

AG41

10/06/2009



# AG41

*Hadrien BEAUFILS*

*Arthur BISCH*

*Hubert LACÔTE*



Help Toody and Muldoon find the shortest round trip route to visit all 33 locations shown on the map.  
All you do is draw connecting straight lines from location to location to show the shortest round trip route.  
**HERE'S THE CORRECT START . . .**  
Begin at Chicago, Illinois. From there, lines show correct route as far as Erie, Pennsylvania. Next, do you go to Carlisle, Pennsylvania or Wana, West Virginia? Check the easy instructions on back of this entry blank for details.

# AG41

- 1. Recherches préliminaires**
- 2. Mise en application de la méthodologie**
- 3. Travail d'optimisation**

# Recherches préliminaires

## 1. Recherches préliminaires

Défauts

Atouts

## 2. Mise en application

Fonctionnement

Application

## 3. Travail d'optimisation

Méthodologie

Profilage

### Le problème de tournées de véhicules avec fenêtres de temps

- Largement traité dans la littérature
- *Vehicle Routing Problem with Time Windows, Part II : Metaheuristics*, Bräysy, Gendreau, 2005

### Méthodologie retenue :

- *A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows*, Taillard et al., 1997
- Fenêtres de temps flexibles

### Défauts de la méthodologie retenue :

- Complexe
- Durée (4 heures)

# Recherches préliminaires

## 1. Recherches préliminaires

Défauts

**Atouts**

## 2. Mise en application

Fonctionnement

Application

## 3. Travail d'optimisation

Méthodologie

Profilage

Avantages de la méthodologie retenue :

- Durée (50 Mhz -> 4 heures, paramétrable)
- Qualité des solutions obtenues selon les paramètres

	20	50	100	Temps (s)
R1	1233,88	1230,48	1220,35	2296 -> 13774
C2	592,75	591,14	590,91	3275 -> 16375

- Temps estimé d'après *Performance of Various Computers Using Standard Linear Equations Software*, J. Dongarra, 2009
  - Sun Sparc 10 de 50 MHz : 23 Mflops/s
  - Pentium 4 de 3.2 Ghz : 873 Mflops/s
- **La méthodologie est applicable** (~= 4500 sec)

# Mise en application de la méthodologie

## 1. Recherches préliminaires

Défauts

Atouts

## 2. Mise en application

Fonctionnement

Application

## 3. Travail d'optimisation

Méthodologie

Profilage

### Algorithme général

**1. Initialisation**  
(solutions mises en mémoire)

**2. Recherche d'une solution  
optimale**  
(utilisation de la mémoire)

**3. Post-optimisation de  
la meilleure solution trouvée**

# Mise en application de la méthodologie

## 1. Recherches préliminaires

Défauts

Atouts

## 2. Mise en application

Fonctionnement

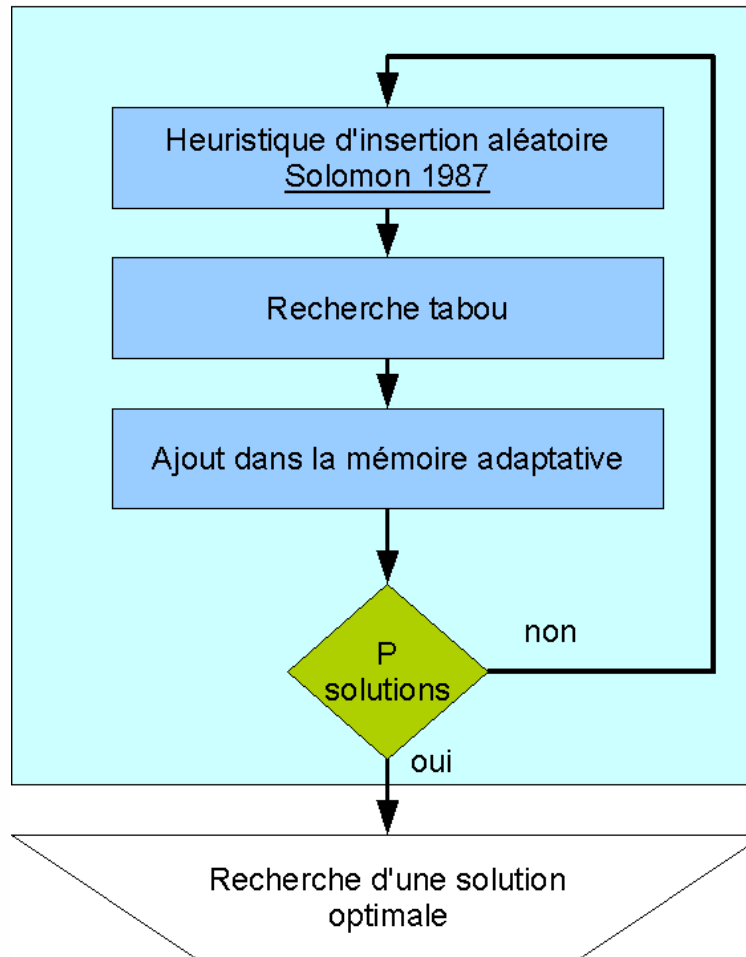
Application

## 3. Travail d'optimisation

Méthodologie

Profilage

### 1. Initialisation



### Mémoire adaptative

- Conserver les routes des meilleures solutions rencontrées
- Explorer toutes les régions de l'espace des solutions

➤ Permet la **diversification**

# Mise en application de la méthodologie

## 1. Recherches préliminaires

Défauts

Atouts

## 2. Mise en application

Fonctionnement

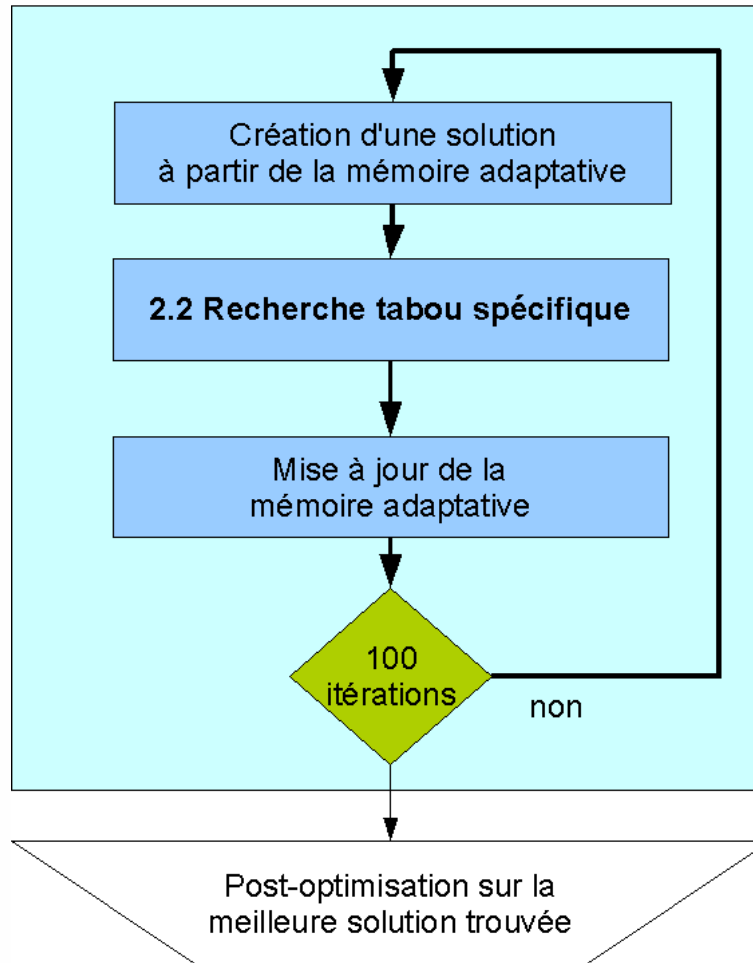
Application

## 3. Travail d'optimisation

Méthodologie

Profilage

## 2. Recherche d'une solution optimale



## Création d'une solution à partir de la mémoire

- Sélectionner par **ranking** une route de la mémoire.
- Enlever toutes les routes non compatibles et recommencer.
- Compléter la solution.

cf. **algorithme génétique**

➤ Croiser plusieurs bonnes solutions -> meilleure solution

# Mise en application de la méthodologie

## 1. Recherches préliminaires

Défauts

Atouts

## 2. Mise en application

Fonctionnement

Application

## 3. Travail d'optimisation

Méthodologie

Profilage

## 2.2. Recherche tabou spécifique

Solution initiale

Décomposition de la solution en C sous-ensembles de routes

Recherche tabou classique

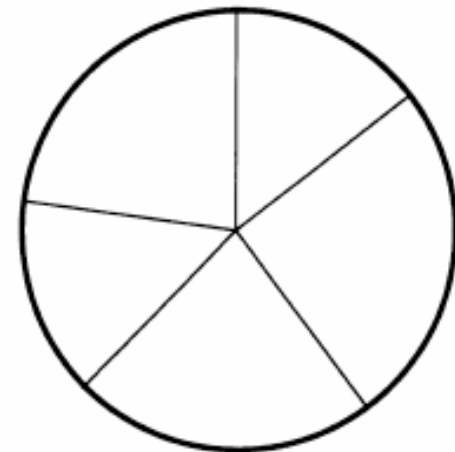
Reconstruction de la solution

I itérations

non

## Décomposition & Reconstruction

Chaque secteur contient approximativement le même nombre de routes.





# Mise en application de la méthodologie

## 1. Recherches préliminaires

Défauts

Atouts

## 2. Mise en application

Fonctionnement

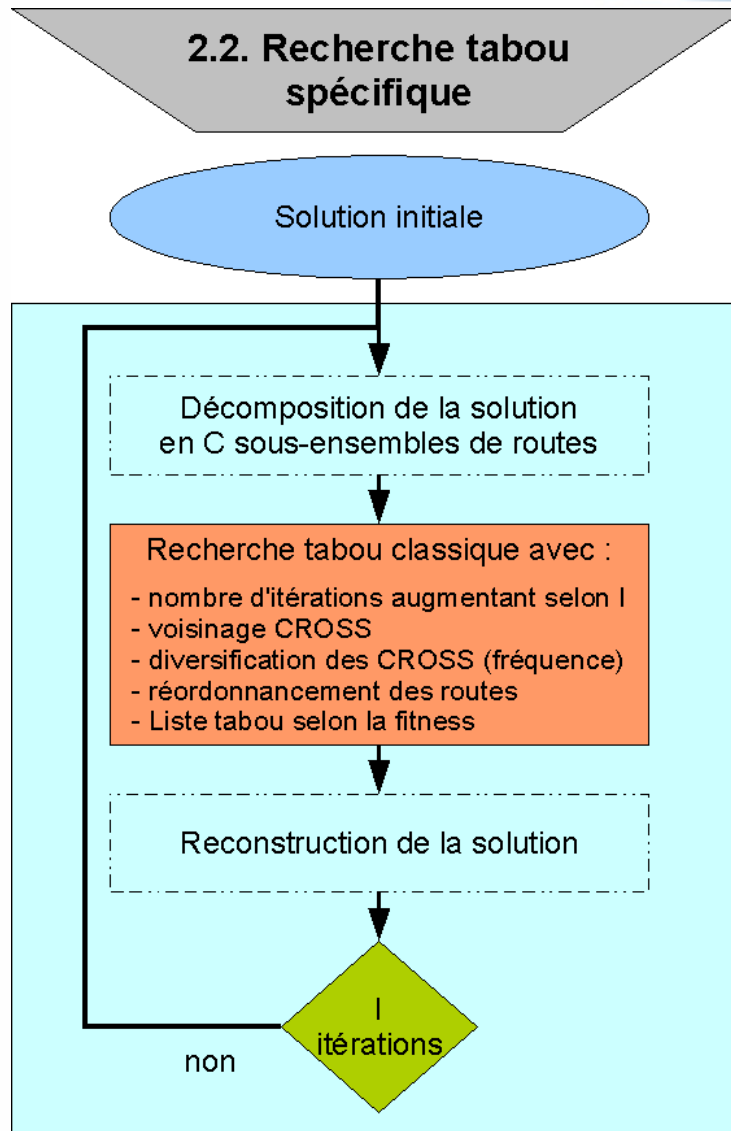
Application

## 3. Travail d'optimisation

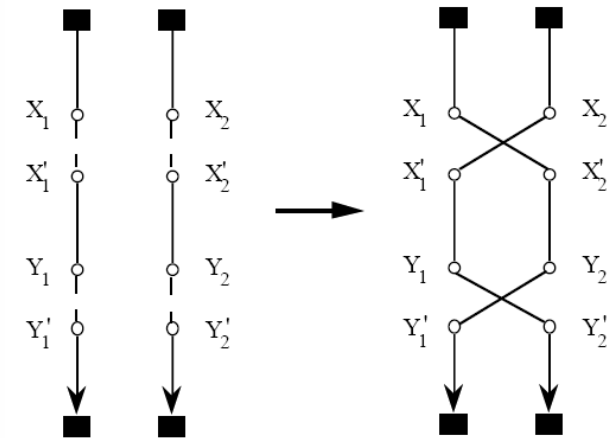
Méthodologie

Profilage

### 2.2. Recherche tabou spécifique



### Voisinage



Voisinage très large.

➤ Réduire le temps de calcul :  
Évaluation avec approximation

### Réordonnancement

**Intensification**

# Mise en application de la méthodologie

## 1. Recherches préliminaires

Défauts

Atouts

## 2. Mise en application

Fonctionnement

**Application**

## 3. Travail d'optimisation

Méthodologie

Profilage

### Application

Choix du langage C++ pour l'implémentation

UML : Diagramme de classes

Résultats ( 20 itérations ) : meilleurs résultats de 5 runs

Type \ Nb cons.	<b>25</b>	<b>50</b>	<b>100</b>
<b>R101</b>	622.5 <i>2.24s</i>	1034.6 <i>13.09s</i>	1681 <i>65.05s</i>
<b>R112</b>	400.3 <i>0.61s</i>	650.5 <i>7.67s</i>	1064.8 <i>57.77s</i>
<b>C208</b>	243.1 <i>0.39s</i>	473.4 <i>5.84s</i>	672.6 <i>50.8s</i>

# Travail d'optimisation

## 1. Recherches préliminaires

Défauts

Atouts

## 2. Mise en application

Fonctionnement

Application

## 3. Travail d'optimisation

**Méthodologie**

Profilage

### Méthodologie :

Partir d'un programme fonctionnel : utilisation d'un algorithme performant

Déterminer et analyser les sections critiques (cf. Profilage)

Optimiser le code : procédures pour éviter de répéter un calcul plusieurs fois, etc.

# Travail d'optimisation

## 1. Recherches préliminaires

Défauts

Atouts

## 2. Mise en application

Fonctionnement

Application

## 3. Travail d'optimisation

Méthodologie

**Profilage**

### Profilage

Classe de profilage de notre programme : analyse des temps d'exécution

```
----- PROFILER -----  
+Tout      - executed during 25.094 s  
0.1.initialisation - executed during 0 s  
0.2.decompositionAndTabu - executed during 25.064 s  
1.decomposition - executed during 0 s  
1.recomposition - executed during 0 s  
2.tabou - executed during 25.064 s  
3.2.isAllowedToCross - executed during 3.21 s  
3.3.CROSS exchange - executed during 9.54096 s  
3.4.Fitness - executed during 2.87 s  
3.5.createNeighborFromSolution - executed during 0.016 s  
3.voisinage - executed during 24.8121 s  
-----
```

Points critiques après analyse : decomposition et tabou, vérification des capacités, opérateur CROSS

# Conclusion

## 1. Recherches préliminaires

Défauts

Atouts

## 2. Mise en application

Fonctionnement

Application

## 3. Travail d'optimisation

Méthodologie

Profilage

### Améliorations à apporter :

- Implémentation du CROSS intra-route
- Limitation de la longueur des segments échangés pour le CROSS à L (L=5 ou L=7)
- Évaluation du voisinage en temps constant (approximations)
- Paramètres s'adaptant automatiquement au type de fichier (R, C, RC ?) pour exploiter pleinement le temps imparti