UPMC
SORBONNE UNIVERSITÉS

Habilitation à Diriger des Recherches

Vers un solveur de programmation mathématique basé sur la recherche locale

Frédéric Gardi

Innovation 24 & LocalSolver

25 novembre 2013
UPMC, Paris

BOUYGUES

Innovation24

Plan

1 – Recherche locale : comment industrialiser ?

2 – Retours d'expériences méthodologiques

3 – Recherche locale pour la programmation non-linéaire 0-1

4 – Vers un solveur d'optimisation d'un nouveau genre

Recherche locale

Comment industrialiser ?

Recherche locale

Une méthode d'amélioration itérative

- Explorer un voisinage de ma solution courante
- Voisinage plus ou moins large
- Prouvée peu efficace dans le pire des cas

Incontournable en optimisation combinatoire

- Cachée derrière de nombreux algorithmes classiques (ex: simplexe, flot max)
- Au cœur de toutes les approches métaheuristiques
- Très utilisée parce que très efficace en pratique

Comment industrialiser ?

2000 – 2005 : prise de conscience

- Ingénieur chez Prologia : planification de personnel
- Challenge ROADEF 2005 : ordonnancement de véhicules chez Renault (1er Prix Junior et Senior avec B. Estellon et K. Nouioua)

Depuis 2005 : comment industrialiser ?

- Retours méthodologiques pour l'optimisation combinatoire
- Retours méthodologiques pour l'optimisation mixte
- Un solveur d'optimisation combinatoire exploitant la recherche locale
- Un solveur d'optimisation hybride, tout-en-un, pour l'optimisation non-convexe en variables mixtes, basé sur la recherche par voisinage

Retours méthodologiques

10 ans de recherche locale

Pourquoi la recherche locale ?

Quand il est vain d'énumérer

- Problèmes très combinatoires, non convexes, de grande taille
- Quand relaxation ou inférence apportent peu
  (ex : relaxation linéaire de mauvaise qualité, très fractionnaire)
- Quand calculer une relaxation ou inférer est coûteux

En accord avec les besoins des clients

- Un optimum local de bonne qualité les contentent
- *Goal programming* et recherche locale font bon ménage
- Technique véloce - solutions en temps très courts - qui passe à l'échelle

Ingrédients de la recette

Rappel : notre objectif est d'industrialiser

Recherche locale = recherche par voisinage

- Voisinages multiples, riches, larges (par union ou composition)
- Exploration randomisée (*first improvement*)
- Évaluation rapide par calcul incrémental
- Stratégie de recherche simple (ex : descente)

Clés méthodologiques

- Privilégier une approche frontale du problème : éviter de décomposer
- *Simple is better* : éviter hybridation et réglages fastidieux, privilégier la qualité (design et fiabilité), éviter toute optimisation prématurée
- Amélioration pas à pas, dirigée par les tests et les retours du client : enrichir/élargir les mouvements et accélérer leur évaluation

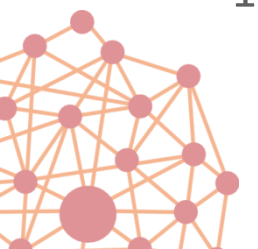
Applications

Optimisation combinatoire

- Ordonnancement de véhicules chez Renault (Challenge ROADEF 2005 : 1er Prix Junior et Senior avec B. Estellon K. Nouioua)
- Planification des interventions techniques chez France Telecom (Challenge ROADEF 2007 : 2ème Prix Senior avec B. Estellon et K. Nouioua)
- Planification de campagnes publicitaires TV chez TF1 (2011)

Optimisation mixte

- Tournées de véhicules avec gestion des stocks chez Air Liquide (2008, avec T. Benoist, B. Estellon, A. Jeanjean) - connu sous le nom d'*Inventory Routing*
- Ordonnancement des mouvements de terre sur chantiers linéaires chez DTP Terrassement (2009, thèse de doctorat A. Jeanjean)
- Planification des arrêts de centrales nucléaires (Challenge ROADEF 2010 : 1er Prix Applications Industrielles ROADEF 2011, avec K. Nouioua)



Conclusions

Revenir à l'essentiel : un bon modèle

- Un modèle réaliste mais parcimonieux
- Ne pas se focaliser sur l'optimum mathématique
- Privilégier objectifs à contraintes : bon pour clients et fournisseurs !

Recherche locale : revenir à l'essentiel

- Ne pas se focaliser sur les aspects « méta »
- Se focaliser sur : enrichir les mouvements, accélérer leur évaluation
- Enrichir en fonction des besoins : se laisser guider par les tests et les retours

⟶ En définitive, produire une recherche locale performante reste une question d'expertise en algorithmique et de dextérité en programmation

LocalSolver

Recherche locale pour la programmation non-linéaire 0-1

Industrialiser : se doter d'outils efficaces

Bibliothèques et frameworks de développement

- Pour les métaheuristiques : EasyLocal++ (2000), Paradiseo (2004)
- Pour l'évaluation incrémentale : Localizer (1997), iOpt (2001)
- Pour les mouvements : OptaPlanner 5.4 (2012)

Solveurs d'optimisation « model & run »

- SAT : Walksat (1996), WSAT-OIP (1999)
- BP/IP : Connolly (1992), Nonobe-Ibaraki (1998), Abramson-Randall (1999)
- CP/CBLS : CP Optimizer 2.0 (2008), Comet 2.1 (2010)
- Les solveurs MIP intègrent de plus en plus d'ingrédients heuristiques depuis une dizaine d'années : Local Branching (2003), RINS (2005)

⟶ En 2007, nous n'avions pas connaissance d'un solveur d'optimisation combinatoire exploitant la recherche locale qui soit largement utilisé

Le projet LocalSolver

2007 : lancement du projet

- Définir un formalisme de modélisation générique (proche du formalisme MIP, connu et pratiqué) adapté à une résolution par recherche locale (*model*)
- Développer un solveur effectif basé sur la recherche locale avec pour premier principe : « qu'il fasse ce qu'un expert ferait » (*run*)

2009-2011 : version 1.x

- Permet d'attaquer des problèmes 0-1 de grande taille de type *assignment*, *partitioning*, *covering*, *packing* hors de portée des solveurs MIP et CP
- Utilisation au sein d'études et d'applications dans le Groupe Bouygues : TF1 Publicité, ETDE, Bouygues Telecom, 1001 Mariages
- Premières utilisations en dehors de Bouygues
- Un papier : 4OR 9(3), pp. 299-316

P-médian

Sélectionner un sous-ensemble P parmi N points minimisant la somme des distances de chaque point dans N au point le plus proche dans P.

```
function model() {
  x[1..N] <- bool() ; // décisions : le point i appartient à P si x[i] = 1

  constraint sum[i in 1..N]( x[i] ) == P ; // contrainte : P points parmi les N

  minDist[i in 1..N] <- min[j in 1..N]
            ( x[j] ? Dist[i][j] : InfiniteDist ) ; // expression : distance au point le plus proche dans P

  minimize sum[i in 1..N]( minDist[i] ) ; // objectif : minimiser la somme des distances
}
```

Rien de plus à écrire : approche “model & run”

Opérateurs mathématiques

Arithmétiques			Logiques	Relationnels
sum	sub	prod	not	==
min	max	abs	and	!=
div	mod	sqrt	or	<=
log	exp	pow	xor	>=
cos	sin	tan	if	<
floor	ceil	round	array + at	>

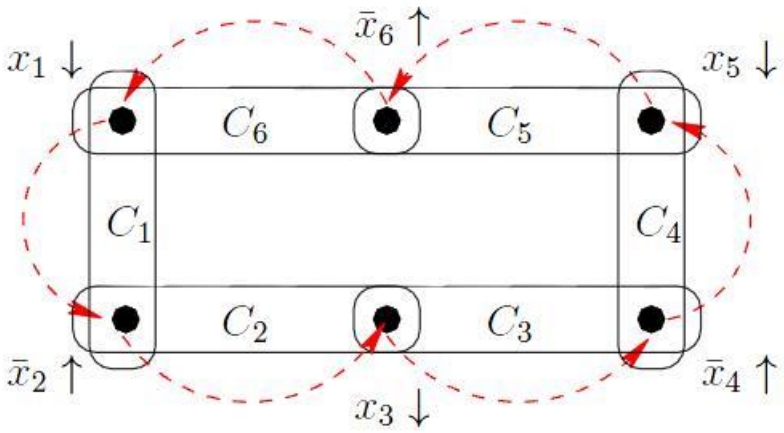
Petits voisinages, très locaux

Le classique en Programmation Booléenne : « k-flips »

- Conduisent majoritairement à des solutions infaisables
- Faisabilité difficile à retrouver : convergence lente

Nos mouvements tendent à préserver la faisabilité

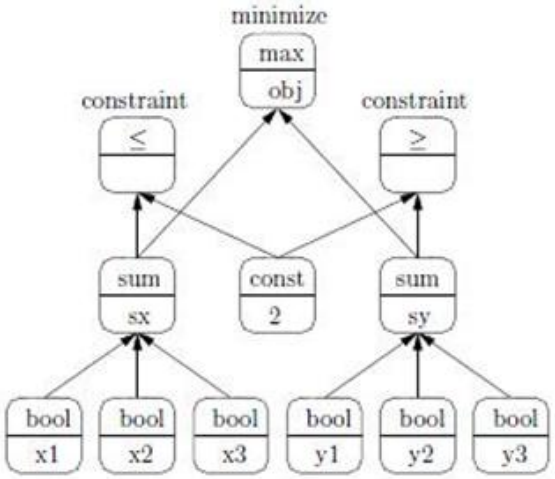
- Approche de type *destroy & repair*
- Chaînes d'éjection dans l'hypergraphe des contraintes
- Spécifiques à certaines structures combinatoires détectées



Petits voisinages, très vite évalués

```
x1 <- bool(); x2 <- bool(); x3 <- bool();
y1 <- bool(); y2 <- bool(); y3 <- bool();
sx <- sum(x1, x2, x3);
sy <- sum(y1, y2, y3);
constraint sx <= 2;
constraint sy >= 2;
obj <- max(sx, sy);
minimize obj;
```

→



Evaluation incrémentale

- Propagation *lazy* des modifications induits par un mouvement dans le DAG
- Exploitation des invariants induits par les opérateurs mathématiques

→ Des millions de mouvements évalués par minute de calcul

LocalSolver 2.x et 3.x

2012 : lancement commercial de la version 2.0

- Faire vivre ce projet R&D à long terme
- Ajuster le logiciel aux besoins des praticiens de la RO
- Diffuser le logiciel (et nos idées) hors de Bouygues et de France

2013 : version 3

- 370 utilisateurs enregistrés dont 270 hors de France
- 500 licences distribuées dont 300 licences académiques
- 15 licences commerciales vendues hors de Bouygues
- En France : Air Liquide, Armée de Terre, MediaTransports, ESIEE
- À l'international : Pasco, Future Architects, Fujitsu, Hitachi, NIES Japan, plusieurs universités chinoises

LocalSolver 2.x et 3.x

Preprocessing

- Réduction et reformulation du modèle
- Élimination de décisions et expressions par inférence
- En temps linéaire en la taille du modèle en entrée

Stratégie de recherche

- Recherche randomisée et multi-cœurs
- Recuit simulé avec *restarts*
- Stratégie de recherche adaptative par apprentissage (ex : sélection adaptative des « bons » mouvements)

LocalSolver 2.x et 3.x

Voisinages

- Plusieurs dizaines de petits voisinages
- Composition de mouvements pour élargir le voisinage exploré
- Mouvements dédiés aux structures de *scheduling & routing*

Algorithmique & implémentation

- Optimisation bas-niveau du code : *cache-aware programming*
- Gestion de la mémoire au niveau du bit
- Multithreading sans impact temps et mémoire

→ Optimisation de la chaîne logistique : production + distribution
5 minutes pour 20 M expressions dont 3 M décisions 0-1
Résolu chaque jour chez Pasco, n°1 de la boulangerie au Japon

Pasco

Panorama d'applications

FUJITSU	Optimisation de la chaîne logistique
MEDIATRANSPORTS	Construction de réseaux d'affichage publicitaire
TF1 PUBLICITE	Media planning TV
AIR LIQUIDE	Tournées de véhicules, clustering logistique
ETDE	Maintenance de réseaux de luminaires
Bouygues Telecom	Optimisations sur le réseau de téléphonie mobile
SIEMENS	Optimisation énergétique de lignes de tramways
EDF	Maintenance des piles nucléaires (QAP)
SSI STRATEGIC SYSTEMS INTERNATIONAL	Planification de personnel hospitalier
ARMEE DE TERRE	Transport de matériel militaire

Séquencement de véhicules

Car sequencing, CSPLIB, prob001: http://csplib.org

Séquencer des véhicules sur un chaîne d'assemblage
Instance 500 = 500 véhicules : 4 000 décisions binaires

Minimisation

10 sec	100	200	300	400	500
Gurobi 5.5	140	274	X	429	513
LocalSolver 3.1	**6**	**8**	**9**	**11**	**24**

60 sec	100	200	300	400	500
Gurobi 5.5	**3**	66	**1**	356	513
LocalSolver 3.1	6	**3**	3	**7**	**10**

600 sec	100	200	300	400	500
Gurobi 5.5	**3**	**2**	**0**	**1**	20
LocalSolver 3.1	6	**2**	1	2	**4**

	100	200	300	400	500
State of the art	*3*	*0*	*0*	*1*	*0*
Décisions 0-1	*500*	*1000*	*1500*	*2000*	*4000*

Séquencement de véhicules chez Renault

Challenge ROADEF 2005 : http://challenge.roadef.org/2005/en

Instances de très grande taille

- 1300 véhicules à ordonnancer : 400 000 décisions binaires

Instance 022_EP_ENP_RAF_S22_J1 : 540 véhicules

- Petite instance : 80 000 variables dont 44 000 décisions 0-1
- État de l'art : 3 109 obtenu par recherche locale spécifique
- Borne inférieure : 3 103

Minimization

Résultats

- Gurobi 5.5 : 3.027e+06 en 10 min | 194 161 en 1 heure
- LocalSolver 3.1 : 3 476 en 10 sec | 3 114 en 10 min

Ordonnancement de processus chez Google

Challenge ROADEF/EURO 2012 : http://challenge.roadef.org/2012/en/

Temps limité à 5 minutes sur un ordinateur standard

Modèle LocalSolver (2.0) de 100 lignes

Classé 25ème sur 82 équipes participantes (30 pays représentés)

Seul solveur model & run qualifié pour la finale (30 équipes)

Instance B1 : 10 M expressions, 300 000 contraintes, 500 000 décisions

Problèmes de tournées

TSP http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95

TSP assymmétrique

LocalSolver 3.1 lancé 5 min sur un ordinateur standard

Écart moyen avec l'état de l'art (algorithmes spécifiques) : 3 %

VRP http://neo.lcc.uma.es/vrp/vrp-instances

CVRPTW, instances de Solomon

LocalSolver 3.1 lancé 5 min sur un ordinateur standard

Écart moyen avec l'état de l'art (algorithmes spécifiques) : 14 %

MIPLIB

Résultats sur les instances les plus dures/larges de la MIPLIB

- On minimise l'objectif
- 5 minutes pour LocalSolver et Gurobi
- Modèle orienté MIP : pas adapté à LocalSolver

Minimisation

Problem	Decisions	LocalSolver 3.1	Gurobi 5.1
ds-big	174 997	9 844	62 520
ivu06-big	2 277 736	479	9 416
ivu52	157 591	4 907	16 880
mining	348 921	- 65 720 600	902 969 000
ns1853823	213 440	2 820 000	4 670 000
rmine14	32 205	- 3 470	-171
rmine21	162 547	- 3 658	- 185
rmine25	326 599	- 3 052	- 161

LocalSolver

Vers un solveur d'optimisation hybride basé sur la recherche par voisinage

Un solveur d'un nouveau genre

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.

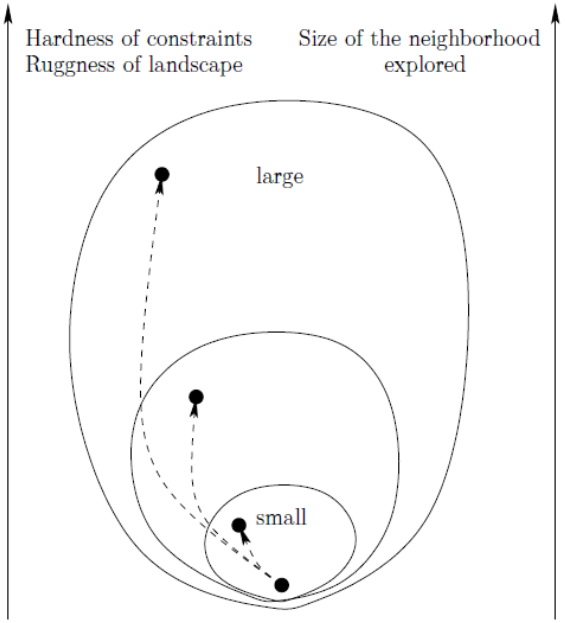
This means there should be no CP solvers, no MIP solvers, and no SAT solvers. All of these techniques should be available in a single system to solve the model at hand.

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

Comment hybrider ?

La recherche par voisinage comme recherche globale

- Permet d'accélérer la recherche par exploration rapide de petits voisinages
- Permet d'adapter le voisinage exploré au besoin : rétrécir, élargir, spécialiser
- Dans ce cadre, une recherche arborescente, et plus largement les techniques de type MIP et CP, ne sont qu'une façon d'explorer un voisinage très large
- Lorsqu'un voisinage met en jeu toutes les décisions (= espace des solutions) et qu'il est exploré de façon exacte, alors la solution trouvée est optimale



Feuille de route

Explorer des voisinages très larges

- Voisinages très larges (= exponentiel) explorés par recherche arborescente
- Utiliser relaxation linéaire et inférence pour explorer efficacement

⟶ Meilleurs optima locaux, voire solutions optimales pour certaines classes de problèmes (ceux pour lesquels les solveurs MIP et CP fonctionnent bien)

Extension à l'optimisation continue et mixte

- Recherche locale = "*direct search*" en programmation non-linéaire
- Ex : méthode du Simplexe de Nelder-Mead (1965)

"Mathematicians hate it because you can't prove convergence;
engineers seem to love it because it often works"

- Petits voisinages et voisinages composés pour décisions continues/mixtes
- Voisinages larges = *successive linear/quadratic programming*
  Utilisation d'approximations de l'espace au premier ou second ordre

Feuille de route

Séparer calcul de solutions et calcul de bornes

- Optimiser et borner sont deux choses différentes
- Borner = propagation de contraintes + relaxation duale linéaire/convexe, plongés dans une stratégie heuristique de type *divide & conquer*

⟶ Optimiser et borner s'enrichissent mutuellement, sans se contrarier

⟶ Fournir un écart d'optimalité voire une preuve d'infaisabilité/optimalité

Modéliser avec des décisions entières

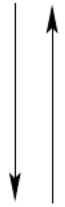
- Des décisions entières comme quantités ou indexeurs d'ensembles
- *Set-variable programming* : une façon de simplifier la modélisation pour les problèmes de type *scheduling & routing* ?
- Volonté de conserver un formalisme de modélisation mathématique et « universel » à la portée du praticien non expert, non informaticien

Architecture cible

Intégrer toutes les techniques d'optimisation opportunes (LS, LP/MIP, CP/SAT, NLP, ...) en un solveur unique pour l'optimisation non-convexe en variables mixtes à grande échelle

Feasibility search

Optimization



Infeasibility proof

Lower bound

Preprocessing	**Neighborhood Search**	**Moves**		
		Combinatorial	**Continuous**	**Mixed**
Model rewriting Structure detection Constraint inference Variable elimination Domain reduction	Simulated annealing Restarts Randomization Learning	Small Compound Large	Small Compound Large	Small Compound Large
	Divide & Conquer	**Propagation**	**Relaxation**	
	Tree search Interval branching	Discrete propagation Interval propagation	Dual linear relaxation Dual convex relaxation	

LocalSolver 4.0

Sortie prévue pour décembre 2013

- Décisions binaires & continues
- Preprocessing et inférence de bornes renforcés
- Heuristique de recherche de faisabilité améliorée
- Voisinages petits et composés pour l'optimisation continue/mixte
- Premiers mouvements à voisinage large basés sur la relaxation linéaire

⟶ Permet d'attaquer des problèmes mixtes comme le « *unit commitment* »

http://www.localsolver.com

UPMC
SORBONNE UNIVERSITÉS

Habilitation à Diriger des Recherches

Vers un solveur de programmation mathématique basé sur la recherche locale

Frédéric Gardi

Innovation 24 & LocalSolver

25 novembre 2013
UPMC, Paris

BOUYGUES

Innovation24