Space Partitioning with multiple models for Parallel Bayesian Optimization

Maxime Gobert, Jan Gmys, Nouredine Melab, Daniel Tuyttens

Space Partitioning with multiple models for Parallel Bayesian Optimization

Maxime Gobert[1], Jan Gmys[2], Nouredine Melab[3] and Daniel Tuyttens[1]

[1] Mathematics and Operations Research Department, University of Mons, Belgium
{maxime.gobert,daniel.tuyttens}@umons.ac.be
[2] Inria Lille - Nord Europe, France
jan.gmys@univ-lille.fr
[3] 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 and motivations

Bayesian Optimization (BO) is a global optimization framework that uses bayesian surrogate models such as Gaussian Processes (GP) to address black-box problems [1], [2] with costly-to-evaluate objective functions. Bayesian models and especially GPs are attractive for their ability to provide the uncertainty over their predictions. Using this information, one can build an indicator of utility for a point to be simulated. This indicator, named Infill Criterion (IC) or Acquisition Function (AF), is used to guide the optimization process and find valuable new point(s) to be exactly evaluated. Based on this procedure, Jones *et al.* [3] introduce the Efficient Global Optimization (EGO) algorithm that uses the Expected Improvement (EI) AF. Many approaches emerged from the idea of making EGO parallel. In particular, Ginsbourger *et al.* [4] introduced the q-dimensional EI criterion (q-EI), able to provide q candidate points when optimized. In [5], Ginsbourger *et al.* write that even though q-EI and its optimization methods are mathematically founded, it is not "mathematically tractable beyond a few dimensions". This motivates the introduction of the Kriging Believer (KB) and Constant Liar (CL) heuristics also presented in [5]. The two heuristics allow to approximate the optimization of q-EI at a much lower time cost. EGO using q-EI is called q-EGO in the following. Many other approaches based on EI or q-EI are constructed. For example, in [6] the authors use Infinitesimal Perturbation Analysis (IPA) to construct a gradient estimator of the q-EI surface, and a multi-start heuristic to find the set of points to evaluate. Zhan *et al.* [7] use a niching strategy to locate several optimal areas of the single point EI. Marmin *et al.* [8] write the analytical form of the multi-point EI gradient to be able to optimize the function with gradient information and reduce the computational cost compared to sequential heuristics or Monte Carlo sampling of [5]. However, none of the methods are efficient on parallel architectures including dozens of processing units. The creation of the batch of candidates is often time consuming, and the GP model fitting cost increases fast. The reference q-EGO algorithm has been experimented and driven to its limits in Briffoteaux *et al.* [9]. The analysis revealed that q-EGO performs well with small budgets (i.e. number of calls to the objective function) and small batches ($q \leq 8$). However, it suffers from early stagnation, poor scalability, and budget seems misspent since increasing q does not necessarily improve the final target for a given number of algorithm iterations (called cycles). The limits of q-EGO comes from at least two aspects: (1) we need to update q times the model to provide q candidate and the Kriging model becomes extremely time consuming to fit as the size of the learning set is increased; (2) the way the candidates are selected is not suited for large batches.

2 The proposed approach

From the previous assessment, our approach called Binary Space Partitioning EGO (BSP-EGO) revisits the way the candidates are selected - i.e. the Acquisition Process (AP). BSP-EGO uses EI as it is known to work consistently well on various problems. To preserve diversity in the candidate selection process BSP-EGO recursively splits the search space $\mathcal{D}$ into sub-domains. The partition can be seen following a binary tree where each node is a sub-domain and the leaves form a cover of $\mathcal{D}$ without overlap. One sub-AP is performed for each leaf using the same surrogate model fitted

Algorithm 1 multi-model BSP-EGO pseudo-code

Input
f: function to optimize, $\mathcal{D}$: search space
n_{init}: initial sample size, n_{cycle}: number of cycles, n_{learn}: learning size
depth: tree depth

1: $\boldsymbol{X},\ \boldsymbol{y} \leftarrow$ initial_sampling$(\mathcal{D}, n_{init}, f)$
2: $\mathcal{T} \leftarrow$ build_tree(*depth*)
3: **for** i in $1 : n_{cycle}$ **do**
4: $\quad\mathcal{B}_{candidates} \leftarrow \varnothing$
5: $\quad$**for** *leaf* in $\mathcal{T}$ **do** ▷ parallel loop
6: $\quad\quad\boldsymbol{X}_{leaf}, \boldsymbol{y}_{leaf} \leftarrow gather\_points(leaf, n_{learn})$
7: $\quad\quad model_{leaf} \leftarrow$ learn_model$(\boldsymbol{X}_{leaf}, \boldsymbol{y}_{leaf})$
8: $\quad\quad\boldsymbol{c} \leftarrow$ find_best_candidate$(\mathcal{D}_{leaf}, model_{leaf}, Infill_criterion)$
9: $\quad\quad\mathcal{B}_{candidates} \leftarrow \mathcal{B}_{candidates} \cup \boldsymbol{c}$
10: $\quad$**end for**
11: $\quad\mathcal{B}_{candidates} \leftarrow$ selection$(\mathcal{B}_{candidates})$
12: $\quad\mathcal{T} \leftarrow$ update_tree$(\mathcal{T}, \mathcal{B}_{candidates})$
13: $\quad\boldsymbol{X} \leftarrow \boldsymbol{X} \cup \mathcal{B}_{candidates}$
14: $\quad\boldsymbol{y} \leftarrow \boldsymbol{y} \cup f(\mathcal{B}_{candidates})$ ▷ parallel evaluation
15: **end for**
16: **return** $\boldsymbol{x}^*,\ y^*$

over all known points, and the best candidates regarding the chosen AF are selected. Then, the partition evolves according to the performance of each sub-domain. If the sub-domain is attractive (in terms of AF), it is split once more while the *worst* is merged to maintain a fixed number of sub-domains. The full description is available in [10]. BSP-EGO is designed to improve parallelism by providing an arbitrary big number of candidates from its AP, and to improve the balance in exploration/exploitation. Indeed, being able to dig deep into a sub-domain allows to sample more this area without neglecting others by forcing the sampling in sub-domains of *a priori* poor interest. Results of [10] show that even though we are able to increase the degree of parallelism, and improve the optimization in restricted time, the fast increasing time of GP fitting is still a bottleneck. Actually, the database is filling even faster as the number of cores increases since n_{batch} points are added per cycle.

This work investigates the use of multiple surrogate models (on per sub-domain) to limit the size of the learning set and lower the computational cost. For each sub-domain, the closest n_{learn} points the closest from the center of the sub-domain are collected to fit a GP model over them. Besides considerably lowering the fitting cost, this approach has several benefits. First, all models are distinct and independent, favoring diversity as experienced in Villanueva *et al.* [11] where space partitions and distinct surrogate models in sub-regions are used to locate several local optima. Second, the whole AP can be performed in parallel, meaning that each worker performs one fast GP fitting. Moreover, search space partitioning is reported as a viable method to deal with non stationary objective functions [12][13]. Algorithm 1 displays the pseudo-code of the multi-model BSP-EGO. It starts with the creation of an initial learning set, and the initialization of the tree $\mathcal{T}$. Then, until the budget runs out, BSP-EGO goes to every leaves (line 6 of Alg. 1) to create a GP model. Note that if the total number of points is smaller than n_{learn}, the model is fitted with all available points. Given an infill criterion, the local AP proposes a candidate for the considered leaf. Finally, a sub-set of candidates is exactly evaluated to be incorporated in the learning set.

3 Results and conclusions

Preliminary experiments have been conducted on three 6-dimensions benchmark functions to measure the walltime and solution quality as a function of the batch size n_{batch} and the size of the learning set n_{learn}. The functions are chosen with very distinct landscapes: the rosenbrock function is regular and unimodal with a large valley, the ackley function is noisy and multimodal, and the alpine function is highly multimodal but not noisy. The two firsts are optimized on the domain $\mathcal{D} = [-32, 32]^6$ and the last one on $\mathcal{D} = [0, 10]^6$. Experiments are performed on Lille site of the

Grid'5000 testbed [14]. Up to 4 nodes using 2 Intel Xeon E5-2630 v4 of 10 cores each are used. For a total of 48 cycles allocated to each experiment (whatever the batch size) we observe that the multi-model approach total execution time does not increase much with the batch size, which indicates a good scalability. With a learning size of 128, and a batch size of 32, this approach is up to 300 times faster compared to the global model one with only an acceptable reduction of the final target quality. To achieve equivalent results in terms of solution quality, we can use a more precise model learnt over 256 points, still with $n_{batch} = 32$, and have a speed up of approximately 54. This approach also makes possible the use of more than 64 computing cores that increases considerably the global evaluation budget (and the final objective value) in a restricted time. For example, running 48 cycles with $n_{batch} = 64$ and $n_{learn} = 128$ takes less than 3 minutes for non-expensive benchmark functions evaluations, which is barely more than the same experiment with $n_{batch} = 8, 16, 32$. The partitioning approach also seems to perform well when the landscape to optimize presents many local optima and needs more exploration. Indeed, each model focuses on its specific sub-region and is not influenced by data of distant regions. On the other side, it makes it difficult to compare the value of the infill criteria coming from different models and can have a negative effect when exploitation must be favored. The BSP-EGO with local-models algorithm is a promising step towards scalable EGO-like approaches. However, increasing the degree of parallelism, the second problem is even more present: how can one take advantage of this scalability to improve the candidate selection procedure (i.e. the AP) ? Recent developments in parallel BO (batch BO) seems to indicate that using different infill criteria working cooperatively is promising. Whereas the local-models BSP-EGO algorithm needs extensive testing and tuning, its framework is highly compatible with the idea of multiple AF operating cooperatively to select a good batch of candidates to exactly evaluate.

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 Jones. A taxonomy of global optimization methods based on response surfaces. *J. of Global Optimization*, 21:345–383, 12 2001.
4. David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. A multi-points criterion for deterministic parallel global optimization based on kriging. 03 2008.
5. David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. Kriging is well-suited to parallelize optimization. 2010.
6. Jialei Wang, Scott Clark, Eric Liu, and Peter Frazier. Parallel bayesian global optimization of expensive functions. 02 2016.
7. Dawei Zhan, Jiachang Qian, and Yuansheng Cheng. Balancing global and local search in parallel efficient global optimization algorithms. *Journal of Global Optimization*, 67(4):873–892, Apr 2017.
8. Sébastien Marmin, Clément Chevalier, and David Ginsbourger. Differentiating the multipoint expected improvement for optimal batch design. pages 37–48, 2015.
9. Guillaume Briffoteaux, Maxime Gobert, Romain Ragonnet, Jan Gmys, Mohand Mezmaz, Nouredine Melab, and Daniel Tuyttens. Parallel surrogate-assisted optimization: Batched bayesian neural network-assisted ga versus q-ego. *Swarm and Evolutionary Computation*, 57:100717, 2020.
10. Maxime Gobert, Jan Gmys, Nouredine Melab, and Daniel Tuyttens. Adaptive Space Partitioning for Parallel Bayesian Optimization. working paper or preprint, January 2021.
11. Diane Villanueva, Rodolphe Le Riche, Gauthier Picard, and Raphael Haftka. Dynamic design space partitioning for optimization of an integrated thermal protection system. 04 2013.
12. B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, Jan 2016.
13. Robert B Gramacy and Herbert K H Lee. Bayesian treed gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483):1119–1130, 2008.
14. Daniel Balouek, Alexandra Carpen Amarie, Ghislain Charrier, Frédéric Desprez, Emmanuel Jeannot, Emmanuel Jeanvoine, Adrien Lèbre, David Margery, Nicolas Niclausse, Lucas Nussbaum, Olivier Richard, Christian Pérez, Flavien Quesnel, Cyril Rohr, and Luc Sarzyniec. Adding virtualization capabilities to the Grid'5000 testbed. In Ivan I. Ivanov, Marten van Sinderen, Frank Leymann, and Tony Shan, editors, *Cloud Computing and Services Science*, volume 367 of *Communications in Computer and Information Science*, pages 3–20. Springer International Publishing, 2013.