Metin Türkay and Erdal Aydın (Eds.)
Proceedings of the 31st European Symposium on Computer Aided Process Engineering
June 6th to 9th, 2021, Istanbul, Turkey.

Data-driven Branch-and-bound Algorithms for Constrained Simulation-based Optimization

Jianyuan Zhai[a], Sachin Shirpurkar[a], Fani Boukouvala[a,*]

[a]*Georgia Institute of Technology, 311 Ferst Dr, Atlanta, GA, 30332, USA*

fani.boukouvala@chbe.gatech.edu

Abstract

The wide use of detailed simulations for complex systems has led to a growing interest for methods that can optimize simulation-dependent problems using data, without explicit equations or derivatives. Due to the lack of derivatives and the dependence on sampling, simulation-based optimization algorithms lack convergence guarantees and often require a significant number of samples to identify an optimal solution with consistency. Moreover, the presence of black-box constraints is an open challenge because it further complicates sampling and identification of unknown feasible spaces. Previously, we have introduced the Data-Driven Spatial Branch-and-Bound algorithm for box constrained problems, which employs data-driven convex underestimators, finds upper and lower bounds on the optimal objective value and progressively prunes suboptimal subspaces until convergence. In this work, we present recent advances of this framework for handling simulation-based and equation-based constraints. We demonstrate the performance of these features with respect to convergence and sampling requirements through benchmark constrained optimization problems.

Keywords: black-box optimization, simulation-optimization, surrogate modelling.

1. Introduction

An increasing amount of optimization applications in research and industry today require the embedding of various forms of data from simulations of various fidelity and/or scale. Such problems are challenging due to the inability to directly use efficient deterministic optimization solvers that require equation-based formulations. As a result, optimization of such problems is often referred to as "black-box" because it relies on input-output data. Many recent contributions from the engineering literature aim to apply and improve the performance of black-box optimization techniques for a wide variety of applications (Bhosekar et al., 2018). Optimization with embedded simulations can be performed with purely sampling-based algorithms, such as direct-search methods, genetic algorithms, particle swarm optimization, and many more that are reviewed in (Rios and Sahinidis, 2013, Boukouvala et al., 2016). Alternatively, a different class of methods employs approximations of data to perform optimization, and these are called model-based or surrogate-based techniques. Such techniques employ novel mechanisms for data collection, fitting of a surrogate model to represent the data, which then allows one to use equation-based optimization solvers. In order to avoid excessive sampling, most techniques in the literature use a small initial set of samples and subsequently adaptively "fit – optimize – resample" the space until their convergence criteria are met. This adaptive scheme is always based on techniques that are designed to balance "exploration" (i.e., searching the feasible space just enough to minimize probability of missing the optimal solution), and "exploitation" (i.e., focusing sampling in promising optimal regions where surrogate models need to be accurate). Recent work focuses on the identification and comparison of different types of surrogate models and their performance for optimization (Garud et al., 2019).

Despite recent advances in the simulation-based optimization literature, several open challenges still exist. First, many of these methods suffer from the curse-of-dimensionality, since sampling requirements increase when decision-variables increase. Second, when a surrogate model is used, there are open challenges when it comes to the selection of a surrogate model that is accurate and has a tractable mathematical representation so that it can in turn be optimized (ideally globally). Specifically, some popular surrogate models result to equations with nonconvex terms, and the number of terms and variables rapidly increase with the number of dimensions, model architecture, and even with number of data points. Recent work on global optimization of Neural Networks as surrogate models are tackling this challenge (Schweidtmann and Mitsos, 2019). Third, we have previously shown (Zhai and Boukouvala, 2020) that surrogate model selection and training leads to significant variation in the optimal results, because if a different surrogate model is used, or different training data are used, many algorithms provide a different result. This inconsistency in performance is highly undesirable for any optimization algorithm. Last, earlier work in black-box optimization led to algorithms that treat the problem as a pure "black-box", often without the ability to consider constraints. However, it is important to treat such problems as "grey-boxes", which allows one to mix known equations that are typically constraints together with an objective and constraints that are reliant on the simulation.

In our previous work, we have proposed an approach that aims to tackle some of the open challenges mentioned above (Zhai and Boukouvala, 2020). Specifically, instead of relying on a known type of surrogate model, we develop convex quadratic underestimators of data. The data may come from high-fidelity simulations, or multiple low fidelity surrogate models. The reason for this approach is to avoid several of the disadvantages of surrogate modelling, such as: (a) variation in parameters and model type with slight changes in data; (b) computational expense of training and selection of best model out of many; and (c) intractability of complex surrogate models when it comes to their global optimization. Moreover, our approach utilizes a spatial branch-and-bound (b&b) framework to search the space and focus sampling in promising spaces, while pruning spaces that are not optimal. This strategy allows us to incorporate more rigorous convergence criteria, such as absolute or relative ε-optimality gap (as opposed to reaching a maximum sampling or CPU limit). This Data-Driven Spatial Branch-and-Bound (DD-SBB) algorithm, also provides estimates of upper and lower bounds on the optimum at any intermediate stopping point. We have found that this approach is promising and obtains optimal solutions with few sampling points, however, it requires a large amount of samples to close the optimality gap.

In this work, we propose techniques for extending the capabilities of the DD-SBB approach in terms of constraints handling. We consider two different forms of constraints: (1) simulation-based constraints (i.e., constraints that are unknown in equation format but are embedded within the simulation) and, (2) equation-based constraints (i.e., constraints that are explicitly known algebraically as a function of decision variables). The type of formulations we aim to solve in this work is shown in Problem 1 (P1), where the objective function f and constraints g_s are embedded within a simulation and thus we do not have explicit equations for them. Constraints g_k are available in equation format. We only consider continuous variables that are bounded.

$$\text{(P1)} \min f(\boldsymbol{x})$$
$$s.t. \quad g_s(\boldsymbol{x}) \leq 0, \qquad s = 1,..,S$$
$$g_k(\boldsymbol{x}) \leq 0, \quad k = 1,\ldots,K$$
$$\boldsymbol{x^{lo}} \leq \boldsymbol{x} \leq \boldsymbol{x^{up}}, \qquad \boldsymbol{x} \in R^N$$

We implement and test different branching and bounds tightening techniques and test this extended capability of the algorithm through a set of benchmark problems. Our approach does not add computational cost to the algorithm because it does not require the fitting of surrogate models for the simulation-based constraints. Concepts from decision-tree techniques are explored for branching the search space cleverly into "feasible" and "infeasible" nodes and the performance of different branching rules are compared with respect to efficiency and convergence.

2. Methods

2.1. Data-Driven Spatial Branch-and-Bound Framework

The DD-SBB framework starts with initial sampling of the entire box-constrained space using Latin Hypercube sampling. Based on these high-fidelity samples from the simulation, a low-fidelity surrogate model can be fit, but this model is only used for ranking the importance of variables and for generation of large amounts of low-fidelity data, to be used to generate the convex underestimators. Convex underestimators are found by solving P2, where M is the total number of samples, and α, b, c are the parameters of the convex quadratic function that underestimates all of the obtained data. In previous work we have reported results on the validity of these bounds as more data is included from high- and low- fidelity samples (Zhai and Boukouvala, 2020).

$$(P2)\ min \sum_{i}^{M} \left(f(\boldsymbol{x_i}) - f_{lb}(\boldsymbol{x_i}) \right)$$
$$s.t.\ \ f(\boldsymbol{x_i}) - f_{lb}(\boldsymbol{x_i}) \geq 0 \quad \forall\, i = 1\ to\ M$$
$$f_{lb}(\boldsymbol{x_i}) = \boldsymbol{a}\boldsymbol{x_i^2} + \boldsymbol{b}\boldsymbol{x_i} + c \quad \forall\, i = 1\ to\ M$$
$$\boldsymbol{a} \geq 0$$

For each space or node of the b&b tree, a lower bound (LB) is found through minimization of the convex underestimator and an upper bound (UB) is the best high-fidelity sample collected. The algorithm then proceeds with branching the space with a selected branching rule, resampling within subspaces and updating the LB and UB of the problem. The branching heuristics originally implemented are equal bisection with respect to branch location. When deciding on which variable to branch on first, the algorithm has options prioritizing the longest side or prioritizing the most important variable with respect to the objective. The presence of constraints requires several modifications to this framework, including novel branching and pruning rules, which will be presented in the next section.

2.2. Branching rules in the presence of constraints

Branch rules are essential to a branch-and-bound algorithm. Besides equal bisection on the longest edge and equal bisection with customized variable selection (Zhai and Boukouvala, 2018, 2020), we implemented two other branch rules targeted to handle constraints. In the presence of constraints, samples are labelled as 0 (infeasible) and 1 (feasible). The proposed branch rules are designed to separate feasible from infeasible regions, with the hypothesis that this will help with faster pruning of infeasible regions. One strategy is to use a weighted Gini impurity score commonly used in classification and regression trees (Kotsiantis, 2013). The Gini impurity score (Eq. (1)) is a measure of the tendency that a randomly chosen sample would be misclassified in a node.

$$G = \sum_{i=0}^{1} p(label = i) * (1 - p(label = i)) \ \forall \ i = 0, 1 \tag{1}$$

where $p(label = i)$ denotes the possibility of a randomly chosen sample with label i in the node. To select the best location to cut, we minimize the weighted Gini impurity score (Eq. (2)), which minimizes the possibility of misclassification if a cut is generated at x_k on edge d.

$$k, d = \underset{k,d}{argmin}(p(x_d \le x_{k,d}) * G_{x_d \le x_{k,d}} + p(x_d \ge x_{k,d}) * G_{x_d \ge x_{k,d}})) \tag{2}$$

where k denotes k^{th} equidistant location on edge d and $p(x_d \le x_{k,d})$ is the possibility of a randomly chosen point lies in the potential subspace.

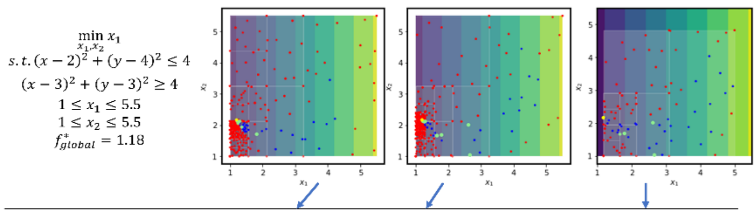
The second strategy is a customized purity score (Eq. (3)) that measures the difference in the fraction of infeasible samples and fraction of feasible samples in each subspace.

$$Purity_{k,d} = | \frac{N_{infeasible_{x \le x_{k,d}}}}{N_{infeasible}} - \frac{N_{feasible_{x \le x_{k,d}}}}{N_{feasible}} | \tag{3}$$

If one subspace contains samples with only one label, the purity score will be 1. Otherwise, the purity score is between 0 and 1. To select the best location, we maximize the purity score (Eq. (4)) to separate purely feasible and infeasible spaces.

$$k, d = argmax_{k,d}(| \frac{N_{infeasible_{x \le x_{k,d}}}}{N_{infeasible}} - \frac{N_{feasible_{x \le x_{k,d}}}}{N_{feasible}} |) \tag{4}$$

where k denotes k^{th} equidistant location on edge d. Note that if one node contains only samples with one label, equal bisection on longest edge will be used.



	Black-box	Grey-box	Glass-box
# Known constraints	0	1	2
Run time (s)	3.12	9.33	2.16
# Level	13	13	6
# Node	80	84	16
# Samples	388	377	92
Upper bound	1.21	1.19	1.22
Lower bound	1.18	1.18	1.18

Figure 1. Motivating example formulation, algorithm performance and results. Red points are infeasible, blue points are feasible.

2.1. Pruning in the presence of constraints

Pruning a search node when the node appears to be less promising to find the global optimum helps the algorithm converge faster. Generally, a node is pruned when the local lower bound is higher than the global upper bound. In the presence of constraints, we implemented additional rules to prune nodes that are infeasible. When the constraints are known, we perform feasibility-based bound tightening at the root node and at each active node. Specifically, we solve a constraint-violation minimization problem to check the feasibility of each node, and if a subspace is found to be infeasible, it is pruned. When the constraints are unknown, node pruning is less straightforward because decisions are made depending on the feasibility labels on the samples instead of explicit mathematical constraints. To reduce the chance of pruning feasible regions, backtracking techniques are incorporated. When the algorithm encounters a node that contains only infeasible samples, it backtracks the parent and grandparent nodes. Only if both the parent and grandparent nodes contain purely infeasible samples, will the node be pruned. By doing so, we avoid pruning nodes prematurely and allow collection of extra samples in that node, to increase the confidence of pruning the node.

3. Results

3.1. Motivating Example

Here we present results on a simple 2-d example that enables us to highlight the algorithm performance. In Figure 1, we show the formulation of the problem, as well as the performance of the algorithm when the formulation is purely black-box (all constraints are simulation-based), grey-box (first constraint is known, second is unknown), to glass-box (all constraints are known). As expected, as more constraints become known, the algorithm converges to the global optimum with less samples. In addition, we observe that the additional cost of feasibility tightening does not significantly increase the cost of the algorithm. On the contrary, due to faster convergence, the algorithm converges faster with respect to run-time when constraints are present. In this motivating example, we keep the pruning and branching rule fixed to equal bisection with branching on most important variable. A comparison of different heuristics is performed in the next section.

3.2. Algorithm benchmarking

In order to test the performance of different branching rules, we test the algorithm on a larger set of benchmark problems. These are 42 problems from the MINLPlib, with 2-5 dimensions and no limit on the number of constraints. Results are reported for the two extreme cases, namely the black-box case where no constraints are known and the glass-box case where all constraints are known (Figure 2). Results show that the algorithm can locate the global optimum for 80% of the problems in the black-box case, and more than 85% of the problems in the glass-box case. Moreover, the algorithm converges with less samples when constraints are explicitly known. With respect to branching rules, we observe that the novel branch rules proposed in this work are able to expedite convergence (i.e., more problems are solved with less number of samples). However, overall, the equal-bisection branching rule solves more problems if more samples can be collected. This implies that these decision-tree rules can expedite pruning, however, they are less conservative and thus increase the chance of removing good optimal solutions.

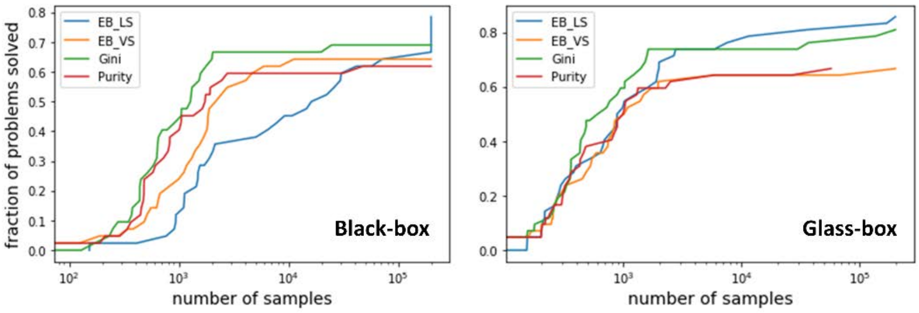


Figure 2. Performance of algorithm and different branching rules for set of 42 benchmarks

4. Conclusions

In this work we presented an extension of our algorithm for data-driven branch-and-bound for constraints handling. We propose and test several branching and pruning rules in the presence of simulation-based constraints that are based on concepts proposed in the literature of decision-trees. In order to treat equation-based constraints, we employ feasibility-based bounds tightening techniques. Results show that it is important to use any known constraints directly, because this expedites algorithm convergence and overall solves more problems without adding computational cost due to reduced sampling requirements. Moreover, branching when using decision-tree heuristics expedites convergence of the algorithm, however, in certain cases prunes valuable solution spaces. As a result, when sampling is not a significant burden, the equal-bisection approach is a more conservative and reliable heuristic.

References

A. Bhosekar, M.G Ierapetritou, 2018, Advances in surrogate based modeling, feasibility analysis, and optimization: A review. Computers & Chemical Engineering 108, 250-267

S.S. Garud, N. Mariappan, I.A. Karimi, 2019, Surrogate-based black-box optimisation via domain exploration and smart placement. Computers & Chemical Engineering 130, 106567

F. Boukouvala, R. Misener, C.A. Floudas, C.A, 2016, Global optimization advances in Mixed-Integer Nonlinear Programming, MINLP, and Constrained Derivative-Free Optimization, CDFO. European Journal of Operational Research 252(3), 701-727

S.B. Kotsiantis, 2013, Decision trees: a recent overview, Artificial Intelligence Review, 39: 261-83.

L. M. Rios and N.V. Sahinidis, 2013, Derivative-free optimization: a review of algorithms and comparison of software implementations. Journal of Global Optimization 56(3), 1247-1293

A.M. Schweidtmann, A. Mitsos, 2019, Deterministic Global Optimization with Artificial Neural Networks Embedded. Journal of Optimization Theory and Applications 180(3), 925-948

J. Zhai, F Boukouvala, 2016, Nonlinear Variable Selection Algorithms for Surrogate Modeling. AIChE Journal 0(ja), e16601. doi:10.1002/aic.16601

J. Zhai and F. Boukouvala, 2020, Data-driven Spatial Branch-and-bound Algorithm for Box-constrained Simulation-based Optimization, In.: Living Archive for Process Systems Engineering. http://psecommunity.org/LAPSE:2020.1164