Méthodes pour les problèmes structurés

Pierre Bonami

M111/M06

30 janvier 2012

Logiciels de résolutions génériques pour la programmation en nombres entiers

- En 20 ans les logiciels de résolution ont été accélérés par un facteur de 1 000 000 000 en moyenne :
  - 1000 : vitesse des processeurs.
  - $\times$ 1000 : codes de programmation linéaires.
  - $\times$ 1000 : méthodes de plans coupants, heuristiques, branchement...
- Un problème résolu en 1 seconde aujourd'hui aurait pris 31 ans avec la technologie de 1990.
- En pratique, on peut résoudre des problèmes ayant des millions de variables et de contraintes.
- Bien sûr, il existe aussi des problèmes de petite taille non résolus.

Évolution de CPLEX

TABLE: Temps de calcul de 12 versions de CPLEX sur un ensemble de 1,734 instances de MIP. Temps normalizés avec CPLEX 11.0. Source [Lodi 2010, Achterberg et Bixby 2009]

version	année	better	worse	temps
11.0	2007	–	–	1.00
10.0	2005	201	650	1.91
9.0	2003	142	793	2.73
8.0	2002	117	856	3.56
7.1	2001	63	930	4.59
6.5	1999	71	997	**7.47**
6.0	1998	55	1060	21.30
5.0	1997	45	1069	22.57
4.0	1995	37	1089	26.29
3.0	1994	34	1107	34.63
2.1	1993	13	1137	56.16
1.2	1991	17	1132	67.90

Évolution de CPLEX

TABLE: Temps de calcul de 12 versions de CPLEX sur un ensemble de 1,734 instances de MIP. Temps normalizés avec CPLEX 11.0. Source [Lodi 2010, Achterberg et Bixby 2009]

version	année	better	worse	temps
11.0	2007	–	–	1.00
10.0	2005	201	650	1.91
9.0	2003	142	793	2.73
8.0	2002	117	856	3.56
7.1	2001	63	930	4.59
6.5	1999	71	997	**7.47**
6.0	1998	55	1060	21.30
5.0	1997	45	1069	22.57
4.0	1995	37	1089	26.29
3.0	1994	34	1107	34.63
2.1	1993	13	1137	56.16
1.2	1991	17	1132	67.90

Coupes de Gomory

Apport de chacune des composantes

TABLE: Variation du temps de calcul avec CPLEX 8 sur un ensemble de 978 instances de MIP en enlevant chaque technique sparment [Bixby, Fenelon, Gu, Rothberg, Wunderling 2005]

Technique	Dégradation
Toutes les coupes	53,7
Coupes Gomory Mixtes	2.52
Coupes MIR	1.83
Coupes Knapsack Cover	1.40
Coupes Disjonctives	1.02
Presolve	10.8
Heuristiques	1.4
Dived Probing	1.1

Problèmes en nombres entiers structurés

Dans certains problmes la matrice A qui définit les contrainte a une structure bien précise :

- Localisation d'entrepôts (Uncapacitated Facility Location).
- Sac à dos
- Voyageur de Commerce
- ...

Dans ces problèmes, peut-on utiliser les structures combinatoires pour améliorer les méthodes génériques ?

(Rappel) Bonnes formulations

Pour Localisation d'entrepôts, formulation avec contraintes désagrégées :

$$
\begin{aligned}
\max \ & \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{j=1}^{n} f_j y_j \\
& \sum_{j=1}^{n} x_{ij} = 1 && i = 1, \ldots, m \\
& x_{ij} \leq y_j && i = 1, \ldots, m \ nj = 1, \ldots, n \\
& y \in \{0, 1\}^n, \ y \geq 0
\end{aligned}
$$

Problème de sac à dos

Formulation

Étant donnés n objet de poids $a_i \geq 0$ et d'utilité $c_i \geq 0$. Quels objets choisir de manière à ne pas dépasser le poids b et à maximiser la somme des utilités :

$$\begin{aligned} \max \; & \sum_{i=1}^{n} c_i x_i \\ & \sum_{i=1}^{n} a_i x_i \leq b \\ & x_i \in \{0, 1\} \qquad i = 1, \ldots, n \end{aligned}$$

Peut-on trouver des inégalités valides pour renforcer la formulation de ce problème ?

Note

Le problème peut être résolu en temps pseudo-polynomial par la programmation dynamique.

Variante difficile : Exploration pétrolière

Une compagnie pétrolière veut forrer n localisation potentielles pour trouver des nouveaux puits de pétroles. Pour chaque localisation, elle dispose d'une estimation de succès c_i, d'une estimation du coût de forage p_i, du nombre de ressources humaines nécessaires h_i et de ressources techniques (machines) t_i. Quelles localisations explorer pour maximiser la probabilité de succés en ne dépensant pas plus de b, en utilisant pas plus de H personnes et T machines ?

$$
\begin{aligned}
\max \sum_{i=1}^{n} c_i x_i & \\
\sum_{i=1}^{n} p_i x_i \leq b & \\
\sum_{i=1}^{n} h_i x_i \leq H & \\
\sum_{i=1}^{n} t_i x_i \leq T & \\
x_i \in \{0,1\} & \qquad i = 1, \ldots, n
\end{aligned}
$$

Inégalités de Cover

On considère l'ensemble des solution d'une contrainte de sac-à-dòs

$$X = \{x \in \{0,1\}^n : \sum_{i=1}^{n} a_i x_i \leq b\}$$

Soit C un sous ensemble de variables telles que $\sum_{i \in C} a_i > b$, alors l'inégalité :

$$\sum_{i \in C} x_i \leq |C| - 1$$

est valide pour X.
On appelle C une cover.

Exemple

$$X = \{x \in \{0,1\}^7 : 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19$$

Les sous-ensembles suivants sont des covers :

- $\{1,2,3\}$
- $\{1,2,6\}$
- $\{1,5,6\}$
- $\{3,4,5,6\}$

Le vecteur $(1,1,\frac{1}{3},0,0,0,0)$ vérifie l'inégalité définissant X mais pas la coupe induite par la cover $\{1,2,3\}$.

Cover minimale

- Soit C une cover et C' tel que $C \subseteq C'$.
- Alors C' est une cover.
- Tout $x \in [0,1]^n$ satisfaisant $\sum_{i \in C} x_i \leq |C| - 1$ satisfait $\sum_{i \in C'} x_i \leq |C'| - 1$.
- Conséquence : il est inutile d'ajouter l'inégalité de cover correspondant à C', on peut se content des covers minimales par inclusion :

$$C \text{ tel que } \sum_{i \in C} a_i > b \text{ et pour tout } i \in C, \sum_{i \in C \setminus \{i\}} a_i \leq b$$

Séparation des inégalités de cover

Soit $\hat{x} \in [0,1]^n$ peut on trouver une cover C telle que l'inégalité de cover correspondant soit violée par $\hat{x}$?
Ou encore :
Existe-t-il C tel que $\sum_{i \in C} a_i > b$ et $\sum_{i \in C} \hat{x}_i > |C| - 1$?

Re-formulation

On peut réecrire l'inégalité $\sum_{i \in C} x_i \leq |C| - 1$ comme $\sum_{i \in C} (1 - x_i) \geq 1$. Le problème se reformule alors comme

$$\min_C \{ \sum_{i \in C} (1 - \hat{x}_i) : \sum_{i \in C} a_i > b \} < 1?$$

Si oui, la solution optimale donne une cover.

Séparation des inégalités de cover II

Le problème

$$\min_{C}\{\sum_{i\in C}(1-\hat{x}_i) : \sum_{i\in C} a_i > b\}$$

se reformule comme un problème en nombres entiers :

$$\begin{aligned} \min \ & \sum_{i=1}^{n}(1-\hat{x}_i)y_i \\ & \sum_{i=1}^{n} a_i y_i \geq b+1 \\ & y \in \{0,1\}^n \end{aligned}$$

Remarque

- Ce problème est à nouveau un problème de sac à dos !
- On peut en réduire sa taille en éliminant tous les i tels que $\hat{x}_i = 0$ (si $\hat{x}_i = 0$ on sait que si $i \in C$ l'inégalité de Cover correspondante n'est pas violée).

Renforcement des inégalités de cover

Une fois une inégalité de Cover séparée on peut la rendre plus forte par des arguments simples.

Cover Étendue

Soit C une cover. On définit l'extention de C $E(C)$ comme

$$E(C) = C \cup \{i : a_i \geq a_j \text{pour tout } j \in C\}$$

L'inégalité de cover peut être renforcée en utilisant $E(C)$:

$$\sum_{i \in E(C)} x_i \leq |C| - 1$$

Exemple

$$X = \{x \in \{0,1\}^7 : 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_2 + 4x_6 + x_7 \leq 19$$

Les sous-ensembles suivants sont des covers :

- C=$\{3,4,5,6\}$: $E(C) = \{1, 23, 4, 5, 6\}$ induit l'inégalité :

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$$

qui est strictement meilleure que

$$x_3 + x_4 + x_5 + x_6 \leq 3$$

Lifting des inégalités de Cover

Étant donnée une Cover C, on cherche les valeurs α_i pour $i \notin C$ les meilleurs pour que l'inégalité

$$\sum_{i \in C} x_i + \sum_{i \notin C} \alpha_i x_i \leq |C| - 1$$

On suppose que les variables $\notin C$ sont ordonnées $i_1, \ldots, i_{n-|C|}$

Lifting séquentielle

A litération t on a une inégalité valide $\sum_{i \in C} x_i + \sum_{j=1}^{t-1} \alpha_{i_j} x_{i_j} \leq |C| - 1$.

Pour calculer α_t on résoud

$$\xi_t = \min \sum_{i \in C} x_i + \sum_{j=1}^{t-1} \alpha_{i_j} x_{i_j}$$

$$\sum_{i \in C} a_i x_i + \sum_{j=1}^{t-1} a_{i_j} x_{i_j} \leq b - a_{i_t}$$

$$x \in \{0,1\}^{|C|+t-1}$$

On prend $\alpha_{i_t} = |C| - 1 - \xi_t$

Algorithme de plans coupants utilisant les cover

1. $\mathcal{C} \leftarrow \emptyset$
2. Résoudre la relaxation continue du problème

$$
\begin{aligned}
\max \ & \sum_{i=1}^{n} c_i x_i & \\
& \sum_{i=1}^{n} p_i x_i \leq b, \ \sum_{i=1}^{n} h_i x_i \leq H, \ \sum_{i=1}^{n} t_i x_i \leq T & \\
& \sum_{i \in E(C)} x_i \leq |C| - 1 & \forall C \in \mathcal{C} \\
& x_i \in [0,1] & i = 1, \ldots, n
\end{aligned}
$$

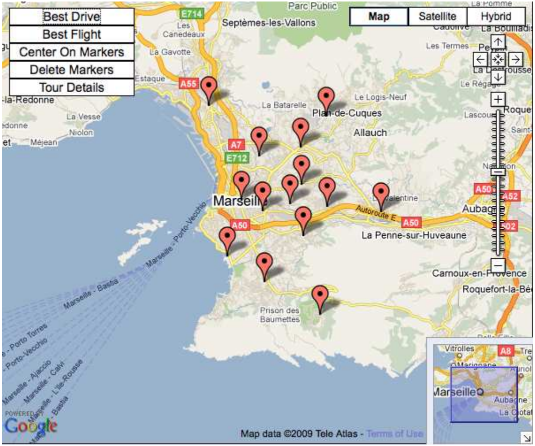
   Soit $\hat{x}$ la solution.

3. Pour chacune des contraintes de ressources résoudre le problème de séparation des cover.
4. Si une ou plusieurs cover C sont trouvées les ajouter à $\mathcal{C}$ et aller en 2. Sinon brancher.

Voyageur de commerce

Formulation

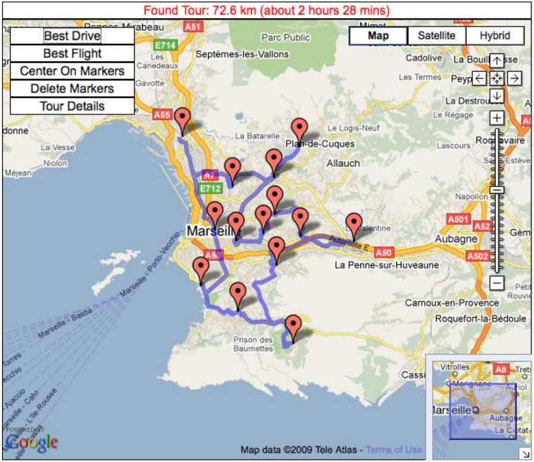
Étant données n points, trouver un tour entre les point de longueur minimale.



Voyageur de commerce

Formulation

Étant données n points, trouver un tour entre les point de longueur minimale[1].



[1]Tour calculé sur http://www.tsp.gatech.edu/maps/index.html

Formulation mathématique

Soit $G = (V, E)$ le graphe complet, on définit les

$$\text{variable} x_e = \begin{cases} 1 & \text{si on utilise l'arète } e = (uv). \\ 0 & \text{sinon.} \end{cases}$$

$$\min \sum_{e \in E} d_e x_e$$

$$\text{t.q. :}$$

$$\sum_{e=(uv)} x_e = 2 \qquad \forall v \in V$$

$$\sum_{e=(uv): u \in S, v \notin S} x_e \geq 2 \quad \forall S \subseteq V$$

$$x_e \in \{0, 1\} \qquad e \in E$$

Formulation mathématique

Soit $G = (V, E)$ le graphe complet, on définit les variable$x_e = \begin{cases} 1 & \text{si on utilise l'arète } e = (uv). \\ 0 & \text{sinon.} \end{cases}$

$$\min \sum_{e \in E} d_e x_e$$

$$\text{t.q. :}$$

$$\sum_{e=(uv)} x_e = 2 \qquad \forall v \in V$$

$$\sum_{e=(uv):u\in S, v\notin S} x_e \geq 2 \qquad \forall S \subseteq V$$

$$x_e \in \{0, 1\} \qquad e \in E$$

Nombre exponentiels d'inégalités de sous tours : Plans Coupants

Relaxation contraintes de degrès

On résoud la relaxation continue avec uniquement les contraintes de degrès :

$$\min \sum_{e \in E}^{n} d_e x_e$$

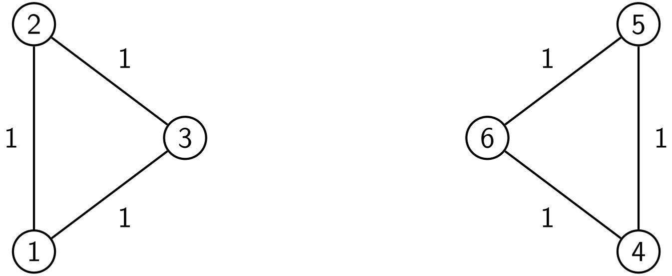
t.q. :

$$\sum_{e=(uv)}^{n} x_e = 2 \quad \forall v \in V$$

$$x_e \in \{0, 1\} \qquad e \in E$$

Une solution de la relaxation

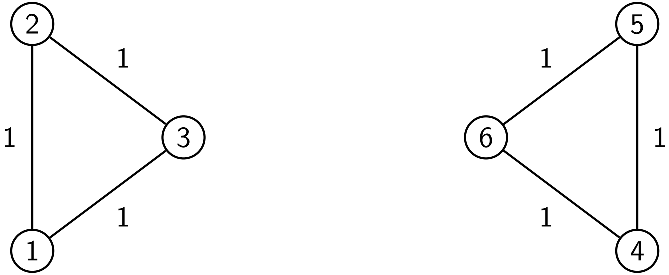
(Sont dessinées les composantes non-nulles de la relaxations)



Trouver S tel que $\sum_{e=(uv):u \in S, v \notin S} x_e < 2$?

Une solution de la relaxation

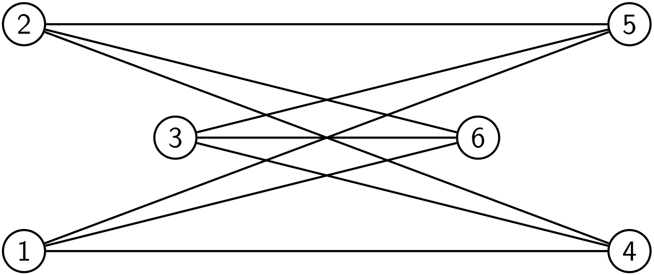
(Sont dessinées les composantes non-nulles de la relaxations)



Trouver S tel que $\sum_{e=(uv):u\in S, v\notin S} x_e < 2$? $S = \{1, 2, 3\}$

Une solution de la relaxation

(Sont dessinées les composantes non-nulles de la relaxations)



≥ 2

Trouver S tel que $\sum_{e=(uv):u\in S, v\notin S} x_e < 2$? $S = \{1, 2, 3\}$ Donne l'inégalité :

$$x_{15} + x_{16} + x_{17} + x_{25} + x_{26} + x_{27} + x_{35} + x_{36} + x_{37} \geq 2$$

Problème de la coupe de poids minimum dans un graphe

Étant donné un graphe $G = (V, E)$ et des poids sur les arètes, trouver un sous ensemble des sommets S tel que la somme des poids des arètes avec une extrémité dans S et une extrémité dans $V \setminus S$ est minimale.

- Problème pouvant être résolu en temps polynomial.
- Algorithme de base : Ford-Fulkerson (Flot maximal, coupe min)
- Algorithme efficace : Gomory-Hu

Séparation des inégalité de sous-tours

- Soit $\hat{x}$ une solution satisfaisant les contraintes de degrés.
- Construire le graphe complet avec les arêtes valuées par $\hat{x}$.
- Chercher une coupe de poids minimum.
- Si cette coupe a un poids < 2, on peut ajouter l'inégalité de sous-tour correspondante :

$$\sum_{i \in S} \sum_{j \notin S} x_{ij} \geq 2$$

- Sinon toutes les inégalités de sous-tours sont satisfaites par $\hat{x}$

Algorithmes de plans coupant pour les sous-tours

1. $\mathcal{S} \leftarrow \emptyset$
2. Résoudre la relaxation :

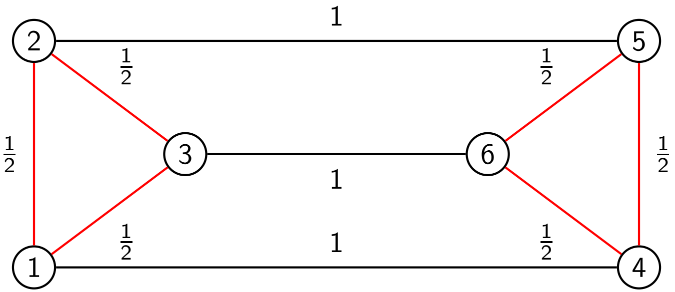
$$\begin{aligned} \min & \sum_{e \in E} d_e x_e \\ & \text{t.q. :} \\ & \sum_{e=(uv)} x_e = 2 && \forall v \in V \\ & \sum_{e=(uv):u \in S, v \notin S} x_e \geq 2 && \forall S \in \mathcal{S} \\ & x_e \in [0,1] && e \in E \end{aligned}$$

   Soit $\hat{x}$ la solution.
3. Résoudre le problème de coupe min avec les valuations $\hat{x}$. Soit S donnant la coupe min.
4. Si la coupe à un poids < 2 FIN. Sinon, ajouter S à $\mathcal{S}$ et aller en 2.

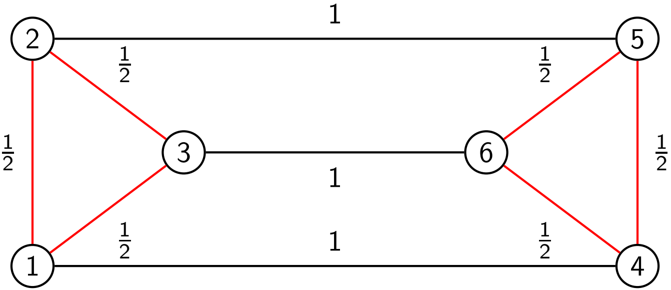
Coupes additionelles

- A la fin de l'algorithmes de plans coupants pour les sous-tours, la solution n'est typiquement pas entière.
- On doit donc soit brancher, soit ajouter de nouvelle inégalités valides.
- De nombreuses classes d'inégalités valides ont été dérivées.

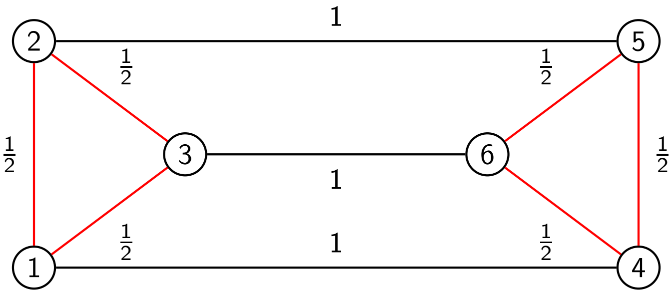
Exemple



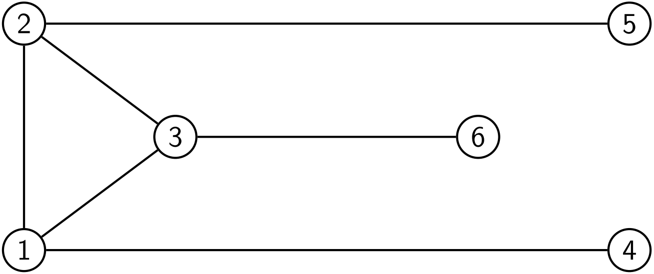
Inégalité 2-Matching



Inégalité 2-Matching



Inégalité 2-Matching



≤ 4

Justification : cette inégalité est une coupe de Chvátal-Gomory

Dérivation

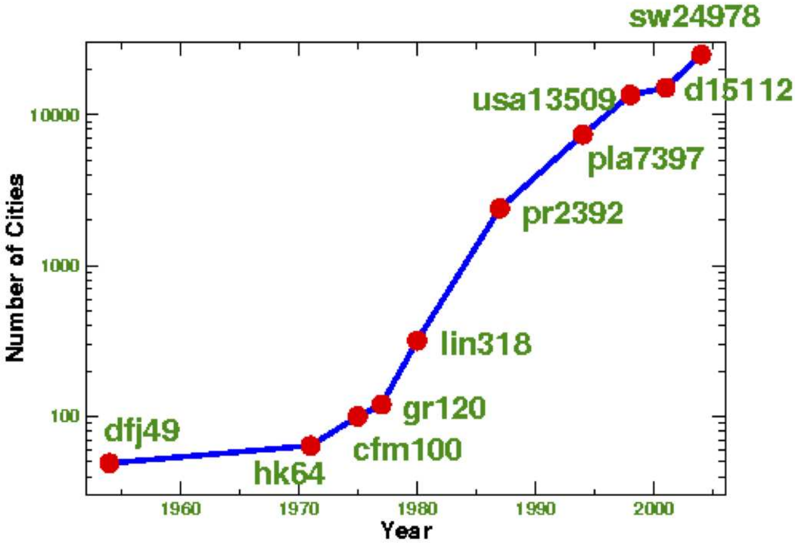
$$\frac{1}{2}(x_{12} + x_{13} + x_{14} + x_{15} + x_{16} \leq 2)$$
$$\frac{1}{2}(x_{12} + x_{23} + x_{24} + x_{25} + x_{26} \leq 2)$$
$$\frac{1}{2}(x_{13} + x_{23} + x_{34} + x_{35} + x_{36} \leq 2)$$
$$\frac{1}{2}(x_{14} \leq 1)$$
$$\frac{1}{2}(x_{25} \leq 1)$$
$$\frac{1}{2}(x_{36} \leq 1)$$
$$\frac{1}{2}(-x_{16} - x_{15} - x_{24} - x_{26} - x_{34} - x_{35} \leq 0)$$

$$x_{12} + x_{13} + x_{13} + x_{14} + x_{25} + x_{36} \leq \lfloor 3 + \frac{3}{2} \rfloor = 4$$

Remarques

- Cette inégalité est une coupe de Chvátal-Gomory mais il existe des algorithmes de séparation efficaces pour cette sous classe.
- De nombreuses généralisations. . .

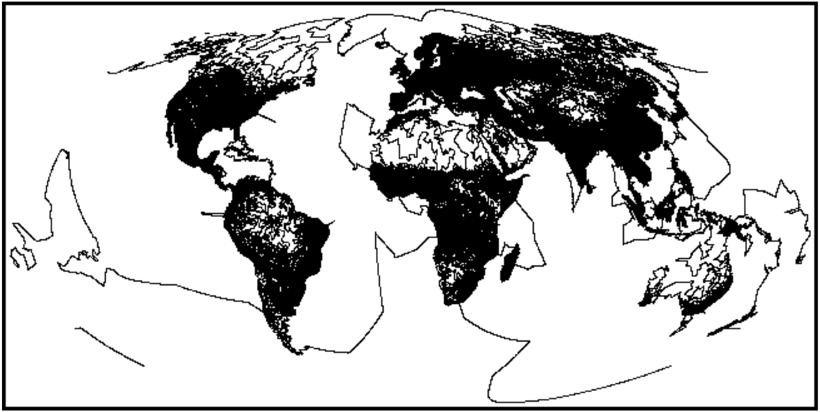
Évolution des résolution des TSP [2]



[2] http ://www.tsp.gatech.edu/

Aujourd'hui

1 904 711 villes du monde :



Meilleure solution : 7 515 778 188 (avril 2011, heuristique)
Meilleure borne : 7 512 218 268 (juin 2007, calculé par concorde)