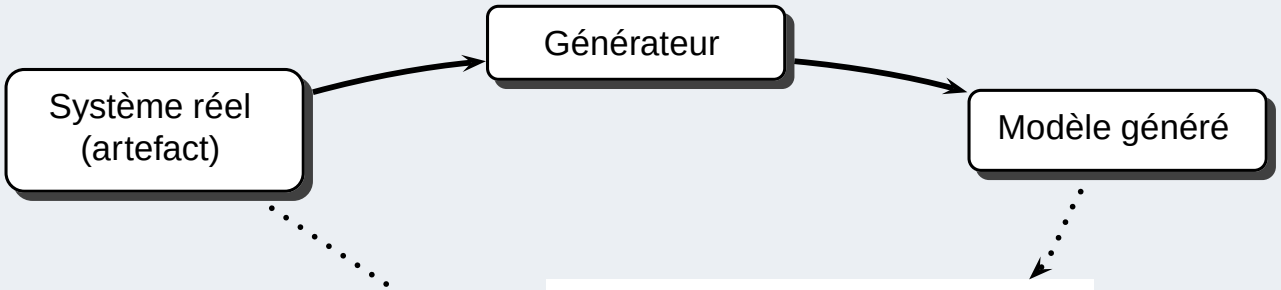
$$\begin{aligned} m_0 &= \emptyset \\ m_k &= \Phi(d_k, m_{k-1}) \end{aligned}$$

Algorithme

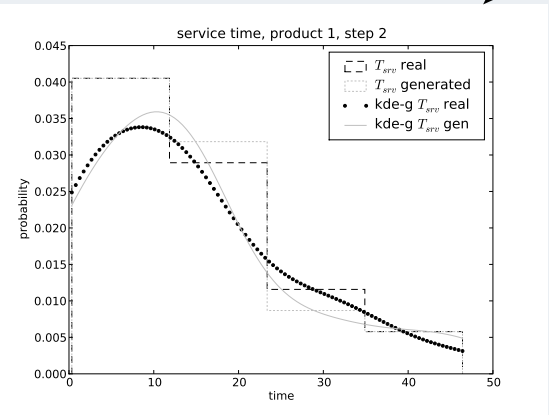
- **Identifier les serveurs** : Points de vitesse nulle (avec récurrence).
  `S=Serveurs(D)`
- **Identifier les trajectoires** : Arcs entre points.
  `A=Trajectoires(D,S)`
- **Identifier produits** : Graphe, $p_j = (A_j, S_j)$.
  `P=Produits(D,S,A)`
- **Générer layout** : Graphe, union de l'ensemble des graphes produits, $L = \cup_j p_j$
  `L=Layout(P)`
- **Distribution des entrées** : pour chaque produit sur chaque serveur.
  `E=Entrees(L,D)`
- **Distribution des temps d'inter-arrivées** : pour chaque produit sur chaque serveur.
  `I=TempsIA(L,D)`
- **Distribution des temps de service** : pour chaque produit sur chaque serveur.
  `T=TempsS(L,D)`
- **Générer modèle**.
  `M=Gen(L,E,I,T)`

for d **do**
  if $\exists\, p \in P : p.i = d.i$ **then**
    if $p.m.s = 1$ **then**
      if $p.m.r = d.r$ **then**
        $p.m.a \leftarrow p.m.a + d.t - p.m.t$
        $p.m.t \leftarrow d.t$
      else
        $m \leftarrow m' \in M : m'.r = p.m.r$
        if $p.m.t - p.m.a < m.l$ **then**
          $p.m.T \leftarrow p.m.t - m.l$
        else
          $p.m.T \leftarrow p.m.a$
        end if
        $m.l \leftarrow p.m.t$
        $p.M \leftarrow p.M \cup \{p.m\}$
        $p.m \leftarrow \check{m}(d.r, d.t)$
      end if
    else
      if $p.m.r = d.r$ **then**
        $p.m.s \leftarrow 1$
        if $\nexists\, m \in M : m.r = d.r$ **then**
          $M \leftarrow M \cup \{p.m\}$
        end if
      else
        $p.m.r \leftarrow d.r$
      end if
      $p.m.t \leftarrow d.t$
    end if
  else
    $m \leftarrow \check{m}(d.r, d.t)$
    $p \leftarrow \check{p}(d.i, m)$
    $P \leftarrow P \cup \{p\}$
  end if
end for

Principe mis en œuvre



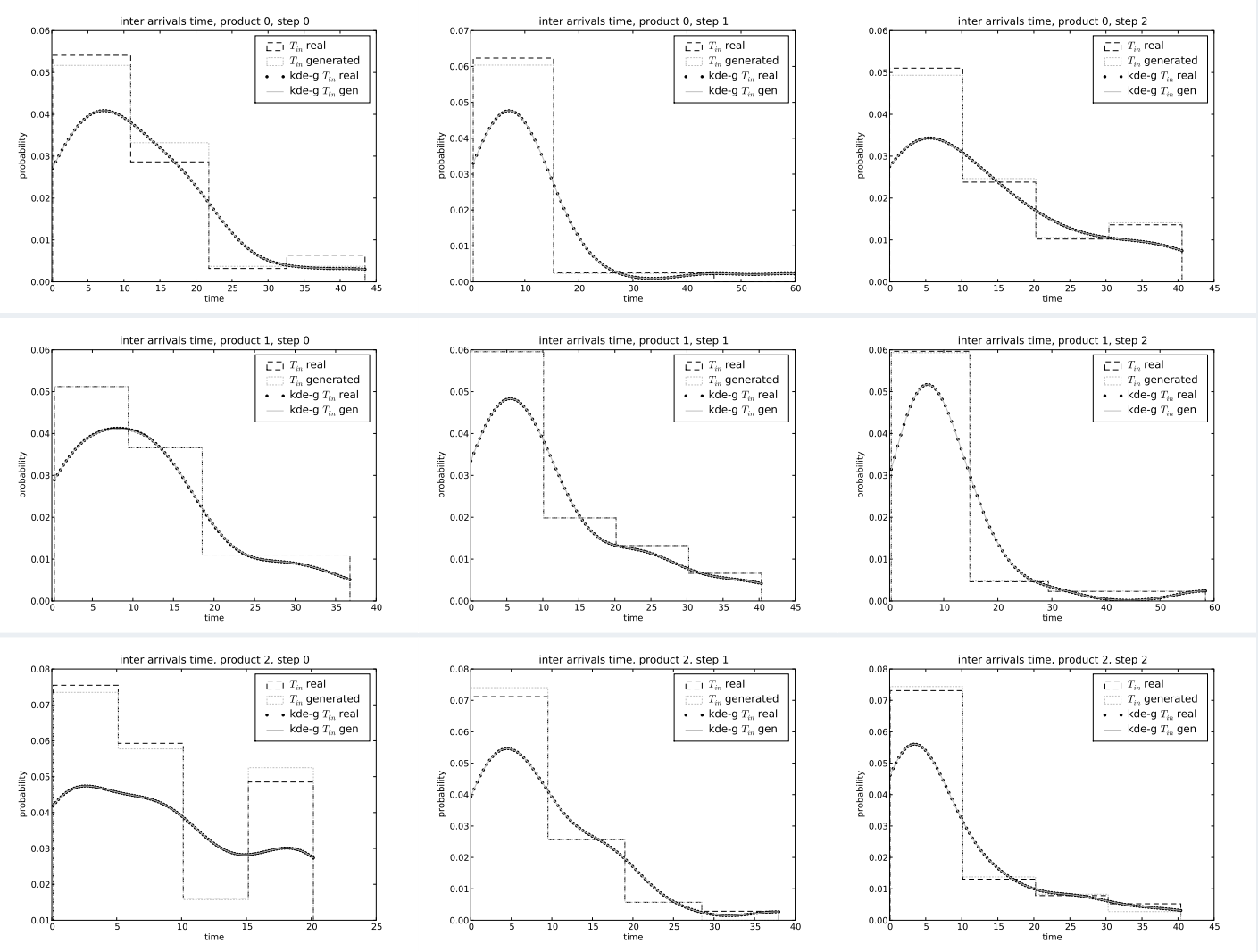
paramètres	valeurs
types de produits	3
nombre de machines	3
nombre d'opérations	3
quantité de produits	100
temps d'inter-arrivées	exp, $\mu = 1/3$
temps de service	exp, $\mu = 13$
nombre d'AGV's	10
vitesse des AGV's	0.44
dimensions de l'atelier	20×20
période d'échantillonnage pour la localisation	0.5



- Module de simulation avec SimPy (Simulation in Python) (Muller et Vignaux, 2003)
- Artefact du système réel pour générer le flux de localisation des produits *D*
- *D* est récupéré par le module générateur de modèle de simulation
- Un module d'analyse des données permet de vérifier les résultats obtenus

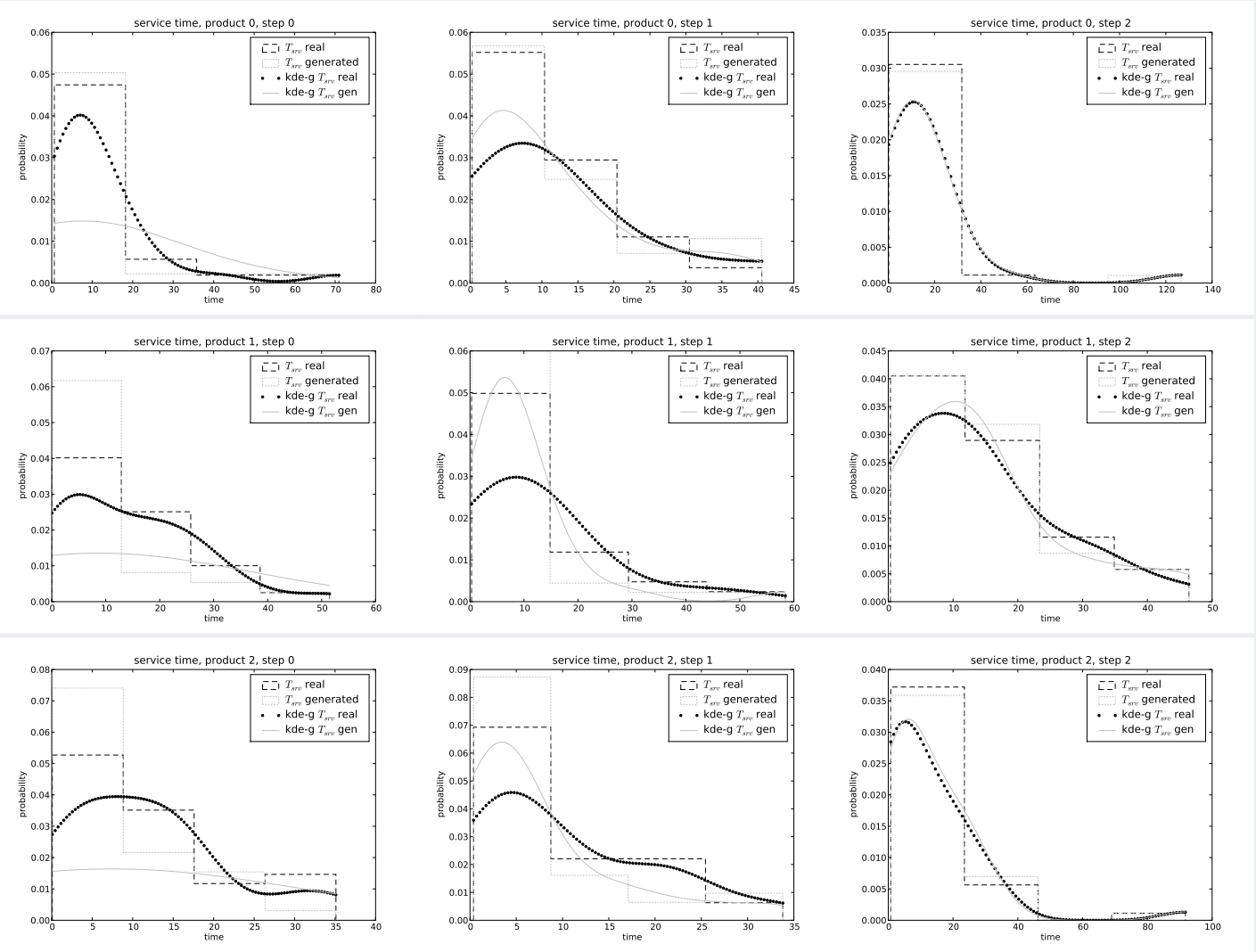
Résultats

Distributions des temps inter-arrivées



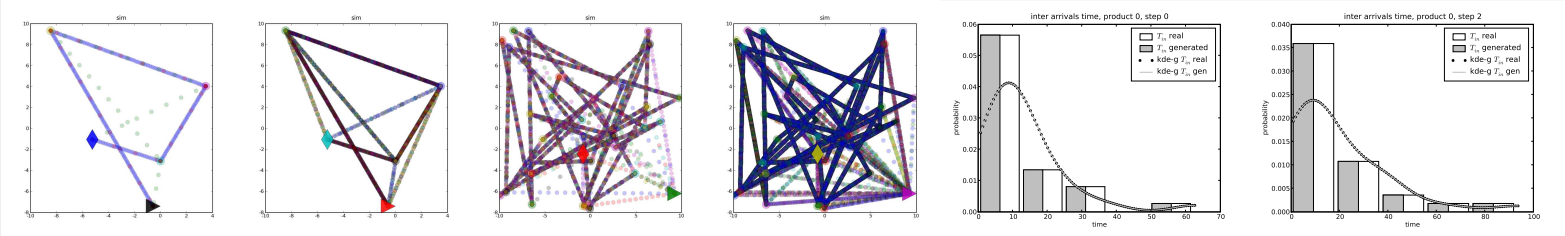
Résultats

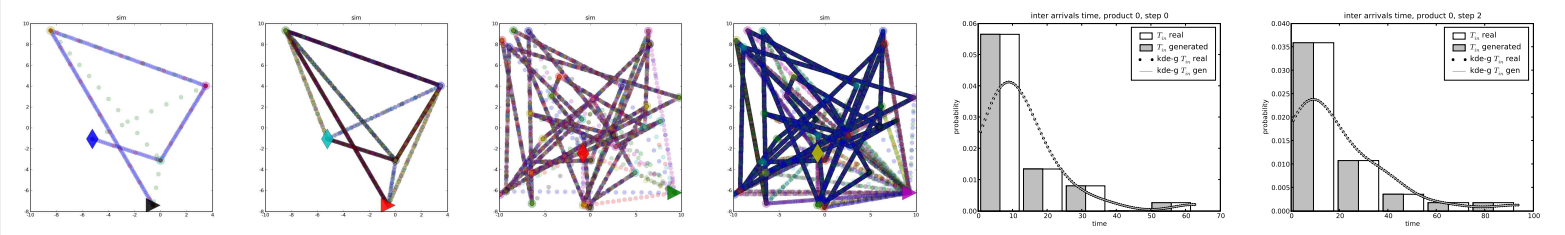
Distributions des temps de service



Conclusion
et Perspectives

- Les résultats obtenus valident la démarche mise en œuvre, ainsi que l'outil développé
- Confirmés par des tests effectués sur des systèmes de taille plus importante (jusqu'à 15 machines, 15 opérations ou phases, 15 types de produits)



Perspectives

- Prendre en compte les imprécisions de mesure de localisation
- Tests et faisabilité technologiques (GPS, accélèromètres, . . .)
- Mise en relation Technologies-Applications
- Modèle de simulation neutre, pour utiliser differents moteurs de simulation
- Validation sur cas réels en vraie grandeur
- Élargir les champs d'application (hors manufacturing)