Soutenance de thèse

Recherche locale pour l'optimisation en variables mixtes : méthodologie et applications industrielles

Antoine Jeanjean

Jury :

- **Philippe Baptiste** - Directeur de recherche, CNRS LIX (Directeur de Thèse)
- **Thierry Benoist** - Ingénieur, Bouygues e-lab (Co-directeur de Thèse)
- **Jacques Carlier** - Professeur, Université Technologique de Compiègne (Rapporteur)
- **Frédéric Gardi** - Ingénieur, Bouygues e-lab (Co-directeur de Thèse)
- **Léo Liberti** - Professeur chargé de cours, Ecole Polytechnique (Examinateur)
- **Alain Quilliot** - Professeur, ISIMA, Université Clermont II (Examinateur)
- **Francis Sourd** - Chef d'équipe RO (HDR), SNCF Innovation & Recherche (Rapporteur)
- **Michel Vasquez** - Enseignant-chercheur (HDR), Ecole des Mines d'Alès (Examinateur)

Lundi 10 octobre – Ecole Polytechnique

Approche par Recherche Locale

- Cette thèse propose une approche directe et pure, par recherche locale, pour les problèmes d'optimisation en variable mixtes

- La méthodologie présentée repose sur :

  - une large variété de transformations travaillant simultanément sur le combinatoire et le continu

  - une évaluation incrémentale exploitant les invariants

  - des algorithmes d’évaluation approchés pour la dimension continue

Plan de l'exposé

1. Une approche par recherche locale pure, directe et randomisée
2. Application sur le problème de planification des mouvements de terre
3. Application sur le problème de tournées avec gestion de stock
4. Conclusion et perspectives

1. **Une approche par recherche locale pure, directe et randomisée**
2. Application sur le problème de planification des mouvements de terre
3. Application sur le problème de tournées avec gestion de stock
4. Conclusion et perspectives

Contexte

Résolution de **problèmes mixtes**

issus de thématiques **industrielles**

à des problèmes de **grande taille**

avec des **temps de résolution court**

pour des solutions **meilleures** que les solutions **existantes**

Approches usuelles

DECOMPOSITIONS « METIER »

- issues méthodes opérationnelles
- exploitent propriétés intrinsèques
- sous-problèmes résolus parfois indépendamment et séquentiellement

Limites

- admissibilité / qualité des solutions
- lien entre les variables combinatoires et continues
- sous-problèmes parfois orthogonaux

DECOMPOSITIONS MATHEMATIQUES

- programmation linéaire généralisée
- algorithmes génération colonnes / lignes
  - décomposition de Dantzig-Wolfe
  - partitionnement de Benders

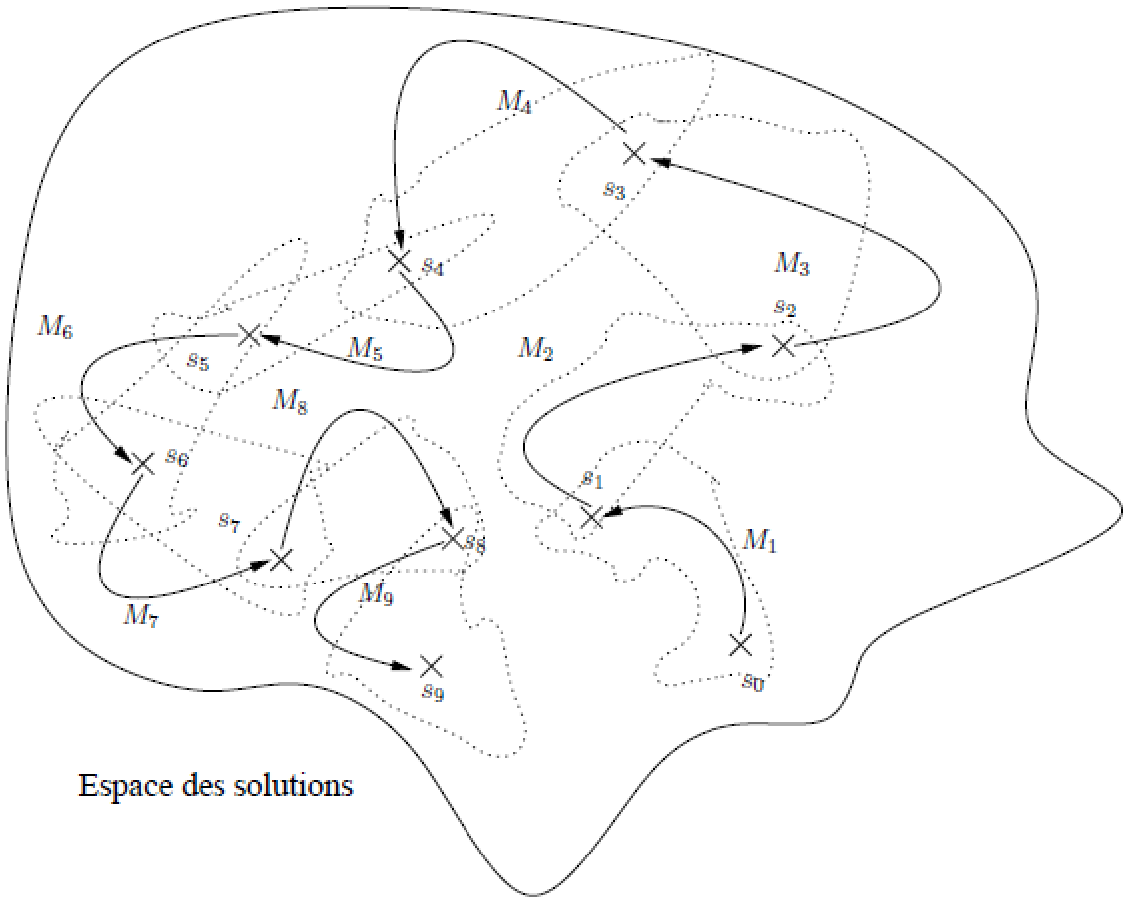
Limites

- faiblesse des relaxations du combinatoire ➔ Branch-Price-Cut
- problème dual souvent non polynomial et formulation non triviale
- délicate mise en œuvre dans un contexte industriel ➔ utilisation d'heuristiques

Approche par recherche locale

RAPPELS

- **solution initiale** dans l'espace de recherche
- amélioration de façon itérative en explorant un **voisinage**
- application de **mouvements** sur la solution courante
- méthode introduite par **Lin et Kernighan** sur le VRP.



S. Lin, B.W. Kernighan. 1973. An effective heuristic algorithm for the Traveling-Salesman Problem. In Operations Research 21, pp. 498-516.

Recherche locale <u>directe</u> et <u>pure</u>

- **Aucune décomposition n'est utilisée**
  - problème attaqué frontalement
  - espace des solutions visitées ≈ espace original des solutions
  - dimensions combinatoire et continue simultanément
  - objectif : n'omettre aucune solution

- **Aucune hybridation n'est utilisée**
  - pas de métaheuristique particulière
  - pas de recherche arborescente
  - descente standard : tous les mouvements non détériorants

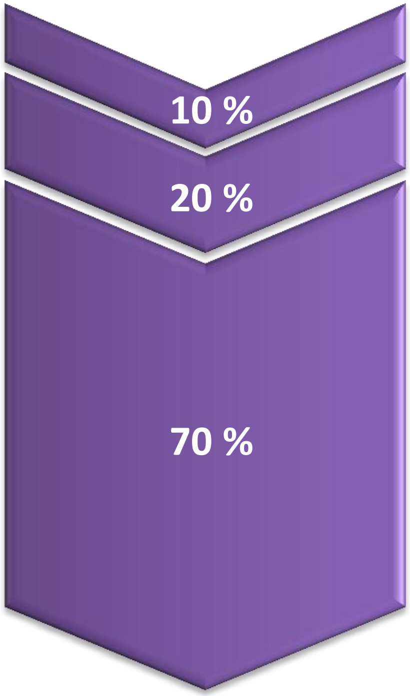
Recherche locale <u>randomisée</u> et <u>agressive</u>

- **Voisinage fortement randomisé**
  - décisions prises de manière pseudo-aléatoire
  - recherche non déterministe
  - union voisinages randomisés → voisinage très large en pratique

- **Recherche agressive**
  - des millions de solutions faisables visitées
  - fait appel à l'évaluation incrémentale
  - utilisation d'algorithme approché mais très efficace
  - exploitation d'invariants

Recherche locale en 3 niveaux

- **Définition de l'heuristique de recherche**
- **Définitions des mouvements**
  ➔ Exploration effective
- **Machinerie d'évaluation**
  ➔ Algorithmie *incrémentale*
  ➔ Principe d'ingénierie logicielle



Contribution pour les problèmes mixtes

➔ Transformations sur le combinatoire et le continu **simultanément**

➔ Evaluation incrémentale et **approchée** pour le continu

B. Estellon, F. Gardi, K. Nouioua (2008). Recherche locale haute performance. In *Actes de ROADEF 2008*, Clermond-Ferrand.

1. Une approche par recherche locale pure, directe et randomisée

2. **Application sur le problème de planification des mouvements de terre**

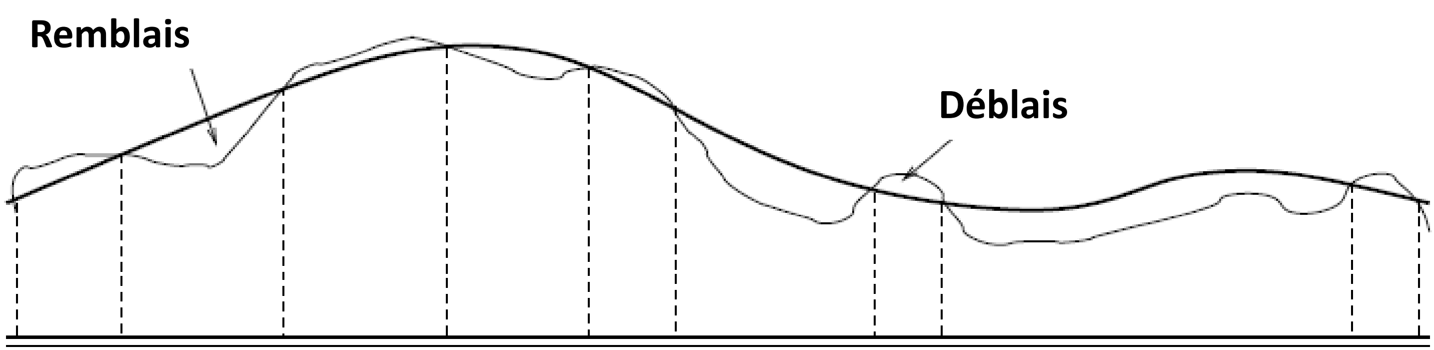
3. Application sur le problème de tournées avec gestion de stock

4. Conclusion et perspectives

Contexte

- **Un problème industriel**
  - **DTP Terrassement** : autoroutes / voies ferrées
  - objectif : optimiser la réponse aux appels d'offre
  - en entrée, résultats du problème de transport optimal de masse

- **Du profil altimétrique initial au profil cible**

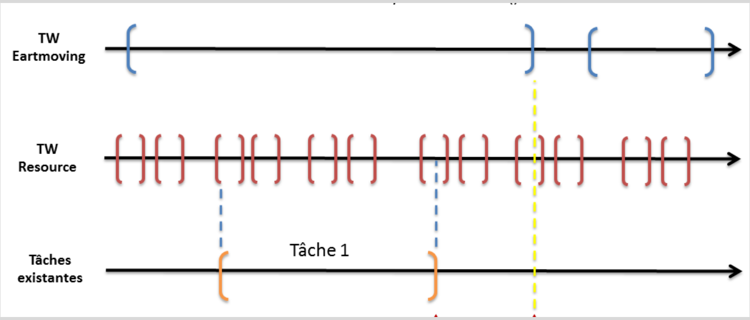


➔ jusqu'à 400 ouvrages, 2000 mouvements, 50 échelons, 2 ans de planning

G. Monge. 1781. Mémoire sur la théorie des déblais et des remblais. Histoire de l'Academie Royale des Sciences, Paris.

Un problème mixte

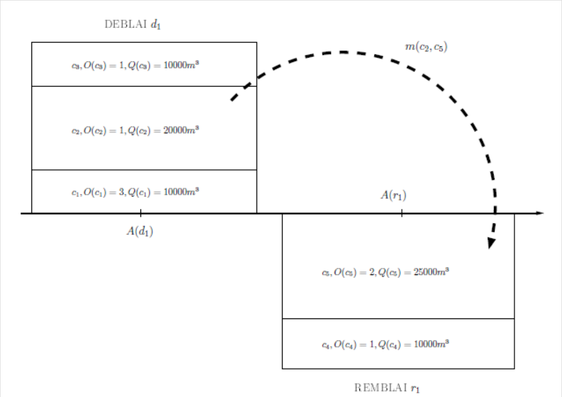
Planification des mouvements



- des tâches (mouvements de terre) ordonnancées sur des machines parallèles (engins de terrassement)

- échelons et mouvements avec fenêtres de temps

Affectation des volumes



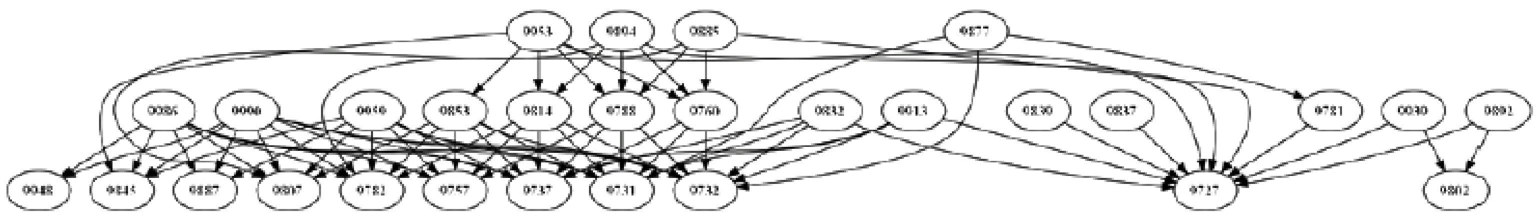
- répartition de la quantité de terre d'un mouvement de terre entre les différentes machines

- chaque mouvement est réalisé à l'aide de plusieurs tâches

SORTIE = un ensemble de tâches de travail
permettent de planifier tous les mouvements de terre sur plusieurs mois
en faisant appel aux ressources disponibles

Modélisation

- **Les contraintes**
  - tous les mouvements sont planifiés
  - fenêtres de temps **autorisées** par les mouvement de terre
  - fenêtres de **disponibilités** des ressources
  - pas de **chevauchement** des tâches
  - respect des contraintes de **précédences**



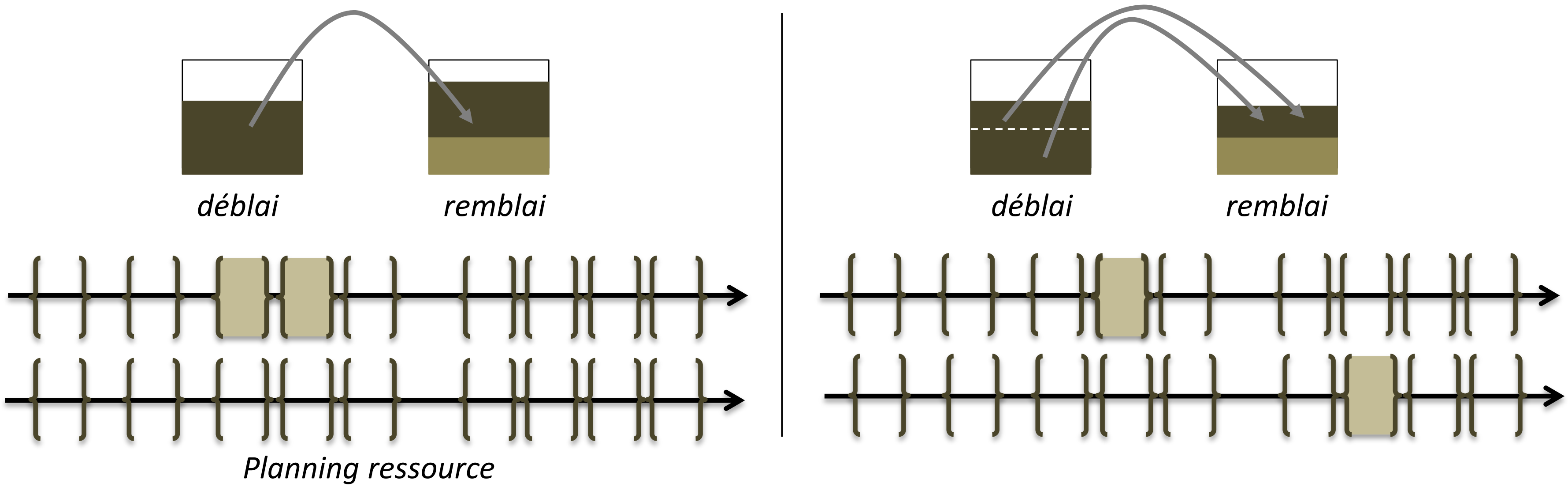
- Minimisation des **coûts des ressources**
  - coût associé au **nombre** de ressources
  - coût **temporel** : temps passé des échelons sur le chantier
  - coût **kilométrique** : distance parcourue source à source

Application de méthodologie

- **Pas de décomposition**

  Approche directe sans séparer la planification de l'affectation des volumes

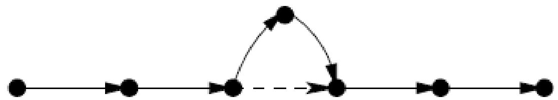
- **Ex : Transformation** travaillant sur le combinatoire et le continu **simultanément**

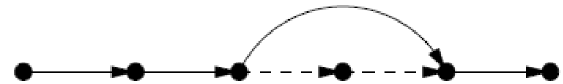


- **Aucun** voisinage large : voisinages explorés ici sont en **O(n²)**

Des transformations variées

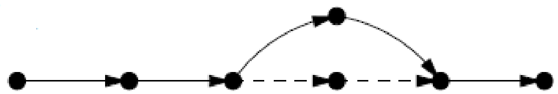
Insertion

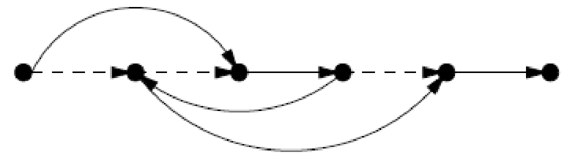




Suppression

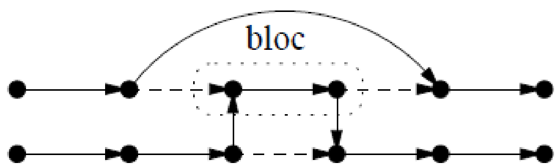
Éjection

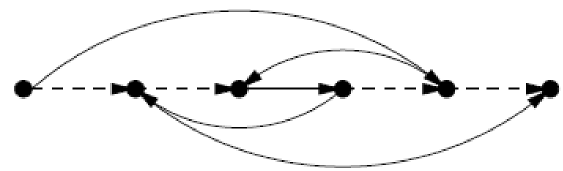




Déplacer au sein d'une ressources

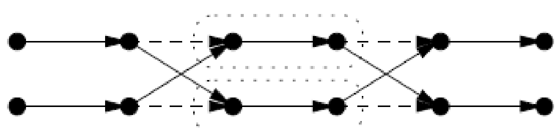
Déplacer entre ressources

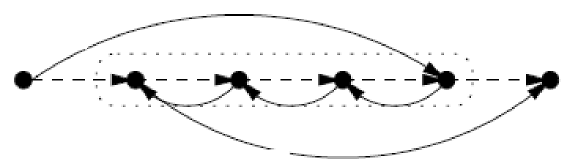




Échanger au sein d'une ressources

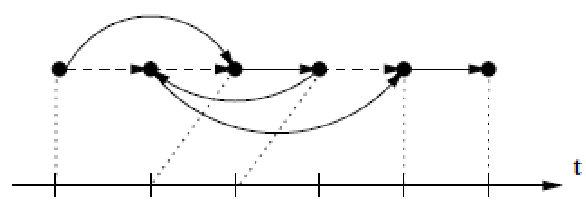
Échanger entre ressources





Miroir 3 tâches

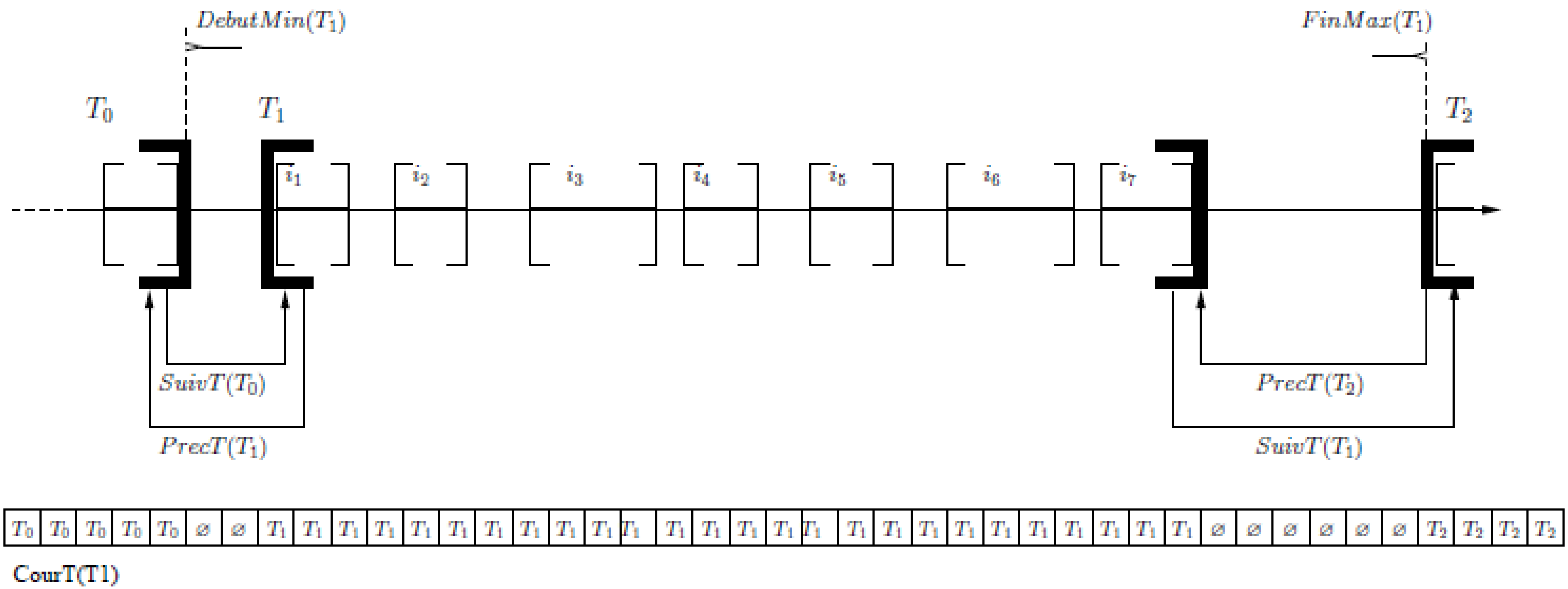
Déplacement par décalage au sein d'une ressource



➔ 100000 transformations à la seconde

Machinerie d'évaluation

- Les précédences rendent l'évaluation **complexe**
- Nécessité de **maintenir des invariants**
- **Pré-calcul et maintien de structures de données**



Complexité en temps / en espace

- Par exemple, l'insertion d'une tâche dans un planning d'une ressource

  - **Algorithme naïf en O(n)**
    parcours de toutes les fenêtres de temps jusqu'à trouver une fenêtre disponible

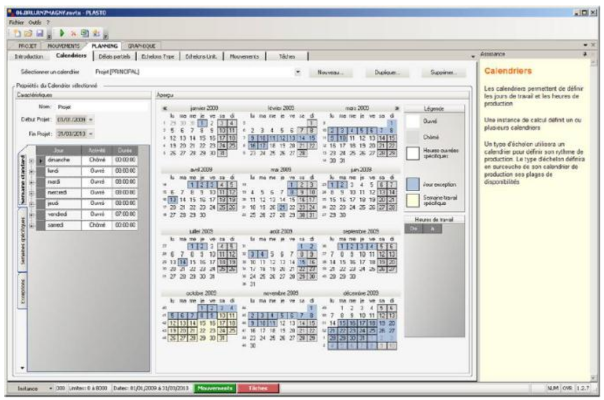
  - **Algorithme en log(n)**
    recherche dichotomique basée sur les durées cumulées de travail

  ➔ On gagne un facteur 100

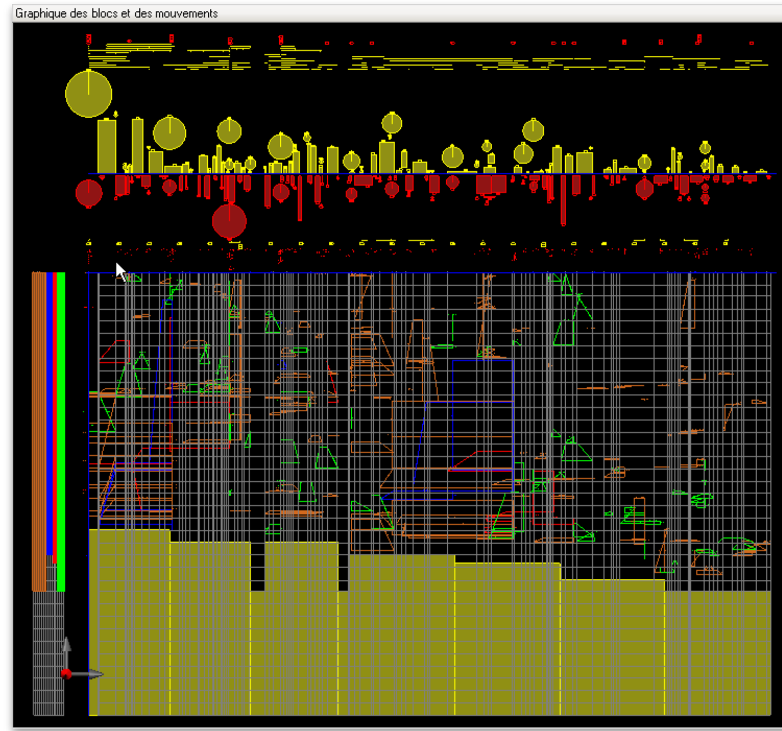
Synthèse

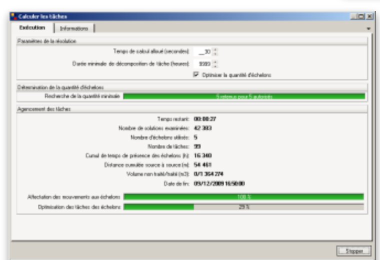
- **Plus d'un million** de transformations évaluées au cours de la résolution
- Taux d'acception constant de 8% : quelques milliers de mouvements acceptés
- Quelques **centaines** de mouvements strictement améliorant
- **Tests sur 11 instances réelles**
- Résultat opérationnel : **logiciel en exploitation depuis 2010**
- Travaux de recherche complémentaires sur le problème mono-ressource avec Thierry Benoist et Vincent Jost en 2010/2011

Création du logiciel PLASTO



Plasto 1.2





1. Une approche par recherche locale pure, directe et randomisée

2. Application sur le problème de planification des mouvements de terre

3. **Application sur le problème de tournées avec gestion de stock**

4. Conclusion et perspectives

Un problème industriel

- partenaire industriel du e-lab en 2007, groupe leader du domaine
- objectif : minimiser les coûts de réapprovisionnement à long terme
- déploiement d'envergure mondiale
- généralisation de l'Inventory Routing Problem (I.R.P.)
- travaux de recherche menés de 2007 à 2010 avec
  Thierry Benoist, Frédéric Gardi et Bertrand Estellon

W. Bell et al. 1983. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. Interfaces 13 (6)

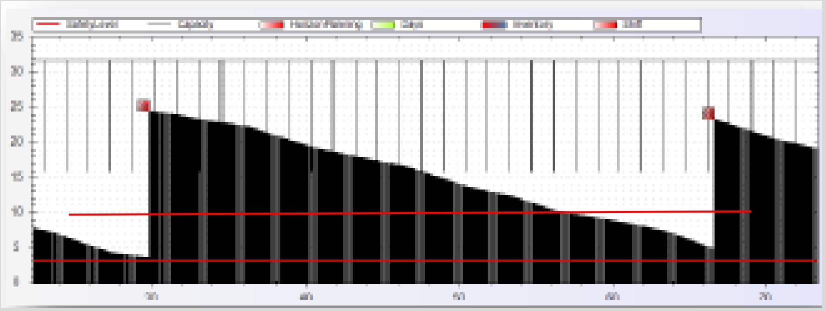
Un problème mixte

Construction des tournées

- Ressources non homogènes
- Fenêtres de temps
- Matrice de compatibilité
- Vitesse des camions
- Règles de conduite stricte
- Plusieurs dépôts
- Gestion des pauses

+

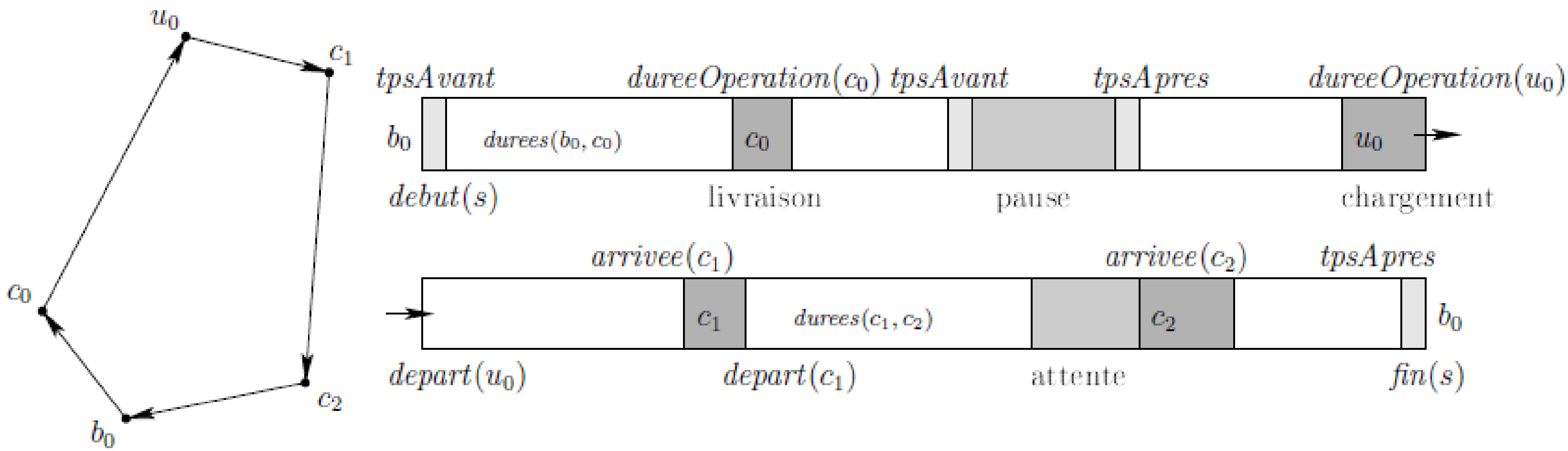
Gestion des stocks



- Capacité de stockage fixe
- Télémétrie + modèle de prévisions (consommations non linéaires)
- Seuil de sécurité (déterministe)
- Heures d'ouvertures de chaque site
- Accès restreints
- Idem pour l'inventaire des usines

SORTIE = un ensemble de tournées de dépôt à dépôt
Une suite d'opérations (date, site, quantité)
réalisées par un triplet (conducteur, tracteur, remorque)

Tournées de livraison de produit



- **Objectifs à long terme**

  Minimiser le ratio logistique *LR = SC / DQ,*

  SC la somme des coûts des tournées et *DQ* la somme des quantités livrées.

- **Processus itératif de planification à court terme**

Complexité du problème

- **NP-difficile** IRP ← VRP ← TSP
- **Échelle des instances en pratique**
  - horizon de planification de 15 jours
  - des centaines de clients (jusqu’à 1500)
  - des dizaines d’usines (jusqu’à 50)
  - des dizaines de dépôts (jusqu’à 50)
  - des dizaines de ressources (jusqu’à 50 par type)
  - temps continu (précision à la minute : 21600 minutes)
  - consommations/productions à l’heure (360 pas de temps)
- Temps de résolution limité à **5 minutes** sur un ordinateur standard
- **Gain minimum** attendu sur solutions métier : **8 % en moyenne**

Application de la méthodologie

- Algorithme de recherche locale pour la planification court terme
  - **pur** : pas de métaheuristique, pas d'hybridation
  - **direct** : le problème n'est pas décomposé pour être résolu

- **Heuristique générale**
  - solution initiale construite par glouton (basé sur l'urgence des clients)
  - *first-improvement descent*
  - choix stochastique des mouvements

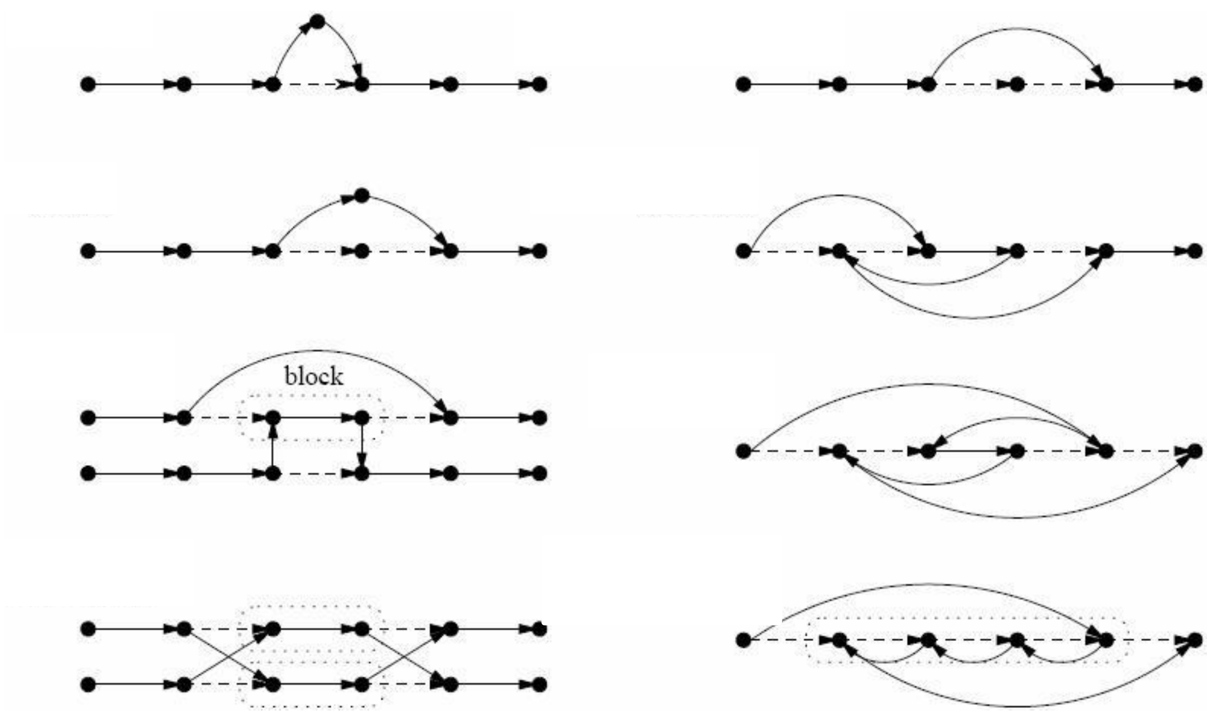
Transformations sur les opérations

Insertion — *Suppression*

Éjection — *Déplacer au sein d'une tournée*

Déplacer entre 2 tournées — *Échange au sein d'une tournée*

Échanger entre 2 tournées — *Miroir*



➔ **voisinage de taille *O*(*n*^{2})** avec *n* le nombre d'opérations dans la solution courante.

(Ré)ordonnancement d'une tournée de *k* opérations

- **Objectif :** minimiser le temps improductif sur la tournée → glouton chronologique
  - prendre la pause le plus tard possible et convertir les temps d'attente devant les sites fermés en pause
  - en temps $O(k)$ si les pauses ne sont pas stockées explicitement
  - calcul avant ou arrière rendu symétrique
  - optimal si aucun temps d'attente n'apparaît sur la tournée
- **Toute tournée impactée par un mouvement** est réordonnancée depuis la première opération impactée jusqu'à la fin (si avant) ou début (si arrière).

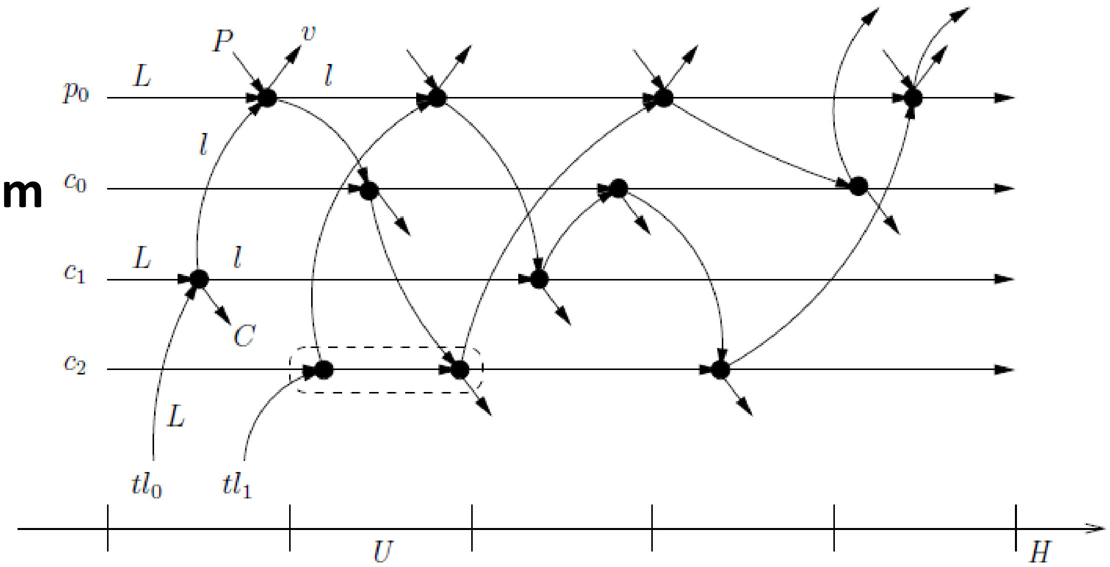
C. Archetti, M. Savelsbergh. 2007. The truckload trip scheduling problem. TRISTAN VI, Phuket Island, Thailand.

(Ré)affectation des volumes sur *n* opérations (1)

- maximiser le volume total livré

➔ critique pour la performance

Résolution exacte par flot maximum



➔ temps $O(n^3)$
➔ TROP LENT !

- Glouton poussant un maximum de flot à chaque nœud du DAG suivant un ordre topologique (= chronologie des opérations) → temps $O(n \log n)$

- **Propriété (intéressante en pratique)**
  optimal si chaque client est livré une seule fois sur l'horizon de planification

<u>(Ré)affectation des volumes</u> sur *n* opérations (2)

- **Difficulté**
  - Réaffecter la quantité livrée à un client à l'instant *t* alors que d'autres opérations ont lieu après *t* → risque de pénurie ou de dépassement de capacité après *t*.

    ➔ mise en œuvre incrémentale délicate :

    les réseaux avant et après transformation sont « superposés » dans la même structure de données.

- **Récompense :** en pratique, le glouton partiel s'exécute en <u>temps quasiment constant</u> en le nombre d'opérations dans le réseau.
  - T(glouton partiel) = T(glouton complet) / 100
  - T(glouton partiel) = T(flot maximum) / 2000
  - Écart moyen à la réaffectation optimale : 2 %

Chiffres clés

- Environ **30 000 lignes de code**, dont 6 000 (20 %) dédiées au *checkers*
- Plus de 10 000 mouvements par seconde
- Près de **10 millions de solutions visitées** en 5 minutes
- Taux d’acceptation quasiment constant durant toute la recherche (diversification naturelle)
- **Taux d’acceptation global** des mouvements **entre 1 et 10 %**
- Un millier de mouvements strictement améliorant
- **Gain moyen de 21 %** sur notre algorithme glouton
- **Gain moyen de 25 %** sur les experts logistiques

1. Une approche par recherche locale pure, directe et randomisée

2. Application sur le problème de planification des mouvements de terre

3. Application sur le problème de tournées avec gestion de stock

4. **Conclusion et perspectives**

Approche par Recherche Locale

- Cette thèse propose une approche directe et pure, par recherche locale, pour les problèmes d'optimisation en variable mixtes

- La méthodologie présentée repose sur :
  - une large variété de transformations travaillant simultanément sur le combinatoire et le continu
  - une évaluation incrémentale exploitant les invariants
  - des algorithmes d'évaluation approchés pour la dimension continue

Tests sur 2 problèmes réels

- 2 **problèmes industriels réels résolus** en utilisant cette méthode :
  - efficacité et la robustesse de notre approche ;
  - **évaluation incrémentale** de ces mouvements : fonctions travaillant aussi bien sur les deux dimensions combinatoires et continues des problèmes ;
  - plusieurs dizaines de millions de mouvements / minute
    → très large **diversification** et une descente rapide vers solutions qualité.
- Gains significatifs observés en production.

Perspectives

- **Perspectives d'applications**
  - Planification des corps d'état secondaire
  - Planification de Partenariat Public-Privé pour l'éclairage public
  - Smart Grid / Pilotage énergétique du bâtiment

- **Algorithmie incrémentale exploitant les invariants**
  - Quelques tentatives de formalisations
  - Beaucoup de travaux éparpillés dans le domaine
  - Un débouché naturel de ces travaux :
    la formalisation dans un solveur de ces concepts (Localsolver)

Merci de votre attention !

- ✓ Antoine Jeanjean (2010). *Resource scheduling optimization in mass transportation problems.* In Project Management and Scheduling, **PMS**, Université de Tours, Avril 2010.
- ✓ T. Benoist, B. Estellon, F. Gardi, A. Jeanjean (2010). *Randomized local search for real-life inventory routing.* **Transportation Science** (to appear).
- ✓ T. Benoist, B. Estellon, F. Gardi, A. Jeanjean (2010). *Recherche locale pour un problème d'optimisation de tournées de véhicules avec gestion des stocks.* In Proceedings of **MOSIM** 2010, Hammamet, Tunisia.
- ✓ T. Benoist, B. Estellon, F. Gardi, S. Jain, A. Jeanjean, E. Patay (2009). *Inventory routing optimization for bulk gas transportation*. In **INFORMS** 2009, Annual Meeting. San Diego, US-CA.
- ✓ T. Benoist, B. Estellon, F. Gardi, A. Jeanjean (2009). *Algorithme de recherche locale pour la résolution d'un problème réel de tournée d'inventaires.* In **JOR** 2009. Paris, France.
- ✓ T. Benoist, B. Estellon, F. Gardi, A. Jeanjean (2009). H*igh-performance local search for solving real-life inventory routing problems.* In Proceedings of **SLS** 2009, Lecture Notes in Computer Science 5752, pp. 105-109. Springer, Berlin, Germany
- ✓ T. Benoist, B. Estellon, F. Gardi, A. Jeanjean (2009). *Recherche locale haute performance pour l'optimisation des livraisons de gaz industriels par camions-citernes.* In Actes de **ROADEF** 2009, pp. 49-50. Nancy, France.
- ✓ T. Benoist, A. Jeanjean, P. Molin (2008). *Minimum Formwork Stock Problem on residential buildings construction sites.* In **4OR**: A Quarterly Journal of Operations Research, Octobre 2009
- ✓ T. Benoist, A. Jeanjean, G. Rochart, H. Cambazard, E. Grellier, N. Jussien (2006). *Subcontractors scheduling on residential buildings construction sites.* In **ISS'06** International Scheduling Symposium, Technical Report, pp. 32-37

Menu

Intro

Plasto

IRP

CCL

Limites Décomp

Annexes

RL & MétaH

PB Mixtes

H.P.

Incr. versus BCP

B & P

IRP Subtitut

Décomp IRP ?

Banches Pb

Banches RL

TSP UP

Recherches locales et Métaheuristiques

- **Lin et Kernighan** ont utilisé la recherche locale en l'appliquant au problème du voyageur de commerce et en formalisant les concepts sous-jacents

- Avènement de la notion de **métaheuristique**, introduite par Glover

- **Le recuit simulé :** on cherche à minimiser l'énergie du système en introduisant une température et en acceptant de ``mauvaises'' solutions pour fuir les optima locaux

- **Les algorithmes génétiques :** reposent sur le concept de sélection naturelle, en se rapprochant par bonds successifs d'une solution

- **Les algorithmes de colonies de fourmis :** concept de rétroactions positives (règle aléatoire de transition proportionnelle entre deux solutions

- **La recherche Tabu**

Problèmes d’optimisation mixte

- **Les problèmes d'optimisation mixte** conjuguent les attributs des problèmes d'optimisation combinatoire et continue

- Certaines variables de décisions du problème d'optimisation mixte sont contraintes de prendre **des valeurs entières, d'autres des valeurs continues**

- Ces problèmes d'optimisation apparaissent dans **diverses applications** :

  - l'optimisation de réseaux de conduites pétrolières
  - l'optimisation des tournées de camions avec réservoir (essence / gaz)
  - l'optimisation des plans de production
  - L’optimisation des arrêts de tranches des centrales nucléaires ou de raffineries
  - l'optimisation du placement des puits en ingénierie de réservoir
  - l'optimisation des plannings et des chargements de porte-containers
  - la construction des horaires de plans de vols (choix avions et gestion aléas)

Branch and Price

- Méthode **d'optimisation combinatoire** pour résoudre des problèmes d'optimisation linéaire en nombres entiers.
- Combine algorithme du Branch & Bound et génération de colonnes.
- Baptisée par Savelsbergh et al, technique initialement proposée par Johnson et implémentée par Desrochers et al.
- **Problème maître** dont on a relâché les contraintes d'intégrité qui sont imposées au fur et à mesure de la descente dans l'arbre du branch and bound
- **Problème esclave** (problème de pricing) qui évalue chaque nouvelle colonne ou groupe de colonnes ajoutées au problème maître.

> **Très efficace sur des problèmes de très grande taille**
> **Complexe à mettre en œuvre : Problème dual est souvent non polynomial**
> **Formulation n'est pas toujours triviale**

Notion de « haute performance » algorithmique

- **The Art of Computer Programming (Knuth et al., 1968)** :
  - Conception de l'algorithme hautement reliée à son analyse
  - L'efficacité d'un algorithme :

    sa vitesse d'exécution, son encombrement mémoire
  - Analyse de l’algorithme : prévision des ressources nécessaires

- **Cormen et al. (1994) :**
  - algorithmes = une ``technologie''
  - Algorithmes dans le compilateur, l'interpréteur, le routage d'informations l'interface graphique et bien sûr dans l'application elle-même

- **Moret (2002) et Helsgaun (2000) :**
  - performance algorithmique / Ingénierie algorithmique
  - algorithmique expérimentale :

    efficacité des algorithmes et structures de données.

Algorithmie incrémentale versus BCP

- **L'algorithmie incrémentale** est certes complexe à implémenter

- Le développement se fait par **amélioration itérative**
- Exploitation par étape de nouveaux invariants
- Améliorant ainsi petit à petit les performances de la mécaniques d'évaluation

- **Au départ :** mécanique d'évaluation mise en place rapidement (algos naïfs)
- Amélioration progresse par analyse algorithmique

➔ Implémentation de l'ensemble **plus simple** que celle d'un BCP

- Les BCP nécessitent une période plus longue de développement
  avant d'obtenir une première solution admissible.

Limites des décompositions mathématiques

- Délicates à mettre en œuvre
- Font appel à des concepts mathématiques avancés
- Nécessite d’intégrer parfaitement l'appel des solveurs
- Faiblesse des relaxations : méthodes complètes type BCP
- Problème dual souvent non polynomial, formulation non triviale
- Aucune garantie d’amélioration de la solution optimale du problème maître.
- Balance gain qualité apportée sur la borne par le dual et gain en temps de calcul global pas facile à calibrer
- L'ensemble initial de colonnes a son importance
- La génération dynamique de coupes dans le problème maître : ajout d'inégalités valides calculées en temps non polynomial.
- Difficile parfois de séparer le problème maître et le dual

Décomposition de l'IRP ?

- Certes, il y a deux routines
- Mais un même mouvement modifie aussi bien le combinatoire et le continue
- Deux routines pour faciliter les tests et l'analyse de performance.
- Le calcul des volumes pourrait tout à fait être intégrer à la routine de construction des dates de la tournée, à chaque site.

➔ Ce regroupement permettrait en plus d'introduire des temps de livraison variable en fonction du volume, contrainte qui n'était pas imposée ici.

Banches – Description du problème

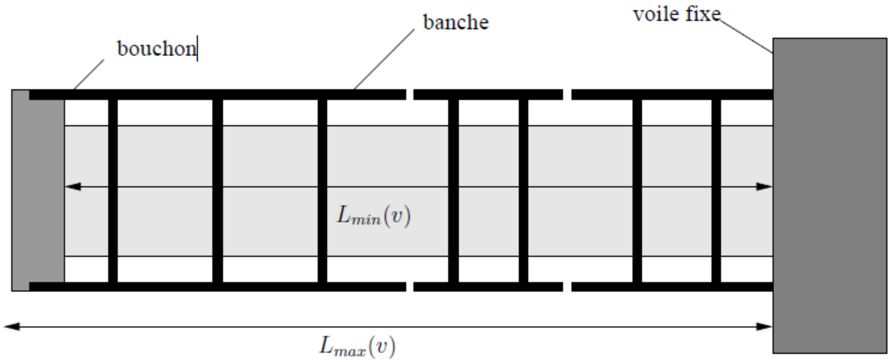
- Problème purement **combinatoire**

- Bouygues **Habitat Résidentiel** – Bureau des méthodes

- **Un chantier =** une dizaine d'étages et une centaine de voiles.
- Chaque étage = une **dizaine** de jours

- **Optimiser** le nombre de banches commandées
  - minimiser les banches non utilisées
  - diminuer les frais de location

- **Contraintes**
  - Couler tous les murs le jour attendu
  - Respecter les longueurs
  - Garantir la stabilité du train



Banches – Recherche Locale

- **Modélisation par couverture d'ensembles**
  - le nombre d'affectations faisables sur chaque voile suffisamment petit pour être calculé de manière préliminaire en une fraction de seconde
  - moins de 2000 combinaisons de trains par murs
- **Recherche locale très simple**
  - Solution initiale à partir de l'optimum fractionnaire du modèle linéaire
  - Un mouvement consistant à choisir un nouveau train de banche pour un mur (parmi la liste des trains possibles pour ce mur)
  - Une simple descente dans ce voisinage donne de bons résultats

Fonction objectif substitution IRP

- Minimiser le ratio logistique à court terme

➔ pas nécessairement bonnes solutions sur le long terme.

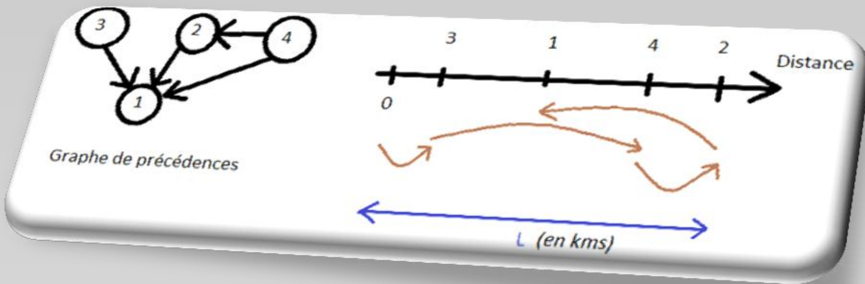
- Introduire une fonction objectif modifiée :

« ne jamais remettre à demain ce qu'on peut faire de façon optimale aujourd'hui »

- Nouvel objectif à court terme : minimiser le coût global additionnel par unité de produit livré, comparé au ratio logistique optimal $RL*$

$$\mathrm{CT}^*(s) = \sum_{client\ clivr\ pendant\ t} RL^*(c) \times quantite(p)$$

$$RL^* = \frac{\sum_s (CT(s) - CT^*(s))}{QT}$$



Le problème du voyageur de commerce unidimensionnel avec précédences

Antoine Jeanjean [1,2], Thierry Benoist [1], Vincent Jost [2]

1. Bouygues e-lab, 40 rue de Washington, 75008 Paris
2. LIX, UMR CNRS 7161, École Polytechnique, F-91128 Palaiseau

{ajeanjean, tbenoist}@bouygues.com, vjost@lix.polytechnique.fr

Définition du problème

Problème du voyageur de commerce unidimensionnel **avec précédences**

Étant donnés :

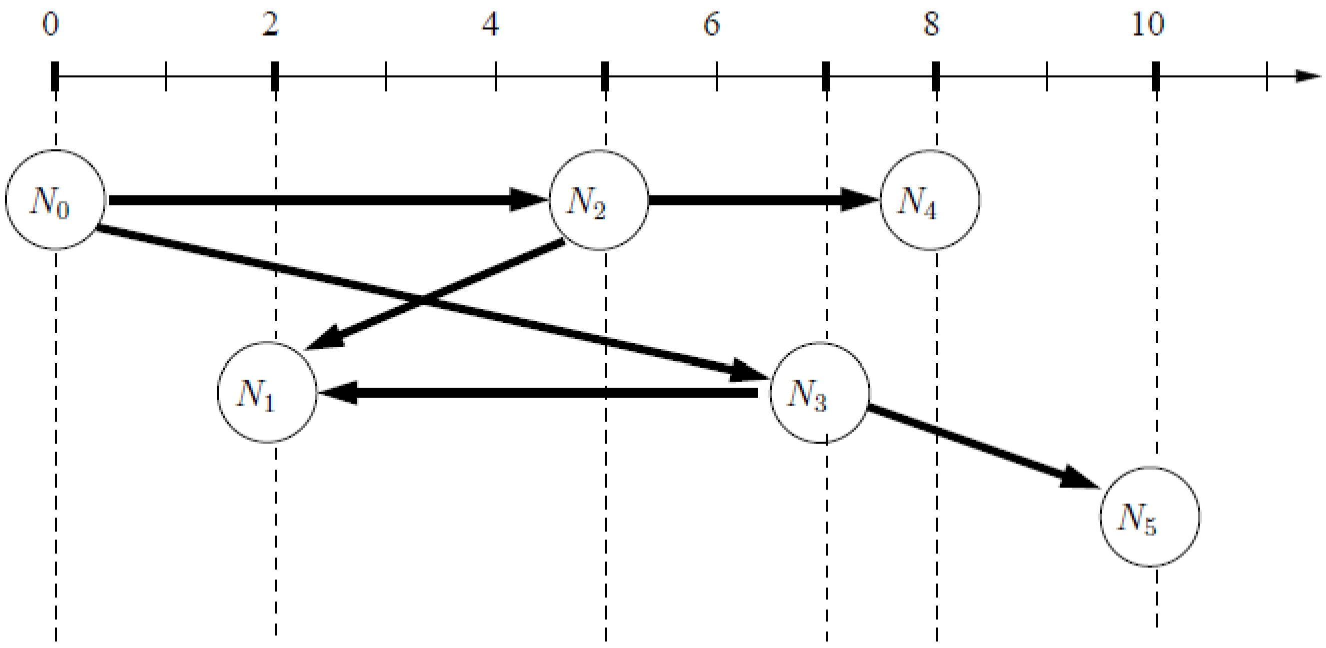
- un ensemble P de n points ayant chacun une abscisse X_i, avec $i \in [1, n]$;
-

Trouver une permutation L des points P telle que :

- $\sum_{i=1}^{n} |X_{L(i)} - X_{L(i-1)}| \leq K$, avec $X_{L(0)} = 0$.
-

➔ **On parlera ici de TSP-UP**

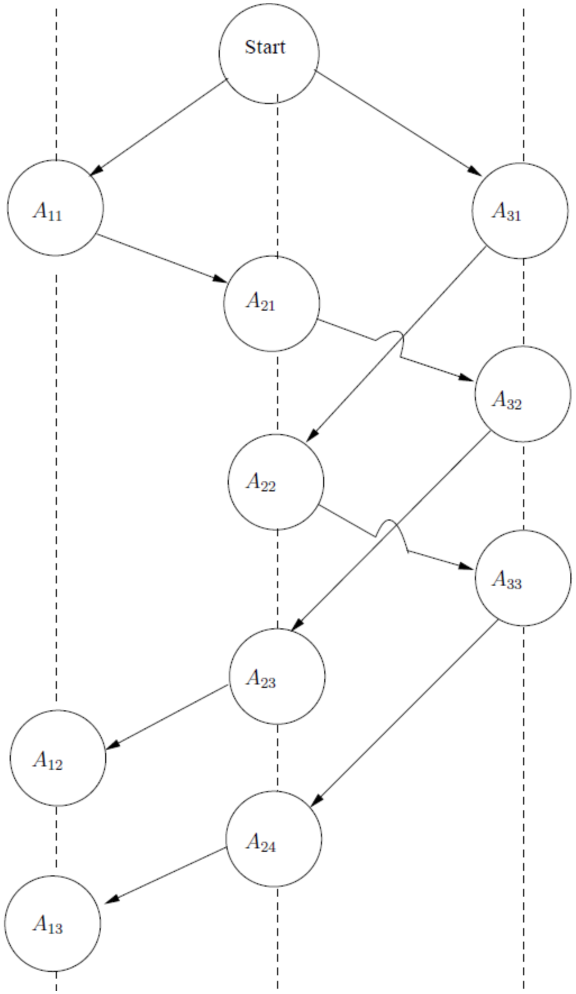
Exemple



Graphe de précédences.

Notons que plusieurs nœuds pourraient avoir la même abscisse

Propriété



- Lorsqu'on passe par une abscisse il est toujours optimal de traiter tous les nœuds « *disponibles* » de cette abscisse.

- Donc une solution est une suite de décisions *droite/gauche.*

  (soit 2^n solutions au plus).

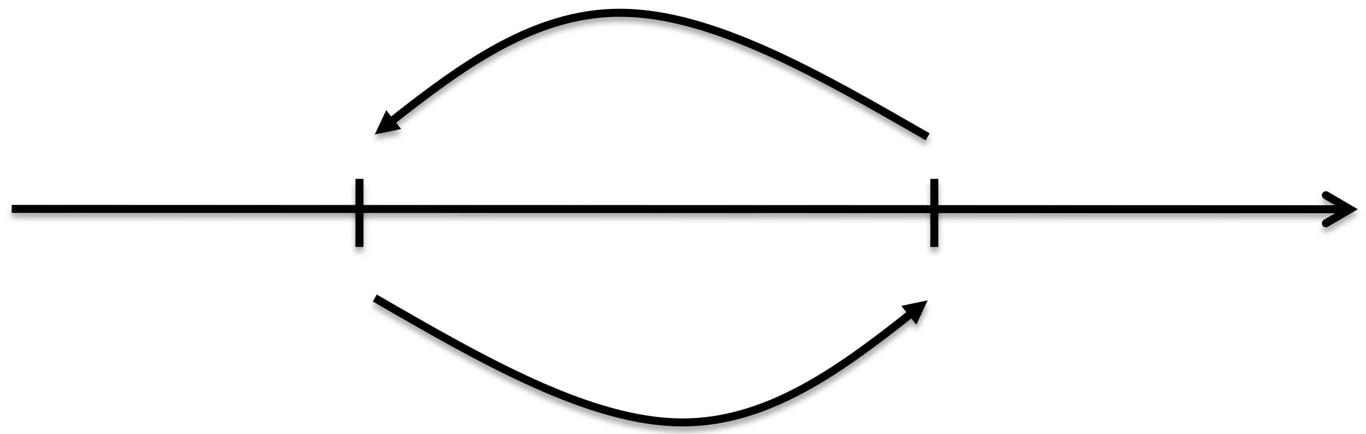
Plusieurs travaux

- Un calcul de borne inférieure
- NP-Complétude
- Une méthode arborescente exacte par programmation dynamique

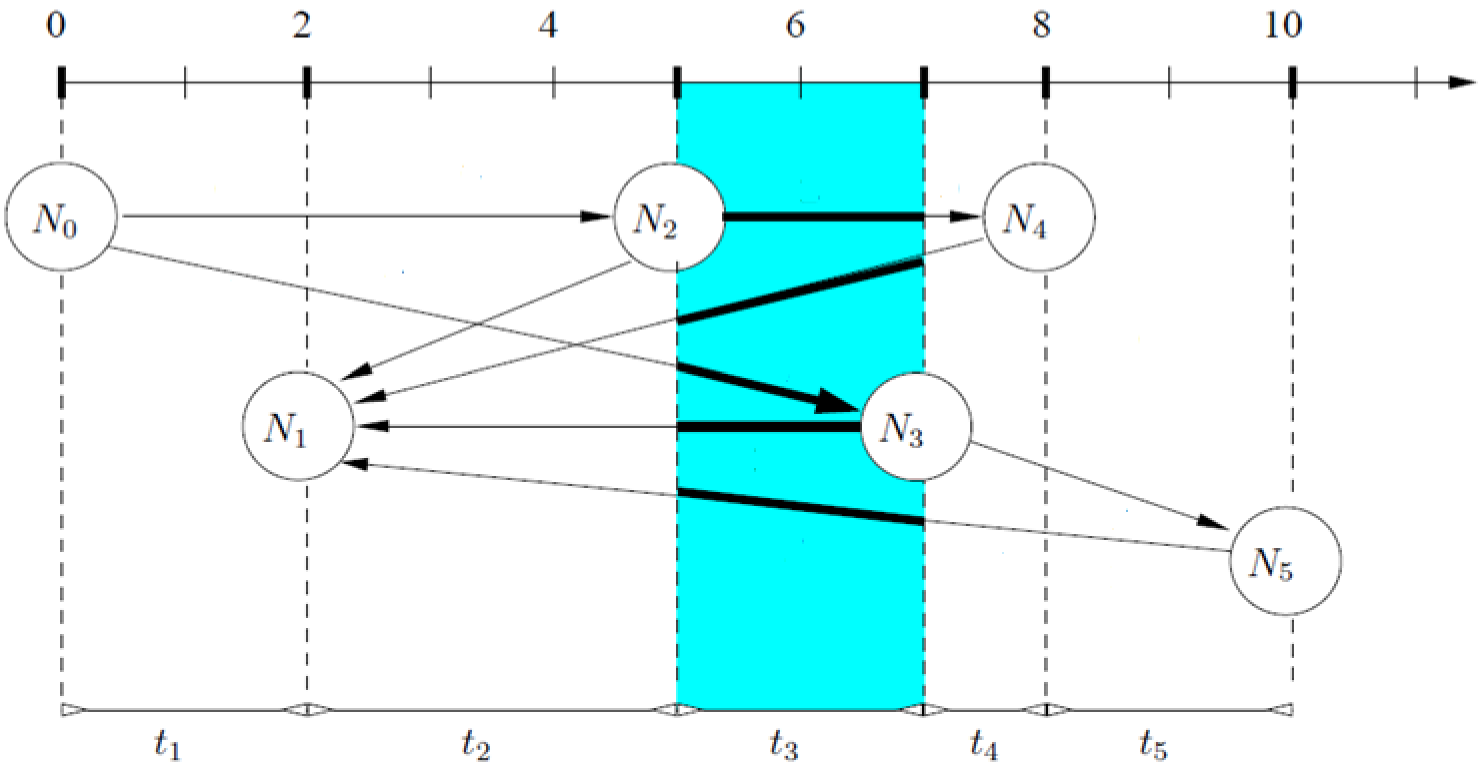
Cas à 2 abscisses

Trivial : une seule solution possible :

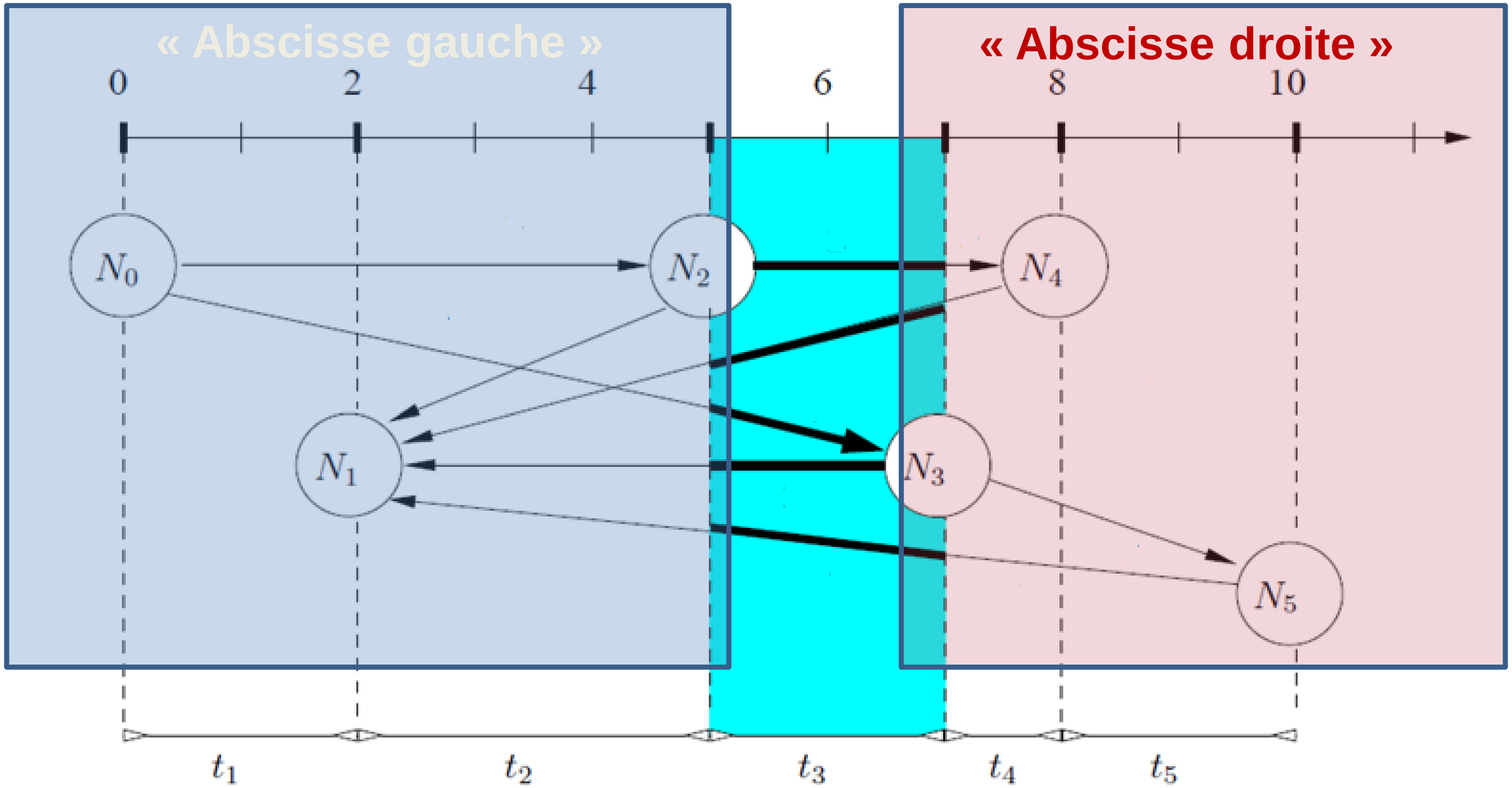
- Quand on est à gauche, on va à droite.
- Quand on est à droite, on va à gauche.



Borne par tronçon

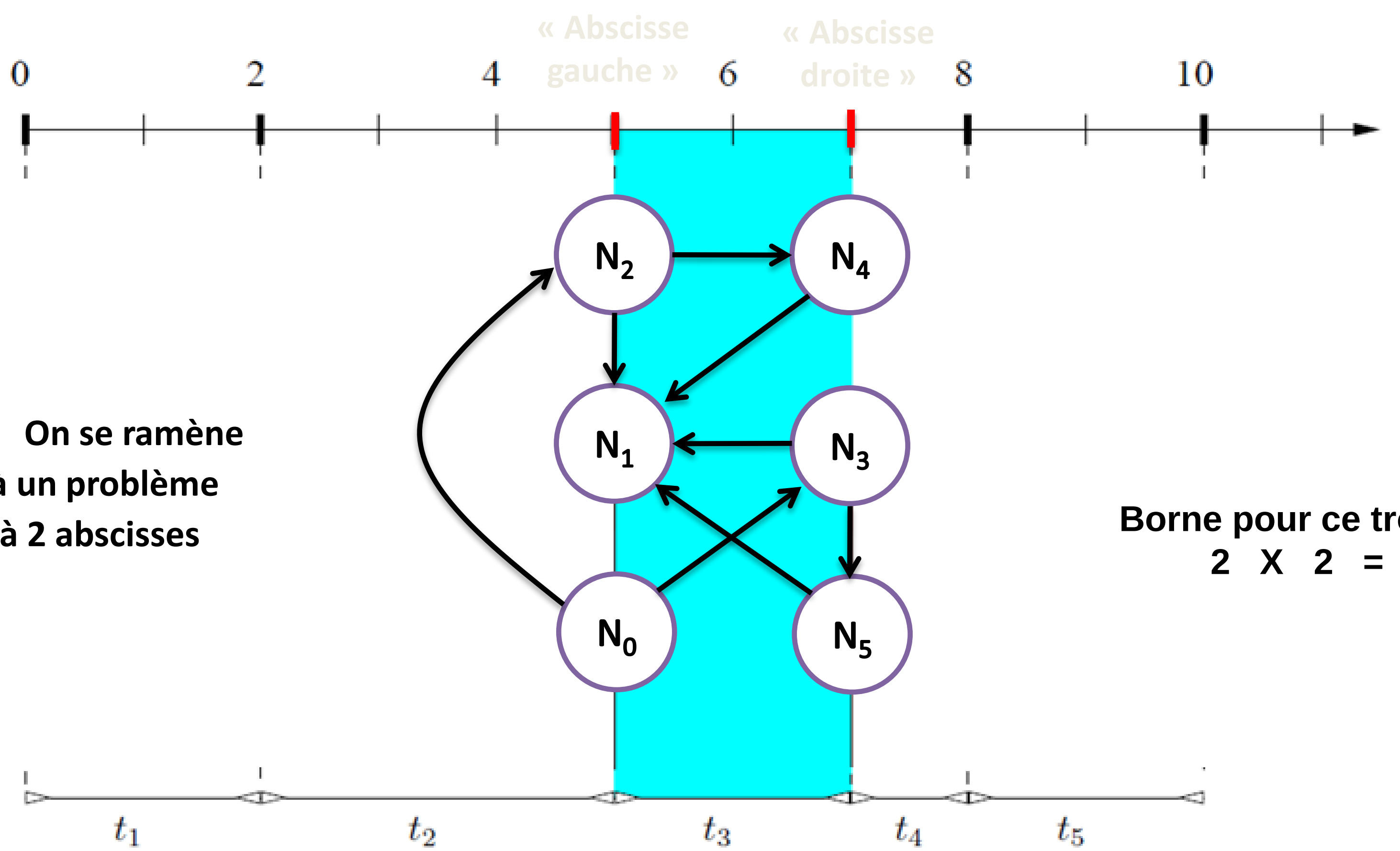


Borne par tronçon



- ➔ **On peut mesurer le nombre minimum de fois qu'un tronçon sera traversé**
- ➔ **On peut améliorer la borne sans surcoût en énumérant les abscisses de fin possible**

Borne par tronçon



➔ On se ramène à un problème à 2 abscisses

Borne pour ce tronçon :
2 X 2 = 4

Borne par tronçon

Complexité du calcul de borne

Nombre de tronçons (au plus N-1) X **Nombre d'arcs (A)**

- On peut **enrichir** le calcul de la borne en la calculant pour chaque nœud terminal possible.

- On multiplie donc la complexité par le nombre de puits candidats P.

- Mais on peut simplifier ce calcul en évitant de re-calculer le coût des tronçons pour tous les puits.

Cas à 3 abscisses

SHORTEST COMMON SUPERSEQUENCE

- INSTANCE: Finite alphabet Σ, finite set R of strings from Σ^*.
- SOLUTION: A string $w \in \Sigma^*$ such that each string $x \in R$ is a subsequence of w, i.e. one can get x by taking away letters from w.
- MEASURE: Length of the supersequence, i.e., $|w|$.

- **NP-complet** même pour un alphabet à 2 lettres {D,G}.

- **Maier** (1978) avait prouvé la NP-Complétude du problème pour un alphabet de taille supérieur à 5. **Räihä et Ukkonen** (1981) ont prouvé la NP-complétude du problème pour un alphabet de taille supérieure ou égale à 2.

Exemple de superséquence

- *Soient deux séquences*

  S1 =< G;D;G;G;D;G >

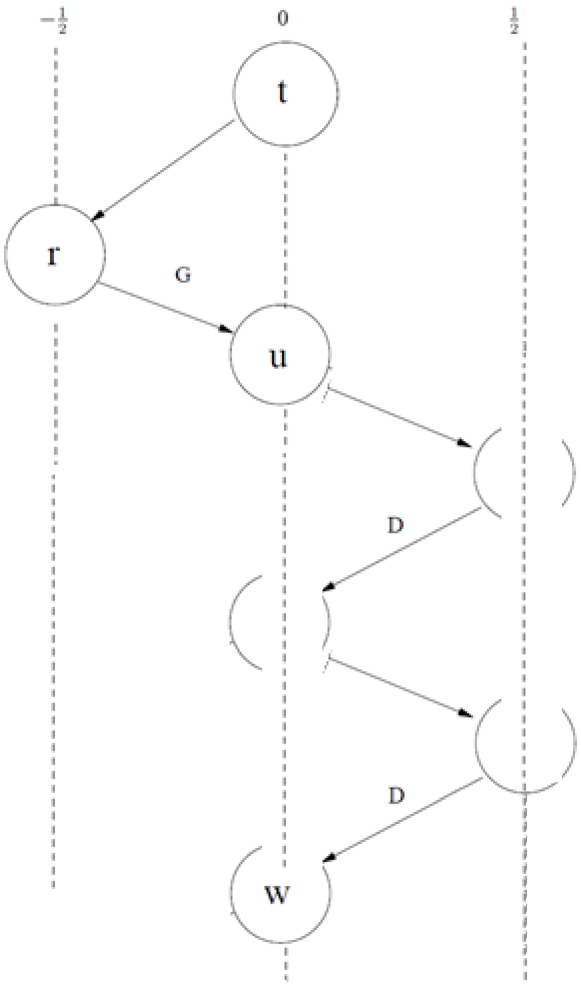
  et

  S2 =< D;D;G;D;G >

- *Une exemple de super-séquence de S1 et S2 est :*

  <G;D;D;G;G;D;G>

Réduction



Pour chaque mot x on crée une chaîne de précédences avec des abscisses :
-1/2 0 +1/2

- Exemple : mot G D D

On montre qu'un chemin est une solution du TSP-UP créée si et seulement si chaque chaîne de précédences apparaît dans le chemin.

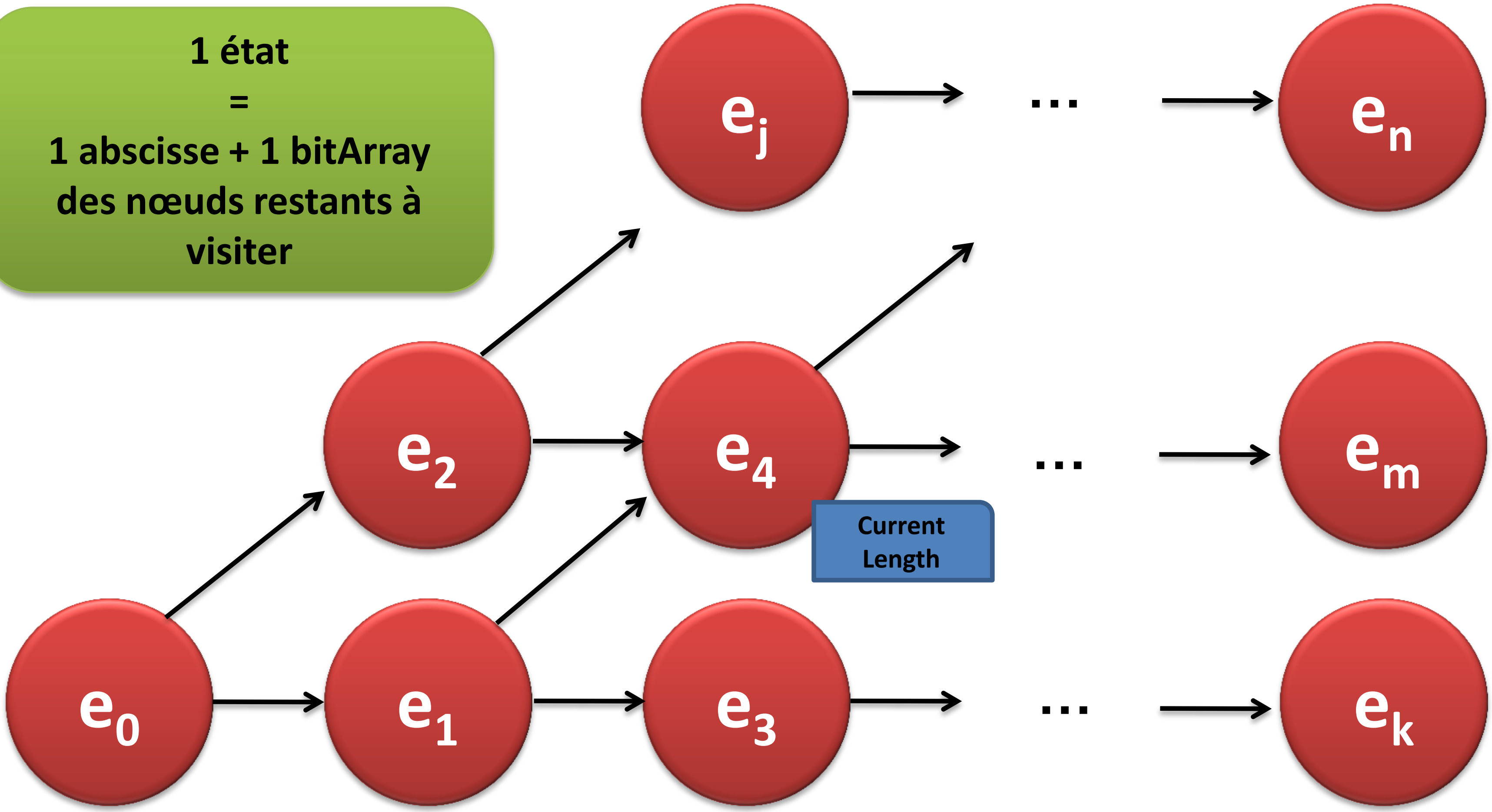
Donc le TSP unidimensionnel avec précédences est NP complet

Cas à 3 abscisses et plus

- **Enumération** (*2^n* solutions au plus).
- **Programmation dynamique** à la Held & Karp :
  - A une abscisse, l'état est défini par l'ensemble des nœuds déjà faits (soit $n2^n$ états).
  - Beaucoup moins en réalité grâce aux précédences et à l'unidimensionalité. (« **on ne saute jamais par-dessus un point »**)
  - ***Call-based dynamic programming :*** Énumération arborescente en stockant les états dans une table de hachage. Permet de couper, de commencer par la meilleure branche, d'avoir une solution avant la fin…

Programmation dynamique

1 état
=
1 abscisse + 1 bitArray des nœuds restants à visiter



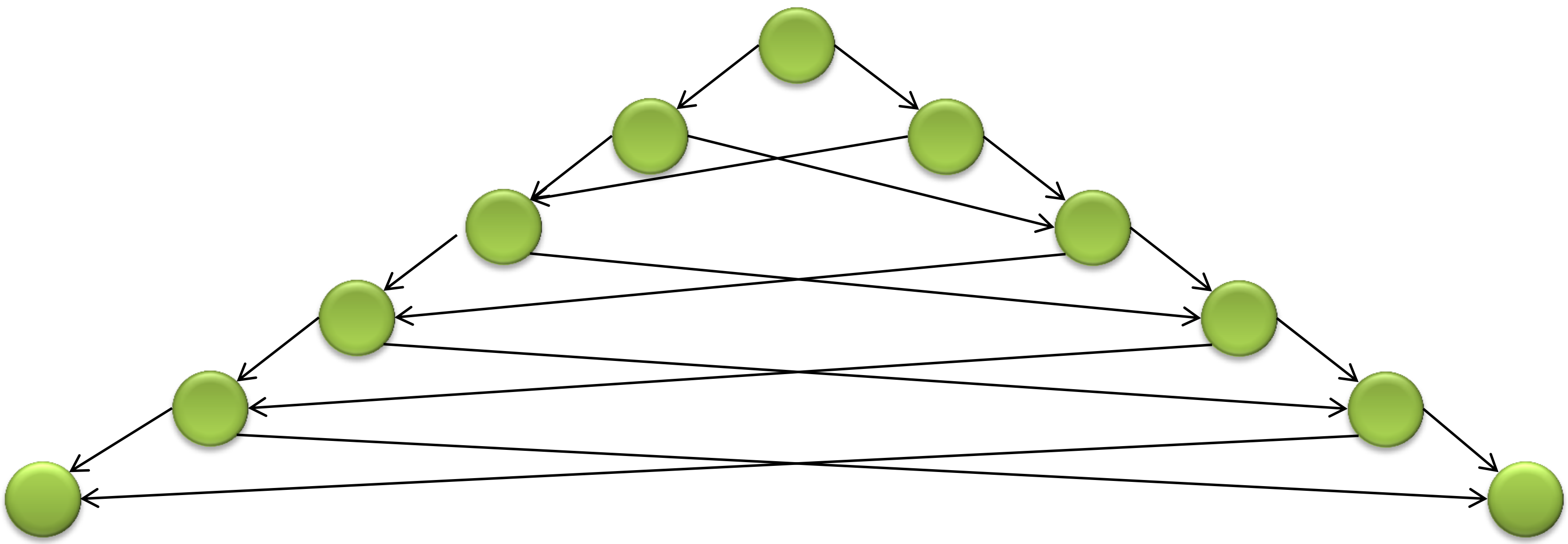
Cas à 3 abscisses et plus

- Exemple : pas de précédences :

- Énumération = n nœuds
- Programmation dynamique = n nœuds

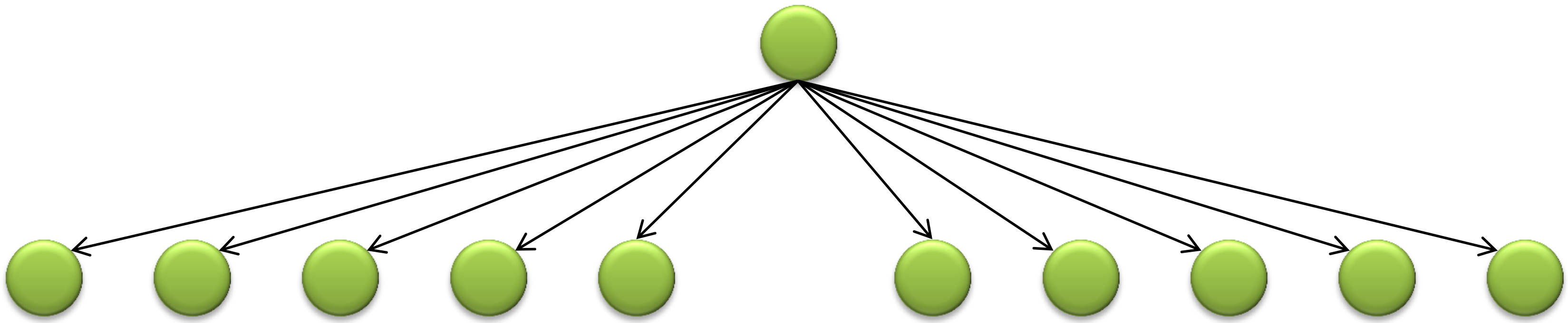
Cas à 3 abscisses et plus

- **Exemple :** Précédences fortes :



- Énumération = 2^n nœuds (1 choix binaire par étage)
- Prog. dynamique = 4n nœuds (4 états par étage)

Exemple



- Énumération = 2^n nœuds
- Programmation dynamique = n^2 nœuds

En « pratique »

Traitement des chantiers linéaires avec une seule ressource

	L (m)	Borne		Programmation Dynamique				Parcours complet			
		Tronçons (m)	t (s)	Prog Dyn (m)	nbPoints	nbNoeuds	t (s)	Fullscan (m)	nbPoints	nbNoeuds	t (s)
Instance 1	33290	50138	4670,253	**66580**	153850	307974	130,721	79654 (STOP)	STOPPEE	-	4500
Instance 2	8000	17152,5	580,206	**18165**	1448	3081	0,533	18165	809327	2278037	1,778
Instance 3	5500	5500	0,99	**5500**	8	19	0,053	5500	8	19	0,018
Instance 4	13215	13410	3,5	**19140**	399	901	0,118	19140	9541	21715	0,076
Instance 5	9820	10290	24,382	**10290**	40	115	0,148	10290	40	115	0,115
Instance 6	2705	2705	0,272	**2705**	12	28	0,0603	2705	12	28	0,0603
Instance 7	5200	12770	1,964	**12770**	141	418	0,176	12770	1634	4284	0,116
Instance 8	3510	3790	0,482	**4070**	15	37	0,076	4070	41	93	0,018
Instance 9	52145	52145	1027,407	**52145**	75	240	1,664	52145	75	240	0,119
Instance 10	48800	68975,5	182,259	**84700**	117	326	0,175	84700	1439	3189	0,012
Instance 11	17350			**30060**	403	1182	2,266	30060	821485	3117437	1,966