# DATA-DRIVEN SPATIAL BRANCH-AND-BOUND ALGORITHMS BLACK-BOX OPTIMIZATION

Jianyuan Zhai, Fani Boukouvala*
Georgia Institute of Technology
Atlanta, GA 30332

*Abstract*

The use of high fidelity simulations is becoming more common in computer-aided design and optimization. Due to the complexity of the simulation models and the lack of closed-form mathematical formulations, the direct use of traditional deterministic optimization tools is prohibitive for such problems. Therefore, data-driven decision-making has become increasingly important, with surrogate-based optimization being one of the most popular approaches. Two of the most important challenges in surrogate-based optimization are the lack of consistent convergence metrics and the variability in the quality of the incumbent solution when a different sampling set or a different surrogate model is used to guide the search. In this work, we propose a strategy to mitigate this uncertainty in the performance of such algorithms. A novel data-driven spatial branch-and-bound framework is proposed that uses stochastic bounds on different surrogate models and partitioning of the search space. The convergence properties of this algorithm are studied through a large set of benchmark problems.

*Keywords*

Data-driven optimization, Surrogate modeling, Branch-and-bound

## Introduction

The rapid development in computer-aided design and optimization have led to the generation and storage of complex models that can capture high-fidelity details of a system, such as models based on large partial differential equation systems of equations (Tsay, Pattison, & Baldea, 2017), and computational fluid dynamics (CFD) (Dowling, Eason, Ma, Miller, & Biegler, 2014; Onel, Niziolek, Butcher, Wilhite, & Floudas, 2017). When it comes to optimization using the above, the use of existing deterministic gradient-based optimization algorithms is prohibitive. Because of the inability to directly use algebraic model-based mathematical programming and deterministic gradient-based optimization algorithms, solving such optimization problems often relies on the information from input-and-output data, and that is why it is referred to as data-driven optimization. Data-driven optimization, also known as derivative-free optimization or black-box optimization, does not assume the availability of the algebraic description of the formulation. There are three main classes of data-driven optimization methods, namely sampling-based methods, model-based methods, and stochastic search methods. Some global sampling-based model-based methods have employed the concept of partitioning the subspace in order to find better solutions. For example, two of the most widely used sampling-based partitioning methods are the DIRECT algorithm (Donald R. Jones, 2009; D. R. Jones, Perttunen, & Stuckman, 1993) and Multilevel Coordinate Search (MCS) (Huyer & Neumaier, 1999). Similarly in the area of model-based methods, Sequential design for optimization (SDO) (Cox & John, 1992), efficient global optimization (EGO) (Donald R. Jones, Schonlau, & Welch, 1998), and stable noisy

* To whom all correspondence should be addressed

optimization by branch-and-fit (SNOBFIT) (Huyer & Neumaier, 2008), are three methods that have combined the concept of fitting a specific type of surrogate model with consequent sampling in subspaces.

Computational studies and reviews of the current existing data-driven optimization software have shown that the performance is problem-dependent and no one solver can solve all types of problems (Amaran, Sahinidis, Sharda, & Bury, 2014; Rios & Sahinidis, 2013). Especially in the model-based optimization field, researchers have devoted great effort on finding the best surrogate modeling strategy and sampling technique (Boukouvala & Floudas, 2017; Cozad, Sahinidis, & Miller, 2014; Eason & Cremaschi, 2014). However, the question on which sampling strategy coupled with a surrogate type is the best for optimization is still an open challenge. In this work, we systematically study the sources of this variability and develop techniques that are less sensitive to this selection. Our overall aim is to develop a general framework that can converge to good solutions for multiple types of surrogate models that do not need to perfectly approximate the black-box problem. Naturally, the selection of the surrogate model will affect the speed of convergence, however, when coupled with space partitioning and bounding, our hypothesis is that consistent convergence can be observed.

First, we identify the reasons that cause this observed variability in the performance of different data-driven approaches. As shown in Figure 1, when fitting a surrogate model to a black-box function, one can easily obtain different realizations of the trained models. This phenomenon can happen for two reasons: a) if slightly different samples are collected, even the same type of surrogate model might have different optimal parameters and as a result, a different functional form, and b) if the same samples are collected but different types of surrogate models are fitted (i.e., one Neural Network, one Gaussian process model, and a polynomial model), the fitted functions will also have differences. Secondly, because of the lack of the algebraic description of the true problem, convergence to a global optimum is less reliable with limited amount of samples. Unlike deterministic global optimization algorithms which can provide finite convergence with $\varepsilon$ tolerance, currently global convergence of data-driven optimization algorithms is guaranteed when approaching the limit of infinite sampling. However, this is usually prohibitive because sampling is often very expensive.

To address the two challenges mentioned above, one promising approach is to adopt the structure of the branch-and-bound and combine it with surrogate-based optimization. As mentioned earlier, many solvers in deterministic global optimization algorithms provide finite $\varepsilon$ convergence to global solution using a branch-and-bound framework, for instance, BARON (Tawarmalani & Sahinidis, 2005), ANTIGONE (Misener & Floudas, 2014), and $\alpha$-BB (Androulakis, Maranas, & Floudas, 1995). Specifically, the finite $\varepsilon$ convergence is referred to the difference between the lower bound (LB) and upper bound (UB) of the objective function. Usually the LB is obtained by solving the convex relaxations of the original nonconvex problem and the UB can be any feasible solution of the original nonconvex problem (Misener & Floudas, 2014; Tawarmalani & Sahinidis, 2004, 2005). By progressively dividing the search space, these algorithms aim to find tighter lower and upper bounds of the global optimum. Although some sampling-based and model-based search methods have the search space partitioning feature, no other method has explored the bounding approach. Methods like SDO, optimize the statistical lower bound of the function but do not branch the search space. EGO (Donald R. Jones et al., 1998) optimizes the expected improvement function in a branch-and-bound framework, while the mathematical form of the expected improvement function is rather too complicated. The DIRECT method, follows a space partitioning approach (Donald R. Jones, 2009); however, it does not use any surrogate approximations or bounding strategies. As a result, although this algorithm has been shown to perform well on a diverse set of applications, algorithm convergence happens when the maximum number of samples has been reached, with no other indication that convergence has truly been achieved. The DIRECT algorithm is the one that has the most similar structure with our proposed framework, thus we will compare the performance of our algorithm with an existing implementation of DIRECT (Johnson, 2018; Donald R. Jones, 2009).

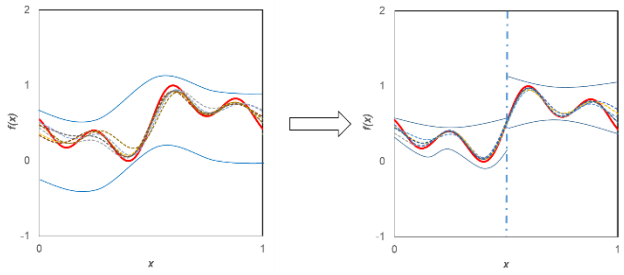


*Figure 1 (a) bounding multiple realizations of surrogate functions, (b) partitioning and bounding multiple realizations of surrogate approximations*

In this work, we propose a data-driven (DD) equivalent spatial branch-and-bound (sBB) algorithm, which uses imperfect surrogate models to solve a wide range of data-driven optimization problems. To handle uncertainty in surrogate modeling, we will use error metrics, statistical bounds and margins to approximately bound the surrogate model realizations (Figure 1). By subdividing the search space, the variance in the data and the approximation of the surrogate model is expected to decrease. As a result, as we branch we expect to obtain tighter bounds and better surrogate approximations, and hypothesize that this will lead to a convergent data-driven branch-and-bound algorithm. Additionally, by pruning some subspaces, sampling can be focused within regions that are more promising and not wasted in regions that cannot improve the solution even in the best case. Our main hypothesis is that by developing such a data-driven branch-and-bound framework, the performance of the algorithm will be less

*Table 1 Summary of SVR and GP models and the strategies for obtaining the lower bound.*

| Model | SVR-RBF |
|---|---|
| Mathematical Form | $\hat{f}(\boldsymbol{x}^*) = b + \sum_{l=1}^{L} w_l \exp(-\gamma \lvert\lvert \boldsymbol{x}^* - \boldsymbol{x_l} \rvert\rvert^2)$ |
| LB | $\hat{f}(x) - \max(e)$ |
| Details | $l$ : index for support vectors<br>$w$: basis function weights<br>$e$ : absolute prediction error |
| **Model** | **GP** |
| Mathematical Form | $\hat{f}(\boldsymbol{x}^*) = \mu + \sum_{n=1}^{N} c_n \exp[-\sum_{i=1}^{C} \theta_i \left(x_i^* - x_i^{(n)}\right)^2]$ |
| LB | $\hat{f}(x) - \sigma$ |
| Details | $n$: index for samples,<br>$i$: index for dimensions<br>$\sigma$: stationary variance |

dependent on the choice of the surrogate model selection.

We also believe that this approach is a first step towards developing a convergent data-driven optimization algorithm. The DD-sBB algorithm demonstrated in this work is developed for multi-variable box-constrained problems. We test the performance of the DD-sBB algorithm with different surrogate models and bounding strategies on a large set of benchmark problems with and without addition of random noise.

## Methods

The algorithm is implemented in Python v3.6.0. using Numpy numerical library (v1.13.3) and Scipy library (v1.1.0). Optimization of the surrogate models is performed in pyomo (W. Hart, Watson, Woodruff, & Watson, 2011; W. E. Hart, Laird, Watson, & Woodruff, 2012) (v5.5.0). The IPOPT (Wächter & Biegler, 2006) (v3.12.10) solver is used to optimize the surrogate models and/or their bounds. Since this is a local solver and our surrogates can be nonconvex, a random multistart local optimization approach is followed, by employing our samples as initial points for the local optimization.

*Sampling*

Latin Hypercube Sampling (LHS) (McKay, Beckman, & Conover, 1979) is used to generate initial sample points (10D+1, where D is the dimension). As the algorithm progresses, augmented LHS is used to add more points (5D+1) (Stein, 1987). In addition to the sample points collected by Latin Hypercube Sampling, the optimal solutions found by IPOPT in each subspace are added to the sampling set. LHS is performed with the pyDOE (v0.3.8) package in python and R(R Core Team, 2016) package 'lhs'(Carnell, 2016) via Python-R interface 'RPy2' (v2.8.5) in python.

*Surrogate Modeling & Bounding*

The two surrogate models currently implemented in the DD-sBB algorithm are: (a) Support Vector Regression (SVR) with a radial basis function (RBF) kernel, and (b) Gaussian Process Modeling (GP) or Kriging. For each of the models, we design a customized bounding strategy. The upper bound is any feasible solution of the surrogate-based optimization, and the lower bound is the perturbation of the upper bound. The summary of the SVR and GP models and the bounding strategies are shown in Table 1. In this work, SVR-RBF models are trained using 'scikit-learn'(Pedregosa et al., 2011) in Python. GP is trained using 'DiceKriging' (Roustant, Ginsbourger, & Deville, 2012) package via 'RPy2' interface.

To train the surrogate models, 5-fold cross-validation is used by randomly splitting the samples into testing sets (20%) and training sets (80%). The model with the lowest test error is picked.

Support vector regression (SVR) is the regression variation of support vector machines (SVM) (Vapnik, 1999). SVR with radial basis functions (RBF) has been shown to efficiently model nonlinear systems. The most distinguishable feature of SVR is that the model only depends on the support vectors, which is a subset of the training data obtained by setting a tolerance on the training error. This feature controls the smoothness and interpolating versus regression nature of SVR models.

One of the main parameters of the SVR model is the margin $\epsilon$, which controls the number of support vectors and smoothness of the final function. In this work, we suggest to scale the standard deviation $s(f(\boldsymbol{x}^*))$ of the sampling points by a scalar $\kappa$, and use this quantity as the margin to train the SVR model. Intuitively, the margin can be used to bound the realization of the SVR model, because most points will lie inside or on the margin. In order to create conservative and more encompassing bounds, a final perturbation of the bound is performed to include all sampled points within the margin. This is done by perturbing (reducing) the lower bound by the maximum absolute prediction error from the SVR model.

The second model currently implemented is based on GP models, which share a similar mathematical form of SVR-RBF. While SVR-RBF only uses a subset of the training points to build the model, GP interpolates all training points by assuming the points are distributed with mean $\mu$ and variance $\sigma^2$, which are estimated from the maximum log-likelihood function (Donald R. Jones, 2001). The standard error associated with the GP interpolation is $s^2(\boldsymbol{x}^*)$:

$$s^2(\boldsymbol{x}^*) = \hat{\sigma}^2[1 - \boldsymbol{r}'\boldsymbol{R}^{-1}\boldsymbol{r} + \frac{(1-\boldsymbol{r}'\boldsymbol{R}^{-1}\boldsymbol{r})^2}{\boldsymbol{1}'\boldsymbol{R}^{-1}\boldsymbol{1}}] \quad (1)$$

where $\boldsymbol{x}^*$ denotes a new point, and $\hat{\sigma}^2$ the stationary variance, $\boldsymbol{R}$ is the correlation matrix between training points, and $\boldsymbol{r}$ is the correlation matrix between the predicting point with the training point, which are related to the covariance matrix $Cov(Y)$:

$$Cov(Y) = \hat{\sigma}^2 \boldsymbol{R} \quad (2)$$

While the statistical lower bound (3) can be directly used to bound the GP model, the complicated form of $s(\boldsymbol{x}^*)$ as shown in (1) leads to a very nonlinear and nonconvex optimization problem.

$$\hat{f}(\boldsymbol{x}^*) - \kappa s(\boldsymbol{x}^*), \text{ where } \kappa \text{ is a constant} \quad (3)$$

Therefore, we propose a new bounding strategy for GP, which is simply based on the stationary variance $\sigma^2$. As presented in (1), $\left[1 - \boldsymbol{r}'\boldsymbol{R}^{-1}\boldsymbol{r} + \frac{\left(1-\boldsymbol{r}'\boldsymbol{R}^{-1}\boldsymbol{r}\right)^2}{\boldsymbol{1}'\boldsymbol{R}^{-1}\boldsymbol{1}}\right] \leq 1$ is always valid. Then, we can obtain the following inequality:

$$s^2(\boldsymbol{x}^*) \leq \hat{\sigma}^2 \quad (4)$$

Consequently, we can obtain a bounding strategy that is very similar to the SVR-RBF bounding (5).

$$\hat{f}(\boldsymbol{x}^*) - \sigma \quad (5)$$

Both of the bounding strategies discussed measure the uncertainty in modeling, and aim to enclose or bound most of the possible realizations of the surrogate models, given the limited amount of samples that have been collected. As the subspaces become smaller, the data variance and modeling uncertainty will decrease, which is a key concept that will drive convergence in our algorithm.

*Branch-and-bound Scheme*

Equal bisection and golden bisection rules based on the ratio of $\frac{-1+\sqrt{5}}{2}$ are implemented in this algorithm. The branching is initialized on the most important variable, ranked by an in-house feature selection algorithm. After the first branch, a different heuristic is employed and the variable with larger range is branched. The overall upper bound $f_{UB}$ is the best solution found, and the lower bound $f_{LB}$ is the minimum lower bound of all active nodes. The subspace is pruned if its lower bound is higher than the overall upper bound.

*Stopping Criteria*

Our first aim is to show the global convergence of our algorithm. Thus, we allow it to run for a long CPU time limit (3000 sec) and the algorithm converges when $|f_{UB} - f_{LB}| \leq \varepsilon$.

## Result and Discussion

*Benchmark Problems*

A total of 220 benchmark problems are tested with DD-sBB algorithm with SVR and GP. 148 problems in the problem set are 2-3D problems. 72 problems are 3-10D problems. These problems are continuous box-constrained problems selected from the Sahinidis test problem set (Puranik & Sahinidis, 2017).

The results are also compared to the performance of the DIRECT algorithm. The comparative profiles are shown in Figure 2. For DIRECT, the stopping criterion is a heuristic that there is no significant improvement over consecutive iterations. As for DD-sBB, the algorithm is set to stop when $|f_{UB} - f_{LB}| \leq 0.05$.

As shown in Figure 2, all three methods can solve over 97% of 2-3D problems and over 90% of 4-10D problems. DIRECT consistently takes less time to solve the problems. One reason is that DIRECT is purely sampling-based, whereas DD-sBB requires training and optimizing surrogate models. However, as the dimensionality increases, DIRECT samples more points than the DD-sBB methods. Moreover, among the DD-sBB methods, GP consistently requires less samples to solve the same amount of problems.

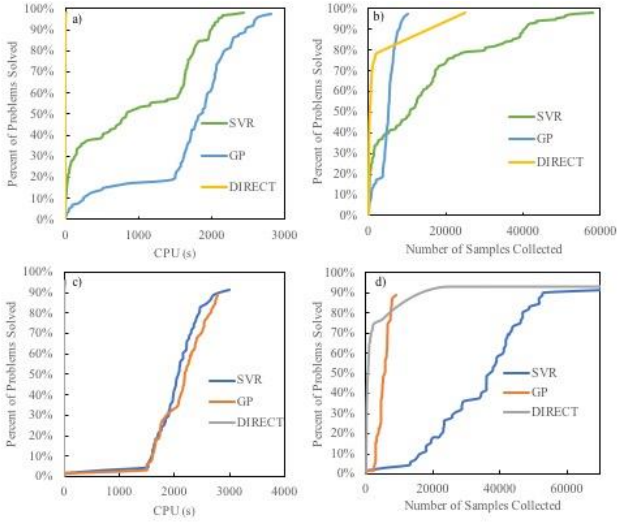


*Figure 2 Performance plots of DD-sBB with SVR and GP, and DIRECT for 2-3D problems (a and b), and 4-10 D problems (c and d)*

As the dimension of the problem increases, the advantage of using GP with DD-sBB is more predominant. This difference in sampling requirements of SVR- and GP-based DD-sBB methods may be tuned by adjusting the width of the SVR bounds and the GP bounds, which results in different efficiency of pruning the subspaces and sampling.

Despite its increased CPU cost, one of the major advantages of our proposed approach is that upper and lower bounds on the best solution are provided. In fact, even in the small subset of the problems that were not solved by the algorithm, the optimal solution is within the final upper and lower bounds. This shows that even if the lower and upper bound tolerance might not have been reached, the current best solution is still a high quality solution.

Finally, we want to study the performance of our algorithm in the case that sampling limitations exist. Thus, we recorded the cpu and sampling requirements for the 2-3D problems, for the point when the algorithm finds the optimal solution with high accuracy (within 0.05 absolute difference from the nominal optimum solution). As shown in Figure 3, GP requires significantly smaller amount of samples to find a good solution compared to SVR. However, the cpu requirements for GP is not significantly

different from that of the SVR method. The potential explanation of this observation is that due to the regularization component of SVR, the model complexity is reduced, and the surrogate models are easier to optimize. Alternatively, when using Gaussian Process models, the model complexity is always dependent on the number of training points but the approximation error is always lower. Therefore, it is observed that better model approximation accuracy increases the CPU requirements; however, it leads to better solutions with fewer samples.

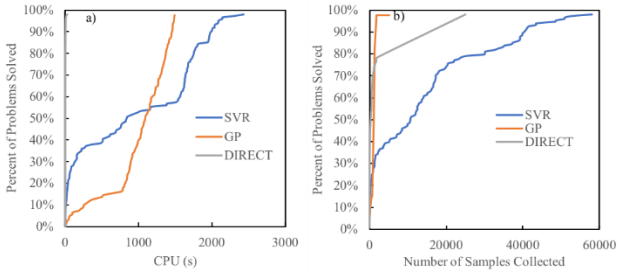


*Figure 3 Performance plot of DD-sBB with SVR and GP based on the the first high quality solution found, and DIRECT for 2-3D problem*

Finally, we also test our algorithms in the presence of up to 10% random noise in the output data. This test aims to show the robustness of the techniques in the case where the simulation output is subject to some uncertainty. It has been previously shown that the use of surrogate models is effective for noisy data, due to the approximation model's ability to filter out some of the noise in the data. In addition, since our algorithm is based on the concept of bounding uncertain surrogates, we expect this framework to perform well even in the presence of noise. As shown in Figure 4, both SVR and GP methods are able to solve over 93% problems in the benchmark set. The addition of random noise increases the required number of samples to solve the problems, especially for the GP method. As expected, noise is less influential on the SVR method since the margin in SVR prevents overfitting of noisy data.

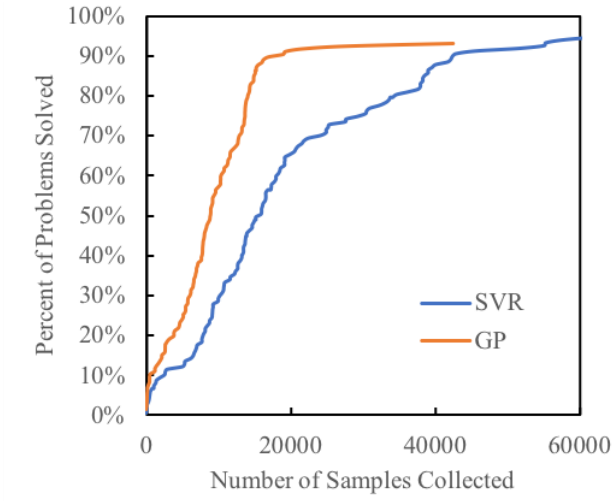


*Figure 4 Performance plot of DD-sBB with SVR and GP subjected to 10% random noise*

## Conclusions

In this work, we present a novel data-driven spatial branch-and-bound algorithm (DD-sBB) for box-constrained black-box problems. The main idea of this algorithm is that by adapting the spatial branch-and-bound scheme from deterministic branch-and-bound algorithms, we can reduce the dependency on sampling and surrogate modeling selection, and convergence of data-driven algorithms can be achieved with imperfect surrogates. We use absolute prediction error for SVR-RBF and stationary variance for GP as the bounding strategies, and compare the lower and upper bounds to prune subspaces. On the large box-constrained benchmark problem set, both DD-sBB with SVR and GP can solve over 90% problems, which is comparable to the DIRECT algorithm. The algorithm is also capable of solving similar amount of problem subject to 10% random noise in the output data. Result shows that the algorithm can find good solutions at an earlier stage before convergence, which indicates that further investigation on the convergence criteria is needed. In the future, we aim to improve the CPU efficiency of the algorithm and investigate less conservative bounding strategies to improve the convergence efficiency. Moreover, we plan to add more surrogate models and bounding strategies to our framework, and extend the algorithm to handle constraints.

## Acknowledgments

The authors acknowledge support by NSF (1805724) and Georgia Tech Startup Funding.

## References

Amaran, S., Sahinidis, N. V., Sharda, B., & Bury, S. J. (2014). Simulation optimization: a review of algorithms and applications. *4OR, 12*(4), 301-333. doi:10.1007/s10288-014-0275-2

Androulakis, I. P., Maranas, C. D., & Floudas, C. A. (1995). αBB: A global optimization method for general constrained nonconvex problems. *Journal of Global Optimization, 7*(4), 337-363. doi:10.1007/bf01099647

Boukouvala, F., & Floudas, C. A. (2017). ARGONAUT: AlgoRithms for Global Optimization of coNstrAined grey-box compUTational problems. *Optimization Letters, 11*(5), 895-913. doi:10.1007/s11590-016-1028-2

Carnell, R. (2016). lhs: Latin Hypercube Samples.

Cox, D. D., & John, S. (1992, 18-21 Oct. 1992). *A statistical method for global optimization.* Paper presented at the [Proceedings] 1992 IEEE International Conference on Systems, Man, and Cybernetics.

Cozad, A., Sahinidis, N. V., & Miller, D. C. (2014). Learning surrogate models for simulation-based optimization. *AIChE Journal, 60*(6), 2211-2227. doi:10.1002/aic.14418

Dowling, A. W., Eason, J. P., Ma, J., Miller, D. C., & Biegler, L. T. (2014). Coal Oxycombustion Power Plant Optimization Using First Principles and Surrogate

Boiler Models. *Energy Procedia, 63*, 352-361. doi:https://doi.org/10.1016/j.egypro.2014.11.038

Eason, J., & Cremaschi, S. (2014). Adaptive sequential sampling for surrogate model generation with artificial neural networks. *Computers & Chemical Engineering, 68*, 220-232. doi:https://doi.org/10.1016/j.compchemeng.2014.05.021

Hart, W., Watson, J.-P., Woodruff, D., & Watson, J. P. (2011). *Pyomo: Modeling and solving mathematical programs in Python* (Vol. 3).

Hart, W. E., Laird, C., Watson, J.-P., & Woodruff, D. L. (2012). *Pyomo - Optimization Modeling in Python*: Springer Publishing Company, Incorporated.

Huyer, W., & Neumaier, A. (1999). Global Optimization by Multilevel Coordinate Search. *Journal of Global Optimization, 14*(4), 331-355. doi:10.1023/a:1008382309369

Huyer, W., & Neumaier, A. (2008). SNOBFIT -- Stable Noisy Optimization by Branch and Fit. *ACM Trans. Math. Softw., 35*(2), 1-25. doi:10.1145/1377612.1377613

Johnson, S. G. (2018). The NLopt nonlinear-optimization package (Version 2.5.0). Retrieved from https://nlopt.readthedocs.io/en/latest/

Jones, D. R. (2001). A Taxonomy of Global Optimization Methods Based on Response Surfaces. *Journal of Global Optimization, 21*(4), 345-383. doi:10.1023/a:1012771025575

Jones, D. R. (2009). Direct global optimization algorithmDirect Global Optimization Algorithm. In C. A. Floudas & P. M. Pardalos (Eds.), *Encyclopedia of Optimization* (pp. 725-735). Boston, MA: Springer US.

Jones, D. R., Perttunen, C. D., & Stuckman, B. E. (1993). Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications, 79*(1), 157-181. doi:10.1007/bf00941892

Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization, 13*(4), 455-492. doi:10.1023/a:1008306431147

McKay, M. D., Beckman, R. J., & Conover, W. J. (1979). Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics, 21*(2), 239-245. doi:10.1080/00401706.1979.10489755

Misener, R., & Floudas, C. A. (2014). ANTIGONE: Algorithms for coNTinuous / Integer Global Optimization of Nonlinear Equations. *Journal of Global Optimization, 59*(2), 503-526. doi:10.1007/s10898-014-0166-2

Onel, O., Niziolek, A. M., Butcher, H., Wilhite, B. A., & Floudas, C. A. (2017). Multi-scale approaches for gas-to-liquids process intensification: CFD modeling, process synthesis, and global optimization. *Computers & Chemical Engineering, 105*, 276-296. doi:https://doi.org/10.1016/j.compchemeng.2017.01.016

Pedregosa, F., Ga, #235, Varoquaux, l., Gramfort, A., Michel, V., . . . Duchesnay, d. (2011). Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res., 12*, 2825-2830.

Puranik, Y., & Sahinidis, N. V. (2017). Bounds tightening based on optimality conditions for nonconvex box-constrained optimization. *J. of Global Optimization, 67*(1-2), 59-77. doi:10.1007/s10898-016-0491-8

R Core Team, R. F. f. S. C. (2016). R: A Language and Environment for Statistical Computing.

Rios, L. M., & Sahinidis, N. V. (2013). Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization, 56*(3), 1247-1293. doi:10.1007/s10898-012-9951-y

Roustant, O., Ginsbourger, D., & Deville, Y. (2012). DiceKriging, DiceOptim: Two R Packages for the Analysis of Computer Experiments by Kriging-Based Metamodeling and Optimization. *2012, 51*(1), 55. doi:10.18637/jss.v051.i01

Stein, M. (1987). Large Sample Properties of Simulations Using Latin Hypercube Sampling. *Technometrics, 29*(2), 143-151. doi:10.1080/00401706.1987.10488205

Tawarmalani, M., & Sahinidis, N. V. (2004). Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming, 99*(3), 563-591. doi:10.1007/s10107-003-0467-6

Tawarmalani, M., & Sahinidis, N. V. (2005). A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming, 103*(2), 225-249. doi:10.1007/s10107-005-0581-8

Tsay, C., Pattison, R. C., & Baldea, M. (2017). Equation-oriented simulation and optimization of process flowsheets incorporating detailed spiral-wound multistream heat exchanger models. *AIChE Journal, 63*(9), 3778-3789. doi:doi:10.1002/aic.15705

Vapnik, V. (1999). *The Nature of Statistical Learning Theory*: Springer New York.

Wächter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming, 106*(1), 25-57. doi:10.1007/s10107-004-0559-y