POSL: A Parallel-Oriented Solver Language

Mémoire présenté en vue de l'obtention du
grade de Docteur de l'Université de Nantes
sous le sceau de l'Université Bretagne Loire

Alejandro Reyes Amaro

École doctorale : Sciences et technologies de l'information, et mathématiques
Discipline : Informatique et applications, section CNU 27
Unité de recherche : Laboratoire d'Informatique de Nantes-Atlantique (LINA)

Directeur de thèse : M. Eric Monfroy, Professeur, Université de Nantes
Co-encadrant : M. Florian Richoux, Maître de Conferences, Université de Nantes

UNIVERSITÉ DE NANTES

Submitted: dd/mm/2016

Jury:

Président :	**M. Salvador ABREU**, *Professeur étranger*, Université d'Évora
Rapporteurs :	**M. Frédéric LARDEUX**, *Maître de conférences*, Université d'Angers
	M. Christophe LECOUTRE, *Professeur*, Université d'Artois
Examinateur :	**M. Arnaud LALLOUET**, *Chercheur industriel*, Huawei Technologies Ltd.

POSL: A Parallel-Oriented Solver Language

Short abstract:

The multi-core technology and massive parallel architectures are nowadays more accessible for a broad public through hardware like the Xeon Phi or GPU cards. This architecture strategy has been commonly adopted by processor manufacturers to stick with Moore's law. However, this new architecture implies new ways of designing and implementing algorithms to exploit their full potential. This is in particular true for constraint-based solvers dealing with combinatorial optimization problems.

Furthermore, the developing time needed to code parallel solvers is often underestimated. In fact, conceiving efficient algorithms to solve certain problems takes a considerable amount of time. In this thesis we present POSL, a Parallel-Oriented Solver Language for building solvers based on meta-heuristic, in order to solve Constraint Satisfaction Problems (CSP) in parallel. The main goal of this thesis is to obtain a system with which solvers can be easily built, reducing therefore their development effort, by proposing a mechanism of code reusing between solvers. It provides a mechanism to implement solver-independent communication strategies. We also present a detailed analysis of the results obtained when solving some CSPs. The goal is not to outperform the state of the art in terms of efficiency, but showing that it is possible to rapidly prototyping with POSL in order to experiment different communication strategies.

Keywords: Constraint satisfaction, meta-heuristics, parallel, inter-process communication, language.

Contents

Part I

Conclusions and future works

1

Conclusion and future works

In this chapter, the conclusions of this thesis is presented, emphasizing on our contribution and obtained results. Future branches to follow are also discussed.

Contents

1.1 Conclusions

The era of parallel computing has opened new and more efficient ways to solve constrained problems. This development is leading us to the multi/many–core technology and massive parallel architectures, which are nowadays more accessible for a broad public through hardware like the Xeon Phi or GPU cards. For that reason, this new architecture implies new ways for designing and implementing algorithms to exploit its full potential.

In this thesis I have presented as a main contribution a Parallel-Oriented Solver Language (POSL) focused in the solution of Constraint Satisfaction Problems, which are very complicated. These problems have huge search spaces, making them intractable through tree-search techniques. POSL propose a language to build meta-heuristic-based solvers, tacking into account the success of these methods solving CSPs. These meta-heuristics are built using the POSL's language following rigorous but well detailed steps, based on the re-usability and coupling small pieces of computation and communication (computation modules and communication modules), designed to the resolution of a broad range of CSPs.

Meta-heuristic methods have some times a lot of parameters to be adjusted. Prior to the POSL's design, Appendix **??** contains a study in which the tool ParamILS was used to tune *Adaptive Search* to solve *Costas Array* and *All-Interval Series* problems. The main goals of that work was studying the performance of the tool, and finding a new and more efficient parameters setting allowing faster resolution of the mentioned benchmark problems. However, after a comparison between obtained results using default parameters found through manually experiments, and obtained results using ParamILS, the conclusion was that, for this implementation of Adaptive Search, the tool is not able to find parameter settings improving obtained results using default parameters. This corroborates the practical intuition that, when the parameters set is not so large, the experience of the scientist is crucial and more accurate than using parameter tunning tools.

The most important characteristic of POSL is allowing the construction of many solvers to work in parallel using the *multi-walk* approach, which has shown very good results solving constrained problems. Into another work prior to POSL's design, I have presented a study of some techniques to improve the performance of algorithms proposed in [1] were a study of the impact of space-partitioning techniques on the performance of parallel local search algorithms is proposed to tackle the *K-Medoids Clustering Problem*. The basic idea of their specific problem is how allocate communication metronodes in order to maximize the client covering. Their solution is based on domain partitioning techniques like *space-filling curves*, and *k-Means* algorithm, but they do not take into account the number of clients associated to each new sub-domain. For that reason, in Appendix **??** are proposed a set of ideas/hypothesis to improve the performance, based on geometrical balancing of the search space. This work was not validated, because it was performed in parallel with the first ideas of POSL, which finally was the main direction of this thesis.

Chapter **??** was dedicated to POSL, the main contribution of this thesis, a Parallel-Oriented Solver Language to build interconnected meta-heuristic-based solvers working in parallel. The language was formally presented by defining each provided operator, as well as the benchmark codification method, and the process of creation/usage of the computation and communication modules.

The most important advantage of POSL is allowing the codding, easily and fast, of many different solvers through a mechanism of module re-usability, and communication strategies through communication operators, which are also formally defined. Hence, as other contribution of this thesis, is presented in Chapter **??** a detailed study of various communication strategies to analyze the behavior and relevance of the information sharing in solving constraint problems.

Solving *Social Golfers Problem*, it was successfully applied an exploitation-oriented communication strategy, in which the current configuration is communicated to focus various solvers in a more promising area. The same idea was applied to solve the *N-Queens Problem*, showing no better results than obtained without communication. However, a deep study of the POSL's behavior during the search process allows to design a communication strategy able to improve the results obtained using non-communicating strategies. It was based on crating *companion* solvers (solvers only searching into a portion of the search space) to accelerate other's solvers search, by communicating the current configuration. The cyclical exchange of this information shows good results for small instances. The *Costas Array Problem* is a very complicated constrained problem, and very sensitive to the methods to solve it. Thanks to some studies of different communication strategies, based on the communication of the current configuration at different times (places) in the algorithm, it was possible to find a communication strategy improving the performance, in comparison with those obtained without communication. Finally, the *Golomb Ruler Problem* was chosen to study a different and innovative communication strategy, in which the communicated information is a potential local minimum to be avoided. This new communication strategy showed to be effective to solve these kind of problems.

Thanks to the operator-based language provided by POSL it was possible to test many different strategies (communicating and non-communicating). The process of building solvers implementing different solution strategies is complex and tedious, but POSL gives the possibility to make communicating and non-communicating solver prototypes and to study them with few efforts. It was possible to show that a good selection and management of inter-solvers communication can play an important role during the search process, working with constrained problems, most of them very complicated.

1.2 Future works

POSL is a tool entirely developed within the context of this thesis. It was completely new and yet under optimization. Although it has shown its first results, I believe that there is still a long way to go.

One of the first steps to do when solving Constraint Satisfaction Problemsusing POSL is precisely the problem modeling. POSL at the language level does not provide a mechanism for benchmark modeling. It currently handles this issue through the low-level framework in C++ programing language, but the creation or integration of problem definition languages is one of the next goals.

During the resolution of *Social Golfers Problem* the success of a communication strategy combining intensification and exploration was showed. In that direction, many other strategies can be analyzed. For instance, the study of the cost history during the search process, in order to find a lower bound for the cost, that indicates the time to communicate the current configuration to all solvers. This strategy allows to focus the search in the same area, launching a generalized intensification.

One of the most costly process during the resolution of *Golomb Ruler Problem* was finding the right *parameters* for the *tabu list*. A set of values for each parameter was proposed, an after a tuning process, those showing the best behavior were chosen. Nevertheless, is clear that in this part there is a lot of room for improvement. The key of the good performance of this strategy is the right choosing of the proximity criterion between configurations. For that reason this issue deserves a deep study in the future.

POSL already has an important library of ready–to–use computation and connection modules, based on a deep study of classical meta-heuristics algorithms for solving combinatorial problems. In the near future we plan to make it grow, in order to increase possibilities of POSL. In such a way, building new algorithms by using POSL will be easier. At the same time we plan to enrich the language by proposing new operators. It is necessary, for example, to improve the solver definition language, allowing to build sets of many new solvers faster and easier. Furthermore, we are aiming to expand the communication definition language, in order to create versatile and more complex and dynamic communication strategies, to allow a communication strategy to change during runtime.

The operators described above only give the possibility to define static communication strategies. However, we aim to improve POSL with more expressive operators in terms of communication between solvers to allow dynamic modifications of communication strategies, that is, having such strategies adapting themselves during runtime. This way, many different communication strategies would be defined in the same solver set. Then, after some time of calculation, an evaluation would be performed in order to make all solvers able to adopt the best strategy until the end of the search process.

In most of the performed experiments, the shared information was the best found configuration. So far, there are no results showing what "a good information to communicate" is. Actually, [2] shows that in fact, the current configuration is not always a relevant information to share among solvers. That is why this subject deserves a deep study. We plan in the near future to investigate other informations to be communicated, such as search directions, search space features, among others.

POSL provides a mechanism of sharing not only data (i.e. configurations, neighborhoods, etc.) but also computation modules. This is the most interesting characteristic of POSL but it was not possible either to be applied or analyzed. With the data exchange, solvers can expand or change their search zones. However, sharing modules they can also mutate their behavior. In that sense, a future direction of this thesis propose the study of other benchmark in order to find the right problem to apply this approach successfully.

2

Bibliography

[1] Alejandro Arbelaez and Luis Quesada. Parallelising the k-Medoids Clustering Problem Using Space-Partitioning. In *Sixth Annual Symposium on Combinatorial Search*, pages 20–28, 2013.

[2] Yves Caniou, Philippe Codognet, Florian Richoux, Daniel Diaz, and Salvador Abreu. Large-Scale Parallelism for Constraint-Based Local Search: The Costas Array Case Study. *Constraints*, 20(1):30–56, 2014.