



LocalSolver 9.5: Nouveautés et améliorations de performances

Julien Darlay

jdarlay@localsolver.com

www.localsolver.com

ROADEF 2020
Montpellier

Optimisation & Aide à la Décision

- optimisation de production
- tournées de véhicules
- planification de projets
- revenue management
- planification de personnel
- conception de réseaux
- etc.

EDITEUR DE LOGICIEL

SOCIÉTÉ DE SERVICES



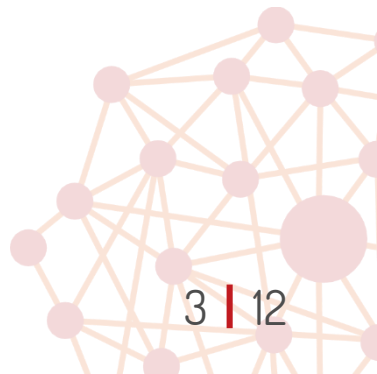
LocalSolver

Global Optimization

Nonlinear & Set-Based Modeling

Heuristics & Exact technics

Fast & Scalable



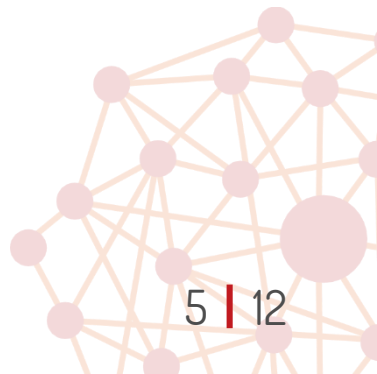
Exemple optimisation de forme

```
#  
# Declares the optimization model  
#  
m = ls.model  
  
# Numerical decisions  
R = m.float(0,1)  
r = m.float(0,1)  
h = m.float(0,1)  
  
# Surface must not exceed the surface of the plain disc  
surface = PI * r**2 + PI * (R+r) * m.sqrt((R - r)**2 + h**2)  
m.constraint(surface <= PI)  
  
# Maximize the volume  
volume = PI * h/3 * (R**2 + R*r + r**2)  
m.maximize(volume)  
  
m.close()  
  
ls.solve()
```

```
Windows PowerShell  
PS C:\localsolver_9_5\examples\optimal_bucket> python .\optimal_bucket.py  
  
Model:  expressions = 28, decisions = 3, constraints = 1, objectives = 1  
Param:  time limit = 2 sec, no iteration limit, nb threads = auto  
  
[objective direction ]:  maximize  
  
[ 0 sec, 0 itr]: 0  
[ optimality gap ]: 100%  
[ 0 sec, 40000 itr]: 0.68709  
[ optimality gap ]: < 0.01%  
  
40000 iterations performed in 0 seconds  
  
Optimal solution:  
obj = 0.68709  
gap = < 0.01%  
bounds = 0.687188  
PS C:\localsolver_9_5\examples\optimal_bucket>
```

Amélioration de performances

Voisinages larges



Ordonnancement

Difficultés (8.5)

- Contraintes de précédences + disjonction
- Pas de voisinages locaux valides

Modélisation mixte

- Liste: séquence des opérations sur une machine
- Entiers: date de début de chaque tâche sur chaque machine

Réparation par propagation sur les entiers

Gains

- Modèles plus concis (JobShop / 4)
- < 4% de l'optimum en 60s (Fisher et Thompson)
- Utile sur d'autres problèmes (packing 3D)

```
function model() {  
  // Integer decisions: start time of each activity  
  // start[j][m] is the start time of the activity of job j which is processed on machine m  
  start[j in 0..nbJobs-1][m in 0..nbMachines-1] <- int(0, maxStart);  
  end[j in 0..nbJobs-1][m in 0..nbMachines-1] <- start[j][m] + processingTime[j][m];  
  
  // Precedence constraints between the activities of a job  
  for [j in 0..nbJobs-1][k in 1..nbMachines-1]  
    constraint start[j][machineOrder[j][k]] >= end[j][machineOrder[j][k-1]];  
  
  // Sequence of activities on each machine  
  jobsOrder[m in 0..nbMachines-1] <- list(nbJobs);  
  
  for [m in 0..nbMachines-1] {  
    // Each job has an activity scheduled on each machine  
    constraint count(jobsOrder[m]) == nbJobs;  
  
    // Disjunctive resource constraints between the activities on a machine  
    constraint and(0..nbJobs-2, i => start[jobsOrder[m][i+1]][m] >= end[jobsOrder[m][i]][m]);  
  }  
  
  // Minimize the makespan: end of the last activity of the last job  
  makespan <- max[j in 0..nbJobs-1] (end[j][machineOrder[j][nbMachines-1]]);  
  minimize makespan;  
}
```

Léa Blaise

Réparation de solutions par propagation de réseaux d'inégalités dans LocalSolver

Modèles mixtes

Difficultés (9.0)

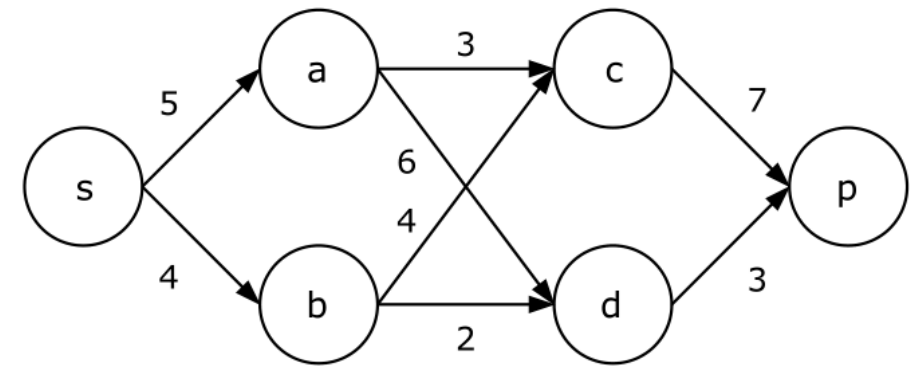
- Variables combinatoires et continues
- Couplées par des contraintes
- Conception de réseau / *Inventory Routing* / *Unit comitment*

Reformulation automatique bi niveaux

- Maître: variables combinatoires (recherche locale)
- Esclave: variables continues (programmation linéaire)
- Implémentation efficace en exploitant des propriétés du simplexe dual

Gains

- 30% sur le *periodic inventory routing problem* en 60s



Olivier Rigal

Résolution de problèmes d'optimisation à variables mixtes dans LocalSolver

VEN 11h50: Salle 36.106

Tournées de véhicules

Difficultés

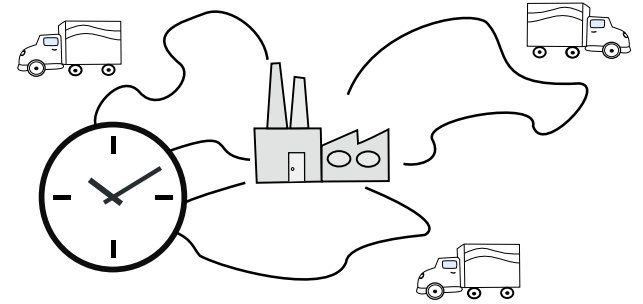
- Voyageur de commerce ✓
- Tournées de véhicules avec contraintes de capacité ✓
- Collecte et livraison ✓
- Fenêtres de temps

Mouvements avec recherche exhaustive

- Enumérer jusqu'à trouver le meilleur voisinage
- Evaluer efficacement les heures d'arrivées et de départ

Gains

- GAP 16% → 2.5% en 60s sur des CVRPTW à 100 clients (Solomon)



Bornes inférieures

Nikolas Stott

Calcul de bornes dans LocalSolver 9.5

Ven 12h10: Salle 36.106

Modèles non linéaires

Amélioration des algorithmes

- Réécriture du *presolve* linéaire
- Réécriture des coupes et ajouts de coupes de la littérature
- Nouvelle propagation de bornes
- Gestion de nouveaux opérateurs: Distance, XOR, min, max

Extension du calcul de bornes

- Traitement du multiobjectifs (linéaire)

Gains

- Gains en temps sur 75% de nos benchmarks linéaires
- Gains + 100 instances fermées sur la MINLPLib

Simon Boulmier

Unconstrained nonlinear relaxations
in global optimization

Modèles ensemblistes

Voyageur de commerce

- Variable de liste
- Contrainte de permutation
- Distances fixes

Reformulation automatique

- Utilisation de la borne de Held-Karp
 - Pénalise la contrainte de degrés
 - Arbre couvrant de poids min

Bornes pour certains problèmes de listes

- Calculées en quelques secondes
- Fermer le GAP en poursuivant la recherche
- Généraliser aux problèmes de tournées

```
Windows PowerShell
PS C:\Users\jdarlay\Projets\localsolver\examples\tsp> localsolver .\tsp.lsp inFileNames=.\instances\pr1002.tsp
LocalSolver 9.5.20200129-win64. All rights reserved.
Load .\tsp.lsp...
Run input...
Run model...
Run param...
Run solver...

Mode]: expressions = 15622, decisions = 1, constraints = 1, objectives = 1
Param: time limit = 5 sec, no iteration limit, nb threads = auto

[objective direction ]: minimize

[ 0 sec, 0 itr]: No feasible solution found (infeas = 12)
[ 1 sec, 50944 itr]: 263081
[ optimality gap ]: 2.75%
[ 2 sec, 96506 itr]: 263081
[ 3 sec, 144894 itr]: 263081
[ 4 sec, 186613 itr]: 263081
[ 5 sec, 240000 itr]: 263081
[ optimality gap ]: 2.75%

240000 iterations performed in 5 seconds

Feasible solution:
obj = 263081
gap = 2.75%
bounds = 255841

Run output...
PS C:\Users\jdarlay\Projets\localsolver\examples\tsp>
```

Alexandre Bontems

Calcul de bornes inférieures pour les problèmes de tournées dans LocalSolver

Ven 11h50: Salle 36.103

Conclusion

Améliorations sur la recherche locale historique

- Voisinages plus larges par propagation / énumération / optimisation
- Gains sur les problèmes de tournées, d'ordonnancement et mixtes (Olivier Rigal Ven 11h50: Salle 36.106)
- Perspective: gérer la planification de production / *unit comitment*

Améliorations sur le calcul de bornes

- Optimisation globale pour le calcul de bornes (Nikolas Stott Ven 12h10: Salle 36.106)
- Modèles ensemblistes (Alexandre Bontems Ven 11h50: Salle 36.103)



LocalSolver 9.5: Nouveautés et améliorations de performances

Julien Darlay

jdarlay@localsolver.com

www.localsolver.com

ROADEF 2020
Montpellier

John N. Hooker (2007)

“Good and Bad Futures for Constraint Programming (and Operations Research)”
Constraint Programming Letters 1, pp. 21-32

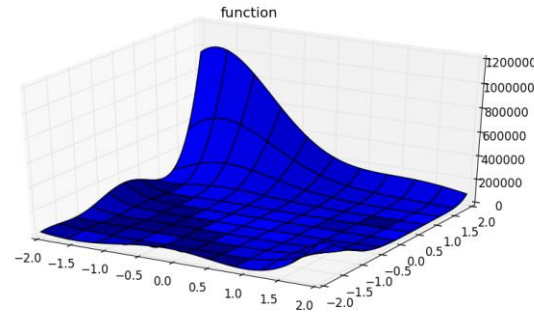
“Since modeling is the master and computation the servant, no computational method should presume to have its own solver.

This means there should be no CP solvers, no MIP solvers, and no SAT solvers. All of these techniques should be available in a single system to solve the model at hand.

They should seamlessly combine to exploit problem structure. Exact methods should evolve gracefully into inexact and heuristic methods as the problem scales up.”

Optimisation boîte noire

- Intégrée dans LocalSolver
- Transparent pour les utilisateurs



Python

- `pip install localsolver -i https://pip.localsolver.com`
- Compatible Anaconda, Apple, Python 3.8 et 3.9

