Towards Adaptive Space Partitioning for Large-scale Parallel Bayesian Optimization

Maxime Gobert, Jan Gmys, Nouredine Melab, Daniel Tuyttens

Towards Adaptive Space Partitioning for Large-scale Parallel Bayesian Optimization

Maxime Gobert[1], Jan Gmys[1], Nouredine Melab[2] and Daniel Tuyttens[1]

[1] Mathematics and Operations Research Department, University of Mons, Belgium
{maxime.gobert,jan.gmys,daniel.tuyttens}@umons.ac.be
[2] Univ. Lille, CNRS, Centrale Lille, Inria, UMR 9189 - CRIStAL - Centre de Recherche en Informatique Signal et Automatique de Lille, F-59000 Lille, France
nouredine.melab@univ-lille.fr

1 Context, motivations and objectives

Black-box optimization refers to the global optimization of an objective function (single real-valued) on which basically no information is available. Black-box optimization algorithms often require a large number of expensive function evaluations. The use of surrogate models and parallel computing – or a combination of both – allows to reduce the optimization time and the number of calls to the objective function. Among available methods, Bayesian Optimization (BO) is of particular interest. Introduced by Kushner *et al.* [1] in 1964 and later developed by Mockus *et al.* [2] in 1974, BO uses an Acquisition Function (or Infill Criterion) to guide the optimization process. The surrogate model is used to locate areas of the search space that will most likely improve the optimum value or the surrogate model. A well-known BO framework is Efficient Global Optimization (EGO), proposed by Jones *et al.* in [3]. It uses a Gaussian Process (GP) based surrogate model (also known as Kriging) which, in addition to a prediction of the objective function values, provides a quantification of its uncertainty. Based on Kriging predictions and their associated uncertainty, the Expected Improvement (EI) infill criterion provides a measure of how desirable it is to add a point to the database of evaluated solutions, balancing exploration and exploitation.

For example, EI is high in regions of high uncertainty (exploration), and in regions with predicted values close to the current optimum (exploitation). So that, adding the candidate that maximizes EI achieves a trade-off between the exploration of regions with low surrogate accuracy and exploitation of areas where the best known solution is located. Based on the assumption that EI can be improved, new infill criteria have been proposed within the context of BO, but none performs significantly better than EI.

The GP and EI coupling is a state-of-the-art method that is still competitive with other recently developed BO methods. However, especially if the objective function is implicitly defined by complex and large-scale numerical simulations, the execution time of the algorithm often remains very high. Therefore, it is important to enable BO algorithms like EGO to efficiently exploit the computational power provided by massively parallel high-performance computing platforms. Due to its inherently sequential nature, the parallelization of the framework is challenging. Several approaches involving batch parallel evaluations of the objective function have been proposed in the literature.

Based on EGO, Ginsbourger *et al.* proposed in [4] a multi-point version of EI (named q-EI) which provides q points per EGO-iteration to be added to the data set, which allows to perform function evaluations in parallel. For large values of q, exactly maximizing q-EI can become prohibitively expensive, so that heuristics are used to provide several distinct points. However, the difficulty to address the multi-point EI optimization (even with heuristics) makes the infill criteria computationally expensive. As a consequence, this way to compute the q-points EI does not allow to efficiently use high number of computational cores.

Multi-point parallel methods for BO can be grossly divided into two groups. The methods in the first group use the same infill criterion and surrogate model to propose q points. For instance, q-EGO from [4] proposes an approximated multi-points EI, gradient-based methods for the maximization of multi-point EI are used in [5] and niching evolutionary algorithms can be used to locate several local optima at once [6]. The methods in the second class combine different selection criteria. For example, Lyu *et al.* [7] use multi-objective optimization to obtain a Pareto front of optimal candidates according to different infill criteria and Wang *et al.* [8] use n infill criteria coupled with multi-points proposal inspired by Kriging Believer from [9] to get $n * q$ candidates per cycle.

Even though EGO and its parallel derivatives achieve excellent results with restricted budgets, several aspects of the approach can be improved. For instance, the scalability of existing parallel approaches suffers from the difficulty to provide q different candidate points to evaluate in parallel. The objective of this work is to tackle these issues by providing a large scale parallel version of EGO using a fast-to-compute EI-based multi-point acquisition function.

2 The proposed approach

As mentioned, a difficulty arising from the optimization of q-EI is that the process becomes computationally expensive for large batch-sizes. Exactly optimizing q-EI is not an option, and even if heuristics allow to lower the computation cost, improvement must still be done to take advantage of larger parallelization ($q > 32$). A common way to tackle large optimization problems is to split the search space and perform optimization on sub-spaces. No occurrence of sub-space decomposition applied to the BO framework has been found in the current literature. Therefore, we propose to investigate splitting strategy to the infill criterion optimization problem in order to provide distinct candidates for evaluation in a reasonable amount of time.

Algorithm 1 presents the idea of BO using space decomposition. As for classical BO, we consider three phases: first the initial sampling is created as shown in lines 1-2; second the surrogate model is fitted (line 6) and optimization of the infill criterion is performed (lines 7 to 11); finally the candidates are evaluated and integrated in the database as shown in lines 12 to 14 of the pseudo-algorithm. The splitting strategy is applied during the second phase so that the infill criterion is optimized in each sub-domain.

When partitioning the search space, the acquisition function optimization can be run in parallel in each sub-domain. For example, considering n computational cores, it is possible to use the exact optimization of two-points EI (available in [4]) in the n sub-domains in order to provide $2 * n$ candidates. A selection operates to keep a batch of n points to evaluate. The multi-point proposal in each sub-domain intends to avoid wasting budget in areas of low expected improvement. Thus, thanks to parallel computing, this infill criterion should remain reasonably costly while providing a large amount of distinct candidates to simulation and enhancing exploration.

Algorithm 1 Framework of the proposed algorithm

f: objective function
$\mathcal{D}$: search space
budget: budget
surrogate: Kriging meta-model
q: number of candidates per batch

1: $\boldsymbol{X} = [\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(n)}] \leftarrow$ initial_sampling(n, $\boldsymbol{D}$)
2: $\boldsymbol{y} \leftarrow f(\boldsymbol{X})$
3: $y_{min} \leftarrow$ get_best_cost($\boldsymbol{Y}$)
4: **while** $budget \neq 0$ **do**
5: $\quad \boldsymbol{X}_c \leftarrow \varnothing$ ▷ batch of candidate solutions
6: $\quad surrogate \leftarrow$ fit_Kriging_model $(\boldsymbol{X}, \boldsymbol{y})$
7: $\quad (\mathcal{D}_1, \dots, \mathcal{D}_n) \leftarrow$ split($\mathcal{D}$)
8: $\quad$ **for** $i \in [1, \dots, n]$ **do** ▷ parallelizable loop
9: $\qquad \boldsymbol{B_i} = ARGMAX_{\mathcal{D}_i}$(q-EI) ▷ one infill criterion per sub-domain
10: $\quad$ **end for**
11: $\quad \boldsymbol{B}_c \leftarrow$ select_best_candidates($\cup_{i=1}^{n} B_i$)
12: $\quad \boldsymbol{y}_c \leftarrow f(\boldsymbol{B}_c)$ ▷ parallel simulations
13: $\quad \boldsymbol{X} \leftarrow \boldsymbol{X} \cup \boldsymbol{B}_c$
14: $\quad \boldsymbol{y} \leftarrow \boldsymbol{Y} \cup \boldsymbol{y}_c$
15: $\quad (\boldsymbol{x}_{min}, y_{min}) \leftarrow$ get_best($\boldsymbol{X}$, $\boldsymbol{Y}$)
16: **end while**
17: **return** $\boldsymbol{x}_{min}$, y_{min}

3 Experimental protocol

The proposed strategy is tested in several classical benchmark functions chosen to represent known difficulties of Global Optimization such as multi-modal or noisy landscapes. We also tackle an optimization problem from a real-world application involving an expensive simulator. The simulator is constructed with the AuTuMN model which is designed to implement tuberculosis control programs and to estimate their efficiency [10]. In order to evaluate the scalability of the approach, experiments are performed on the Grid5000 [11] platform using up to 8 compute nodes with a total of 256 CPU cores. Extensive experiments are being conducted and the results will be presented at the OLA'20 conference.

4 Conclusion and ongoing works

Several space partitioning methods exist and can be compared in this context, furthermore it could be beneficial to dynamically adapt the partition size according to the number of high EI points found in a specific partition. Indeed, splitting the search space increases the exploration, but some areas are clearly not worth to be explored. To that end, two attached partitions with low EI may be merged while a region of high EI can be split. We are currently investigating such an approach. Furthermore, the use of parallelization makes the size of the data base increase fast, so that the Kriging model fitting might reach penalizing cost (growing cubically with the data set size). In this condition, alternate surrogate models such as Bayesian Optimization could replace the Kriging at a lower cost thanks to its incremental training. As an example, Snoek *at al.* [12] use deep neural networks to overcome this issue.

References

1. H. J. Kushner. A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise. *Journal of Fluids Engineering*, 86(1):97–106, 03 1964.
2. J. Močkus. On bayesian methods for seeking the extremum. In G. I. Marchuk, editor, *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*, pages 400–404, Berlin, Heidelberg, 1975. Springer Berlin Heidelberg.
3. Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, Dec 1998.
4. David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. A Multi-points Criterion for Deterministic Parallel Global Optimization based on Gaussian Processes. Technical report, March 2008.
5. Sébastien Marmin, Clément Chevalier, and David Ginsbourger. Differentiating the multipoint expected improvement for optimal batch design. pages 37–48, 2015.
6. Hao Wang, Thomas Bäck, and Michael Emmerich. Multi-point efficient global optimization using niching evolution strategy. 06 2014.
7. Wenlong Lyu, Fan Yang, Changhao Yan, Dian Zhou, and Xuan Zeng. Batch Bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3306–3314, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.
8. Yuan Wang, Zhong-Hua Han, Yu Zhang, and Wenping Song. Efficient global optimization using multiple infill sampling criteria and surrogate models. 01 2018.
9. David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. Kriging is well-suited to parallelize optimization. 2010.
10. James McCracken Trauer, Romain Ragonnet, Tan Nhut Doan, and Emma Sue McBryde. Modular programming for tuberculosis control, the "AuTuMN" platform. *BMC Infectious Diseases*, 17(1):546, Aug 2017.
11. Raphaël Bolze et al. Grid'5000: A large scale and highly reconfigurable experimental grid testbed. *The International Journal of High Performance Computing Applications*, 20(4):481–494, 2006.
12. Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Md. Mostofa Ali Patwary, Prabhat, and Ryan P. Adams. Scalable bayesian optimization using deep neural networks, 2015.