10 ans de LocalSolver:

Rétrospective et feuille de route

Julien Darlay

jdarlay@localsolver.com

www.localsolver.com

ROADEF 2020
Montpellier

Pour en savoir plus sur LocalSolver

Jeudi 10h (Dumontet): Journée GdR-RO Retour d'expérience industriel 1

Naissance de LocalSolver : de l'idée au produit

Vendredi 10h20 (36.101): Sur les meilleures pratiques de programmation en RO

LocalSolver 9.5 : nouveautés et améliorations des performances

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

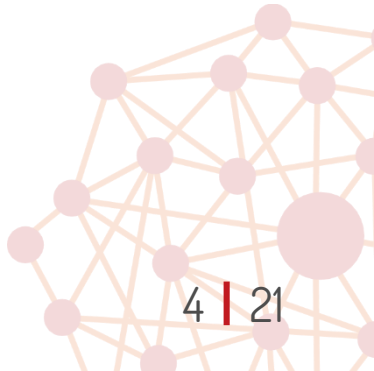
LocalSolver

Global Optimization

Nonlinear & Set-Based Modeling

Heuristics & Exact technics

Fast & Scalable



Ambition

John N. Hooker (2007)

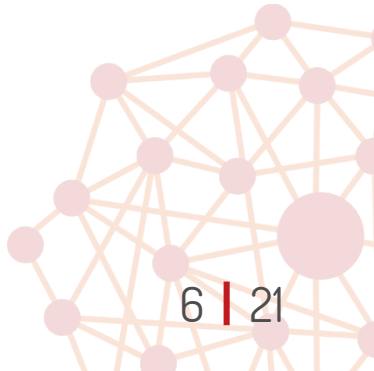
"Good and Bad Futures for Constraint Programming (and Operations Research)"
Constraint Programming Letters 1, pp. 21-32

"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."

Retrospective



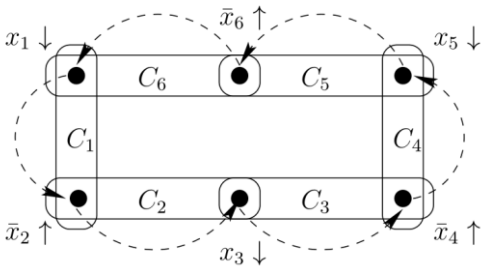
Premier travaux

De : Antoine JEANJEAN
Envoyé : lundi 16 juillet 2007 17:04
À : Frédéric GARDI; Guillaume ROCHART; Thierry BENOIST
Objet : LocalSolver : compte rendu journée 1

Voici quelques notes prises suite à notre première journée (n'hésitez pas à corriger ou à ajouter des choses).
Le but du solveur basé sur la recherche locale serait de résoudre des problèmes d'optimisation mathématique posés de manière abstraite, afin d'optimiser un objectif sous contraintes, sans faire une recherche complète de solutions. Les problématiques qui nous intéressent sont des problèmes industriels, avec beaucoup de variables de décisions, binaires dans un premier temps, peut -être des variables avec domaines dans un second temps. Notre volonté est de se comparer à des solveurs existants (CPLEX, XPRESS, COIN, GLPK, PPC, SAT, ...) en se basant sur des benchmarks tels que ceux de la CSPLib, de la OR Library, etc. Pour cela, nous nous concentrerons sur trois aspects : la stratégie de recherche, les transformations locales et l'algorithmique d'évaluation. Ces trois aspects étant très liés, nous tâcherons de veiller à leur bonne adéquation. Le langage choisi semble être le C++.
Les transformations. Il est important de randomiser les transformations. On pourra réaliser de l'apprentissage sur les résultats de chacune de ces transformations. Ainsi, au fur et à mesure du déroulé de la recherche, on utilisera à chaque itération les meilleures transformations pour optimiser et diversifier la recherche.
[…]

LocalSolver 1.0 (2009-2011)

- Solveur booléen
- Recherche locale
- Modèle mathématique non linéaire
- Problèmes combinatoires
- Efficacité algorithmique
- Multithread
- Ciblage



LocalSolver 1.x
A black-box local-search solver for 0-1 programming

Thierry Benoist · Bertrand Estellon · Frédéric Gardi · Romain Megel · Karim Nouioua

LocalSolver 2.0 (2012)

Introduction d'un langage de modélisation

- Faciliter la modélisation
- Pas de nécessité de linéarisation

Prétraitement automatique du modèle

- Correction des erreurs de modélisation
- Simplification du modèle

Amélioration des performances

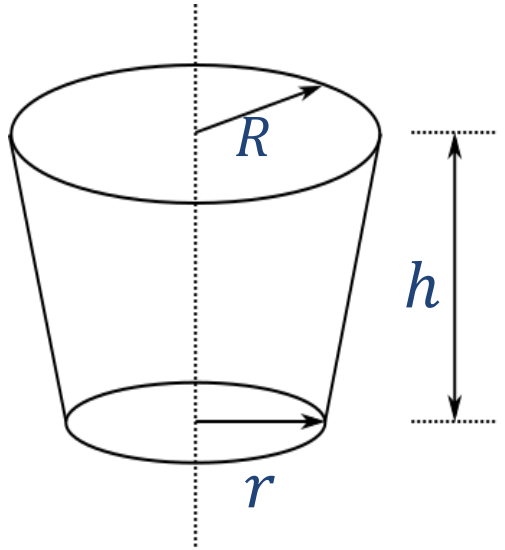
- Finalistes du challenge ROADEF 2012

```
function model() {
  // 0-1 decisions
  x[0..nbProcesses-1][0..nbMachines-1] <- bool();

  // each process must be assigned to a single machine
  for [p in 0..nbProcesses-1]
    constraint sum[m in 0..nbMachines-1](x[p][m]) == 1;

  // capacity constraints
  for [m in 0..nbMachines-1][r in 0..nbResources-1] {
    u[m][r] <- sum[p in 0..nbProcesses-1](require[p][r] * x[p][m]);
    constraint u[m][r] <= capacity[m][r];
  }

  // conflict constraints
  processByService[0..nbServices-1] = {}; // assigns an empty map
  for [p in 0..nbProcesses-1] {
    s = service[p];
    processByService[s][count(processByService[s])] = p;
  }
  for [s in 0..nbServices-1][m in 0..nbMachines-1]
    constraint sum[p in processByService[s]](x[p][m]) <= 1;

  //...
```

Vers un formalisme MINLP

Modélisation

- Variables de décision continues
- Variables de décision entières
- Opérateurs non linéaires: `puissance / cos / floor / piecewise`
- Fonctions définies par l'utilisateur



Résolution

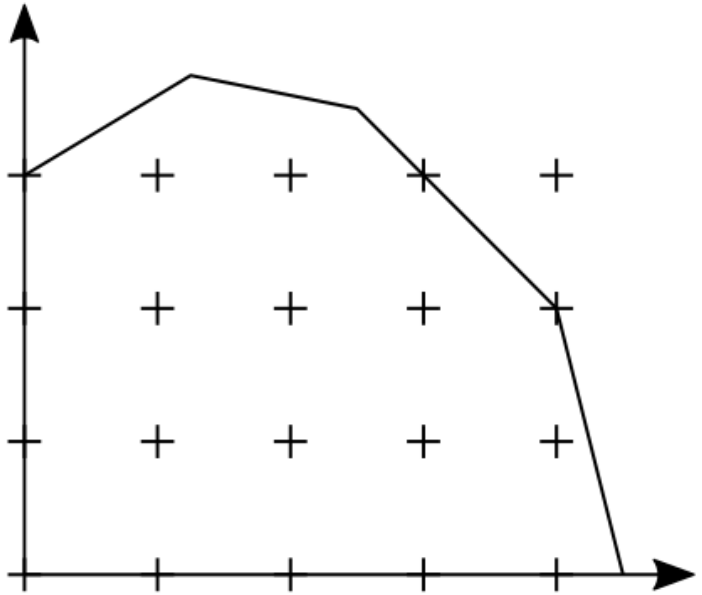
- Voisinages continus et entiers
- Réduction de domaines
- Apprentissage des meilleurs voisinages
- Reformuler les modèles à la MIP

```
function model() {

  R <- float(0,1);
  r <- float(0,1);
  h <- float(0,1);

  V <- PI * h / 3.0 * (R*R + R*r + r*r);
  S <- PI * r * r + PI*(R+r) * sqrt(pow(R-r,2) + h*h);

  constraint S <= PI;
  maximize V;
}
```

Ajout des techniques MIP

Ajout d'un solveur linéaire puis MIP

- Simplexe dual
- Branch & Cut
- Résolution des programmes linéaires
- Résolution des MIP simples



Utilisé comme sous-routine

- Exploration de voisinages larges
- Calcul de bornes inférieures
- Résolution de problèmes mixtes
- Heuristiques primales

Problèmes de tournées

Problèmes de tournées

- $x_{ij} = 1$ si le client i apparait à la position j
- Modèle avec $O(n^2)$ variables

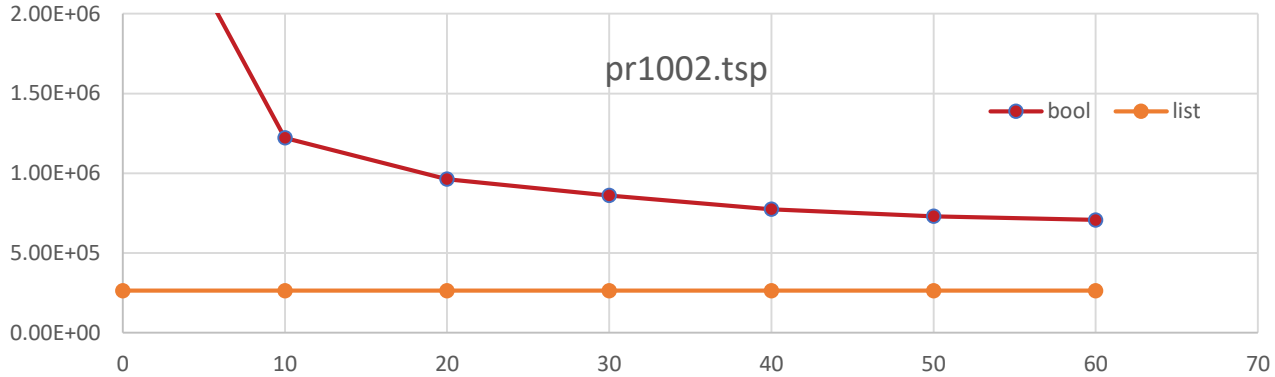
Introduction des *list*

- Sous-ensemble ordonné des entiers
- Une variable qui représente $O(n!)$ états
- Contraintes globales (partition, disjonction)

Voisinages spécifiques

- k-opt, insertions, échanges…
- Evaluation efficace

```
function model() {
  x <- list(N) ;
  constraint count(x) == N;

  minimize sum[i in 1..N-1](distance(x[i-1], x[i])) +
distance(x[N-1],x[0]);
}
```



Tiphaine Rougerie

Modélisation de problèmes de tournées de véhicules avec LocalSolver

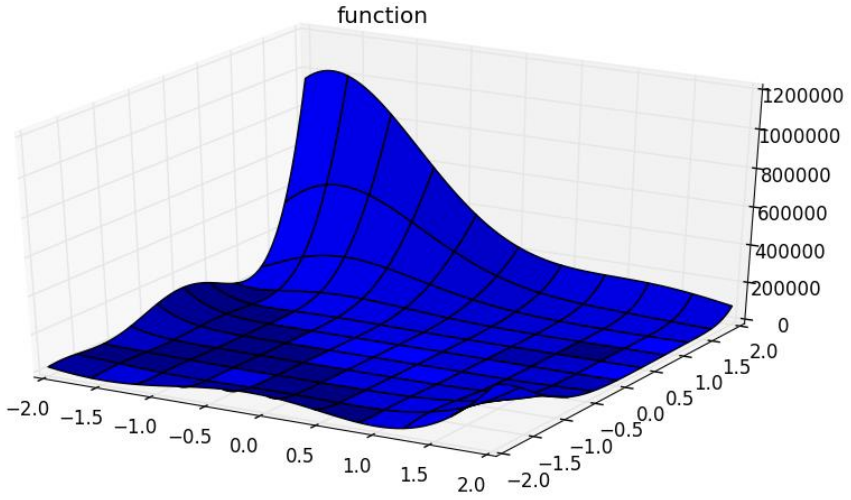
MER 14h00: Salle 36.103

LocalSolver

Optimisation boite noire

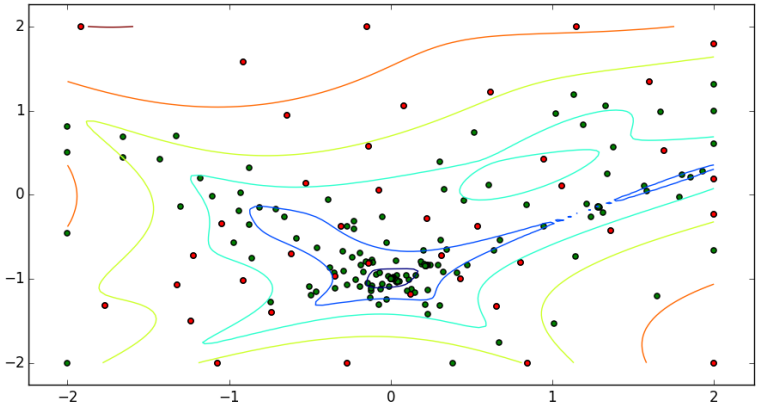
Contexte

- Optimisation d'un oracle « boite noire »
- Couteux à évaluer
- Variables continues ou combinatoire

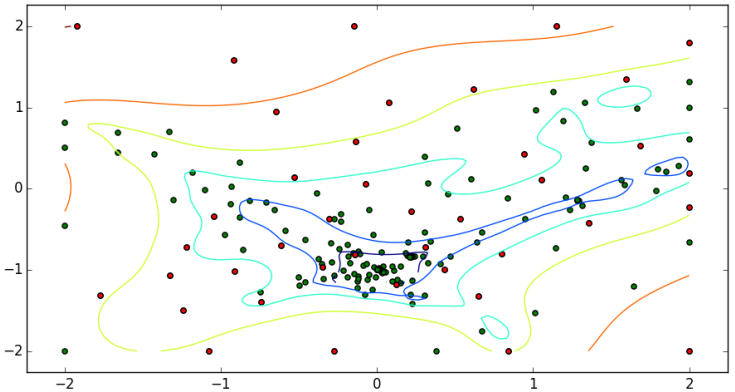


Résolution

- Apprentissage d'un modèle de l'oracle
- Optimisation / Diversification via recherche locale
- Comparable à l'état de l'art (RBFOpt / Nomad)



Oracle



Modèle

Problèmes de packing

Problèmes d'affectation

- $x_{ij} = 1$ si l'objet i est dans la boite j
- $\sum_j x_{ij} = 1$ chaque objet i est dans une boite
- Modèle en $O(nm)$

Introduction de l'opérateur *set*

- Sous-ensemble non ordonné des entiers
- Une variable qui représente $O(2^n)$ états
- Contraintes globales (partition, disjonction)

Voisinages spécifiques

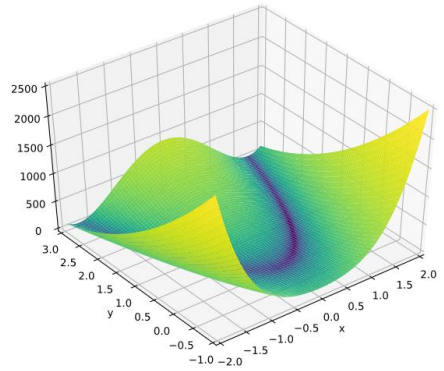
- Insertion / Echange
- Contraintes d'exclusion, de liaison, etc.

```
function model() {
    bins[k in 1..nbMaxBins] <- set(U);
    constraint partition[k in 1..nbMaxBins](bins[k]);

    for [k in 1..nbMaxBins] {
        constraint sum(bins[k], u => w[u]) <= binCapacity;
    }

    binsUsed[k in 1..nbMaxBins] <- count(bins[k]) > 0;
    totalBinsUsed <- sum[k in 1..nbMaxBins](binsUsed[k]);
    minimize totalBinsUsed;
}
```

Calcul de bornes

Solveur MINLP

- Algorithme de points intérieurs
- Reformulation automatique
- Convexification
- Branch & Reduce
- Calcul de bornes



Simon Boulmier

Unconstrained nonlinear relaxations in global optimization

JEU 11h40: Salle 36.07

Exploitation des structures

- Linéarisation automatique
- Détection de structures

Nikolas Stott

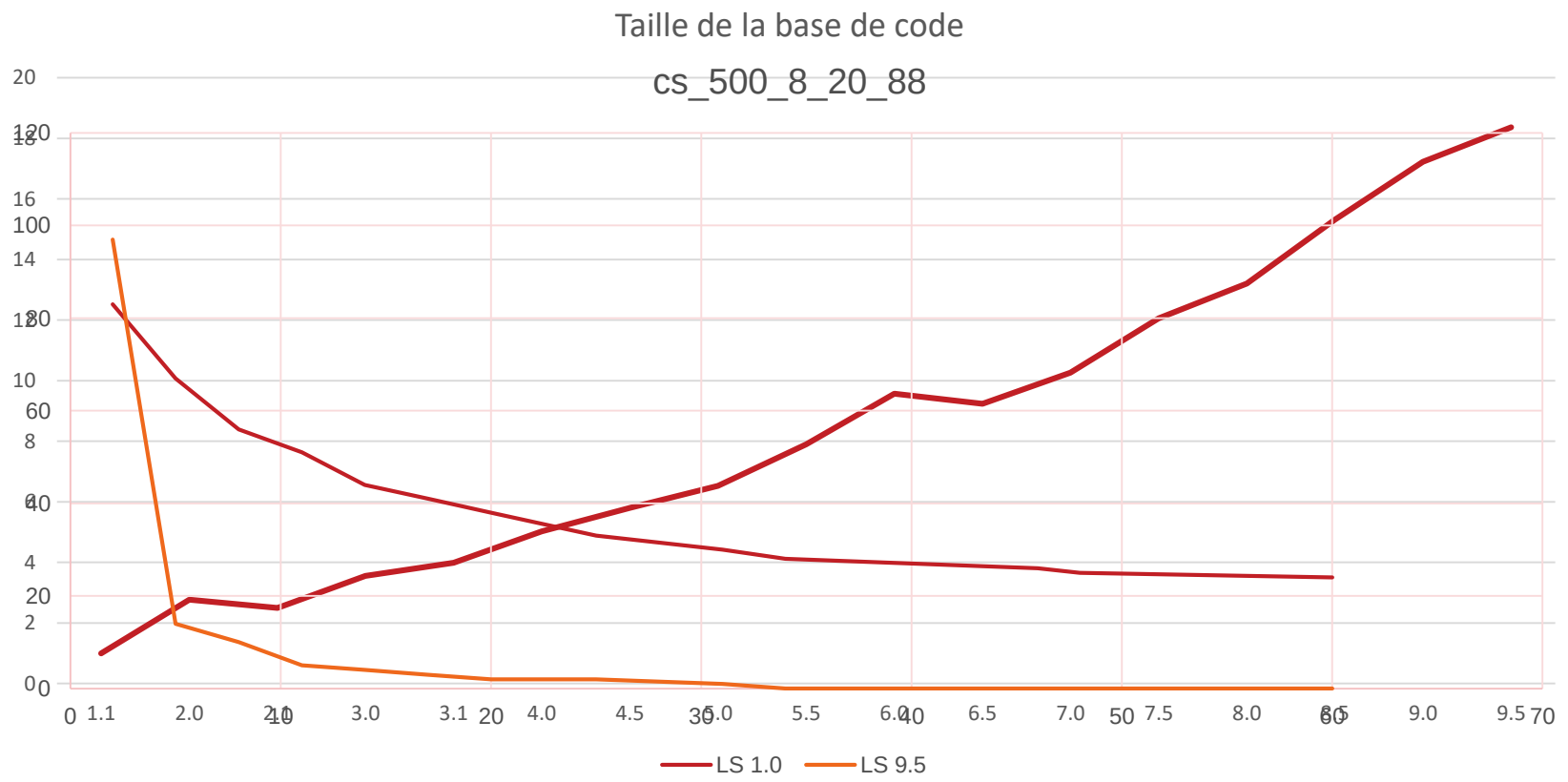
Calcul de bornes dans LocalSolver 9.5

VEN 12h10: Salle 36.106

Evolutions de LocalSolver

En développement permanent

Piloté par les besoins de nos utilisateurs



Feuille de route

Axée sur les besoins de nos utilisateurs

Planification de production

Problème d'ordonnancement

- Comment les modéliser efficacement ?
- Gestion des temps morts
- Ajouter les mouvements nécessaires

Léa Blaise

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

JEU 11h40: Salle 36.05

Problèmes mixtes

- Mixte variables combinatoires et continues
- Exemple: conception de réseau, IRP...

Olivier Rigal

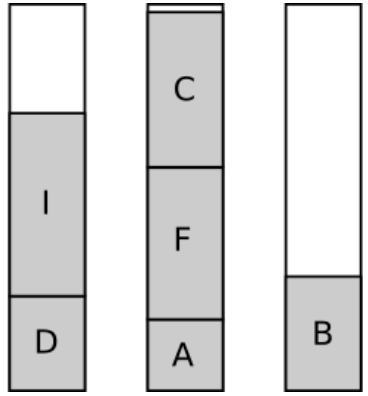
Résolution de problèmes d'optimisation à variables mixtes dans LocalSolver

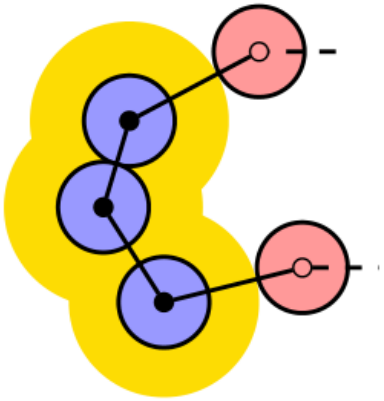
VEN 11h50: Salle 36.106

Bornes sur les structures ensemblistes

Structures ensemblistes

- *list, set* + contraintes globales
- Problèmes de tournées, de *packing* et de *clustering*





Algorithmes spécifiques

- Exploitation de la structure du sous problème

Programmation mathématique

- Plans coupants
- Formulation étendue + génération de colonnes
- Réutilisation des solutions primales de la recherche locale

Conclusion

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."

10 ans de LocalSolver:

Rétrospective et feuille de route

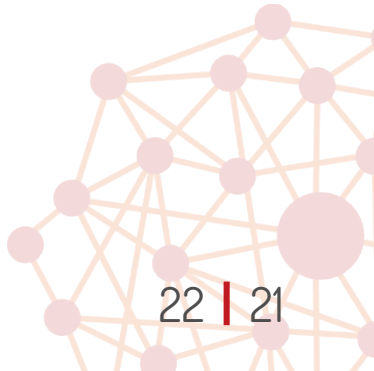
Julien Darlay

jdarlay@localsolver.com

www.localsolver.com

ROADEF 2020
Montpellier

Conclusion



LocalSolver

R&D pilotée par les besoins des utilisateurs

Donner accès aux méthodes de l'état de l'art à travers un modèle haut niveau

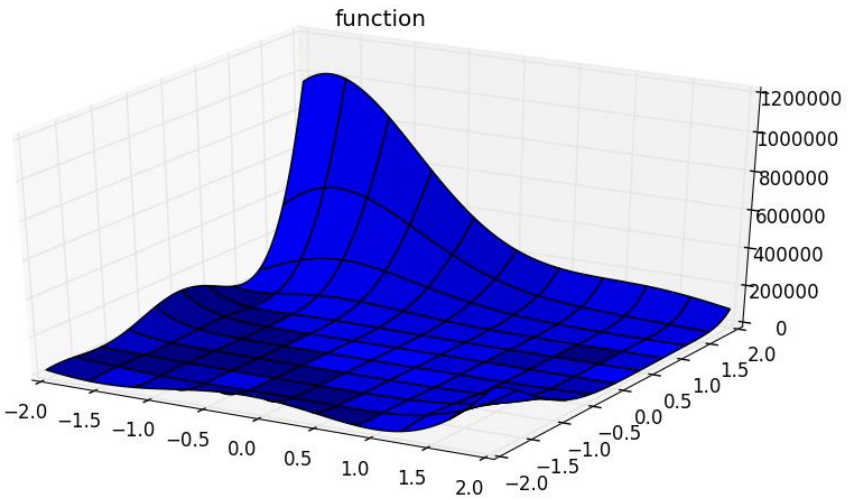
Amélioration des fonctionnalités / performances

Tournées de véhicules

- Grandes instances (> 500 clients) avec fenêtres de temps

Optimisation boite noire

- Gestion des contraintes
- Multiobjectifs



LocalSolver