

---

# Introduction à la recherche avec tabous

---

Michel Gendreau

CIRRELT et MAGI, École Polytechnique de Montréal

Journées Informatique en Région Centre 2015

Bourges – 28 mai 2015

---

# Plan de la présentation

1. Deux problèmes de référence
2. Motivation et concepts de base
3. Concepts intermédiaires
4. Sujets avancés
5. Conclusion

# Deux problèmes de référence

---

# Problème de confection de tournées de véhicules (CVRP)

Donnée du problème:

1. Graphe orienté  $(V, A)$
2. Sommets: un dépôt et des clients
3. Arcs: déplacements possibles (temps de parcours)
4. Au dépôt, une flotte de  $m$  véhicules de capacité  $Q$ .
5. Les clients ont des demandes  $d_i$ .

# Problème de confection de tournées de véhicules (CVRP) (2)

On cherche un ensemble de routes telles que:

1. Chaque route débute et se termine au dépôt.
2. Chaque client est visité exactement une fois par une seule route.
3. La demande totale des clients affectés à chaque route respecte la capacité des véhicules.
4. La durée totale de chaque route (temps de transport + temps de service) ne dépasse pas une valeur limite spécifiée  $L$ .
5. Le coût total des routes est minimisé.

# Problème de localisation d'usine avec capacités (CPLP)

- $I = \{ \text{clients avec des demandes } d_i \}$
- $J = \{ \text{sites potentiels pour des usines} \}$
- $f_j = \text{coût fixe d'ouverture de l'usine } j$
- $K_j = \text{capacité de l'usine } j$
- $c_{ij} = \text{coût de transport unitaire de } i \text{ à } j$
- On cherche l'ensemble d'usines à ouvrir et le plan de distribution pour réussir à satisfaire la demande des clients à moindre coût.

# Problème de localisation d'usine avec capacités (CPLP) (2)

$$\text{(CPLP) Minimiser } z = \sum_{j \in J} f_j y_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

$$\text{s.l.c.} \quad \sum_{j \in J} x_{ij} = d_i, i \in I$$

$$\sum_{i \in I} x_{ij} \leq K_j y_j, j \in J$$

$$x_{ij} \geq 0, i \in I, j \in J$$

$$y_j \in \{0,1\}, j \in J$$

# Problème de localisation d'usine avec capacités (CPLP) (3)

Pour tout vecteur  $\tilde{\mathbf{y}}$  de variables de localisation, on peut calculer un plan de distribution optimal  $\mathbf{x}(\tilde{\mathbf{y}})$  en résolvant le problème de transport associé:

$$\begin{aligned} \text{(TP)} \quad & \text{Minimize } z(\tilde{\mathbf{y}}) = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \\ & \text{s.t.} \quad \sum_{j \in J} x_{ij} = d_i, \quad i \in I \\ & \quad \sum_{i \in I} x_{ij} \leq K_j \tilde{y}_j, \quad j \in J \\ & \quad x_{ij} \geq 0, \quad i \in I, \quad j \in J \end{aligned}$$

Si  $\tilde{\mathbf{y}} = \mathbf{y}^*$ , vecteur de sites optimal, la solution optimale du problème original de CPLP est donnée par  $(\mathbf{y}^*, \mathbf{x}(\mathbf{y}^*))$ .



# Problème de localisation d'usine avec capacités (CPLP) (4)

Une solution optimale du problème original peut toujours être trouvée en un point extrême du polyèdre des flux de produits défini par:

$$\sum_{j \in J} x_{ij} = d_i, i \in I$$

$$\sum_{i \in I} x_{ij} \leq K_j, j \in J$$

$$x_{ij} \geq 0, i \in I, j \in J$$

On obtient le vecteur  $y^*$  optimal en posant à 1 les  $y_j$  pour lesquels  $\sum_{i \in I} x_{ij} > 0$ , et à 0 sinon.

# Motivation et concepts de base

---

# Un peu d'histoire...

- Les problèmes combinatoires difficiles ont depuis longtemps attiré l'attention des chercheurs (p.e.: commis-voyageur).
- Début des années '70: théorie de la complexité
  - Notion de problèmes NP-difficiles
  - Peu d'espoir de résoudre **efficacement** de nombreux problèmes importants.
  - **QUE FAIRE?!?**
- **UTILISER DES APPROCHES HEURISTIQUES**
  - Heuristiques constructives (p.e., méthodes gloutonnes)
  - Heuristiques d'amélioration itératives

# Méthodes classiques de recherche locale

- Prototype: méthodes d'échange (*r-Opt*, *Or-Opt*)  
(Lin, Or, Lin and Kernighan,...)
- Principe:
  - Partir d'une solution initiale réalisable
  - Appliquer une série de **modifications locales** à la solution, tant que celles-ci améliorent l'objectif.
- Problème:
  - Ces méthodes arrêtent quand elles rencontrent un optimum local.
  - La qualité des solutions (et les temps de calcul) dépend de la « richesse » des transformations permises.

# Un peu plus d'histoire...

- 1983: Kirkpatrick, Gelatt et Vecchi présentent le **recuit simulé**:
  - ❑ une heuristique de recherche locale probabiliste
  - ❑ capable de surmonter les optima locaux
  - ❑ et avec des propriétés de convergence!!!
- Intérêt renouvelé pour le développement de nouveaux types d'heuristiques (on parle souvent de **métaheuristiques**)
- Très souvent basées sur des analogies avec des phénomènes naturels:
  - ❑ recuit simulé (physique des solides)
  - ❑ algorithmes génétiques (évolution des espèces)
  - ❑ méthodes de fourmis
  - ❑ etc.

# La méthode de recherche avec tabous

- Glover (1977, 1986); Hansen (1986: *SAMD*)
- Une métaheuristique qui contrôle une heuristique **interne** définie pour le problème spécifique à traiter.
- Concepts d'intelligence artificielle: maintenir une histoire de la recherche dans des **mémoires**.
- **Principe de base:** permettre dans une recherche locale des mouvements non améliorants pour transcender les optima locaux.
- **PROBLÈME:** comment éviter le cyclage?
- **SOLUTION:** En utilisant l'histoire de la recherche pour éviter les retours en arrière (mouvements tabous)

# Espaces de recherche et voisinages

- Recherche avec tabous: une généralisation des méthodes classiques de recherche locale
- La recherche s'effectue dans un espace de recherche constitué de toutes les solutions possibles du problème
  - **élément-clé de la conception de l'heuristique**
- À chaque itération, les transformations locales permises par l'heuristique interne définissent un **voisinage  $N(S)$**  de la solution courante  $S$ .

# Exemples d'espace de recherche pour le CVRP

- Ensemble des solutions réalisables (naturel)
- Le même ensemble auquel on ajoute des solutions violant les contraintes de capacité ou de durée des tournées.
- Le même ensemble auquel on ajoute des solutions où les clients ne sont pas tous servis.



# Exemples d'espace de recherche pour le CPLP

- Ensemble des solutions réalisables exprimées tant par rapport aux variables  $\mathbf{x}$  que  $\mathbf{y}$ .
- Ensemble des solutions dans l'espace des variables de localisation  $\mathbf{y}$ .
- Ensemble des points extrêmes du polyèdre des flux réalisables  $\mathbf{x}$ .

# Exemples de voisinages pour le CVRP

- Déplacement d'un seul client
  - les insertions peuvent être faites de façon plus ou moins complexe (ex: insertions GENI)
- Échange de clients
- Mouvements et échanges simultanés de clients ( $\lambda$ -interchange d'Osman, 1993)
- Mouvement coordonné de clients entre plusieurs routes (chaînes d'éjection)
- Échange de suites de clients entre deux routes (Cross exchange de Taillard et al., 1997)

# Exemples de voisinages pour le CPLP

- Les voisinages sont contingents de l'espace de recherche choisi.
- Quand on travaille sur les variables de localisation, on peut utiliser des voisinages classiques pour la localisation:
  - ajout/retrait (« add/drop »)
  - échange (« swap »)
- Quand on travaille sur les variables de flux, on peut considérer le voisinage induit par les pivots qu'on peut effectuer à partir du point extrême courant du polyèdre « complet ».

# Les tabous

- Une mémoire à court terme de la recherche (on ne garde en général qu'une quantité fixe – et limitée – d'information)
- Plusieurs possibilités:
  - une liste des dernières solutions visitées (cher et peu fréquent);
  - une liste des dernières modifications effectuées à la solution courante; on interdit les modifications inverses (le type de tabous le plus courant);
  - une liste d'attributs clés des solutions ou des modifications récentes (parfois plus efficace).

# Exemples de tabous

- Supposons que, pour le CVRP, on utilise des mouvements qui consistent à déplacer un client  $i$  d'une route  $R'$  à une route  $R''$ .
- Alors, on pourrait interdire le mouvement inverse et noter cela en mémoire  $(i, R'', R')$ .
- On pourrait interdire tous les mouvements ramenant  $i$  dans la route  $R'$  :  $(i, R')$ .
- Finalement, on pourrait tout simplement interdire tous les mouvements du client  $i$ .

# Plus sur les tabous

- Il est souvent utile d'utiliser plusieurs listes de tabous simultanément.
- Les tabous « simples » peuvent être implantés au moyen de listes circulaires de longueur fixe.
- Les tabous de durée fixe ne peuvent pas toujours prévenir le cyclage: plusieurs auteurs ont proposé diverses techniques pour varier la longueur de la liste des tabous en cours d'exécution (Skorin-Kapov, Taillard).
- Autre solution: les **étiquettes tabous aléatoires**, qui imposent des tabous dont la durée est générée aléatoirement à leur création.
- Autre solution encore: les **tabous activés aléatoirement** (à chaque itération, un nombre aléatoire indique jusqu'où regarder dans la liste des tabous).

# Critères d'aspiration

- Les tabous sont parfois trop puissants:
  - ils interdisent des mouvements intéressants, même s'il n'y a aucun risque de cyclage;
  - ceci peut provoquer une stagnation de la recherche.
- Les critères d'aspiration sont des éléments algorithmiques qui permettent **d'annuler des tabous** dans certaines circonstances.
- Le plus simple des critères d'aspiration consiste à toujours permettre un mouvement qui permet d'atteindre une solution supérieure à la meilleure solution connue.
- Des critères plus compliqués ont été proposés et parfois appliqués.
- **RÈGLE D'OR:** S'il n'y a pas de danger de cyclage, on peut ignorer les tabous!

# Recherche avec tabous « simple »

## ■ Notation

- ❑  $f(\cdot)$ , la fonction que l'on veut maximiser
- ❑  $S$ , la solution courante
- ❑  $S^*$ , la meilleure solution connue,
- ❑  $f^*$ , sa valeur
- ❑  $T$ , la liste des tabous
- ❑  $N(S)$ , le voisinage de  $S$
- ❑  $\bar{N}(S)$ , le voisinage « admissible » de  $S$



# Recherche avec tabous « simple »

## ■ Initialisation

- Choisir (construire) une solution de départ  $S_0$
- Poser  $S := S^* := S_0$ ,  $f^* := f(S_0)$ ,  $T := \emptyset$

## ■ Recherche

- ***Tantque*** critère d'arrêt non satisfait ***faire***
  - $S := \arg \max_{S' \in \overline{N}(S)} [f(S')]$
  - ***si***  $f(S) > f^*$ , ***alors***  $f^* := f(S)$ ,  $S^* := S$ ;
  - enregistrer le tabou pour le mouvement courant;

# Critère d'arrêt

- En théorie, l'algorithme pourrait ne jamais s'arrêter (sauf si valeur optimale était connue a priori).
- En pratique, il faut bien s'arrêter:
  - après un nombre fixe d'itérations ou un temps CPU déterminé;
  - après un certain nombre d'itérations sans amélioration de l'objectif;
  - quand l'objectif atteint une valeur seuil déterminée.
- Dans les implantations plus complexes, l'algorithme comporte souvent un certain nombre de **phases** dont la durée est déterminée par ces critères.

# Recherche avec tabous probabiliste

- Il est souvent avantageux de ne pas explorer tout le voisinage de la solution courante, car celui-ci est trop grand.
- On peut souvent se contenter d'examiner un sous-ensemble aléatoire.
- Ceci agit comme mécanisme anti-cyclage, mais peut faire rater la solution optimale.

# Concepts intermédiaires

# Intensification de la recherche

- ***Idée: Explorer plus attentivement les portions « prometteuses » de l'espace de recherche!***
- De temps à autre, on arrête le processus normal de recherche pour passer à une phase d'intensification.
- Souvent basée sur des mémoires à **moyen terme**
  - p.e, des mémoires qui enregistrent la durée depuis laquelle divers « éléments » sont présents dans la solution courante.

# Intensification de la recherche (2)

- Souvent relancée de la meilleure solution connue.
- Techniques:
  - ❑ « geler » les « bons » éléments de la solution courante,
  - ❑ augmenter de l'échantillon quand on fait de la RT probabiliste,
  - ❑ changer d'algorithme interne ou en modifier les paramètres.

# Diversification de la recherche

- Il est fréquent que le processus de recherche n'explore qu'une partie restreinte de l'espace de recherche.
- On peut ainsi rester loin de l'optimum.
- ***Diversification***: un mécanisme pour « forcer » l'exploration de régions jusqu'alors inexplorées.
- Habituellement basée sur des **mémoires à long terme**
  - **mémoires de fréquence** qui enregistrent la présence de divers « éléments » dans la solution courante.

# Diversification de la recherche (2)

- Techniques:
  - **diversification radicale:** forcer dans la solution quelques éléments peu fréquents et repartir la recherche;
  - **diversification continue:** biaiser l'évaluation des mouvements par l'ajout d'un petit terme correctif relié à la fréquence des éléments;
  - **oscillation stratégique:** prochain sujet.



# Gestion des contraintes

- La prise en compte de toutes les contraintes d'un problème peut sévèrement restreindre le processus de recherche et mener à des solutions médiocres.
- **Il est souvent utile de relaxer des contraintes!**
- Ceci « ouvre » l'espace de recherche et permet souvent l'utilisation de voisinages plus simples.
- Les violations de contraintes sont pondérées et ajoutées à l'objectif.

# Gestion des contraintes (2)

- **Comment trouver les bons poids?!?**
- **On peut utiliser des pénalités auto-ajustantes:**  
les poids sont ajustés dynamiquement en fonction de l'histoire récente de la recherche
  - On augmente les poids quand toutes les solutions récentes sont non réalisables;
  - On diminue les poids dans le cas inverse.
- **Oscillation stratégique:** modifier les poids de façon systématique pour induire de la diversification.

# Fonctions objectifs auxiliaires

- Quand l'évaluation de l'objectif est très coûteuse, on peut évaluer les voisins avec un objectif de remplacement
  - corrélé avec le véritable objectif,
  - moins exigeant à calculer;
  - on ne calcule le véritable objectif que pour un petit nombre de solutions candidates « prometteuses ».
- Dans certains problèmes, la plupart des voisins peuvent avoir une même valeur de l'objectif, comment choisir?
  - En utilisant une fonction objectif auxiliaire qui mesure des attributs désirables des solutions.

---

# Sujets avancés et tendances plus récentes

---

# Parallélisme

- L'utilisation du calcul parallèle ouvre de nombreuses avenues de recherche intéressante pour la RT.
- **Parallélisme bas niveau**
  - utilisation du calcul parallèle pour accélérer les phases plus exigeantes en calcul (évaluation des voisinages)
- **Parallélisme haut niveau**
  - exécution en parallèle de plusieurs trajectoires de recherche qui échangent des informations pour trouver de meilleures solutions
- Chapitre dans le livre d'Enrique Alba (2005).

# Hybrides

- Utilisation de la RT en conjonction avec d'autres méthodes d'optimisation:
  - branch-and-bound (calcul de bornes)
  - algorithmes génétiques, méthodes de fourmis (amélioration des solutions – algo. mémétiques)
  - Programmation par contraintes
  - autres méthodes de recherche local
- Actuellement, les méthodes les plus performantes pour de nombreux problèmes.

# Hybrides (2)

- Deux « architectures »:
  - **Approches unifiées** où on combine dans un même algorithme des composantes de diverses approches.
  - **Hybrides parallèles** qui juxtaposent des processus qui font appel à des implantations « pures » de méthodes différentes pour le même problème.

# Utilisation différente de l'information

- **Idée générale:** tirer plus de l'exploration de l'espace de recherche que les solutions elles-mêmes.
- **Des concepts:**
  - ❑ Reactive Tabu Search
  - ❑ Path relinking
  - ❑ Candidate list and elite solutions
  - ❑ Hashing and Chunking
  - ❑ Vocabulary building



# Des sujets d'application différents

- Programmation en nombres entiers
- Optimisation continue
- Optimisation multi-critère
- Programmation stochastique
- Prise de décision en temps réel

---

# Convergence

- Certains résultats par Hanafi et Glover.
- Pas la préoccupation de l'utilisateur moyen de la RT.

# L'analyse en profondeur

- Nouveau domaine de recherche lancé par J.-P. Watson et ses coauteurs depuis 3-4 ans.
- L'idée n'est plus de mettre au point de façon intuitive et souvent artisanale de nouvelles implantations de RT, mais plutôt de **modéliser** le comportement empirique d'implantations spécifiques pour **comprendre!**

# Conclusion

- La recherche avec tabous a démontré son efficacité sur de nombreux problèmes difficiles au cours des 20 dernières années.
- À l'heure actuelle, elle subit une vive concurrence de nombreuses autres métaheuristiques (l'imagination humaine n'a pas de limites...!), mais demeure très compétitive.
- Cependant, il faut bien connaître et comprendre le problème auquel on s'attaque si l'on veut espérer mettre au point une approche de résolution vraiment efficace.
- La recherche avec tabous, tout comme les autres métaheuristiques, n'est pas une panacée!