

First month progress report

Parallélisation d'une recherche arborescente hybride et partielle

Lieu : Unité Mathématiques et Informatique
Appliquées de Toulouse, INRA,
Castanet-Tolosan

Dates : du 18/03/2018 au 15/09/2018

Abdelkader BELDJILALI

18 avril 2019

1 Objectif du projet de fin d'étude

Le projet porte sur la parallélisation d'un algorithme de recherche arborescente par évaluation et séparation (Branch and Bound : B&B) hybride baptisé HBFS (Hybride Best First Search). HBFS combine, en effet, un parcours de l'arbre de recherche de type "Best first search", ou BFS, à une recherche en profondeur bornée, DFS ou "Bounded Depth first search". Cet algorithme est implémenté dans le solveur C++ pour réseaux de fonctions de coût *toulbar2* (<https://github.com/toulbar2/toulbar2>, <http://www7.inra.fr/mia/T/toulbar2/>). HBFS fournit une solution exacte si elle existe. HBFS a été développé par l'équipe SaAB [2] et a été récemment étendu dans [9].

Description succincte du formalisme : Le problème de satisfaction de contraintes valuées (VCSP) est décrit dans [17],[13]. C'est un formalisme général qui permet, contrairement aux problèmes de satisfaction de contraintes *dures*, d'exprimer des préférences entre variables et donc entre solutions ou de traiter des problèmes sur-contraints pour lesquels certaines contraintes peuvent éventuellement être violées. Le problème de satisfaction de contraintes pondérées (*Weighted Constraint Satisfaction Problem (WCSP)*) constitue un cas particulier de VCSP. Les réseaux de contraintes valuées ou réseaux de fonctions de coût (CFN : Cost Function Network) définissent une classe de modèles graphiques. En effet, l'ensemble X des variables, D de leurs domaines respectifs, F des fonctions de coût et d'une structure de valuation V peuvent être représentés sous

forme d'un hypergraphe plus ou moins détaillé selon qu'on veuille capturer la structure globale du problème ou sa micro-structure qui intègre l'information sur les domaines des variables et les fonctions de coût [13]. Un exemple de graphe décrivant la micro-structure d'un problème est donné dans [18].

Fonction de coût globale Les fonctions de coût locales, i.e. qui ne concernent qu'un sous-ensemble de variables, sont agrégées pour obtenir une "valuation" globale du problème, c'est à dire, une fonction de coût globale en les variables du problème. Il s'agit alors de trouver le, ou les tuples, de valeurs à associer à toutes les variables du problème, c'est à dire de trouver *une affectation complète*, qui minimise la fonction de coût globale ou valuation globale.

$$Val(a_1, \dots, a_n) = w_\emptyset + \sum_{i=1}^n f_i(a_i) + \sum_{f_{ij} \in C} f_{ij}(a_i, a_j)$$

Exemple de fonction de coût dans le cas simple de contraintes unaires :

Le fait de remplacer ces contraintes dures par des fonctions de coûts d'arité 1, appelées également **contraintes valuées** ou **contraintes souples** permet par exemple de modéliser une préférence pour certaines des valeurs que peuvent prendre les variables du problème d'où le terme de contraintes souples ; une fonction de coût telle que $f_x(b) = 0$ modélise le fait que la valeur b pour la variable X est autorisée. A contrario, la valeur b est interdite si on écrit la contrainte souple $f_x(b) = k$ où k est un entier positif maximal fixé plus ou moins arbitrairement en fonction du problème. k est parfois noté $+\infty$. Ainsi, les fonctions de coût de valeurs 0 et k modélisent des contraintes dures. Les valeurs entre 1 et $k-1$ permettent quant à elles de graduer les préférences liées à la nature du problème à modéliser. Par exemple, $f_x(c) = 2$ signifierait que l'on n'interdit pas la valeur 2 pour x mais qu'on lui préfère la valeur b . Ceci permet en particulier de modéliser des problèmes sur-contraints mais dont on voudrait quand même une solution car les contraintes n'ont pas la même importance et le décideur ou modélisateur peut faire des concessions sur certaines d'entre elles.

Propagation de contraintes souples dans l'hypergraphe des fonctions de coût :

Des algorithmes de propagation de contraintes souples ont été développés par l'équipe SaAB, en particulier *l'existential directional arc consistency (EDAC)* et implémentés dans toulbar2 qui se basent sur des opérations d'extensions et de projections, opérations qui conservent la fonction de coût globale, et qui permettent in fine d'obtenir une borne inférieure de cette fonction de coût globale [7]. L'idée est de concentrer les coûts sur une seule variable ce qui peut déclencher une augmentation de f_\emptyset via une projection unaire. f_\emptyset est une fonction de coût constant positif, i.e. d'arité nulle donc indépendante de toute variable. f_\emptyset est une borne inférieure de l'optimum car la fonction de coût globale s'écrit comme la somme de f_\emptyset et d'une quantité positive Q : $Val = f_\emptyset + Q$. Il s'agit de trouver la bonne heuristique qui donnera la bonne séquence de projections et extensions pour maximiser f_\emptyset et donc se rapprocher de l'optimum. Parfois, la propagation suffit à trouver l'optimum. f_\emptyset , ou w_\emptyset , selon la notation choisie, sont calculées pour chaque nœud de l'arbre de recherche. Dans [18] est présenté la notion de propagation de contraintes souples à l'aide de 3 types de transformations : projection, extension et projection unaire comme en figure 1.

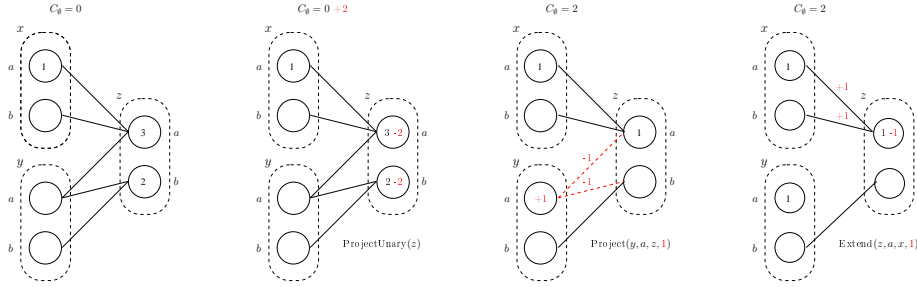


FIGURE 1 – Trois types de transformations de base

Utilisation du minorant obtenu par propagation dans un branch and bound Cette propagation par extension et projection dans l'hypergraphe, ou réseau, des fonctions de coût fournit une borne inférieure w_\emptyset de l'optimum en chaque nœud de l'arbre de recherche du B&B [2]. w_\emptyset est un minorant de l'optimum du sous-problème associé à un nœud. Par ailleurs dès qu'on obtient une affectation complète des variables du problème, i.e on arrive à une feuille de l'arbre de recherche, on peut calculer une valeur V pour la fonction de coût globale. La borne supérieure UB du problème global qui constitue la meilleure solution courante, ou solution titulaire, ou incumbent value est mise à jour avec cette valeur V si $UB \geq V$. Dès que $w_\emptyset \geq UB$, on pourra élaguer l'arbre de recherche, ie supprimer le nœud en question qui ne peut donner de solution et tous ses descendants non plus car les w_\emptyset des descendants sont tous supérieurs ou égaux au w_\emptyset du nœud parent.

Livrables demandés par le service Il s'agit dans un premier temps de fournir une version parallélisée de l'algorithme HBFS (Hybrid Best First Search) dans un contexte de mémoire distribuée en utilisant par exemple le standard Message Passing Interface (MPI) ainsi que les benchmarks associés (itération 1) et d'assurer le transfert de compétences nécessaires au fur et à mesure du projet. D'autres développements pourront suivre, notamment concernant l'algorithme BTB-HBFS en fonction du temps disponible.

2 Environnement de travail

Le stage se déroule au sein de l'équipe Statistiques et Algorithmique pour la Biologie (SaAB) qui fait partie de l'Unité Mathématiques et Informatique Appliquées (MIAT) de l'INRA, située à Castanet-Tolosan dont on trouvera ci-après en figure 2 l'organigramme. L'encadrement est assuré par les docteurs Simon De Givry et David Allouche. Les moyens administratifs, informatiques et logistiques sont parfaitement adaptés aux exigences du stage en particulier un PC de travail performant, des serveurs et la présence d'un data center sur le site.

L'équipe SaAB a pour objectif de développer et de mettre à disposition des biologistes des méthodes mathématiques, statistiques et informatiques permettant de contribuer à la compréhension du vivant. Elle s'intéresse à la localisation et l'identification d'éléments fonctionnels dans les génomes des bactéries, plantes

et animaux, et de façon croissante aux interactions qui existent entre ces différents éléments. Pour traiter ces problèmes, l'équipe mobilise et développe des méthodes en mathématiques, statistiques, probabilités (modélisation, inférence, modèles de mélanges de lois, régression pénalisée, modèles graphiques stochastiques, processus) et en informatique (modélisation, optimisation combinatoire, réseaux de contraintes, modèles graphiques déterministes, algorithmique) avec le but de valoriser les méthodes développées dans des outils logiciels directement utilisables par nos partenaires biologistes et rendant compte le mieux possible de la complexité et de la variété des données utilisables et en capitalisant les développements méthodologiques dans des logiciels génériques, éventuellement déclinés ensuite sur différentes applications.

L'équipe développe en particulier des méthodes originales dans le domaine de l'optimisation combinatoire, en s'appuyant sur les réseaux de contraintes pondérées, aussi appelés "réseaux de fonctions de coût", un modèle graphique dédié à l'optimisation et généralisant les réseaux de contraintes utilisés en programmation par contraintes et proches des Champs de Markov. Ces techniques, implémentées dans l'outil *toulbar2* (développé dans l'équipe et très bien placé dans différentes compétitions internationales), sont ensuite mises en œuvre sur des problèmes issus de la bioinformatique (design de protéines, localisation d'ARNs de familles connues, diagnostics de pedigrees complexes de grande taille...).



FIGURE 2 – organigramme MIAT-INRA

3 Planning prévisionnel

Le travail effectué jusqu'ici à la date de rédaction du présent rapport d'étape porte sur la bibliographie relative ou connexe au sujet du stage et à la recherche de stratégies de parallélisation en lien avec les encadrants.

Il est prévu de procéder par itérations successives. L'itération 1 visera à produire un code fonctionnel répondant au cahier des charges du projet concernant HBFS. Les itérations suivantes tenteront le cas échéant de trouver des solutions plus performantes pour le HBFS tirées des réflexions et de l'expérience acquise ou de l'analyse d'éventuels goulots d'étranglement et de l'adapter au BT-D-HBFS (Backtracking with Tree Decomposition-HBFS).

Mars-Juin : itération 1

1. Bibliographie ([12], [1], [16], [10], [15], [4])
 - wscsp
 - consistance d'arc souple
 - propagation de contraintes souples
 - parallélisation de recherches arborescentes

- Architectures logiciel : MPI, OpenMPI, OpenMP, Pthread
 - Architectures matériel du laboratoire : serveurs MIAT, cluster genotoul
2. prise en main de l'environnement de travail et du code.
 3. Recherche et analyse de stratégies de parallélisation.
 4. codage/benchmarks/analyse des performances. Les benchmarks sont disponibles à <https://forgemia.inra.fr/thomas.schiex/cost-function-library>.

Juillet-Août : itérations suivantes et autres selon temps disponible

1. Comparaison avec un travail similaire mené dans le cadre des modèles graphiques([10])
2. Comparaison à des travaux issus de la recherche opérationnelle.
3. Piste de recherche 1 : adaptation de la méthode à une variante proposée dans (Larrosa, 2016, [11]).
4. Piste de recherche 2 : intégration dans une métaheuristique parallèle [15]
5. rédaction rapport, beamer

Août :

1. rédaction rapport (suite)
2. Présentation beamer
3. Fin codage/benchmarks. Les benchmarks sont disponibles à <https://forgemia.inra.fr/thomas.schiex/cost-function-library>.

4 Approche initiale pour traiter le sujet

Parmi les approches possibles, il sera exploré lors de la première itération l'approche Embarrassingly Parallel Search (EPS) décrite dans [16] et le paradigme Master-Worker[8].

Références

- [1] D. ALLOUCHE, S. De GIVRY et T. SCHIEX. *Towards Parallel Non Serial Dynamic Programming for Solving Hard Weighted CSP*.
- [2] David ALLOUCHE et al. “Anytime Hybrid Best-First Search with Tree Decomposition for Weighted CSP”. In : *Principles and Practice of Constraint Programming*. Sous la dir. de Gilles PESANT. Cham : Springer International Publishing, 2015, p. 12-29. ISBN : 978-3-319-23219-5.
- [3] Martin COOPER et al. “Soft arc consistency revisited”. In : *Artificial Intelligence* 174 (mai 2010), p. 449-478. DOI : [10.1016/j.artint.2010.02.001](https://doi.org/10.1016/j.artint.2010.02.001).
- [4] Youssef EDITORS : HAMADI et Lakhdar (Eds.) SAIS. *Handbook of Parallel Constraint Reasoning - , Springer, 2018*. Springer, 2018.
- [5] Ian P. GENT et al. *A Preliminary Review of Literature on Parallel Constraint Solving*.
- [6] Simon de GIVRY, Thomas SCHIEX et Gérard VERFAILLIE. “Exploiting Tree Decomposition and Soft Local Consistency In Weighted CSP”. In : *AAAI*. 2006.
- [7] Simon de GIVRY et al. “Existential arc consistency : Getting closer to full arc consistency in weighted CSPs”. In : 2005.
- [8] “Parallel Solvers for Mixed Integer Linear Optimization”. In : *Handbook of Parallel Constraint Reasoning*. Sous la dir. d’Youssef HAMADI et Lakhdar SAIS. Springer International Publishing, 2018. Chap. Parallel Solvers for Mixed Integer Linear Optimization, p. 283-336.
- [9] Philippe JÉGOU, Hélène KANSO et Cyril TERRIOUX. “Adaptive and Opportunistic Exploitation of Tree-Decompositions for Weighted CSPs”. In : nov. 2017. DOI : [10.1109/ICTAI.2017.00064](https://doi.org/10.1109/ICTAI.2017.00064).
- [10] Akihiro KISHIMOTO, Radu MARINESCU et Adi BOTEAN. “Parallel Recursive Best-first AND/OR Search for Exact MAP Inference in Graphical Models”. In : *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. NIPS’15. Montreal, Canada : MIT Press, 2015, p. 928-936. URL : <http://dl.acm.org/citation.cfm?id=2969239.2969343>.
- [11] Javier LARROSA, Emma ROLLON et Rina DECHTER. “Limited Discrepancy AND/OR Search and Its Application to Optimization Tasks in Graphical Models”. In : *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. IJCAI’16. New York, New York, USA : AAAI Press, 2016, p. 617-623. ISBN : 978-1-57735-770-4. URL : <http://dl.acm.org/citation.cfm?id=3060621.3060708>.
- [12] Bernard MANS, Thierry MAUTOR et Catherine ROUCAIROL. “A parallel depth first search branch and bound algorithm for the quadratic assignment problem”. In : *European Journal of Operational Research* 81.3 (mar. 1995), p. 617-628. URL : <https://ideas.repec.org/a/eee/ejores/v81y1995i3p617-628.html>.
- [13] Simon de Givry MARTIN C. COOPER et Thomas SCHIEX. “Valued Constraint Satisfaction Problems”. In : sous la dir. de SPRINGER. T. 2. Chap. 7.

- [14] Tarek MENOUEUR, Bertrand LECUN et P VANDER-SWALMEN. “Parallélisation d’un solveur de contraintes avec le framework parallèle BOBPP”. In : jan. 2013.
- [15] Abdelkader OUALI et al. “Iterative Decomposition Guided Variable Neighborhood Search for Graphical Model Energy Minimization”. In : *Conference on Uncertainty in Artificial Intelligence, UAI 2017*. Sydney, Australia, août 2017. URL : <https://hal.archives-ouvertes.fr/hal-01628162>.
- [16] Jean-Charles RÉGIN, Mohamed REZGUI et Arnaud MALAPERT. “Embarassingly Parallel Search”. In : *Principles and Practice of Constraint Programming*. Sous la dir. de Christian SCHULTE. Berlin, Heidelberg : Springer Berlin Heidelberg, 2013, p. 596-610. ISBN : 978-3-642-40627-0.
- [17] Thomas SCHIEX, Helene FARGIER et Gerard VERFAILLIE. “Valued Constraint Satisfaction Problems : Hard and Easy Problems”. In : *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*. IJCAI’95. Montreal, Quebec, Canada : Morgan Kaufmann Publishers Inc., 1995, p. 631-637. URL : <http://dl.acm.org/citation.cfm?id=1625855.1625938>.
- [18] *Weighted constraint satisfaction problem*. URL : https://en.wikipedia.org/wiki/Weighted_constraint_satisfaction_problem.