

La RO, c'est quoi : des théorèmes ou des logiciels ?

Frédéric Gardi ^{1 2}

Bouygues e-lab & LocalSolver, Paris

<http://pageperso.lif.univ-mrs.fr/~frederic.gardi>

Après qu'Olivier Spanjaard m'eut proposé d'écrire un article pour le bulletin de la ROADEF, je fus quelque peu embarrassé. En effet, j'avais déjà été sollicité pour écrire un article sur LocalSolver dans le bulletin précédent. Que dire en quelques pages qui n'ait déjà été dit ? Que dire en quatre pages qui puissent intéresser le plus grand nombre ? Ayant eu quelques retours positifs lors de mon exposé intitulé "C'est quoi la RO : des théorèmes ou des logiciels ?" lors de la remise du Prix Robert Faure 2012 à Angers, j'ai pensé que cela pourrait être intéressant d'en faire une note manuscrite. Ainsi, je vous propose un panorama de mes travaux en RO, avec en filigrane et en conclusion un retour d'expérience à destination des praticiens et des chercheurs, notamment les plus jeunes. Pour plus de détails sur les sujets exposés, j'invite le lecteur à se reporter aux articles cités, disponibles sur ma page web à l'adresse mentionnée ci-dessus.

2000-2005 : DEA et thèse de Doctorat

J'ai découvert la RO à Marseille, alors étudiant en informatique à la Faculté des Sciences de Luminy. Cette petite faculté est sans doute inconnue de nos jeunes lecteurs, mais les anciens se souviendront qu'elle est le berceau de Prolog, fameux langage de programmation logique qui fit les beaux jours de l'intelligence artificielle durant les années 80. Après mon DEA d'informatique (option structures discrètes et recherche opérationnelle), je suis entré comme ingénieur dans la société Prologia, tout en débutant une thèse de Doctorat au Laboratoire d'Informatique Fondamentale de Marseille (LIF). Bien qu'éprouvante, cette expérience – un pied dans le monde de l'entreprise et l'autre dans le milieu de la recherche publique – a été fort enrichissante et m'a profondément marqué. Alors filiale

du Groupe Air Liquide, Prologia était une PME d'une trentaine de personnes proposant des produits et services en aide à la décision et optimisation, notamment dans les domaines de la gestion de crédits immobiliers et de la planification de ressources humaines. C'est sur ce dernier sujet que j'ai débuté mes travaux en RO. J'ai travaillé sur les moteurs d'optimisation au cœur du progiciel Bamboo de planification de personnel, commercialisé auprès d'une quinzaine de sociétés clientes, notamment dans le domaine aéroportuaire (par exemple Corsair, Servisair, Servair, Tunisair Handling).

Parallèlement à ces travaux de recherche très opérationnelle, je consacrais ma thèse à des questions d'algorithmique et de théorie des graphes extraites de ces problématiques de planification, notamment le problème de la coloration bornée de graphes d'intervalles ou de classes apparentées [1, 2, 3, 6, 7, 9, 13]. C'est ainsi que j'ai compris ce que prouver signifie véritablement, et combien rédiger un article scientifique est un art... laborieux voire douloureux ! Mais dont l'apprentissage est d'une nécessité vitale pour le jeune chercheur, d'autant plus s'il souhaite embrasser une carrière académique.

2003-2007 : applications bancaires

En 2003, Prologia décrocha un grand projet (8000 jours-hommes) : le développement d'une application d'instructions de prêts immobiliers pour la Société Générale. Je fus dépêché pour réaliser un certain nombre de composants analytiques au sein de cette application. Un des composants semblait particulièrement intéressant d'un point de vue RO : l'optimisation du plan de financement immobilier. Je ne soupçonnais pas que celui-ci serait une de mes plus belles réalisations, tant sur le plan technique

1. Bouygues e-lab, 24 avenue Hoche, 75008 Paris. fgardi@bouygues.com - <http://e-lab.bouygues.com>

2. LocalSolver, 24 avenue Hoche, 75008 Paris. fgardi@localsolver.com - <http://www.localsolver.com>

que d'un point de vue métier.

Si certaines banques peuvent proposer des financements immobiliers adossés à des assemblages de prêts, ceux-ci demeurent limités tant par la technicité de ces prêts que par les outils de calcul classiques qui aident à les composer. Dans leur grande majorité, les plans de financement immobiliers se contentent ainsi d'un seul prêt dont l'échéance est constante sur la durée de celui-ci. La Société Générale, qui avait préalablement fait évoluer sa gamme de produits immobiliers vers des technicités permettant plus de souplesse dans le remboursement, souhaitait pratiquer des assemblages de prêts afin de proposer les meilleurs plans de financement possibles. Il s'avère qu'un bon assemblage peut permettre de réduire le coût d'un emprunt immobilier de plus de 10 %, ce qui n'est pas négligeable pour des particuliers. Pour aider les chargés de clientèle et être certain de proposer les meilleures solutions à ses clients, la Société Générale souhaitait se doter d'un véritable outil d'optimisation – plus que de construction – de plans de financement immobiliers. Cet outil, basé sur une approche par programmation linéaire mixte [5, 11], est progressivement entré en exploitation dans le réseau des 2500 agences de la banque début 2006. Plus de 10000 chargés de clientèle ont été formés à l'utilisation de cet outil, qui permet de construire de façon automatique et optimisée plusieurs milliers de plans de financement chaque jour. Depuis, la Banque Postale a elle aussi intégré ce composant d'optimisation au sein de son système d'information. Ainsi, près de 15 milliards d'euros de crédit à l'habitat sont planifiés chaque année à l'aide de ce logiciel.

Dans un tel contexte, les contraintes techniques sont très fortes : quelques secondes de temps de calcul, près de 300 000 exécutions par an (un taux de panne de 1 % signifie alors 3000 anomalies par an, soit 10 par jour), une durée de vie du logiciel envisagée à plus de 10 ans avec la nécessité de pouvoir s'adapter aux évolutions de la réglementation. C'est durant ce projet que j'ai compris que la RO en pratique, c'est beaucoup d'informatique. Et je reste convaincu que la qualité (fonctionnelle et surtout technique) des logiciels développés par les ingénieurs en RO est une des clés du succès de la discipline dans les entreprises.

2003-2007 : recherche locale (combinatoire)

Fin 2003, Karim Nouioua, un collègue docteur du LIF, me proposa de participer au Challenge ROADEF 2005 avec lui. Le challenge portait sur un problème d'ordonnement de véhicules [16] posé par Renault. Jeunes apprentis, nous

étions (ou plus exactement avions été) convaincus de la toute puissance des techniques de programmation linéaire en nombres entiers (PLNE). Nous nous sommes donc lancés à l'assaut de ce problème avec notre langage de programmation préféré (GNU C) et un solveur de PLNE libre (GLPK). Nous nous sommes ainsi qualifiés pour la finale, mais plutôt en bas de tableau. En fait, la PLNE fonctionnant mal (et c'est sans doute pour cela que Renault avait posé le problème), nous avons fini par faire – sans le savoir à l'époque – une recherche locale à voisinage large exploré par PLNE. Suite à notre qualification, Bertrand Estellon, doctorant tout juste sorti du DEA, nous rejoignit et se proposa d'implémenter quelques voisinages en complément des nôtres. Des voisinages classiques, de petite taille, qu'il avait trouvés dans la littérature : échanges, déplacements, ou encore inversions de blocs de véhicules. Quelle ne fût pas notre surprise : il battait tous nos records, en quelques dizaines de secondes seulement ! Pas si surprenant si l'on y réfléchit sans a priori : vaut-il mieux faire des mouvements ayant une forte probabilité d'amélioration mais très coûteux en temps (10 par minute), ou bien des mouvements ayant une faible probabilité d'amélioration mais à l'exécution extrêmement efficace (10 million par minute) ? La réponse était ici sans ambiguïté. En omettant les mouvements larges et en travaillant les petits mouvements (notamment leur complexité algorithmique), les résultats dépassaient nettement les meilleurs résultats obtenus par l'ensemble des compétiteurs. En définitive, les résultats furent si bons qu'ils nous permirent de remporter le challenge, catégories Junior et Senior confondues. Renault décida de déployer notre solution au sein de ses 17 usines dans le monde. Nous écrivîmes deux articles sur le sujet [4, 8], en sus de nos travaux de thèse respectifs. Quelle aventure extraordinaire ! Débutée par une simple question – “Cela te dirait de faire le challenge avec moi ?” – dont je ne pouvais me douter jusqu'où m'emmènerait la réponse...

En 2006, nous sommes repartis à l'assaut du nouveau Challenge, posé par France Telecom sur la planification d'interventions et de techniciens de maintenance. Vu la combinatoire du problème et l'échelle des instances, il nous sembla judicieux d'attaquer le problème par recherche locale. Nous avons ainsi remporté la deuxième place en appliquant la même “recette” que précédemment : une heuristique assurant la diversification de la recherche, des mouvements de petite taille mais riches par leur nombre et leur diversité, une machinerie d'évaluation incrémentale à l'efficacité redoutable (un million de mouvements par seconde). Il nous parut alors inté-

ressant de formaliser cette méthodologie afin qu'elle profite aux praticiens désireux de résoudre efficacement de grands problèmes combinatoires [10]. Dans le même temps, après dix années passées à Marseille, je montais à Paris. J'avais décroché un poste au Bouygues e-lab, le département Innovation et Optimisation du Groupe Bouygues, dont le rayonnement au sein de la communauté française de RO m'avait ébloui.

2008-2012 : recherche locale (mixte)

En 2008, un grand groupe industriel français sollicitait le Bouygues e-lab pour résoudre un problème d'optimisation de tournées de véhicules avec gestion des stocks. Ce problème a la particularité de combiner des décisions combinatoires (quels clients livrer ? avec quelles ressources les livrer ?) et des décisions continues (quelles quantités de produit livrer ?). La complexité du problème est gigantesque : des centaines de clients, des dizaines de ressources, un horizon temporel d'une dizaine de jours avec un pas de temps à l'heure. Tandis que les temps d'exécution étaient limités (5 minutes), les exigences de qualité étaient élevées : les performances de l'algorithme de résolution devaient dépasser celle d'un prototype déjà développé. Bien que les applications de la recherche locale soient rares concernant l'optimisation en variables mixtes, nous avons décidé d'attaquer le problème en adaptant notre fameuse recette [14]. Une des caractéristiques de notre approche est qu'elle est pure (sans hybridation) et directe (sans décomposition). En effet, face à des problèmes très combinatoires, notre expérience est qu'il est préférable d'optimiser de façon heuristique (par recherche locale) sur l'espace originel des solutions, que de façon exacte sur un espace largement tronqué. Surgit alors une difficulté lors de l'évaluation des mouvements. Une fois les décisions combinatoires modifiées, des contraintes portant sur les décisions continues sont violées. Il faut alors recouvrer la faisabilité en modifiant des décisions continues. Cela revient à résoudre un sous-problème induit par ces décisions. Dans le cas présent, ce sous-problème est un problème de flot maximum. Facile, me direz-vous : un flot maximum se calcule en temps polynomial. Mais une fois encore, le rapport entre la qualité du voisinage et le temps passé à l'explorer est crucial : dépenser un temps $O(n^3)$ à chaque itération pour obtenir les quantités de livraison optimales eut ralenti fortement la convergence de la recherche locale. Nous avons donc préféré un algorithme incrémental (s'appuyant sur le flot courant) et approché (retournant un flot non nécessairement maximal) dont la complexité en temps est

quasi constante en pratique. Cet algorithme, à défaut d'être exact, s'exécute 2000 fois plus vite qu'un algorithme classique de flot maximum, tandis que le flot retourné est à moins de 2 % de l'optimum. De cette façon, nous sommes parvenus à explorer 10 millions de solutions en 5 minutes. La qualité des solutions produites par la recherche locale s'est avérée excellente : d'un coût inférieur à 20 % à celui des solutions fournies par un algorithme classique d'insertion.

Cette approche pour l'optimisation mixte a été rééditée à l'occasion du Challenge ROADEF 2010 posé par EDF sur la planification des arrêts de centrales nucléaires [12]. Ici le sous-problème continu est encore plus grand (1 milliard de variables) du fait de centaines de scénarios servant à modéliser les aspects stochastiques du problème. De plus, il n'est pas linéaire, ce qui complique encore les choses. Beaucoup d'équipes ont traité celui-ci en le réduisant à un programme linéaire (de type flot) sur un ensemble de scénarios représentatifs. De notre côté, nous l'avons résolu par un algorithme incrémental, approché et randomisé, en pratique 10 000 fois plus rapide qu'une approche par programmation linéaire. C'est ce qui nous a sans doute permis d'obtenir les meilleurs résultats sur les instances A et B... avant de chuter à cause d'un bogue dans notre code. Cela me permet de revenir à la question de la qualité logicielle : vous n'optimiserez rien du tout avec des résultats erronés ! Au contraire, vous créerez plus de problèmes que vous n'apporterez de solutions. Aussi les méthodes itératives comme la recherche locale demandent-elles une rigueur toute particulière lors du développement. Je résumerai simplement notre approche de la qualité logicielle aux deux assertions suivantes. Tout résultat (produit par un code) qui n'est pas vérifié (par un autre code) est faux avec une probabilité de presque 1. Si vous n'attrapez pas une anomalie à l'itération où elle se produit, alors la probabilité que vous l'attrapiez un jour est de presque 0.

À travers cette méthodologie pour la recherche locale, nous avons cherché à industrialiser notre développement de solutions d'optimisation [17]. Cela nous a permis de produire en quelques dizaines de jours, tout au plus quelques centaines, des logiciels d'optimisation offrant des retours sur investissement se chiffrant en millions d'euros pour nos clients. Tout comme l'informatique, la RO doit s'industrialiser afin d'être moins chère, moins risquée, plus fiable, plus profitable pour ses clients, entreprises privées ou services publics. Au risque de surprendre voire de choquer, la résolution des problèmes que nous rencontrons au Bouygues e-lab

nous posent peu de difficultés techniques, aussi complexes et grands soient-ils. Nous parvenons à satisfaire nos clients que ce soit en leur fournissant des solutions optimales ou pas : l'essentiel pour eux étant le gain entre les solutions qu'ils avaient et les solutions que nous leur fournissons.

2007-2012 : LocalSolver

Suite à ces expériences fructueuses autour de la recherche locale et cette volonté d'industrialiser nos développements, nous est venue l'idée d'utiliser la puissance de cette technique au sein d'un solveur de programmation mathématique. En effet, nous avons remarqué qu'au fil des projets nos recherches locales se ressemblaient : les mouvements étaient souvent les mêmes (insérer, supprimer, déplacer, échanger), de même que les opérations dont nous exploitons les invariants (somme, minimum, maximum, et, ou).

Durant l'été 2007, nous profitons d'un creux d'activité au Bouygues e-lab pour débiter avec Thierry Benoist le développement d'un solveur d'un type nouveau, fondé non plus sur les techniques classiques de recherche arborescente mais sur la recherche locale. Le projet LocalSolver [15] débutait. Très vite, mes collègues marseillais Bertrand puis Karim nous rejoignaient sur le projet. Après quelques expériences, nous comprîmes que le formalisme standard de la PLNE n'est pas adapté à une résolution par recherche locale. Il n'est pas assez expressif : les variables de décision ne sont même pas clairement identifiées. Or il est crucial que les variables de décision soient distinguées des variables intermédiaires, qui ne servent qu'à poser les contraintes et les objectifs du modèle. Une fois ce formalisme défini, une difficulté majeure est que des mouvements du type flip de variables 0-1 sont impuissants à faire progresser efficacement la recherche, que ce soit pour trouver une solution admissible ou bien pour optimiser. Les solveurs SAT basés sur la recherche locale sont confrontés à cette même difficulté : c'est une des raisons pour lesquelles ils sont peu performants face à des problèmes combinatoires très structurés, comme fréquemment rencontrés en pratique. Une remarque importante est que des flips réalisés de façon quelconque ne simulent en aucun cas les mouvements qu'implémenterait le praticien averti. Or nous étions (et restons) convaincus que la clé pour parvenir à un solveur par recherche locale efficace est "qu'il fasse ce qu'un expert ferait", tant du point de vue des mouvements que de la machinerie d'évaluation. Pour les mouvements, cela signifie qu'à chaque itération, les décisions doivent être modifiées

de façon à maintenir la faisabilité des contraintes, en particulier celles qui induisent la structure combinatoire du problème. C'est ainsi que nous avons implémenté nos premiers mouvements "autonomes", qui peuvent être vus comme des chaînes ou des cycles d'éjection dans l'hypergraphe de dépendances entre décisions et contraintes : après avoir flippé une décision 0-1, on répare les contraintes violées suite à cette modification en flippant d'autres décisions. Ce principe peut être généralisé (imaginez des cheminement quelconques dans cet hypergraphe, notamment arborescents) ou bien spécialisé en fonction du type des contraintes violées (contraintes d'égalité et d'inégalité se réparent différemment, par exemple). Du concept simple on bascule alors dans une ingénierie algorithmique complexe...

Début 2010, LocalSolver devenait opérationnel : sa version 1.0 était lancée à l'occasion du congrès ROADEF. Elle était intégrée au sein d'une première application pour TF1 Publicité. Puis les applications s'enchaînèrent dans le Groupe Bouygues (ETDE, Bouygues Telecom, Bouygues SA), mais aussi en dehors avec la société de services Eurodecision. Fort de ses succès, nous avons décidé avec l'appui de nos tutelles (Bouygues, Aix-Marseille Université, CNRS) de transformer le projet scientifique en produit commercial. Bien qu'induisant des contraintes et objectifs d'une tout autre nature, l'aventure n'en est que plus passionnante à mon sens. Début 2012, nous avons donc sorti la version commerciale, LocalSolver 2.0, qui néanmoins reste gratuit pour l'enseignement et la recherche (jetez un coup d'œil à localsolver.com). Cette version permet de résoudre des problèmes d'optimisation combinatoire comportant plusieurs millions de variables, et ce en quelques minutes. Mais il nous reste fort à faire : nous prévoyons de sortir des versions 2.1 (juin 2012) et 2.2 (octobre 2012) comportant de nombreuses améliorations techniques (plus de performance) et fonctionnelles (plus d'expressivité). Quant à la version 3.0 qui sera présentée au congrès ROADEF 2013, elle offrira la possibilité d'attaquer des problèmes non-linéaires en variables mixtes de très grande taille. Cela devrait éveiller la curiosité des plus sceptiques...

Retours d'expérience

Pour conclure, je vous propose quelques retours d'expérience, à destination des plus jeunes notamment. Loin de moi l'idée de vous donner des leçons. J'ai découvert certaines choses au fil des années, qui m'auraient été utiles de savoir plus tôt. Je me permets d'en présenter quelques unes ici.

Un bon praticien doit maîtriser les fondements

mathématiques et algorithmiques de la RO. Ils sont les outils intellectuels et pratiques, et par conséquent indispensables, de l'ingénieur en RO. Néanmoins, il est important de ne pas confondre la RO et les mathématiques de la RO. La RO sur le terrain, c'est 80 % d'informatique (notamment en gestion des données et génie logiciel). Enfin, les qualités premières d'un bon ingénieur (RO ou pas) sont le sens du service et le pragmatisme, ne l'oubliez jamais. J'invite le lecteur intéressé par ces sujets à consulter [17].

Concernant les jeunes chercheurs, je ne leur donnerai ici que deux conseils. Apprenez à écrire des articles... seul ! La rédaction d'articles scientifiques (en anglais) est une composante majeure du métier de chercheur, aujourd'hui plus que jamais. Mais la communication scientifique, c'est aussi de la "com". N'ayez donc pas peur d'en faire et utilisez pour cela tous les canaux : colloques, magazines, internet. Gardez tout de même à l'esprit que sans fond scientifique, vous ne serez pas crédible. Cela me permet d'enchaîner avec mon second conseil. Soyez ambitieux, soyez utiles : intéressez-vous aux sujets majeurs, que ce soient des sujets de recherche fondamentaux, appliqués ou méthodologiques.

En définitive, je m'aperçois que nous n'avons pas répondu à la question initiale : la RO, c'est quoi ? Des théorèmes ou des logiciels ? Voici une jolie réponse donnée par Francis Sourd après mon exposé à Angers : des théorèmes *pour* des logiciels. Pour ma part, je ne sais pas trop ce que c'est. Je citerai simplement Miles Davis : *I don't know what jazz is. My experience is that... If the music is played well, it's good. If it's not played well, it's not so good.*

Remerciements. Je ne remercierai jamais assez les personnes avec qui j'ai collaboré ces dix dernières années, grâce à qui j'ai tant appris, grâce à qui j'ai vécu de si beaux projets. Certaines sont déjà citées dans le texte ci-dessus ; d'autres apparaissent dans les références ci-dessus. Enfin, j'ai une pensée pour mon père, Max Gardi (1947–2011), qui loin du monde de la recherche m'a appris l'essentiel...

Là où il y a une volonté, il y a un chemin.

Gaston Rébuffat (1921–1985)
alpiniste français, né à Marseille

Références

- [1] F. Gardi (2003). Efficient algorithms for disjoint matchings among intervals and related problems. In *Proceedings of DMTCS 2003*, LNCS 2731, pp. 168-180. Springer.
- [2] F. Gardi (2004). On partitioning interval and circular-arc graphs into proper interval subgraphs with applications. In *Proceedings of LATIN 2004*, LNCS 2976, pp. 129-140. Springer.
- [3] F. Gardi (2006). Mutual exclusion scheduling with interval graphs or related classes : complexity and algorithms. *4OR* 4(1), pp. 87-90. (PhD abstract)
- [4] B. Estellon, F. Gardi, K. Nouioua (2006). Large neighborhood improvements for solving car sequencing problems. *RAIRO Operations Research* 40(4), pp. 355-379.
- [5] F. Gardi, A. David (2007). Optimisation de plans de financement immobiliers : de la recherche opérationnelle en actuariat bancaire. *Bulletin Français d'Actuariat* 7(13), pp. 107-121.
- [6] F. Gardi (2007). The Roberts characterization of proper and unit interval graphs. *Discrete Mathematics* 307(22), pp. 2906-2908.
- [7] F. Gardi (2008). Mutual exclusion scheduling with interval graphs or related classes. Part II. *Discrete Applied Mathematics* 156(5), pp. 794-812.
- [8] B. Estellon, F. Gardi, K. Nouioua (2008). Two local search approaches for solving real-life car sequencing problems. *European Journal of Operational Research* 191(3), pp. 928-944.
- [9] F. Gardi (2009). Mutual exclusion scheduling with interval graphs or related classes. Part I. *Discrete Applied Mathematics* 157(1), pp. 19-35.
- [10] B. Estellon, F. Gardi, K. Nouioua (2009). High-performance local search for task scheduling with human resource allocation. In *Proceedings of SLS 2009*, LNCS 5752, pp. 1-15. Springer.
- [11] F. Gardi (2010). Optimisation de plans de financement immobiliers. *RAIRO Operations Research* 44(3), pp. 207-239.
- [12] F. Gardi, K. Nouioua (2011). Local search for mixed-integer nonlinear optimization : a methodology and an application. In *Proceedings of EvoCOP 2011*, LNCS 6622, pp. 167-178. Springer.
- [13] F. Gardi (2011). On partitioning interval graphs into proper interval subgraphs and related problems. *Journal of Graph Theory* 68(1), pp. 38-54.
- [14] T. Benoist, B. Estellon, F. Gardi, A. Jeanjean (2011). Randomized local search for real-life inventory routing. *Transportation Science* 45(3), pp. 381-398.
- [15] T. Benoist, B. Estellon, F. Gardi, R. Megel, K. Nouioua (2011). LocalSolver 1.x. A black-box local-search solver for 0-1 programming. *4OR* 9(3), pp. 299-316.
- [16] B. Estellon, F. Gardi. Car sequencing is NP-hard : a short proof. (à paraître dans *Journal of the Operational Research Society*)
- [17] T. Benoist, F. Gardi, A. Jeanjean. Lessons learned from 15 years of operations research for French TV channel TF1. (à paraître dans *Interfaces*)