# Problème du placement géographique de labels

Auteur : Grégory Burri
Prof. responsable : Éric Taillard

École d'ingénieurs du canton de Vaud eivd
Route de Cheseaux 1
CH-1400 Yverdon-les-bains, Suisse

## Sommaire

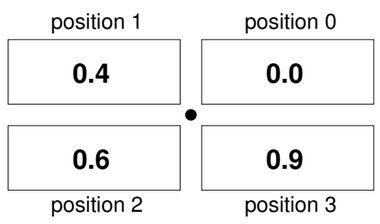

– Introduction ;
– Recherche avec tabous ;
  – Approche ;
  – Étalonnage des constantes ;
  – Complexité.
– POPMUSIC+TS ;
  – Approche ;
  – Modification ;
  – Complexité.
– Comparaison entre la recherche avec tabous et POPMUSIC+TS ;
– Comparaison de POPMUSIC+TS avec d'autres méthodes ;
– Pondération des labels ;
– Démonstration ;
– Conclusion.

# Introduction

Le but est de disposer des légendes autour de points de manière à minimiser le coût.

$$\sum_{i=0}^{n-1} C\left(i\right)$$

Le nombre de positions du label autour de son point est fini. Quatre positions sont utilisées par la suite.



## La recherche avec tabous (1)

1. Créer une configuration initiale.
2. Pour un certain nombre d'itérations, effectuer.
   (a) Créer la liste des candidats.
   (b) Pour chaque candidat calculer sa meilleure position.
   (c) Prendre le candidat le meilleur, c'est à dire celui qui dégrade le moins ou qui améliore le plus la solution. Si il n'améliore pas la solution et se trouve dans la liste tabou alors on passe à l'itération suivante.
   (d) Ajouter le candidat à la liste tabou si il ne s'y trouve pas déjà.
   (e) Mettre à jour la solution courante en effectuant la modification.
3. FIN.

## La recherche avec tabous (2)

Le calcul du coût d'un label.

$$C(i) = \alpha_1 \cdot nbChevauchement(i) + \alpha_2 \cdot préférence(i)$$

La taille de la liste tabou est calculée dynamiquement.

$$taille = min_t + int(p_t \cdot nLCh)$$

Une liste de candidats, elle contient les labels qui coûtes le plus. Sa taille est calculée dynamiquement.

$$taille = min_c + int(p_c \cdot nLch)$$

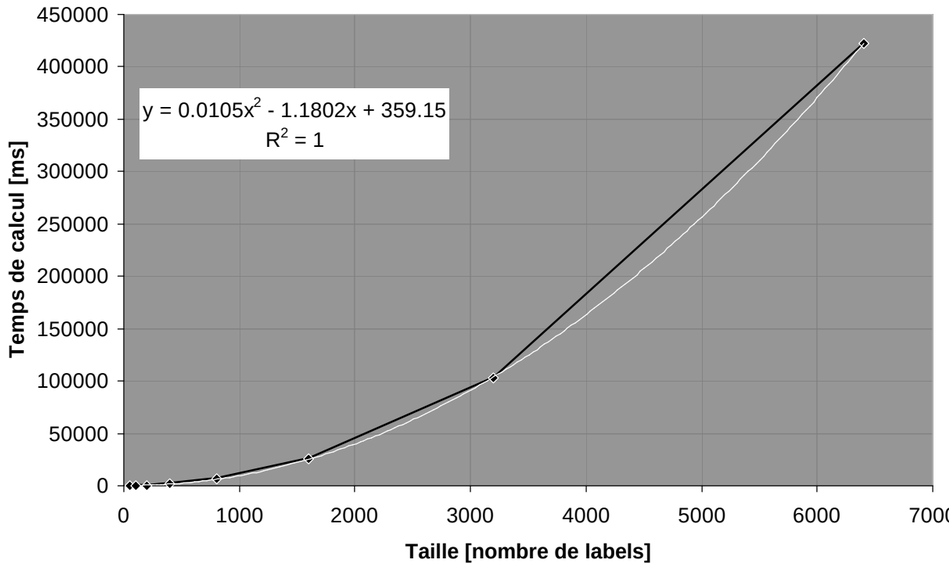
## La recherche avec tabous (3)

Étalonnage des constantes, utilisation d'une méthode d'optimisation.

- $p_{base}$ est le facteur minimum de pondération de la taille de la liste des candidats, il est le $p_c$ initial $[0.1, 0.8]$ ;
- $f_r$ est le facteur de réduction du facteur $p_c$ $[1.1, 1.5]$ ;
- $f_a$ est le facteur d'agrandissement du facteur $p_c$ $[5, 20]$ ;
- $p_t$ est le facteur de pondération de la taille de la liste tabou $[0.1, 0.8]$ ;
- $m$ définit la fréquence de recalcul de la taille des listes tabous et de candidats $[10, 50]$ ;
- $min_c$ est la taille minimum de la liste des candidats $[2, 20]$ ;
- $min_t$ est la taille minimum de la liste tabou $[2, 10]$.

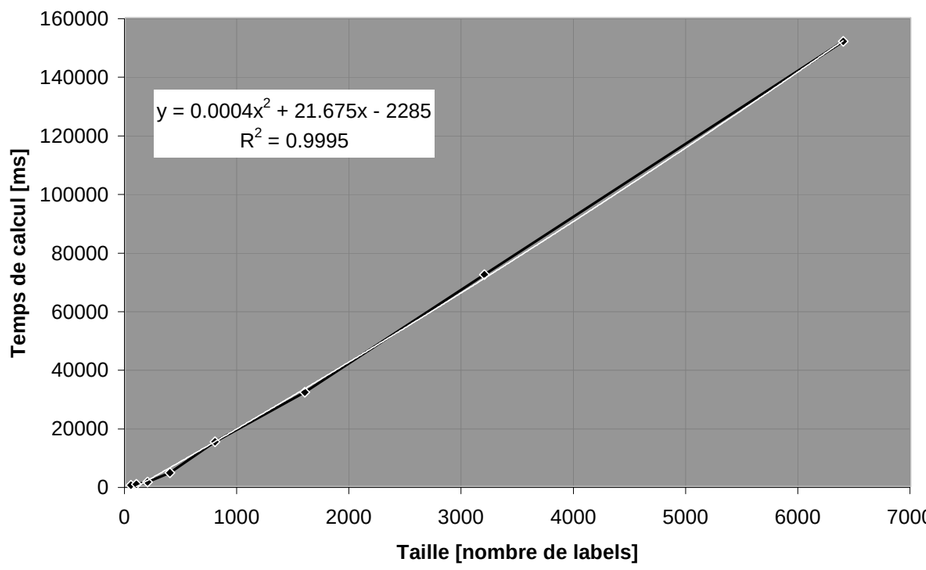
# La recherche avec tabous (4)

Phase d'initialisation, création de la structure de données représentant tous les chevauchements possibles.
Sa complexité est en $O(n^2)$.



# La recherche avec tabous (5)

Dans le cas d'une densité constante de label, la complexité est en $n \cdot log(n)$.

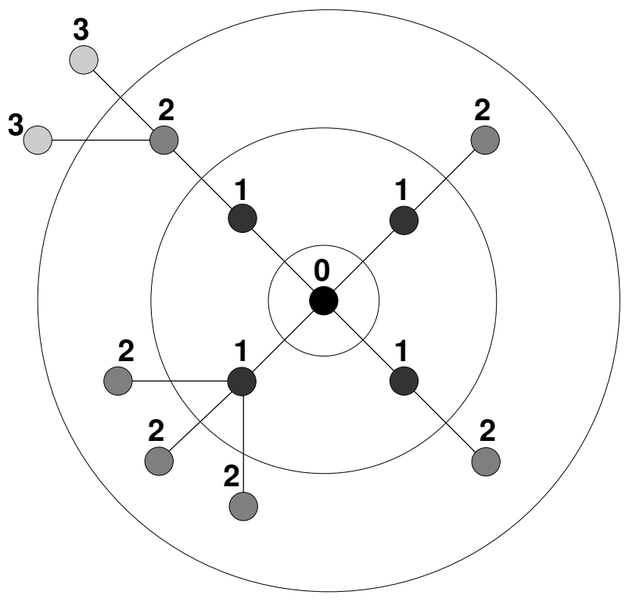


## POPMUSIC+TS (1)

1. $O$ est un ensemble de points de la solution, initialement vide.
2. $r$ est la taille maximale d'un sous-problème.
3. Tant que $O$ ne contient pas tous les points, répéter.
   (a) Choisir un point de façon aléatoire qui ne se trouve pas dans $O$.
   (b) Créer un sous-problème dont la taille n'excède pas $r$ à partir du point choisi.
   (c) Appliquer la recherche avec tabous sur le sous-ensemble.
   (d) Si la solution globale est meilleure alors on la met à jour avec les changements effectués sur le sous-ensemble et on vide $O$, si elle est moins bonne ou égale alors on ajoute le point choisi à $O$.
4. FIN.

# POPMUSIC+TS (2)

Création des sous-problèmes.



## POPMUSIC+TS (3)

Modification de POPMUSIC.

Au lieu d'ajouter uniquement le point qui a servi de germe on ajoute tout le sous-ensemble à $O$. Cette modification, appelée POPMUSIC+TS version 2, donne des résultats un peu moins bon que la version initiale mais en des temps nettement meilleurs. Ceci se verra lors de la comparaison avec d'autres méthodes.

## POPMUSIC+TS (4)

Etalonnage des constantes de la recherche avec tabous pour POPMUSIC+TS.

Paramètres définis de manière empirique :

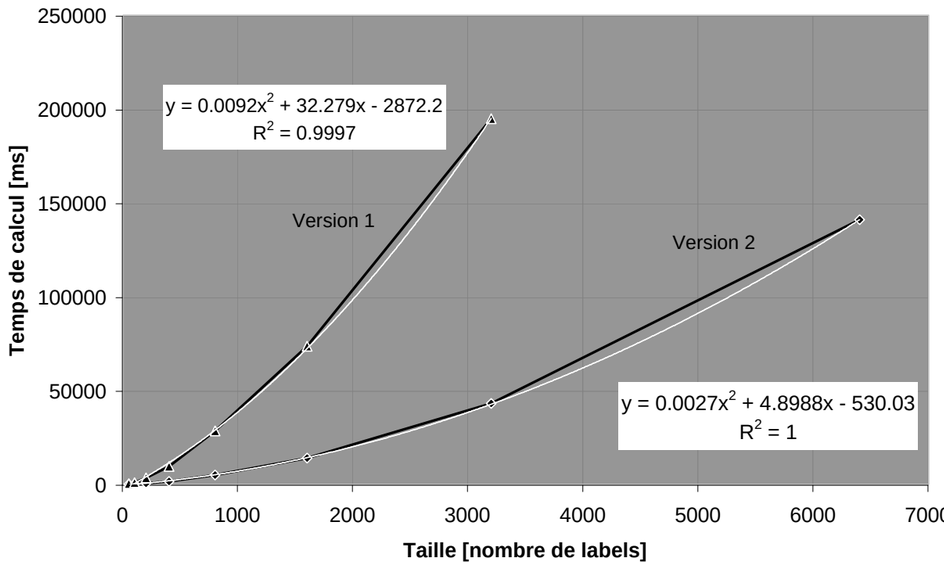
- $nombre\ itération = 70$
- $taille\ sous\ problème = 75$

Utilisation de la méthode vu précédement.

- $p_{base}$ : 0.73 ;
- $f_r$ : 1.3 ;
- $f_a$ : 15 ;
- $p_t$ : 0.77 ;
- $m$ : 47 ;
- $min_c$ : 18 ;
- $min_t$ : 9

# POPMUSIC+TS (5)

La complexité de POPMUSIC+TS est en $Omega(n)$.

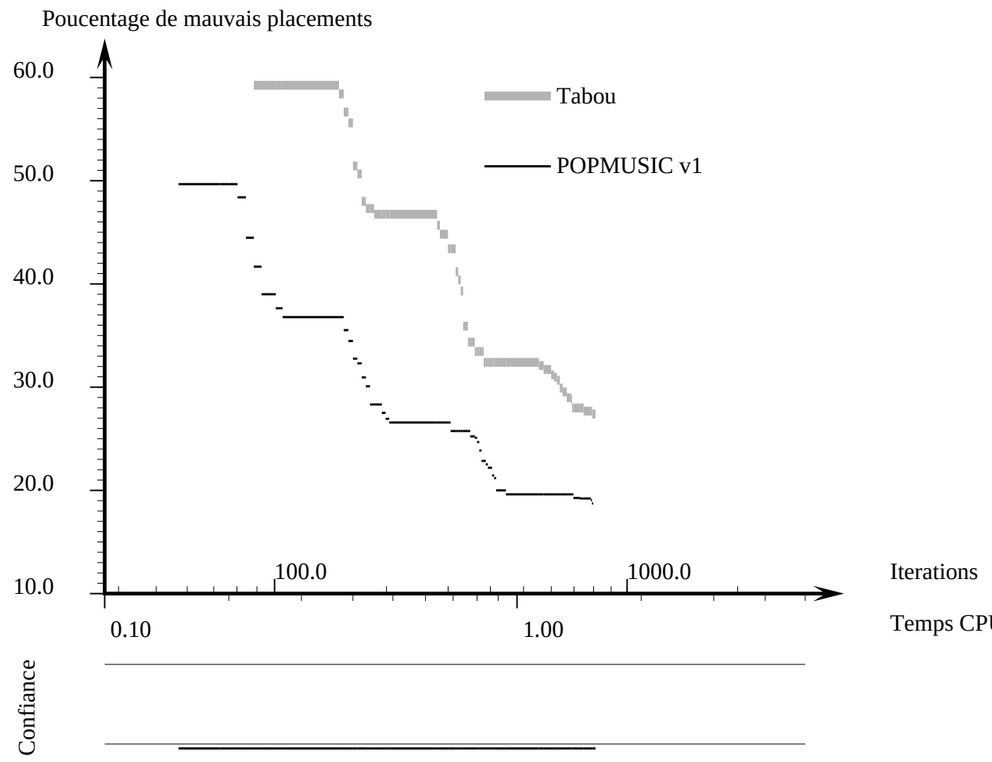


## Comparaison entre la recherche avec tabous et POPMUSIC+TS (1)

Elle est effectuée avec un outil de comparaison de méthodes itératives non-détérministes :
*http ://ina.eivd.ch/collaborateurs/etd/codes.dir/comparaison.dir/comparaison.htm.*

- Le nombre d'itération va de 10 à 2'560 par pas de facteur 2.
- chaque calcul est reproduit 15 fois avec des configurations initiales différentes.
- La taille des problèmes est de 500.

# Comparaison entre la recherche avec tabous et POPMUSIC+TS (2)



# Comparaison de POPMUSIC+TS avec d'autres approches de la littérature (1)

Qualité de la solution (pourcentage de bon placement).

| **Algorithme** | **100** | **250** | **500** | **750** | **1000** |
|---|---|---|---|---|---|
| POPMUSIC + TS version 1 | 100.0 | 100.0 | 99.6 | 97.4 | 92.3 |
| POPMUSIC + TS version 2 | 100.0 | 100.0 | 99.5 | 97.2 | 91.6 |
| $\text{CGA}_{best}$ | 100.00 | 100.00 | 99.6 | 97.1 | 90.7 |
| $\text{CGA}_{average}$ | 100.00 | 100.00 | 99.6 | 96.8 | 90.4 |
| Tabu search | 100.00 | 100.00 | 99.2 | 96.8 | 90.00 |
| GA with masking | 100.00 | 99.98 | 98.79 | 95.99 | 88.96 |
| GA | 100.00 | 98.40 | 92.59 | 82.38 | 65.70 |
| Simulated Annealing | 100.00 | 99.90 | 98.30 | 92.30 | 82.09 |
| Zoraster | 100.00 | 99.79 | 96.21 | 79.78 | 53.06 |
| Hirsh | 100.00 | 99.58 | 95.70 | 82.04 | 60.24 |
| 3-Opt Gradient Descent | 100.00 | 99.76 | 97.34 | 89.44 | 77.83 |
| 2-Opt Gradient Descent | 100.00 | 99.36 | 95.62 | 85.60 | 73.37 |
| Gradient Descent | 98.64 | 95.47 | 86.46 | 72.40 | 58.29 |
| Greedy | 95.12 | 88.82 | 75.15 | 58.57 | 43.41 |

## Comparaison de POPMUSIC+TS avec d'autres approches de la littérature (2)

Vitesse d'exécution en seconde.

| **Algorithme** | **100** | **250** | **500** | **750** | **1000** |
|---|---|---|---|---|---|
| POPMUSIC + TS version 1 | 0.0 | 0.0 | 0.3 | 3.5 | 20.0 |
| POPMUSIC + TS version 2 | 0.0 | 0.0 | 0.2 | 1.3 | 4.4 |
| $\text{CGA}_{best}$ | 0 | 0.6 | 21.5 | 228.9 | 1227.2 |
| $\text{CGA}_{average}$ | 0 | 0.6 | 21.5 | 195.9 | 981.8 |
| Tabu search | 0 | 0 | 1.3 | 76.0 | 352.9 |

## Pondération des labels

Effectuée comme une deuxième passe après POPMUSIC+TS.

1. Trier les étiquettes par ordre de poids, du plus grand au plus faible.
2. Pour chaque étiquette triée.
   (a) elle n'est pas masquée alors masquer toutes les étiquettes qui la chevauchent.
3. FIN.

# Démonstration

## Conclusion

La méthode POPMUSIC+TS appliquée au problème du placement de labels donne de très bons résultats, que ce soit en terme de qualité de la solution qu'en terme de vitesse d'exécution.

Le projet entier peut être téléchargé ici :
*http ://www.euphorik.ch/placement_labels*