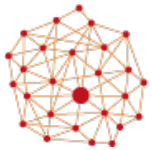


LocalSolver

Calcul de bornes inférieures pour les problèmes de tournées dans LocalSolver

Alexandre Bontems

www.localsolver.com

ROADEF 2020

Tour rapide de LocalSolver

- Solveur “model & run”
  - Outils de modélisation simples
  - Pas besoin de paramétrage
- Variables de décision riches (bool, int, float, set, list)
- Grand nombre d’opérateurs (min, max, sum, prod, comparaisons, etc.)
- Combine plusieurs techniques d’optimisation
  - NLP/MINLP
  - LP/MILP
  - Recherche locale
  - CP/SAT

Problèmes de tournées dans LocalSolver

Variable de décision structurée: `list(n)`

- Sous-ensemble ordonné de valeurs dans $\{0, \ldots, n-1\}$
- Chaque valeur est unique dans la liste

Pour interagir avec la liste :

- `count(l)`: Nombre d'éléments dans la liste
- `at(l, i)`: Valeur à l'indice i dans la liste
- `indexOf(l, v)`: Indice de la valeur v dans la liste
- `contains(l, v)`: Vrai si v est dans la liste

Modélisation du Voyageur de Commerce

TSP.lsp

```
nodes <- list(nbNodes);

constraint count(nodes) == nbNodes;

minimize sum(1..nbNodes-1, i => distance[nodes[i-
    1]][nodes[i]]) + distance[nodes[nbNodes-1][nodes[0]];
```

Modélisation du Voyageur de Commerce

TSP.lsp

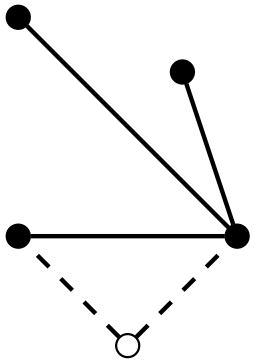
```
nodes <- list(nbNodes);

constraint count(nodes) == nbNodes;

minimize sum(1..nbNodes-1, i => distance[nodes[i-
    1]][nodes[i]]) + distance[nodes[nbNodes-1][nodes[0]];
```

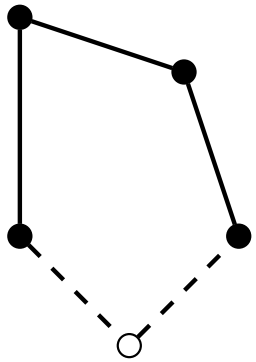
- Comment calculer des bornes pour ce modèle ?

Relaxation des 1-trees

1-tree de coût minimum

- MST dans $V \setminus \{i\}$ + deux arêtes incidentes à i de coût minimum
- Une tournée est un 1-tree avec $\delta(i) = 2,\ \forall i \in V$





Algorithme de sous-gradient

Relaxation lagrangienne (Held and Karp, 1970)

- Perturbation des coûts : $\bar{c}_{ij} = c_{ij} + \pi_i + \pi_j$
- Borne inférieure : $w(\pi) = c(T) - 2\sum_{i \in V} \pi_i$

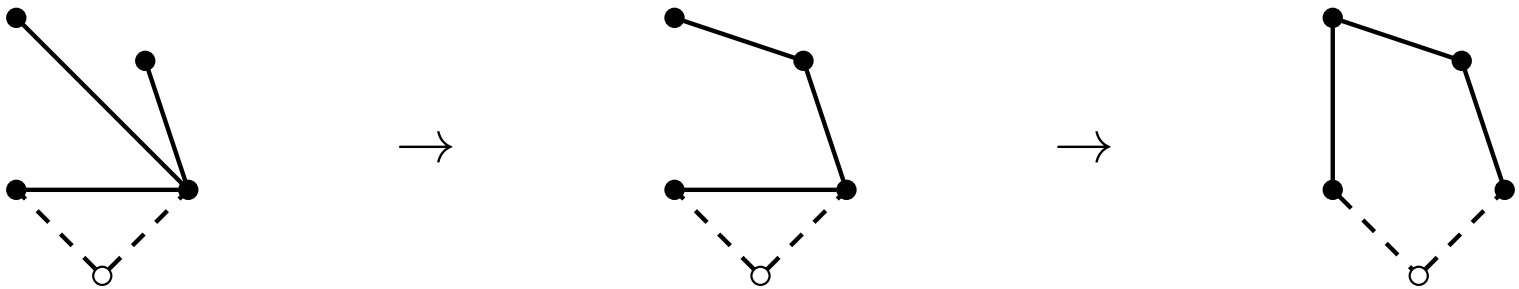
Algorithme de sous-gradient

Relaxation lagrangienne (Held and Karp, 1970)

- Perturbation des coûts : $\bar{c}_{ij} = c_{ij} + \pi_i + \pi_j$
- Borne inférieure : $w(\pi) = c(T) - 2\sum_{i \in V} \pi_i$

Algorithme

- Trouver $\max w(\pi)$
- Calculer une série de 1-tree de coût min.
- Mettre à jour les poids : $\pi_i^{m+1} = \pi_i^m + t^m(\delta(i)^m - 2)$



Amélioration de l'algorithme

Itérations sur sous-graphe (Valenzuela and Jones, 1997)

- Itération sur le sous-graphe des plus proches voisins
- Calcul de la borne finale sur le graphe complet
- d1291 $\rightarrow$ de 13.380s à 0.989s

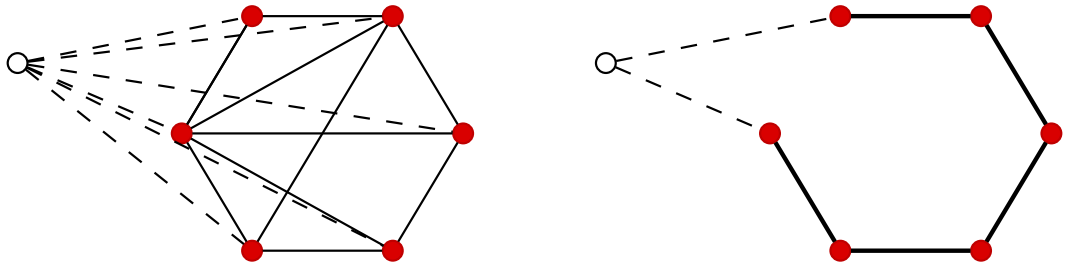
Amélioration de l'algorithme

Itérations sur sous-graphe (Valenzuela and Jones, 1997)

- Itération sur le sous-graphe des plus proches voisins
- Calcul de la borne finale sur le graphe complet
- d1291 $\rightarrow$ de 13.380s à 0.989s

Borne pour le chemin hamiltonien de coût minimum

- Ajout d'un sommet fictif f avec $c_{if} = 0,\ \forall i \in V$



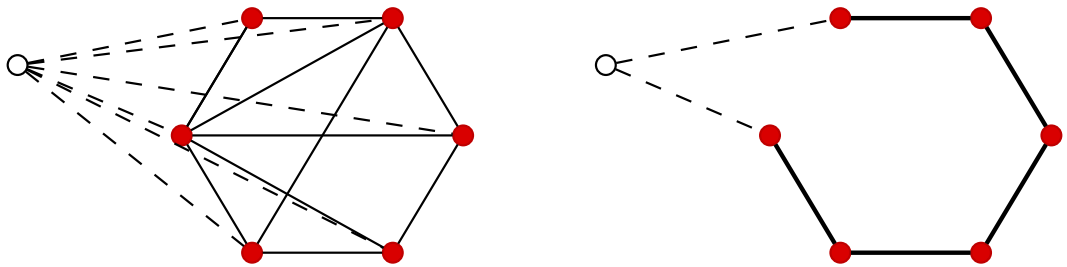
Amélioration de l'algorithme

Itérations sur sous-graphe (Valenzuela and Jones, 1997)

- Itération sur le sous-graphe des plus proches voisins
- Calcul de la borne finale sur le graphe complet
- d1291 $\rightarrow$ de 13.380s à 0.989s

Borne pour le chemin hamiltonien de coût minimum

- Ajout d'un sommet fictif f avec $c_{if} = 0,\ \forall i \in V$



Cas asymétrique

- Deux jeux de poids pour chaque sommet

Results

Instance	Opt	Bound	Gap (%)	Time (secs)
gr48	5 046	4 950	1.9	0.020
st70	675	670	0.7	0.027
ch130	6 110	6 065	0.7	0.054
si175	21 407	21 321	0.4	0.130
a280	2 579	2 562	0.6	0.155
rat575	6 773	6 713	0.8	0.353
d1291	50 801	49 136	3.2	0.989
br17	39	39	0.0	0.005
ftv70	1 950	1 906	2.2	0.037
kro124p	36 230	35 924	0.8	0.053
ftv170	2 755	2 703	1.8	0.072

Extension de la méthode au CVRP (Fisher, 1994)

Capacitated Vechicle Routing Problem

- Calcul de K-trees $\rightarrow$ arbre couvrant avec K + n arêtes
- Dualisation des contraintes de degré
- Dualisation des contraintes de capacité
  - Ajout dynamique de variables duales correspondants aux contraintes de capacité violées par un K-tree
- Algorithme pour calculer un K-tree au degré contraint

Utilisation des solutions primales issues du solveur

Programme linéaire des sous-tours

$$(P) \quad \begin{cases} \min & \displaystyle\sum_{(i,j) \in E} x_{ij} c_{ij} & \\ \text{s.t.} & \displaystyle\sum_{(i,j) \in \delta(S)} x_{ij} \geq 2 & \forall S \subset V,\ S \neq \emptyset \\ & 0 \leq x_{ij} \leq 1 & \forall (i,j) \in E \end{cases}$$

- Contrainte serrée $\rightarrow$ sommets adjacents dans la solution optimale

Utilisation des solutions primales issues du solveur

Programme linéaire des sous-tours

$$(P) \quad \begin{cases} \min & \displaystyle\sum_{(i,j) \in E} x_{ij} c_{ij} & \\ \text{s.t.} & \displaystyle\sum_{(i,j) \in \delta(S)} x_{ij} \geq 2 & \forall S \subset V,\ S \neq \emptyset \\ & 0 \leq x_{ij} \leq 1 & \forall (i,j) \in E \end{cases}$$

- Contrainte serrée $\rightarrow$ sommets adjacents dans la solution optimale

Première approche

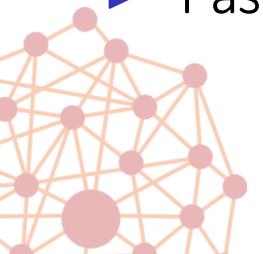
- Ajouter les contraintes correspondants à des sommets adjacents dans une solution primale
- Limite sur la taille des sous-ensembles de sommets considérés

Utilisation des solutions primales issues du solveur

Résultats

Instance	Opt	Bound	Gap(%)	Time(secs)
gr48	5 046	4 769	5.4	0.025
st70	675	629	6.8	0.058
ch130	6 110	5 582	8.6	0.211
si175	21 407	21 140	1.2	0.417
a280	2 579	2 541	1.4	0.965
rat575	6 773	6 669	1.5	4.165
d1291	50 801	47 337	6.8	23.618
pr2392	378 032	359 672	4.8	236.016

- Passage à l'échelle problématique
- Pas de gain de performance par rapport à l'algorithme de sous-gradient



Utilisation des solutions primales issues du solveur

Programme dual

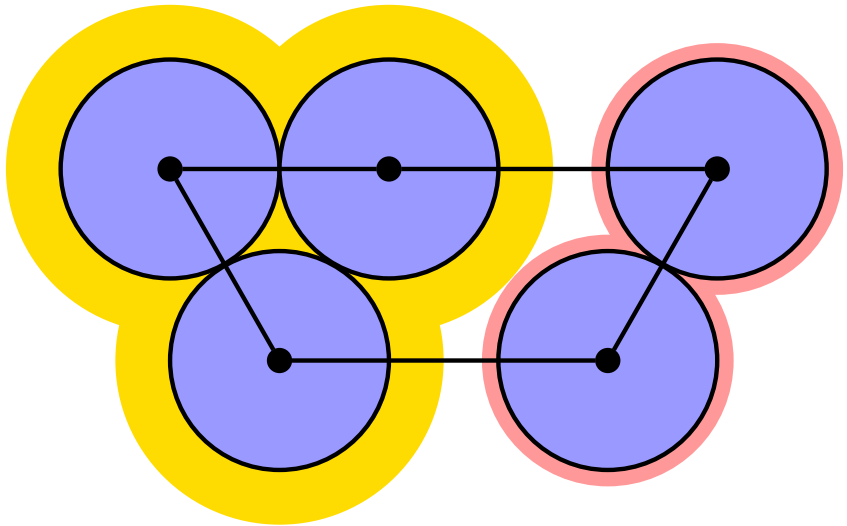
$$(D) \quad \begin{cases} \max & \displaystyle\sum_{S \subset V,\, S \neq \emptyset} 2\pi_S & \\ \text{s.c.} & \displaystyle\sum_{S:(i,j) \in \delta(S)} \pi_S \leq c_{ij} & \forall (i,j) \in E \\ & \pi_S \geq 0 & \forall S \subset V,\, S \neq \emptyset \end{cases}$$

- $\pi_S^* > 0 \rightarrow$ contrainte serrée dans P
- $x_{ij}^* > 0 \rightarrow$ contrainte serrée dans D

Zones de contrôle (Cook et al., 1998)

Programme dual

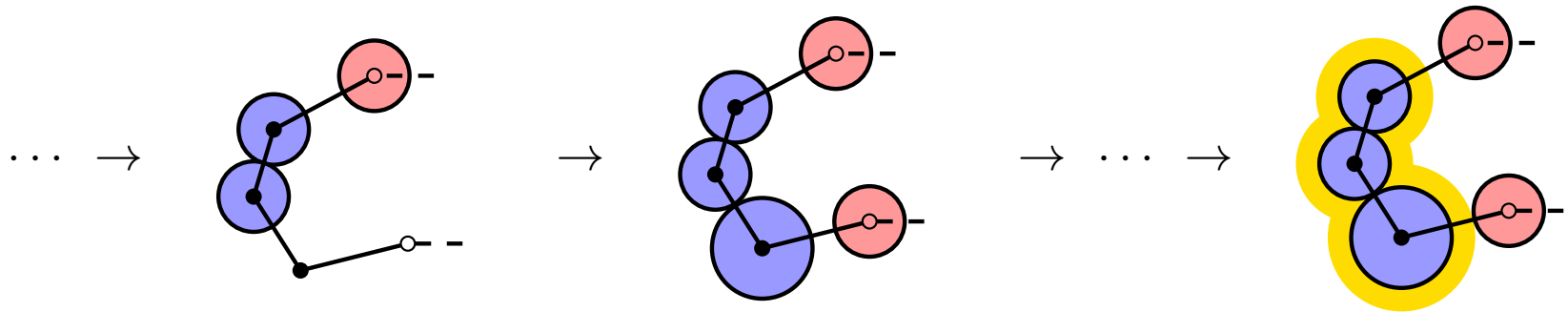
- Chaque tournée doit passer par les zones de contrôle de chaque sous-ensemble de sommets
- Les zones de contrôles correspondent aux contraintes serrées dans P



Un simple algorithme glouton

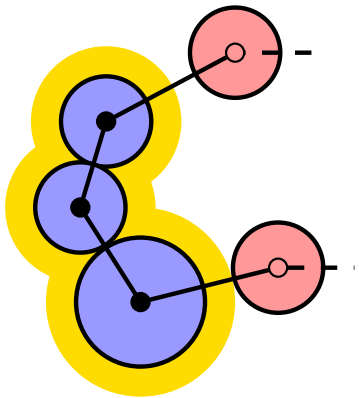
- Itérer sur les arêtes d'une solution primale
- Augmenter la valeur des variables duales π_S de manière gloutonne

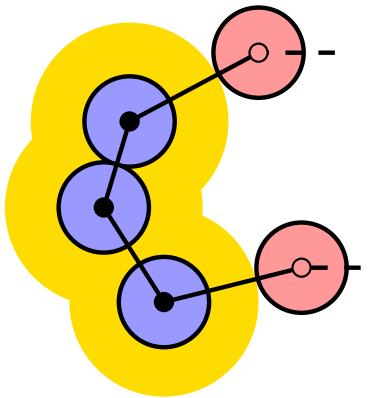
Exemple d'exécution



Résultats

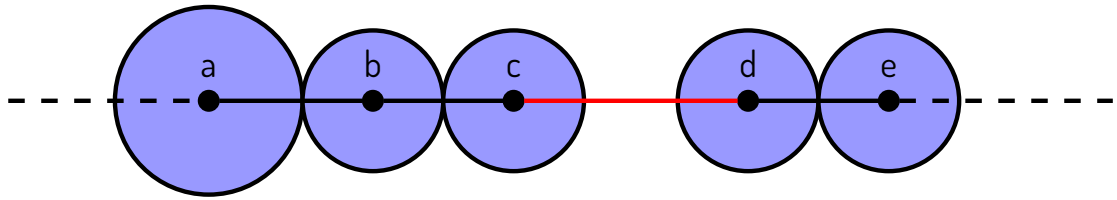
Instance	Opt	Sol. value	Bound	Gap (%)	Time (secs)
gr48	5 046	5 046	4 466	11.4	0.0001
st70	675	677	597	11.5	0.0001
ch130	6 110	6 161	5 453	10.7	0.002
si175	21 407	21 414	20 829	2.7	0.003
a280	2 579	2 586	2 459	4.6	0.008
rat575	6 773	6 902	6 346	6.3	0.044
d1291	50 801	52 792	45 120	11.1	0.251
pr2392	378 032	396 320	345 604	8.5	1.026





Optimisation des variables π_S

- Résolution itérative de sous-problème du dual
- Sous-problème initial contient tous les $\pi_{\{i\}}$
- Arête non saturée $\rightarrow$ Ajouter des variables π_S au sous-problème.



- Limite sur le nombre de variables qui impliquent un sommet donné

Résultats

Instance	Opt	Sol. value	Bound	Gap (%)	Time (secs)
gr48	5 046	5 046	4 819	4.4	0.035
st70	675	677	628	6.9	0.090
ch130	6 110	6 161	6 039	1.1	0.418
si175	21 407	21 414	21 185	1.0	0.775
a280	2 579	2 586	2 555	0.9	1.238
rat575	6 773	6 902	6 644	1.9	12.909
d1291	50 801	52 792	48 229	5.0	58.016
pr2392	378 032	396 320	363 462	3.8	359.282

- Meilleurs résultats mais passage à l'échelle toujours problématique

Conclusion

- Calcul automatique de bornes pour le TSP en LS 9.0+
- Future intégration de la méthode Held-Karp pour le CVRP

Conclusion

- Calcul automatique de bornes pour le TSP en LS 9.0+
- Future intégration de la méthode Held-Karp pour le CVRP

Références

William J. Cook et al. Combinatorial Optimization. 1998.

Marshall L. Fisher. "Optimal Solution of Vehicle Routing Problems Using Minimum K-Trees". In: Operations Research 42.4 (1994), pp. 626–642.

Michael Held and Richard M. Karp. "The Traveling-Salesman Problem and Minimum Spanning Trees". In: Operations Research 18.6 (1970), pp. 1138–1162.

Christine L. Valenzuela and Antonia J. Jones. "Estimating the Held-Karp lower bound for the geometric TSP". In: European Journal of Operational Research 102.1 (1997), pp. 157–175.