Derivative-Free Optimization: Linking Algorithms and Applications

Matt Menickelly (Argonne National Laboratory),
Warren Hare (University of British Columbia),
Charles Audet (Polytechnique Montréal),
Ana Lusa Custdio (Universidade Nova de Lisboa)

July 18 - July 22, 2022

1 Overview of the Field

The following excerpt is taken from the description of the workshop, as was posted on the BIRS website:

> *Optimization, the study of minimizing or maximizing a function subject to satisfying constraints, arises in virtually every area of science and engineering. Modern applications of optimization take many forms, each with their own unique challenges. One important form is optimization problems where the objective and/or constraint functions are provided via a "black-box". Evaluation of the black-box function yields a function value for the specified input at a given point, but no other information (such as derivative information). This occurs, for example, whenever the objective function relies on an executable-only computer simulation. With the rapid increase of computational science and engineering problems, and associated computer simulations, problems of this form have become ubiquitous in modern applications.*
>
> *Derivative-free optimization (DFO) is the design and study of optimization algorithms that use only function values without derivatives. Such algorithms provide the perfect tool for optimization of applications based in computer simulation. DFO has made significant advances over the past decade and represents one of the most rapidly expanding fields of nonlinear optimization research. Novel algorithms have expanded the reach of DFO applications. In*

this workshop, we forge new connections between research focused in DFO algorithm design and novel applications of DFO.

In this initial description, we deliberately used terminology "black-box" and "derivative-free optimization", and felt it necessary to provide definitions. This choice was, in part, because there is notably *not* a general consensus on the use of these terms, but people within this community tend to understand these as nebulous umbrella terms. As a side objective of this workshop, we attempted to crowdsource guidelines for the appropriate uses of these words, which we will document in the Side Objectives section below.

However, using the definitions provided in the initial abstract, an overview of the field of DFO algorithms can be largely characterized by two classes of methods - model-based methods and direct search methods. Model-based methods aim to construct local surrogates of the objective function via some model function (e.g., an interpolated function) and iteratively optimize these surrogates to obtain candidate points for evaluation by the true objective function (which could be a black-box). Direct search methods, on the other hand, do not "in spirit" construct models of the objective function, but rather iteratively generate batches of trial points around an incumbent point (e.g., by sampling a grid in a calculated manner) and update incumbents if function decrease is realized at one of the trial points. As we shall see in the Principal Objective section below, the theoretically-motivated talks presented in this workshop often cleanly divide into this split, with a few notable exceptions.

2 Workshop Principal Objective

In the initial abstract, the workshop's goals were to: "*forge new connections between research focused in DFO algorithm design and novel applications of DFO*."

As a collection of mathematicians in a BIRS-sponsored workshop, we naturally had no difficulty in finding experts in algorithmic design, which we will outline below in this section. We were also fortunate to have several talks (e.g., talks by Audet, Lucet, and Le Digabel) extremely focused on motivating applications for DFO algorithms. However, given the attendance constraints imposed by the global COVID-19 pandemic, we could not achieve the broad audience we had desired. As such, there was a slight imbalance between theory-driven versus application-driven talks. We want to stress that we do not view this imbalance as a failure. Indeed, many talks discussed specific applications, which we call out in summaries below. Moreover, as is particularly salient in the field of DFO, most of the considerations made in algorithmic design are driven by our particularly complicated setting presented by black-box information. Therefore, when compared with other fields of applied mathematics, much of the work in DFO algorithms is a crystallization of hard work and respect for the difficulties presented by real applications. DFO rarely exists in a "perfect vacuum". Thus, many of the workshop participants were familiar with the unique challenges of DFO applications, and were able to participate meaningfully in discussions of the intersection of theory and application.

In the workshop goals, we do want to stress the word *connections*. A general sense felt among the participants was that a workshop with mostly physical (as opposed to virtual) attendance felt great following two years of virtual conferences and workshops necessitated by the pandemic. Moreover, due to our open workshop schedule – we deliberately left 15 minute “coffee breaks” between every talk – spontaneous discussion and conversation was abundant. Such conversations would absolutely not have been reproducible in a virtual setting. Seeing other researchers in the same field fostered new connections, and was generally a welcome late-stage-pandemic respite.

2.1 Model-Based DFO Algorithm Presentations

Three talks in particular (Jarry-Bolduc, Menickelly, and Larson) exhibited a common theme of exploiting additional information about a black-box, when such information is available in a practical setting. Gabriel Jarry-Bolduc presented a talk on the use of calculus rules – e.g., product rule, quotient rule, and chain rule – for building models of products, quotients, and compositions of black-boxes. Matt Menickelly presented on a means of constructing “stochastic average models”, emulating SAG(A) algorithms, when the black-box is in fact a sum of black-boxes; this work was motivated by problems in nuclear model calibration. Jeff Larson presented a model-based “manifold sampling” method for optimizing (nonsmooth) compositions of analytic functions with black-box functions; this work was motivated by problems in physical beamline calibration.

Other talks in model-based methods highlighted some of the unique challenges faced in more general settings of DFO. Ana Custdio presented a model-based trust-region method for derivative-free multiobjective optimization, a particularly difficult generalization of derivative-free optimization problems that seeks Pareto front approximations, as opposed to a single minimizer. Stefan Wild presented a method for iteratively sampling random subspaces over which to build models of the objective function using only black-box evaluations. Stefan’s work also puts a fine point on the particular challenges of DFO, in that DFO algorithms generally scale poorly with dimension, encouraging the exploration of down-sampling in dimension.

Whereas the previous five talks on model-based methods were related to *local* optimization methods, we had several presentations on *global* model-based optimization. In particular, Youssef Diouane discussed work in enhancing typical methods of Bayesian optimization (known to be costly in high dimensions) with trust-region model-based methods to enable less-expensive local searches. Juliane Muller discussed the use of global surrogate methods with stochastic black-boxes tailored to hyperparameter optimization for tuning DL model architectures; this work was particularly motivated by problems in particle physics. Christine Shoemaker discussed novel global surrogate methods for the difficult problem of multiobjective optimization. Everton Silva presented scalarization conditions for global optimality within multiobjective optimization and used these conditions to propose a (local) scalarized method for multiobjective derivative-free optimization that ought to exhibit improved global performance. This work was the subject of his Master thesis.

2.2 Direct-Search DFO Algorithm Presentations

As discussed in the overview of the field, direct-search methods generate a set of trial points, which may become future incumbents. One such commonly employed (and analyzed set) are positive bases and/or positive spanning sets. Warren Hare gave a theoretically-minded presentation on positive bases, and algorithms for determining if a given set of vectors specifies a good quality positive basis, as well as desirable structures within positive bases. Sebastien Kerleau gave a talk on a novel extension of positive spanning sets, called positive k-spanning sets, which exhibit nice properties that may be exploitable in future development of direct search algorithms.

Some talks offered extensions of existing direct search methods, notably, mesh-adaptive direct search (MADS) methods, for specialized challenging settings. Charles Audet discussed a method for guiding the generation of poll directions in MADS methods, when an objective monotonic trend is suspected in some variables; this work was heavily influenced by a real hydroelectric dam optimization problem, where engineers were implicitly considering the existence of such a trend in making heuristic decisions in operating conditions. Solène Kojtych discussed DiscoMADS, an extension of MADS designed to identify and appropriately deal with discontinuities in the black-box epigraph, a known challenge in real practical settings. Kwassi Joseph Dzahini discussed an extension of a progressive-barrier variant of MADS to settings where the black-box objective and constraints are stochastic, and probabilistic constraint violation is an appropriate form of constraint-handling. Ludovic Salomon discussed an extension to multiobjective MADS frameworks to handle general additional inequality black-box constraints.

2.3 Other DFO Algorithm Presentations

While model-based and direct-search algorithms create a clean split in principle, not all of development in DFO algorithms can be partitioned so cleanly. Indeed, in this workshop, some of the discussion centered on algorithms that are either a hybrid of the two classes of methods, or cannot be cleanly described as either.

Giampaolo Liuzzi presented on extensions of a derivative-free *line search method* in the setting of an interior point method designed to handle general black-box constraints. Derivative-free line search methods cannot be cleanly classified as either a direct search method or a model-based method - they essentially operate like a derivative-based line search method using derivative-free directions. Positive spanning sets are a possibility for it, like was the case of the talk of Liuzzi, but replacing the gradient with a gradient approximation of dynamically adjusted gradient accuracy can also be considered. One of such gradient approximations was the subject of a presentation by Yiwen Chen, a recent undergraduate degree awardee. Yiwen discussed the approximation quality of centered simplex gradients under the assumption of “misaligned points”, which may prove useful in settings of expensive black-boxes, where one does not wish to expend additional effort on computing perfectly centered simplex gradients.

Hybrid methods were also present at this meeting. Dominic Huang, recently awarded a Masters degree, presented on a hybrid of direct search methods, quadratic model-based methods, and derivative-free line-search methods, and showed its practical potential in radiation dosimetry problems. In terms of software employing hybrid methods, Margherita Porcelli discussed BFO 2.0, a Matlab package for black-box optimization that combines a particular direct search (also compatible with categorical variables) with model-based search steps. Although not a hybrid method, Francesco Rinaldi discussed both direct-search and model-based methods when proposing improved means of satisfying probabilistic tail bounds with stochastic black-box oracles in stochastic variants of DFO methods.

2.4 Application Presentations

Two talks were significantly more focused on applications rather than algorithms. Sbastien Le Digabel discussed a benchmarking tool that he and collaborators have made available called SOLAR - it is a collection of *real* black-box optimization problems representing a wealth of distinct challenges that are derived from a real code for a solar plant simulation. Yves Lucet also presented extensive comparisons and practical experience of applying various derivative-free multiobjective optimization methods to real problems in road construction.

3 Workshop Side Objectives

As mentioned in the overview of the field, we wanted to take advantage of the unique opportunity of having so many experts in the same scientific area, gathered in one place for a week, with few outside distractions, to discuss larger problems within the field. Towards this end, we had guided "lunch discussions". Each day, participants were provide a question and encouraged to discuss their thoughts over lunch. A Google Doc page was created for participants to record their opinions and reactions to their conversations. In this section, we reproduce these questions and provide a summary of responses. Our intention is not to offer definitive answers to the questions or even to suggest a consenus, but rather to record a collection of opinions that were gathered. We would also like to remark that we are particularly grateful for the participation of John Dennis, one of the founding fathers of the DFO scientific area, who also attended to this workshop – many of his insightful comments and experiences are woven throughout this summary.

1. **DFO (Derivative-free Optimization) vs BBO (Black-box Optimization) vs ZOO (Zeroth-order Optimization): Are there differences between these terms? What are those differences?**

   Consider that *linear* optimization is the study of optimization problems with linear objectives and constraints. Similarly, *nonlinear* optimization is the study of optimization problems with nonlinear objectives or constraints. In this sense, "linear" and "nonlinear" define properties of a *problem* rather than properties of an *algorithm*. Because we are united under problems defined by a black-box, the optimization we are performing is *black-box optimization* (BBO), if we are to continue this analogy. *Derivative-free*, on the other hand, is an appropriate descriptor of an algorithm. It describes what the algorithm is given access to; in this case, the word derivative-free emphasizes that the algorithm has no access to derivatives. The implied opposite of the word derivative-free is *derivative-based*, which would describe virtually every non-derivative-free algorithm for nonlinear smooth optimization, since their convergence fundamentally relies on Taylor's theorem. In light of this, it is perhaps appropriate to attempt to rebrand DFO as DFA, and refer to the algorithms we produce as derivative-free algorithms.

This distinction and suggestion was met with some acceptance, with a few caveats. For instance, it was pointed out that many talks in this workshop were not entirely black-box, and that the word black-box potentially excludes these gray-box situations. An additional concern noted that some researchers in nonlinear optimization consider an oracle capable of returning both function and gradient values as a black-box oracle (but the mechanism or analytic form that returned those values is unknown); such a setting is certainly not within the standard scope of what many of us consider black-box optimization, which implicitly means only zeroth-order information is returned.

In terms of ZOO, there seemed to be some consensus that this is a weird term produced by the computer science community that is redundant with "derivative-free", because it is a description of the information available to solve the problem. John Dennis also commented that in his long view of optimization, referring to the order of a method is a vestige of a simpler time when optimization algorithms were easily categorized as zeroth, first, or second order methods. However, with the now-abundance of methods like BFGS (which dynamically generates subspace approximations of second order information using only first order information), and coordinate descent (which only uses partial derivative information), this classification has become cumbersome and is perhaps time to retire. This in line with our view that ZOO is generally an unnecessary term.

2. **Evolutionary algorithms/metaheuristics : Why don't our communities align? Are they aligned? Should we be aligned? What's the "threshold" for inclusion?**

Initially, it was the participants' view that an evolutionary algorithm (EA) or metaheuristic is not properly an algorithm, if one believes the definition of algorithm should include something about returning a certificate of solving a problem within a finite stopping time. As an example, gradient descent, when applied to a sufficiently regular smooth function and with appropriate step size selection, is guaranteed to stop in a quantifiably finite (expressed as a function of $\epsilon > 0$) number of iterations and return a point with objective gradient ϵ-small. By definition of being heuristics, EAs and metaheuristics cannot do this. Thus, because we as mathematicians are so keen on having theoretically obtainable results concerning the performance of algorithms, it seems that this gap is too wide.

It was pointed out, however, that there have been efforts to force convergence guarantees on heuristics by using various safeguards - work in employing heuristics in the search step of direct search methods comes to mind. To search for deeper connections between these two communities, it was noted that the majority of EAs are based on some aspect of nature and attempts to establish some analogy to nature's ability to "optimize" (e.g., ant colony optimization attempts to mimic the forging behaviour of an ant colony). If the real performance benefit of EAs is coming from the inherent randomization of update rules, this is more a case for the continued and further study of randomized algorithms rather than a specific aspect of any one metaheuristic.

The desire to reconcile the heuristic and algorithmic communities doesn't come from a vacuum. It has been repeatedly documented that heuristics are often remarkably useful in practical settings. In a cute analogy, John Dennis referred to his former hobby of maintaining an aquarium. In John's view, while we all look at the "fancy fish" (algorithms with many assumptions, like Newton's method) in the aquarium, the optimization landscape fundamentally relies on the bottom-feeders (EAs) to keep the whole ecosystem going. The bottom-feeders are ugly, but when solving a real problem, they are sometimes necessary to hold things together.

Another observation that was generated as a result of this discussion is that EAs and metaheuristics tend to come from the computer science community, which (perhaps surprisingly, to an external viewer) does not frequently intersect with the applied mathematics community. Indeed, several people noted that the majority of EAs papers they had read or EAs talks they had attended typically only discussed the implementation of a specific method with a specific parameter tuning applied to their specific problem, and documented its relative success. This is perhaps bolstered by the differences in publication style between the two communities. There was some agreement that mathematicians tend to move slower and target journal publications, whereas computer scientists tend to target conference proceedings, which historically have faster turn-around times and publication rates. There was even a sense that this rush to publish quickly in the computer science community leads to a general disinterest among computer scientists to intersect with research trends that are evolving outside of their own community. Some workshop participants even went so far as to note a frustration in their own personal endeavors to interact with computer scientists, but found that computer scientists were frequently too pressed for time to have meaningful collaborations with mathematicians.

3. **Finite difference gradients: Recently, there's been more research employing finite difference gradients (and other similar gradient approximations) within gradient-based algorithms. Is this derivative-free research?**

   Whereas the responses to the previous two questions were a bit more one-sided, this one did not seem to elicit too many strong opinions. Our initial question was driven by the observation that one, ostensibly, could propose any inexact gradient (or higher-derivative) method, spend an entire paper analyzing it, and then in the last section of a paper claim "These results can be realized, for instance, by finite differencing". In this sense, these methods have very little to do with the actual black-box nature of the problem, and thus seem outside the scope of traditional derivative-free algorithms. As an idealized point, consider the Optimization Toolbox in Matlab - if one does not supply objective or constraint gradients, then the default nonlinear optimization solvers will simply create a function that returns finite-difference gradients, and proceeds to treat this created function as a true gradient within the context of the usual derivative-based algorithms.

However, perhaps this view of derivative-free algorithms is in fact too limited. Implicit in the question that we asked was the assumption that a gradient approximation was meaningful, which implicitly assumes that the black-box function for which we are constructing a gradient approximation is smooth. This is in fact one of the arguments for the initial development of direct search methods. A finite difference gradient method should be expected to struggle in a neighborhood of a point of non-differentiability, whereas a direct search method may find more success. It is important to remember that convergence proofs for (not just derivative-free) algorithms are intended to be proofs of correctness for an algorithm applied to a particular class of problems; convergence proofs alone *do not characterize the practical scope of an algorithm*. With this lens, the difference between what we have traditionally labeled derivative-free methods and these inexact gradient methods is that the inexact gradient methods innately mean that the *practitioner* - as opposed to a mathematician - has made a smoothness assumption on the black-box. Whereas traditional derivative-free methods (but especially direct search methods) do not require that assumption on the part of a practitioner.

4. **The future of derivative-free algorithms. What are the exciting trends? What should we be working on that we aren't?**

   We should certainly note that in this workshop the amount of work that has been done recently when assuming stochastic black-boxes and when dealing with multi-objective optimization. These are important theoretical steps towards capturing the real complicated structure of black-box optimization problems. John Dennis noted that it is very important for theoreticians to continue to provide formal justifications for why and when algorithms ought to work under sets of assumptions, because this is exactly what helps steer practitioners towards better problem definitions. Several participants stressed that both the stochastic/noisy and multiobjective settings are exactly the settings that will enable the solution of more real problems.

   Another form of structure that tends to be frequently overlooked is optimization with discrete variables. We know that integer optimization has been accelerated by advanced decomposition techniques and exploitable parallelism. As BBO is often applied to hyperparameter optimization in machine learning settings, which naturally deal with (hierarchical) categorical discrete variables, there is both a need and opportunity for massive algorithmic development in this field.

   Massive parallelism also lends itself to the solution of potentially larger black-box optimization problems. As noted previously, derivative-free algorithms scale poorly with problem dimension, but coupled with algorithmic ideas like subspace methods (Wild's talk) and problem separability (Porcelli's talk), increased parallelism will certainly improve the practical performance of derivative-free methods.

   Finally, we identified that not enough research has been performed in "budget-aware" multi-fidelity optimization. In particular, given the availability of multi-fidelity objective and constraint functions, and a rough understanding of the time costs and resource availability of evaluating these functions, how can or should one go about

Figure 1: Workshop participants at the beautiful UBC Okanagan campus.

distributing a budget of core hours towards the solution of an optimization problem so that the best solution found at the end of the time horizon is taken as the output of the algorithm? This is an intriguingly difficult problem that generated a lot of opinions.

4 Outcome of the Meeting

Overall, the workshop was successful in its aims of bringing together an international collection of experts and rising experts (students) in the field of DFO. As documented in Sections 2 and 3, both the objectives and side-objectives of the meeting were achieved.

We gratefully acknowledge our funding from BIRS, without which this excellent meeting could not have occurred. A second meeting is already planned to take place in 2024, in Italy.

References

[1] C. Audet and W. Hare, *Derivative-free and black-box optimization*, Springer Series in Operations Research and Financial Engineering, Springer International Publishing, Cham, 302 pages, 2017.

Figure 2: A subset of workshop participants at a hiking excursion at Big White.

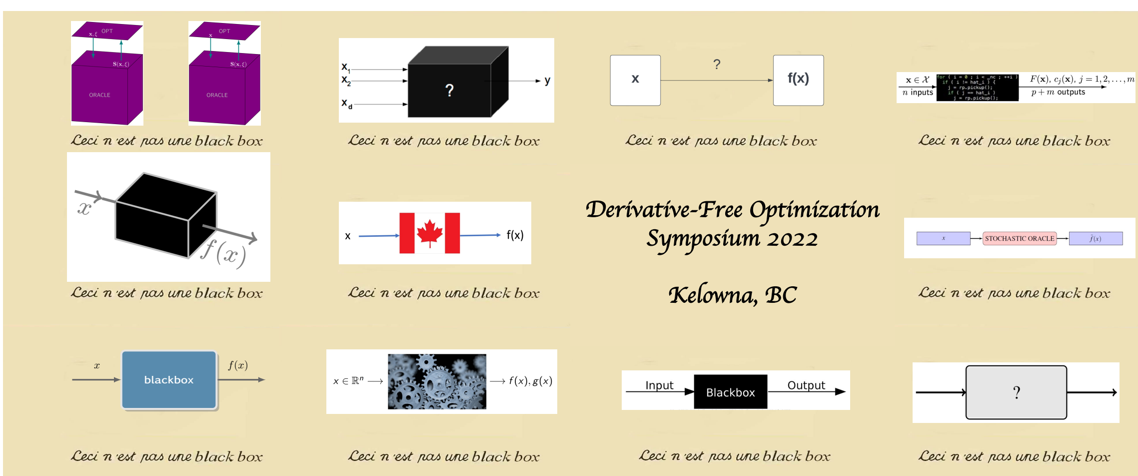

Figure 3: Because we deal with black-box optimization, many workshop participants included an image of a "black-box" in their talks. We collected these disparate representations here, and arranged them in an artful manner.