Unconventional Optimization Algorithms: An Introduction*

Jason Brownlee
jasonb@CleverAlgorithms.com
The Clever Algorithms Project
http://www.CleverAlgorithms.com

January 18, 2010
Technical Report: CA-TR-20100118-1

Abstract

The Clever Algorithms project aims to describe a large number of Artificial Intelligence algorithms in a complete, concise, and centralized manner. The techniques selected for inclusion in the project are predominately intended for function optimization and have been described as unconventional optimization algorithms. This report provides an introduction to unconventional optimization algorithms in terms of a description of the general class of problem to which they are suited and an overview of the intent, concerns, broader fields and conceptualization as to how they operate. A set of general attributes of difficult problems are described that motivate the need for heuristic methods. Two general classes of problems are introduced that include function optimization and function approximation, that latter of which can be phrased as an optimization problem. So-called unconventional optimization algorithms are introduced as those algorithms drawn from the fields of Computational Intelligence and Metaheuristics that are black box, stochastic, and operate using an inductive learning process of adaptive generate and test.

Keywords: `Clever, Algorithms, Function, Optimization, Approximation, Unconventional, Optimization, Black-box, No-Free-Lunch, Stochastic, Induction`

1 Introduction

The Clever Algorithms project aims to describe a large number of Computational Intelligence and Natural Computation algorithms in a complete, consistent, and centralized manner [6]. This report provides an overview of so-called 'unconventional optimization algorithms'. These are stochastic optimization algorithms drawn from the fields of Computational Intelligence, Biologically Inspired Computation, and Metaheurustics [8].

Section 2 provides an overview of the general features that make computational problems difficult in the fields of Computational Intelligence and Metaheuristics and introduces two general classes of problem: function optimization and function approximation. Section 3 provides an introduction to unconventional optimization that considers subjects such as black box methods, the 'no free lunch theorem' of search and optimization, the importance of randomness in the broader field of stochastic optimization, an and general strategy of inductive learning. Finally, Section 4 relates the properties of unconventional global optimization to the approaches to be described in the Clever Algorithms project and highlights some areas for future consideration.

2 Problems

Algorithms from the fields of Computational Intelligence, Natural Computing, and Metaheuristics are applied to difficult problems, to which more traditional approaches may not be suited. Michalewicz and Fogel propose five reasons why problems may be difficult [17] (page 11):

- The number of possible solutions in the search space is so large as to forbid an exhaustive search for the best answer.
- The problem is so complicated that just to facilitate any answer at all, we have to use such simplified models of the problem that any result is essentially useless.
- The evaluation function that describes the quality of any proposed solution is noisy or varies with time, thereby requiring not just a single solution but an entire series of solutions.
- The possible solutions are so heavily constrained that constructing even one feasible answer is difficult, let alone searching for an optimal solution.
- The person solving the problem is inadequately prepared or imagines some psychological barrier that prevents them from discovering a solution.

This section introduces two problem formalisms that embody many of the most difficult problems faced by Artificial and Computational Intelligence. They are: Function Optimization (in Section 2.1) and Function Approximation (in Section 2.2). Each class of problem is described in terms of its general properties, a formalism, and a set of specialized sub-problems. These problem classes provide a tangible framing of the algorithmic techniques drawn from the fields of Computational Intelligence, Natural Computing, and Metaheuristics.

2.1 Function Optimization

Real-world optimization problems and generalizations thereof can be drawn from most fields of science, engineering, and information technology (for a sample see [1, 21]). Importantly, optimization problems have had a long tradition in the fields of Artificial Intelligence in motivating basic research into new problem solving techniques, and for investigating and verifying systemic behavior against benchmark problem instances.

2.1.1 Problem Description

Mathematically, optimization is defined as the search for a combination of parameters commonly referred to as decision variables ($x = \{x_1, x_2, x_3, \dots x_n\}$) which minimize or maximize some ordinal quantity (c) (typically a scalar called a score or cost) assigned by an objective function or cost function (f), under a set of constraints ($g = \{g_1, g_2, g_3, \dots g_n\}$). For example, a general minimization case would be as follows: $f(x\prime) \leq f(x), \forall x_i \in x$. Constraints may provide boundaries on decision variables (for example in a real-value hypercube $\Re^n$), or may generally define regions of feasibility and in-feasibility in the decision variable space. In applied mathematics the field may be referred to as Mathematical Programming. More generally the field may be referred to as Global or Function Optimization given the focus on the objective function (for more general information on optimization, see [13]).

2.1.2 Sub-Fields of Study

The study of optimization is comprised of many specialized sub-fields, generally based on an overlapping taxonomy that focuses on the principle concerns in the general formalism. For example, with regard to the decision variables, one may consider univariate and multivariate optimization problems. The type of decision variables promotes the specialities for continuous, discrete, and permutations of variables. Dependencies between decision variables under a cost function define the fields of Linear Programming, Quadratic Programming, and Nonlinear Programming. A large class of optimization problems can be reduced to discrete sets and are considered in the field of Combinatorial Optimization, to which many theoretical properties are known, most importantly that many interesting and relevant problems cannot be solved by an approach with polynomial time complexity (so-called NP-complete, for example see [18]).

The topography of the response surface for the decision variables under the cost function may be convex, which is an important class of functions to which many important theoretical findings have been made, not limited to the fact that location of the local optimal configuration also means the global optimal configuration of decisional variables has been located [5]. Many interesting and real-world optimization problems produce cost surfaces that are non-convex or so called multi-modal[1] (rather than uni-modal) suggesting that there are multiple peaks and valleys. Further, many real-world optimization problems with continuous decision variables cannot be differentiated given their complexity or limited information availability meaning that derivative-based gradient decent methods that are well understood are not applicable, requiring the use of so-called 'direct search' (sample or pattern-based) methods [16]. Real-world objective function evaluation may be noisy, discontinuous, dynamic, and the constraints of real-world problem solving may require a viable or approximate solution in limited time or resources, motivating the need for heuristic approaches.

2.2 Function Approximation

The phrasing of real-world problems in the Function Approximation formalism are among the most computationally difficult considered in the broader field of Artificial Intelligence for reasons including: incomplete information, high-dimensionality, noise in the sample observations, and non-linearities in the target function. This section considers the Function Approximation Formalism and related specialization's as a general motivating problem to contrast and compare with Function Optimization.

2.2.1 Problem Description

Function Approximation is a problem of finding a function (f) that approximates a target function (g), where typically the approximated function is selected based on a sample of observations (x, also referred to as the training set) taken from the unknown target function. In machine learning, the function approximation formalism is used to describe general problem types commonly referred to as pattern recognition, such as classification, clustering, and curve fitting (called a decision or discrimination function). Such general problem types are described in terms of approximating an unknown Probability Density Function (PDF), which underlies the relationships in the problem space, and is represented in the sample data. This 'function approximation' perspective of such problems is commonly referred to as statistical machine learning and/or density estimation [10, 4].

2.2.2 Sub-Fields of Study

The function approximation formalism can be used to phrase some of the hardest problems faced by Computer Science, and Artificial Intelligence in particular such as natural language processing and computer vision. The general process focuses on (i) the collection and preparation of the observations from the target function, (ii) the selection and/or preparation of a model of the

[1] Taken from statistics referring to the centers of mass in distributions, although in optimization it refers to 'regions of interest' in the search space, in particular valleys in minimization, and peaks in maximization cost surfaces.

target function, and (ii) the application and ongoing refinement of the prepared model. Some important problem-based sub-fields include:

- *Feature Selection* where a feature is considered an aggregation of attributes, where only those features that have meaning in the context of the target function are necessary to the modeling process [15, 12].
- *Classification* where observations are inherently organized into labelled groups (classes) and a supervised process models an underlying discrimination function to classify unobserved samples.
- *Clustering* where observations may be organized into inherent groups based on common features although the groups are unlabeled requiring a process to model an underlying discrimination function without corrective feedback.
- *Curve or Surface Fitting* where a model is prepared that provides a 'best-fit' (called a regression) for a set of observations that may be used for interpolation over known observations and extrapolation for observations outside what has been observed.

The field of Function Optimization is related to Function Approximation, as many-sub-problems of Function Approximation may be defined as optimization problems. As such, many of the technique paradigms used for function approximation are differentiated based on the representation used and/or the optimization process used to minimize error or maximize effectiveness on a given approximation problem. The difficulty of Function Approximation problems centre around (i) the nature of the unknown relationships between attributes and features, (ii) the number (dimensionality) of of attributes and features, and (iii) general concerns of noise in such relationships and the dynamic availability of samples from the target function. Additional difficulties include the incorporation of prior knowledge (such as imbalance in samples, incomplete information and the variable reliability of data), and problems of invariant features (such as transformation, translation, rotation, scaling and skewing of features).

3 Unconventional Optimization

This selection provides an overview of so-named unconventional optimization algorithms for the function optimization class of problem. These types of algorithms comprise the majority of approaches selected for description in the Clever Algorithms project drawn from the fields of Computational Intelligence, Natural Computation, and Metaheuristics [7]. These algorithms are referred to as 'unconventional' to differentiate them from the more traditional approaches, not limited to mathematical optimization algorithms (such as Newton's method and Gradient descent that uses derivatives to locate a local minimum) and direct search methods (such as the Simplex method and the Nelder-Mead method that use a search pattern to locate optima).

Unconventional optimization algorithms are designed for the more difficult problem instances, the attributes of which were introduced in Section 2.1. This section introduces some common attributes of this class of algorithm including: black box methods (Section 3.1), the no-free-lunch theorem (Section 3.2), the broader field of stochastic optimization (Section 3.3), and general approach of inductive learning (Section 3.4).

3.1 Black Box Algorithms

Black Box optimization algorithms are those that exploit little, if any, information from a problem domain in order to devise a solution. They are generalized problem solving procedures that may be applied to a range of problems with very little modification [9].

Domain specific knowledge refers to making use of known relationships between solution representations and the objective cost function. Generally speaking, the less domain specific information incorporated into a technique, the more flexible the technique, although the less efficient it will be for a given problem. For example, ‘random search’ is the most general black box approach and is also the most flexible requiring only the generation of random solutions for a given problem. Random search also has a worst case behavior that is worse than enumerating an entire search domain given the freedom it has to resample the domain. In practice, the more prior knowledge available about a problem, the more information that should be exploited by a technique in order to efficiently locate a solution for the problem, heuristically or otherwise. Black box methods are those methods suitable for those problems where little information from the problem domain is available to be used by a problem solving approach.

3.2 No Free Lunch

The *No Free Lunch Theorem* of search and optimization by Wolpert and Macready proposes that all black box optimization algorithms are the same for searching for the extremum of a cost function when averaged over all possible functions [23, 22]. The theorem has caused a lot of pessimism and misunderstanding, particularly in relation to the evaluation and comparison of computational intelligence algorithms.

The implication of the theorem is that searching for the ‘best’ general-purpose black box optimization algorithm is theoretically impossible, as no such procedure exists. The theory applies to stochastic and deterministic optimization algorithms as well as to algorithms that learn and adjust their search strategy over time. It is invariant to the performance measure used and the representation selected. Perhaps the catalyst for cynicism related to algorithm benchmarking is a comment accompanying the proof suggesting that: “*. . . comparisons reporting the performance of a particular algorithm with a particular parameter setting on a few sample problems are of limited utility*” [23].

The theorem is an important contribution to computer science, although its implications are theoretical. The original paper was produced at a time when grandiose generalizations were being made as to algorithm, representation, or configuration superiority. The practical impact of the theory is to encourage practitioners to bound claims of applicability for optimization algorithms. Wolpert and Macready encouraged effort be put into devising practical problem classes and into the matching of suitable algorithms to problem classes. Further they compelled practitioners to exploit domain knowledge in optimization algorithm application, which is now an axiom in the field.

3.3 Stochastic Optimization

Stochastic optimization algorithms those that use randomness to elicit non-deterministic behaviors, contrasted to purely deterministic procedures. most algorithms from the fields of Computational Intelligence, Natural Computation, and Metaheuristics may be considered to belong to sub-fields of the field of stochastic optimization. Algorithms that exploit randomness, are not random in behavior. Rather, they sample a problem space in a biased manner, focusing on areas of interest and neglecting less interesting areas [20]. A class of techniques that focus on the stochastic sampling of a domain are called Markov Chain Monte Carlo (MCMC) algorithms that provide good average performance, quickly, and generally offer a low chance of the worst case performance. Such approaches are suited to problems with many coupled degrees of freedom, for example large, high-dimensional spaces. MCMC approaches involve stochastically sampling from a target distribution function similar to Monte Carlo simulation methods using a process that resembles a biased Markov chain.

Monte Carlo methods are used for selecting a statistical sample to approximate a given target probability density function and are traditionally used in statistical physics. Samples are drawn sequentially and the process may include criteria for rejecting samples and biasing the sampling locations within high-dimensional spaces. Markov Chain processes provide a probabilistic model for state transitions or moves within a discrete domain called a walk or a chain of steps. A Markov system is only dependent on the current position in the domain in order to probabilistically determine the next step in the walk.

MCMC techniques combine these two approaches to solve integration and optimization problems in large dimensional spaces by generating samples while exploring the space using a Markov chain process, rather than sequentially or independently [2]. The step generation is configured to bias sampling in more important regions of the domain. Three examples of MCMC techniques include the Metropolis-Hastings algorithm, Simulated annealing for global optimization, and the Gibbs sampler which are commonly employed in the fields of physics, chemistry, statistics, and economics.

3.4 Inductive Learning

Many unconventional optimization algorithms employ a process that includes the iterative improvement of candidate solutions against an objective cost function. This process of adaptation is generally a method by which the process obtains characteristics that improve the system's (candidate solution) relative performance in an environment (cost function). This adaptive behavior is commonly achieved through a 'selectionist process' of repetition of the steps: generation, test, and selection. The use of non-deterministic processes mean that the sampling of the domain (the generation step) is typically non-parametric, although guided by past experience.

The method of acquiring information is called inductive learning or learning from example, where the approach uses the implicit assumption that specific examples are representative of the broader information content of the environment, specifically with regard to anticipated need. Many unconventional optimization approaches maintain a single candidate solution, a population of samples, or a compression thereof that provides both an instantaneous representation of all of the information acquired by the process, and the basis for future learning.

This method of simultaneously acquiring and improving information from the domain and the optimization of decision making (where to direct future effort) is called the k-armed bandit (two-armed and multi-armed bandit) problem from the field of statistical decision making or game theory [19] (for a contemporary treatment see [3]). This formalism considers the capability of a strategy to allocate available resources proportional to the payoff the strategy is expected to receive. The classic example is the 2-armed bandit problem used by Goldberg to describe the behavior of the genetic algorithm [11]. The example involves an agent that learns which one of the two slot machines provides more return by pulling the handle of each (sampling the domain) and biasing future handle pulls proportional to the expected utility, based on the probabilistic experience with the past distribution of the payoff. This formalism may also be used to understand the properties of inductive learning demonstrated by the adaptive behavior of most unconventional optimization algorithms.

The limited use of prior knowledge from the domain (black box) coupled with the stochastic sampling process mean that the adapted solutions are created without top-down insight or instruction can sometimes be interesting, innovative, and even competitive with decades of human expertise [14]. The stochastic iterative process of generate and test can be computationally wasteful, potentially re-searching areas of the problem space already searched, and requiring many trials or samples in order to achieve a 'good enough' solution.

4 Conclusions

The selection of algorithms to be described in the Clever Algorithms project are predominately unconventional optimization algorithms, meaning they are black box heuristics that use stochastic processes of iterative generate and test to adapt a candidate solution for a problem domain. This report has clarified the relationship between the selected algorithms and the function optimization general class of problem, distinct from function approximation. This work has also clarified what is meant by the phrase 'unconventional optimization algorithm' and pointed out the general philosophy behind black-box algorithms, highlighted the broader field of stochastic optimization, the concerns of the no-free-lunch theorem, and suggested some explanations at how the approaches generally produce solutions.

This report was not exhaustive in either describing the two general classes of problems, or with the introduction of unconventional optimization algorithms. Sample problem instances may be provided to ground the introduction of the problem classes including both commonly cited benchmark instances as well as real world examples. The introduction of unconventional algorithms may further benefit from a more detailed explanation of subjects such as the state space paradigm, search, adaptation, and satisficing. Future efforts may consider an introduction into the classes of algorithm suited to different sub-classes of function optimization problems such as global and local, parallel, cooperative, hybrid and meta optimization.

References

[1] M. M. Ali, C. Storey, and A. Trn. Application of stochastic global optimization algorithms to practical problems. *Journal of Optimization Theory and Applications*, 95(3):545–563, 1997.

[2] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50:5–43, 2003.

[3] D. Bergemann and J. Valimaki. Bandit problems. Cowles Foundation Discussion Papers 1551, Cowles Foundation, Yale University, January 2006.

[4] C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[6] Jason Brownlee. The clever algorithms project: Overview. Technical Report CA-TR-20100105-1, The Clever Algorithms Project http://www.CleverAlgorithms.com, January 2010.

[7] Jason Brownlee. Data-driven optimization algorithm selection. Technical Report CA-TR-20100112-1, The Clever Algorithms Project http://www.CleverAlgorithms.com, January 2010.

[8] Jason Brownlee. A gentle introduction to AI: Natural computation, computational intelligence and metaheuristics. Technical Report CA-TR-20100115-1, The Clever Algorithms Project http://www.CleverAlgorithms.com, January 2010.

[9] Stefan Droste, Thomas Jansen, and Ingo Wegener. Upper and lower bounds for randomized search heuristics in black-box optimization. *Theory of Computing Systems*, 39(4):525–544, 2006.

[10] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, 1990.

[11] David Edward Goldberg. *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley, 1989.

[12] I Guyon and A Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[13] R. Horst, P. M. Pardalos, and N. V. Thoai. *Introduction to Global Optimization.* Kluwer Academic Publishers, 2nd edition, 2000.

[14] John R. Koza. *Genetic programming IV: routine human-competitive machine intelligence.* Springer, 2003.

[15] M. Kudo and J. Sklansky. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33:25–41, 2000.

[16] R. M. Lewis, V. T., and M. W. Trosset. Direct search methods: then and now. *Journal of Computational and Applied Mathematics*, 124:191–207, 2000.

[17] Zbigniew Michalewicz and David B. Fogel. *How to solve it: modern heuristics.* Springer, 2004.

[18] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Courier Dover Publications, 1998.

[19] H. Robbins. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.*, 58:527–535, 1952.

[20] James C. Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control.* John Wiley and Sons, 2003.

[21] A. Trn, M.M. Ali, and S. Viitanen. Stochastic global optimization: Problem classes and solution techniques. *Journal of Global Optimization*, 14:437–447, 1999.

[22] D. H. Wolpert and W. G. Macready. No free lunch theorems for search. Technical report, Santa Fe Institute, Sante Fe, NM, USA, 1995.

[23] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(67):67–82, 1997.