RECLAMATION

Managing Water in the West

Technical Report S&T-2009-5992

Scoping Report-- New Algorithms for Hydropower Optimization

U.S. Department of the Interior
Bureau of Reclamation
Technical Service Center
Denver, Colorado

November 15, 2011

Legal Notice

Technical Report S&T-2009-5992

Scoping Report-- New Algorithms for Hydropower Optimization

David A. Harpman
Natural Resource Economist
Economics and Resource Planning Team
U.S. Bureau of Reclamation (86-68212)
Technical Service Center
Denver, Colorado 80225

Project Funding

This scoping investigation was supported in fiscal year 2009 by the U.S. Bureau of Reclamation's Science and Technology Program, project identification (ID) number 5992.

Contents

Page

Tables

Page

Figures

Page

This page is deliberately blank.

Introduction

In recent years, a variety of new optimization heuristics have been described in the power engineering literature. The bulk of these are evolutionary algorithms. These approaches rely on innovative search techniques, drawn from biological and physical processes. Although computationally intensive, these methods can solve difficult constrained optimization problems, like the optimal economic dispatch problem, quickly and reliably.

This scoping document describes some of the specialized terms encountered in the evolutionary algorithm literature, compares and contrasts traditional calculus based optimization approaches with evolutionary algorithms and describes one such algorithm, particle swarm optimization (PSO), in detail. A review of the pertinent literature was undertaken and an extensive list of citations is included. This document is designed to inform future research efforts focused on the optimal hydropower economic dispatch problem.

Selected Terms

Like any branch of science, there are some terms used to describe mathematical optimization approaches which are not commonly encountered in other fields. As an aid to understanding the narrative which follows, it will be useful to define some of these terms.

Algorithm

"A detailed sequence of actions to perform to accomplish some task. named after an Iranian mathematician, Al-Khawarizmi. Technically, an algorithm must reach a result after a finite number of steps, thus ruling out brute force search methods for certain problems, though some might claim that brute force search was also a valid (generic) algorithm. The term is also used loosely for any sequence of actions (which may or may not terminate)" (Computer Dictionary Online 2010).

Heuristic

"A rule of thumb, simplification, or educated guess that reduces or limits the search for solutions in domains that are difficult and poorly understood. Unlike (true) algorithms, heuristics do not guarantee optimal, or even feasible, solutions and are often used with no theoretical guarantee" (Computer Dictionary Online 2010).

In practice, the term algorithm is often used interchangeably with the term heuristic. However, mathematicians typically reserve their use of the word algorithm to describing optimization approaches for which there is a theoretical mathematical basis for expecting a favorable result. Typically, mathematicians employ the term heuristic to describe any of the non-traditional optimization approaches not supported by mathematical theory.

Objective Function

The object of mathematical optimization is to minimize or maximize a specified mathematical expression. This expression is known as an objective function.

Penalty

Many applied mathematical optimization problems have natural or logical constraints on the values which can be considered in the solution. For example, physical (quantity) measurements are typically non-negative.

One approach to characterizing constraints in a constrained mathematical optimization problem is to arithmetically disadvantage, or penalize, solution results which violate a constraint. This topic is discussed in much greater detail in subsequent sections of this document. A penalty function is used to compute the numerical magnitude of the disadvantage caused by one or more constraint violations. A penalty is the value returned by a penalty function.

Fitness

In cases where penalty functions are used to characterize constraint violations, a fitness function is maximized or minimized instead of an objective function. A fitness function returns the numerical value of the fitness—defined as the objective function value plus the value of the penalties for constraint violations, if any.

Optimization Approaches

Taxonomy of Optimization Approaches

For purposes of this document and the discussion which follows, it will prove useful to provide some type of taxonomy or classification scheme to illustrate the relationship. Figure 1 provides some structure for this discussion.

As shown in Figure 1, optimization approaches can be divided into traditional (calculus based) optimization algorithms and heuristic algorithms. The latter class of optimization methods may also be described as metaheuristics or heuristic optimizers, depending on the author and the source.

The focus of this research is on a sub-set of optimization methods which are classified as heuristic algorithms. Even so, comparison and understanding of these methods is facilitated by some familiarity with traditional methods and approaches.

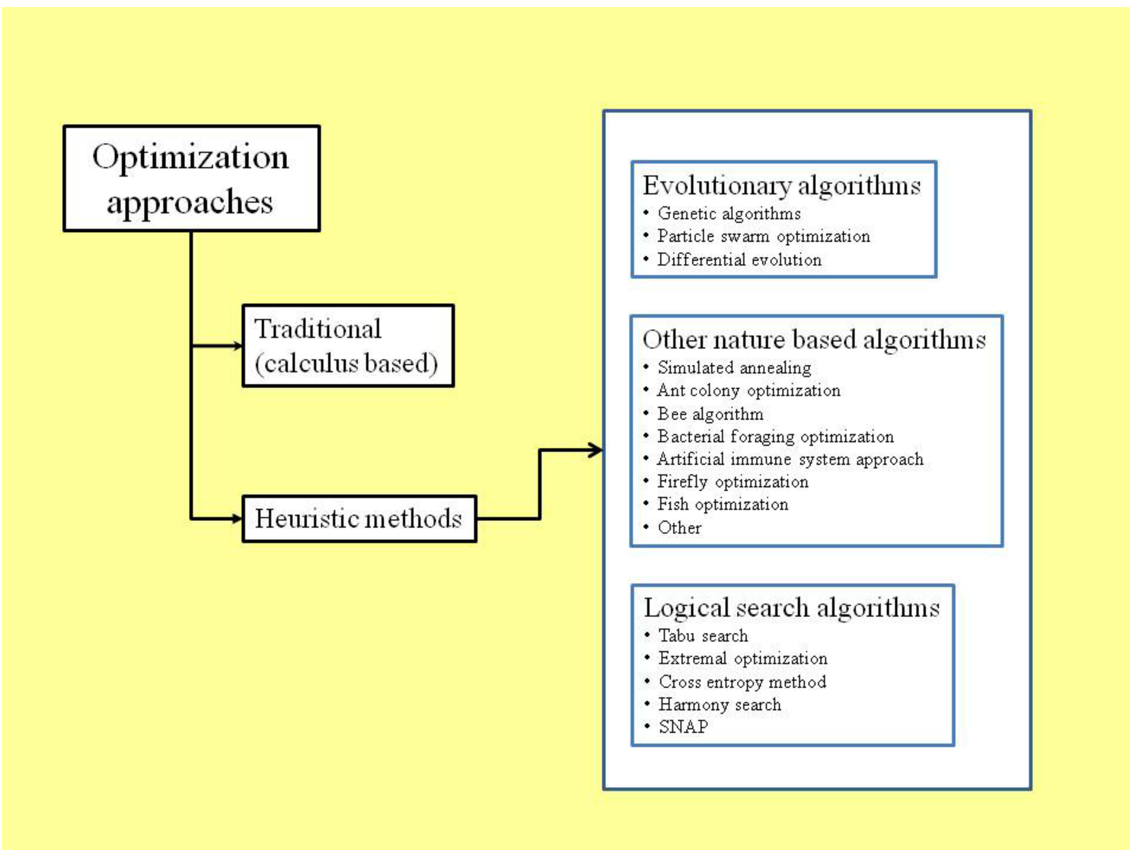

Figure 1. Taxonomy of Optimization Approaches

Traditional Solution Algorithms

Optimization problems have traditionally been addressed with a variety of traditional calculus based methods and throughout the remainder of this document, these approaches will be referred to as “traditional” or calculus based approaches. Calculus based optimization approaches are routinely taught to all engineers and economists. Most students of these disciplines will surely have fond memories of the many hours they devoted to mastery of this topic!

Since the time of Sir Isaac Newton (circa 1400), mathematicians, economists and engineers have collectively devoted vast amounts of effort to the study of optimization, with a particular focus on convex optimization problems with constraints. There are many books devoted to this subject, one of the many modern examples being the tome by Boyd and Vandenberghe (2006).

Numerical solution of convex optimization problems is typified by the Newton-Raphson approach and its many variants. This approach has been taught to engineers and economists since the early 1950’s (for example, see Wood and Wollenberg (1996) or Rau (2003)).

As described in Press et al (1989) and Judd (1999), the Newton-Raphson approach has been largely supplanted by some of its recent and more advanced variants. At the present time, two approaches are in the forefront of current calculus based optimization technology. These are the sequential quadratic programming (SQQ) method, and, the generalized reduced gradient (GRG) method. Both of these methods are aptly described in Rau (2003). The SQQ method is often used in high-end commercially available optimization platforms, such as LINGO (www.lindo.com). The GRG method has found its niche as the optimization solver incorporated in all currently shipping versions of Microsoft Excel (Fylstra et al 1998). As such, it may well be the world’s most frequently used optimization algorithm. In any case, it is almost certainly the most widely installed optimization package! As bundled with the ubiquitously available Excel program, the solver is broadly employed in graduate and undergraduate teaching (for example, see Weber 2007).

Heuristic Optimization Methods

The focus of this research is on the application of a subset of the heuristic optimization methods shown in Figure 1. Heuristic optimization approaches are based on the application of rules and logic which reduce the search space and allow for solution of difficult optimization problems. Generalizing rather broadly, we can classify these methods into the three categories shown; evolutionary algorithms, other nature based algorithms and logical algorithms.

Evolutionary algorithms explicitly characterize crossover, mutation and selection operators (Engelbrecht 2005). As might be expected by their name, evolutionary algorithms are based on the concept of biological evolution. These approaches are based on the improvement of an artificial population of individuals over a series of generations or iterations. Each individual carries a solution to the optimization problem. At each generation, the most fit individuals in the population reproduce and their offspring survive into the next generation, the less fit individuals die and their inferior genes are lost. The fitness of the population and the quality of the solutions found, improve over time. Genetic algorithms, differential evolution and particle swarm optimization fall into this category of algorithms.

There are an amazing variety of optimization heuristics which are related to organisms, their behavior or some natural physical phenomenon. Among these are ant colony optimization, bee optimization, firefly optimization and a host of others. Typically, these algorithms are predicated on the collective food location strategies typified by the species.

The realm of optimization heuristics is rather broad. As might be expected, not all of them are based on natural systems. For purposes of this document, we will classify these remaining approaches as logical heuristic search algorithms. While these may be very different from one another in search strategy, they are based on logical insights, experience and in-depth knowledge of one or more types of optimization problems. As shown in Figure 1, this category includes such well-known heuristics as Tabu search and Extremal optimization. It also includes some less well known but quite effective algorithms such as the Substitution-based Non-linear Approximation Procedure (SNAP) algorithm developed by Veselka, Schoepfle and Mahalik (2003)

Comparison of Approaches

Much of the research effort described in this report is focused on the application of evolutionary algorithms to two common hydropower optimization problems. A comparison of these two classes of algorithms and their respective suitability to these problems will provide both some background and rationale. Table 1 compares a number of pertinent characteristics of these two types of approaches.

The hydropower problems examined here are inherently nonlinear with both nonlinear and linear constraints. Both traditional and evolutionary algorithms can be applied to these types of problems. Very fast and incredibly reliable traditional algorithms are available for solving problems with linear objective functions and constraints. However, traditional algorithms are typically less efficient when applied to nonlinear objectives and nonlinear constraints. They typically require longer solution times and can fail to identify a solution more frequently in this setting.

Table 1. Traditional and Evolutionary Algorithms

	Traditional Algorithms	Evolutionary Algorithms
Problem formulation	Linear or nonlinear	Linear or nonlinear
Mathematical requirements	Smooth, continuous and twice differentiable	Can be piecewise, discontinuous and non-differentiable
Allowable constraints	Equality, inequality, linear or non-linear.	Equality, inequality, linear or non-linear.
Mathematical requirements	Calculus, linear and matrix algebra operations	Primitive mathematical operators only (add, subtract, multiply, divide)
Function return	Single solution	Multiple solutions
Nature of outcome	Deterministic	Stochastic
Optimal point	Extremal point closest to starting position usually identified. This may or may not be the global optima.	Extremal point within search range usually identified. This is more likely to be the global optima.
Memory requirements	Extensive	Modest
Convergence characteristics	Slow large-scale search Fast local convergence	Fast large-scale search Slow local convergence
Solution time	Short	Often lengthy
Code implementation	Complex (very)	Unsophisticated

Many commonly encountered hydropower problems are nonlinear, nonconvex, and have discontinuities. This includes the dynamic economic dispatch problem and the unit dispatch problem examined here. Perhaps the chief strength of evolutionary programs is their applicability to these types of real-world hydropower problems, a factor which largely motivated this research effort. The mathematical requirements for applying traditional optimization algorithms are rather restrictive. Typically, traditional algorithms can only be employed when the objective function and the constraints are smooth, continuous and twice differentiable. In contrast, evolutionary algorithms can solve a much wider range of problems including those which are discontinuous, piecewise, are not convex and which cannot be differentiated.

Both traditional and evolutionary algorithms can solve constrained optimization problems with various types of constraints including equality, inequality, linear and nonlinear constraints. Traditional algorithms are less well suited to solving optimization problems with nonlinear constraints. The solution of problems with one or more equality constraints can be problematic for evolutionary algorithms.

The mathematical requirements for implementing evolutionary algorithms are far less onerous than they are for traditional (calculus based) algorithms. In both philosophy and practice evolutionary algorithms are not based on calculus and do not use calculus constructions for obtaining a solution. In fact, some authors consider this to be their greatest strength! Evolutionary algorithms use only primitive mathematical operators such as addition, subtraction, multiplication and division. Traditional algorithms are, of course, founded in calculus concepts. As a result, they use not only gradients vectors (vectors of first partial derivatives) and hessian matrices (matrices of second partial derivatives), but also have advanced linear algebra requirements. These advanced mathematical constructs are error prone to derive and code, difficult to implement numerically and require an extremely high degree of knowledge and skill on the part of the researcher/programmer. Judd, a master of understatement, writes “Many readers could write acceptable unconstrained optimization code, but it is much more difficult to write good, stable, reliable code for constrained optimization (Judd 1999, page 142)

Traditional (calculus based) optimization algorithms return one single solution. It is *the* solution to the problem, as every economics and engineering student is acutely aware. A fundamental difference between traditional and evolutionary algorithms is that evolutionary algorithms return a population of solutions. This difference in solution paradigm is both unfamiliar and potentially confusing.

To expand upon this concept, we must recall that evolutionary algorithms characterize a population of individuals. This population is of say size, np, which could consist of from 5 to 100 individuals or more. Fundamentally, each of these np individuals stores a solution (in some cases, more than one). The stored solution consists of not only the optimal function value, but the vector of values which produces it. As the evolutionary process proceeds, each of these np solutions evolves and becomes better, or more “fit.” When the evolutionary process terminates, the result is np, not necessarily unique, individual solutions--not one single solution. As a practical matter, the analyst will often choose to report the best of these np individual solutions as *the* solution. Since evolutionary algorithms are probabilistic in nature, each new run will produce slightly different results (in contrast with a traditional algorithm which produces identically the same result for a given starting condition). In the case of evolutionary algorithms, it is customary to undertake multiple runs and report the mean and other descriptive statistics for the outcomes.

Many real-world optimization problems have more than one optimal or extremal point. At an extremum, the first order necessary conditions (FOCs) for a minimum or maximum are satisfied. In the case of a traditional calculus based algorithm, the specific extrema identified by the algorithm depends primarily on the starting conditions specified by the analyst. These types of functions are the bane of researchers everywhere! In the absence of detailed knowledge about the optimal surface, the usual procedure is to restart the traditional algorithm at many different points in the solution space and search for the global optimum point. Problems which exhibit multiple local optima can often be solved by these calculus based methods. However, there is no theoretical or practical way to guarantee the solution identified by the researcher is the global solution to the problem.

Evolutionary algorithms are sometimes described as global optimizers owing to their well-documented ability to identify the global optima within the given search space. Notwithstanding the published glowing reports, an equal body of published evidence suggests this behavior is not universally observed. Furthermore, it cannot be proved theoretically that they can be relied upon to identify the global best solution. It is most certainly true that relative to traditional algorithms, evolutionary programs carry more solutions through the iteration process and have much greater exploratory ability. These two characteristics enable evolutionary algorithms to more exhaustively traverse the solution space. Consequently, they are much more likely than traditional algorithms to identify the global optima.

Traditional optimization algorithms make heavy use of vectors, matrices and linear algebra operations, which themselves exact a huge computer memory overhead. Consequently, traditional optimization algorithms require extensive amounts of computer memory, especially for the solution of sizable problems. As little as ten years ago the practical usage of traditional optimization algorithms was restricted by the amount of physical and virtual memory addressable by existing microcomputers. In contrast, evolutionary algorithms do not make use of vectors, matrices or other advanced mathematical structures or operators. Their memory requirements are quite modest for similar size problems.

In cases where they can be applied, traditional calculus based optimization algorithms are known for their rapid converge properties. This is especially true in the case of convex functions with linear constraints. Experiments show that for traditional optimization algorithms, the initial phases of search are quite slow. Once they have identified the region where the optima resides, local convergence to the final solution is often very fast. Evolutionary algorithms on the other hand, exhibit behavior which is very much the opposite. Experiments on evolutionary algorithms demonstrate the initial search phase is very fast—the algorithms quickly and efficiently locate the region of the optima. However, the local convergence of these algorithms is slow, in some cases, painfully so. Typically, large amounts of time are required for the population to converge on an optimal point, after the region where it is located has been isolated.

The computational resources required by traditional calculus based algorithms and evolutionary algorithms differ profoundly. Not surprisingly, the time required to achieve convergence is vastly different. Traditional algorithms require large amounts of memory but typically require less than 100 major iterations to converge to a solution. Evolutionary algorithms often require thousands or tens of thousands of iterations to converge to a solution. While it is true that evolutionary algorithms utilize only primitive mathematical operations—it is no understatement to say they do so intensively! Prior to the advent of microcomputers, the lack of sufficient computing power and sheer cost of computer resources precluded the use of evolutionary algorithms for civilian purposes.

One of the advantages of evolutionary algorithms is their ease of implementation. Unlike traditional algorithms, effective cutting-edge evolutionary algorithms are routinely developed by researchers and hobbyists. As of December 2010, there a number of toolboxes and working computer codes are available. Even so, many researchers with limited resources, develop research grade evolutionary algorithms using high level computer languages such as C++, C, Fortran, Java, Visual Basic and Delphi. This is rarely the case for traditional calculus based algorithms.

Evolutionary Algorithms

Evolutionary algorithms (EAs) belong to a larger class of algorithms best described as being inspired by natural phenomenon, particularly the behavior of different organisms. These are often called nature based, nature inspired, or in some cases, biological algorithms. The universe of nature inspired algorithms is large and creative. Nature inspired algorithms span the realm from bacteria (Kim, Abraham and Cho 2007), to fireflies (Yang 2009), raindrops (Shah-Hosseini 2009), ants (Dorigo and Stutzle 2004) and beyond. Newly described algorithms appear in the literature on a regular basis. A selection of the more common and better documented nature inspired algorithms is shown in Table 2.

The evolutionary algorithms, including genetic algorithms, particle swarm optimization, and differential evolution are a sub-category of the nature inspired optimization algorithms. Evolutionary algorithms and their characteristics are the focus of this research and are discussed in greater detail in subsequent sections of this document.

Research on nature inspired algorithms is ongoing and active. There have been several evaluations and performance comparisons of nature inspired algorithms. These have typically focused on the less-esoteric members of this algorithm class. The most expansive of these evaluations is found in the book by Wahde (2008). Readily obtainable studies by Potter et al (2009) and Mezura-Montes and Lopez-Ramirez (2007) are also very useful contributions to this line of research.

Table 2. Selected Nature Inspired Optimization Algorithms

Algorithm	References
Ant colony optimization (ACO)	Dorigo and Stutzle (2004)
Artificial immune system optimization	Cutello and Nicosia (2002)
Bacterial foraging optimization	Kim, Abraham and Cho (2007)
Bee optimization	Karaboga and Bosturk (2007) Pham et al (2006)
Cuckoo algorithm	Yang and Deb (2009, 2010)
Differential evolution (DE)	Storn and Price (1995, 1997)
Firefly optimization	Yang (2010)
Fish optimization	Huang and Zhou (2008)
Genetic algorithms (GA)	Haupt and Haupt (2004)
Particle swarm optimization (PSO)	Eberhart and Kennedy (1995) Kennedy and Eberhart (2001)
Raindrop optimization	Shah-Hosseini (2009)
Simulated annealing	Kirkpatrick, Gelatt and Vecchi (1983)

Particle Swarm Optimization

Introduction

Particle swarm optimization (PSO) is one of the more promising examples of an evolutionary algorithm. It was invented by Kennedy and Eberhart (1995) who developed the concept by observing the behavior of flocking birds. Since that time, there have been an impressive number of PSO applications encompassing at least three books (Kennedy and Eberhardt 2001, Engelbrecht 2005, Clerc 2006) and over one thousand published articles.

Description of PSO

The PSO approach exploits the behavior of np-independent virtual particles, which "fly" through the search domain, have a memory and are able to communicate with other members of their "swarm." Each particle has a single purpose—to better its fitness—and thereby identify the optimum (minimum or maximum) of a function.

Although computationally intensive, PSO has many advantages over traditional optimization methods. It can accommodate continuous, discrete, nonlinear and complex objective functions as well as many forms of constraints. PSO is more likely to identify a global extrema and less prone to converge on a local optima. This approach is especially well suited for complex optimization problems characterized by multiple local extrema.

PSO Terms

There are several PSO specific terms commonly used in the literature. Among these are the following.

- Fitness function- objective function value plus penalties, if any.
- Fitness- value of the fitness function
- Personal best (p)– a particle's (own) best fitness
- Global (or neighborhood) best (g) – best fitness achieved by the swarm (or neighborhood sub-swarm)
- Velocity (v)– change in location from one iteration to the next along a single dimension

Individual Components

Each of the np particles in the swarm consists of the following components, where d is the number of dimensions in the problem:

- Coordinates of its position: $x=(x_1 \dots x_d)$
- Current velocity: $v=(v_1 \dots v_d)$
- Personal best position: $p=(p_1 \dots p_d)$
- Global (or neighborhood) best position: $g=(g_1 \dots g_d)$

Operationally each particle is typically coded as either an object, in object oriented programming languages such as C#, or as a record type.

Basic PSO Algorithm

The basic PSO algorithm is relatively straightforward as illustrated in Figure 2. First, each of the np particles in the swarm is created and their positions and velocities are initialized. The PSO iterative process then begins. During each of these iterations, (a) the fitness each of the 1…np particles is evaluated, (b) the personal best and global (or neighborhood) best of each particle in the swarm are updated, and, (c) a new velocity and a new particle position are computed. A test is then applied to determine if the swarm has converged. If the swarm has converged, the iterative process is terminated and the results are reported. If the swarm has not converged, a new iteration is undertaken. This process continues until the swarm has either converged or the maximum number of iterations has been completed.

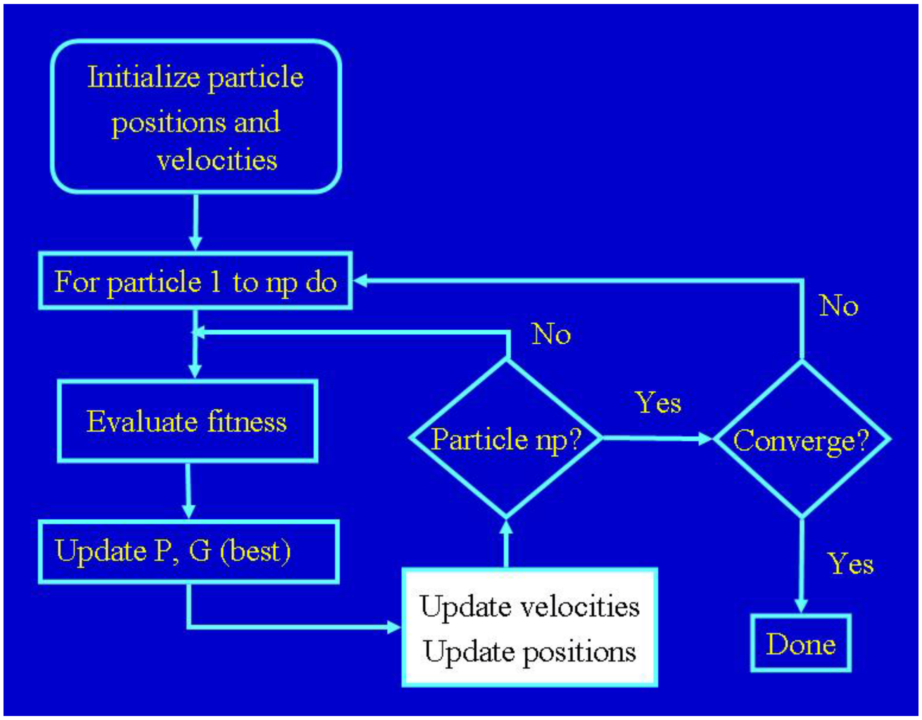

Figure 2. The basic PSO algorithm.

The velocity, or change in the location of each particle in a given dimension, is updated according to the rule illustrated in equation (1).

(1) $$v_d(t) = w[v_d(t-1)] + c_1 rand_1[g_d - x_d] + c_2 rand_2[p_d - x_d]$$

Where:

w = inertia coefficient
c1,c2 = cognitive and social weights
rand = uniform random value
v = velocity x = current location
g = global best p = personal best
t=iteration counter or index.

The new velocity of each particle depends on the velocity in the previous iteration, an inertia coefficient (w), the cognitive weight (c1), a social weight (c2), the particle's current location in each of the d-dimensions (x_d), two random uniform deviates, the particle's own personal best position (p_d), and the global (or neighborhood) best position (g_d).

After the particle's velocity has been updated, its position is updated using equation (2).

(2) $$x_d(t) = x_d(t-1) + v_d(t)$$

where:

v = velocity

x = current location

As shown, each particle's new position depends on its position in the previous iteration and the new (updated) velocity.

Conclusion

A fairly extensive review of emerging heuristic optimization algorithms was undertaken. Several promising evolutionary algorithms (EA's) were identified in the process, including the real coded genetic algorithm (RGGA), differential evolution (DE) and particle swarm optimization (PSO). A narrative and feature comparison between traditional calculus based optimization approaches and evolutionary algorithms was made. The particle swarm optimization (PSO) algorithm is described in some detail and a flowchart was constructed to allow for future code development. In aggregate, the available literature suggests these algorithms could successfully be applied to the optimal hydropower economic dispatch problem.

Literature Cited

Ao, Youyun and Hongqin Chi. "Dynamic Differential Evolution for Constrained Real-Parameter Optimization." *Journal of Advances in Information Technology* Vol.1 No. 1. (February 2010):43-51.

Banks, Alec, Jonathan Vincent and Chukwudi Anyakoha. "A Review of Particle Swarm Optimization. Part I: Background and Development. *Natural Computing* Vol 6 No. 4 (December 2007): 467-484.

Banks, Alec, Jonathan Vincent and Chukwudi Anyakoha. "A Review of Particle Swarm Optimization. Part II: Hydridization, Combinatorial, Multicriteria and Constrained Optimization, and Indicative Applications." *Natural Computing* Vol 7 No. 1 (March 2008): 467-484.

Boyd, Stephen and Lieven Vandenberghe. *Convex Optimization*. New York City, New York: Cambridge University Press. 2004. (revised 2006). 730 pages.

Brest, Jamez, Viljem Zumer and Mirjam Sepesy Maucec. "Self-Adaptive Differential Evolution Algorithm in Constrained Real-Parameter Optimization." Pages 919-926 in, Proceedings of the 2006 IEEE Congress on Evolutionary Computation. Vancouver, British Columbia, Canada. July 16-21, 2006.

Blackwell, Tim Jurgen Branke and Xiaodong Li. "Particle Swarms for Dynamic Optimization Problems." In, *Swarm Intelligence—Introduction and Applications.* Springer Natural Computing Series. Leiden Center for Natural Computing. Christian Blum and Daniel Merkle, Editors. Springer-Verlang: Berlin, Germany. 2008.

Blum, Christian and Xiaodong Li. "Swarm Intelligence in Optimization." In, *Swarm Intelligence—Introduction and Applications.* Springer Natural Computing Series. Leiden Center for Natural Computing. Christian Blum and Daniel Merkle, Editors. Springer-Verlang: Berlin, Germany. 2008.

Boyang Li, Yew-Soon Ong, Minh Nghia Le, Chi Keong Goh: Memetic Gradient Search. IEEE Congress on Evolutionary Computation 2008: 2894-2901.

Bratton, Don and Tim Blackwell. "A Simplified Recombinant PSO." In, *Swarm Intelligence—Introduction and Applications.* Springer Natural Computing Series. Leiden Center for Natural Computing. Christian Blum and Daniel Merkle, Editors. Springer-Verlang: Berlin, Germany. 2008.

Bucknall, Julian. "Ant Colony Optimizations." *The Delphi Magazine* Issue 136 (December 2006):17-22.

Bucknall, Julian. “Round & Round—How Random are Your Numbers?” *The Delphi Magazine* Issue 33 (May 1988):18-25.

Caldwell, Chris. “The Prime Pages—Prime Number Research, Records and Resources.” University of Tennessee. http://primes.utm.edu . Last accessed on 12/17/2009.

Carlisle, Anthony and Gerry Dozier. “An Off-The-Shelf PSO.” Proceedings of the Workshop on Particle Swarm Optimization. Indianapolis, IN. 2001.

Chakraborty, Uday K (Editor), *Advances in Differential Evolution.* Studies in Computational Intelligence, Volume 143. Springer-Verlang: Belin, Germany. 2008.

Chandrum, K. N. Subrahmanyam and M. Sydulu. “Brent Method for Dynamic Economic Dispatch with Transmission Losses.” *Iranian Journal of Electrical and Computer Engineering* Vol 8 No. 1 (Winter-Spring 2009):16-22,

Coelho, Leandro do Santos and Viviana C. Mariani. “Improved Differential Algorithms for Handling Economic Dispatch Optimization with Generator Constraints.” *Energy Conversion and Management.* Vol. 48 No. 5 (May 2007):1631-1639.

Coello Coello, Carlos A. “Theoretical and Numerical Constraint-Handling Techniques Used With Evolutionary Algorithms: A Survey of the State Of the Art.” *Computer Methods in Applied Mechanics and Engineering* 191 No. 11-12 (January 2002): 1245-1287.

Computer Dictionary Online. Web based dictionary of computer terms. Located at: www.computer-dictionary-online.org Last accessed on 28 September 2010.

Clerc, Maurice. “Initializations for Particle Swarm Optimization.” Unpublished manuscript. 24 December 2008. Available from: http://clerc.maurice.free.fr/pso/ . Last accessed on 12/31/2009.

Clerc, Maurice. “Confinements and Biases in Particle Swarm Optimization.” Unpublished manuscript. 12 March 2006. Available from: http://clerc.maurice.free.fr/pso/ . Last accessed on 01/04/2010.

Clerc, Maurice. *Particle Swarm Optimization*. London, England: ISTE Publishing Company. 2006.

Clerc, Maurice and James Kennedy. “The Particle Swarm—Explosion, Stability and Convergence in a Multidimensional Complex Space.” *IEEE Transactions on Evolutionary Computation* Vol 6 No. 1. (February 2002): 58-73.

Cutello, Vincenzo and Giuseppe Nicosia "An Immunological Approach to Combinatorial Optimization Problems" in, *Lecture Notes in Computer Science*, Vol. 2527. Proceedings of the 8th Ibero-American Conference on AI: Advances in Artificial Intelligence. 2002 pp. 361–370.

Das, Swagatam, Ajith Abraham and Amit Konar. “Particle Swarm Optimization and Differential Evolution Algorithms: Technical Analysis, Applications and Hybridization Perspectives.” *Advances of Computational Intelligence in Industrial Systems* Vol. 116. Ying Liu et al. (Eds.), Studies in Computational Intelligence, Springer Verlag, Germany, 2008. pp. 1–38.

Deb, Kalyanmoy. “An Efficient Constraint Handling Method for Genetic Algorithms.” *Computer Methods in Applied Mechanics and Engineering* Vol. 186 No. (2-4) (June 2000): 311-338.

Deep, Kusum and Manoj Thakur. “A New Crossover Operator for Real Coded Genetic Algorithms.” *Applied Mathematics and Computation* Vol. 188 No. 1 (May 2007):895-911.

De Jong, Kenneth A. “Analysis of the Behavior of a Class of Genetic Adaptive Systems.” Unpublished Ph.D. Dissertation. Computer and Information Sciences. University of Michigan, Ann Arbor. 1975.

Dorigo, Marco and Thomas Stutzle. *Ant Colony Optimization*: MIT Press, Inc. July 2004. 319 pages.

Eberhart, Russell C. and James Kennedy. “A new optimizer using particle swarm theory.” Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, October 1995; 39-43.

Edwards, Brian K. *The Economics of Hydroelectric Power*. New Horizons in Environmental Economics. Northampton, Massachusetts: Edward Elgar Publishing Inc. 2003.

Edwards, Brian K., Silvio J. Flaim, and Richard E. Howitt. “Optimal Provision of Hydroelectric Power Under Environmental and Regulatory Constraints.” *Land Economics*. Vol 75 No. 2 (May 1999):267-283.

Edwards, Brian K., Richard E. Howitt, and Silvio J. Flaim. “Fuel, Crop, and Water Substitution in Irrigated Agriculture.” *Resource and Energy Economics* 18 No. 3 (October 1996):311-331.

Engelbrecht, Andries P. *Fundamentals of Computational Swarm Intelligence*. Hoboken, New Jersey: John Wiley & Sons, Ltd. 2005.

Farmani, Raziyeh and Jonathan A. Wright. "Self-Adaptive Fitness Formulation for Constrained Optimization." *IEEE Transactions on Evolutionary Computation* Vol. 7 No. 5 (October 2003):445-455.

Feoktistov, Vitaliy. *Differential Evolution—In Search of Solutions* Volume 5 in Optimization and its Applications Series. Panos M. Pardalos, Managing Editor. Springer, New York, NY. 2006.

Forsund, Finn R. *Hydropower Economics* International Series in Operations Research and Management Science. Frederic S. Hillier, Series Editor. New York, NY. Springer. 2010.

Fylstra, Daniel, Leon Lasdon, John Watson and Allen Warren. "Design and Use of the Microsoft Excel Solver." *Interfaces* 28 No. 5 (September-October) 1998:29-55.

General Electric Energy Corporation. *Western Wind And Solar Integration Study* Prepared for the National Renewable Energy Laboragory. GE Energy. Schenectady, NY. May 2010. 536 pages.

General Electric Energy Corporation. MAPS™ Multi-Area Production Simulation Model Product Description. GE Energy. Schenectady, NY. March 2008. 33 pages.

Goldberg, David E. *Genetic Algorithms in Search Optimization, and Machine Learning*. Reading, Massachusetts: Addison-Wesley Professional Inc. 1989 (reissued). 432 pages.

Gong, Wenyin, Alvaro Fialho and Zhihua Cai. "Adaptive Strategy Selection in Differential Evolution." Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2010. July 7-11, 2010. Portland, Oregon. ACM Press, Inc. 2010.

Haupt, Randy L. and Sue Ellen Haupt. *Practical Genetic Algorithms*. 2nd Edition. John Wiley and Son, Inc, New York, N.Y. 2004. 192 pages.

Harpman, David A. "Assessing the Short-Run Economic Cost of Environmental Constraints on Hydropower Operations at Glen Canyon Dam." *Land Economics* 75 No. 3 (August 1999):390-401.

Helwig, Sabine and Rolf Wanka. "Particle Swarm Optimization in High-Dimensional Bounded Search Spaces." Proceedings of the 2007 IEEE Swarm Intelligence Symposium. 1-5 April 2007 Honolulu, HI, Pages 198-205.

Holland, John H. *Adaption in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press. 1975.

Hu, Xiaohui and Russell Eberhart. “Solving Constrained Nonlinear Optimization Problems with Particle Swarm Optimization.” Pages 203-206 in, Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002), Orlando, USA. 2002.

Huang, Hua-Juan and Yong-Quan Zhou. “Hybrid Artificial Fish Swarm Algorithm For Global Optimization Problems” *Journal of Computer Applications*. Vol. 28, no. 12 (December 2008): 3062-3064.

Judd, Kenneth L. *Numerical Methods in Economics*. Second Printing. Cambridge, Massachusetts: MIT Press. 1999.

Karaboga, Dervis and Bahriye Basturk. “A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm.” *Journal of Global Optimization* 39 No. 3 (November 2007):459-471.

Kennedy, James and Rui Mendes. “Population Structure and Particle Swarm Performance.” Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress. Honolulu, HI , USA. 12 May 2002 - 17 May 2002. pages 1671 – 1676.

Kennedy, James and Russell C. Eberhart. *Swarm Intelligence*. San Francisco, CA: Morgan Kaufmann Academic Press, 2001.

Kim, Dong Hwa, Ajith Abraham and Jae Hoon Cho. “A Hybrid Genetic Algorithm and Bacterial Foraging Approach for Global Optimization.” *Information Sciences* Vol 177 No. 18 (September 2007): 3918–3937.

Kirkpatrick, Scott, Charles D. Gelatt, M. P. Vecchi. “Optimization by Simulated Annealing” *Science*, New Series, Vol. 220 No. 4598 (May 1983): 671-680.

Klimasauskas, Casimir C. “Not Knowing Your Random Number Generator Could be Costly: Random Generators-- Why They are Important.” *Personal Computer Artificial Intelligence* Vol 16 No. 3 (May/June 2002):52-59.

Knuth, Donald E. *The Art of Computer Programming: Seminumerical Algorithms* Vol 2. 3rd Edition. Massachusetts: Addison-Wesley Press, Inc. 2002.

Kumar, Awadhesh. “Dynamic Economic Dispatch Using Particle Swarm Optimization.” Unpublished Masters Thesis. Electrical and Instrumentation Engineering Department, Thapar University, Patiala. June 2009.

Lee, Kwang Y. and Jong-Bac Park. “Application of Particle Swarm Optimization to Economic Dispatch Problem: Advantages and Disadvantages.” in, proceedings of the Power Systems Conference and Exposition, 2006. Atlanta, GA. Oct. 29 - Nov. 1 2006. pages 188 – 192.

Li, Boyang, Yew-Soon Ong, Minh Nghia Lee and Chi Keong Goh. "Memetic Gradient Search." *Evolutionary Computation 2008*. Proceedings of the 2008. IEEE World Congress on Computational Intelligence. Hong Kong, China. June 1-6, 2008. pages 2894 – 2901.

Liu, Hui, Zixing Cai and Yong Wang. "Hybridizing Particle Swarm Optimization with Differential Evolution for Constrained Numerical and Engineering Optimization." *Applied Soft Computing* Vol 10 No. 2 (March 2010):629-640.

Matsumoto, Makoto and Takuji Nishimura. "Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudorandom Number Generator." *Association for Computing Machinery (ACM) Transactions on Modeling and Computer Simulations* Vol 8 No. 1 (January 1998):3-30.

Mezura-Montes, Efren and Blanca Cecilia Lopez-Ramirez. "Comparing Bio-Inspired Algorithms in Constrained Optimization Problems." Pages 662-669 in, Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2007, 25-28 September 2007, Singapore. 2007

Mezura-Montes, Efren, Jesus Velazquez-Reyes and Carlos A. Coello Coello. "Modified Differential Evolution for Contrained Optimization" Pages 25-32 in, Proceedings of the 2006 IEEE Congress on Evolutionary Computation. Vancouver, British Columbia, Canada. July 16-21, 2006.

Mezura-Montes, Efren and Jorge Isacc Flores-Mendoza. "Improved Particle Swarm Optimization in Constrained Numerical Search Spaces." in Raymond Chiong (Editor), *Nature-Inspired Algorithms for Optimisation*, pages: 299-332, Springer-Verlag, Studies in Computational Intelligence Series Vol. 193, 2009.

Michalewicz, Zbigniew. *Genetic Algorithms + Data Structures = Evolution Programs.* 3rd Ed. Springer. 1996. 387 pages.

Michalewicz, Zbigniew and David B. Fogel. *How to Solve it: Modern Heuristics.* 2nd Ed. Springer. 2010. 554 pages.

Mishra, S.K. "Global Optimization by Differential Evolution and Particle Swarm Methods Evaluation on Some Benchmark Functions." MPRA Paper No. 1005, posted 07. November 2007. http://mpra.ub.uni-muenchen.de/1005 .

Miranda, Mario J. and Paul L. Fackler. *Applied Computational Economics and Finance.* Cambridge, Massachusetts: MIT Press. 2006.

Monson, Christopher K. and Kevin D. Seppi. "Linear Equality Constraints and Homomorphous Mappings in PSO." in, Proceedings of the 2005 IEEE World

Congress on Evolutionary Computation, Vol. 1 pages 73-80. Edinburgh, Scotland. September 5, 2005.

Nelder, John A. and Roger Mead. “A Simplex Method for Function Minimization” *Computer Journal* 7 No. 4 (January 1965):308-313.

Neri, Ferrante and Ville Tirronen. “Recent Advances in Differential Evolution: A Survey and Experimental Analysis.” *Artificial Intelligence Review* 33 Nos. 1-2 (February 2010): 61-106.

Nguyen, Q. H., Yew-Soon Ong, Natalio Krasnogor. “A Study on the Design Issues of Memetic Algorithm.” IEEE Congress on Evolutionary Computation 2007: 2390-2397

Noman, Nasimul and Htoshi Iba. “Accelerating Differential Evolution Using Adaptive Local Search.” *IEEE Transactions on Evolutionary Computation* Vol. 12 No. 1. (February 2008):107-125.

Omran, Mahamed G.H. “Using Opposition-based Learning with Particle Swarm Optimization and Barebones Differential Evolution.” Chapter 23 in, *Particle Swarm Optimization*. Edited by Aleksandar Lazinica. Vienna, Austria: INTECH-Education and Publishing. January 2009. 476 pages.

Omran, Mahamed G.H., Andries P. Engelbrecht and Ayed Salman. “Bare Bones Differential Evolution.” *European Journal of Operations Research* Vol 196 No. 1 (July 2009):128-139.

Paquet, Ulrich and Andries P. Engelbrecht. “Particle Swarms for Linearly Constrained Optimization.” *Fundamenta Informaticae* Vol. 76 No. 1-2 (March 2007):147-170.

Paquet, Ulrich and Andries P. Engelbrecht. “A New Particle Swarm Optimiser for Linearly Constrained Optimization.” in, proceedings of the 2003 IEEE World Congress on Evolutionary Computation Vol. 1 pages 227-233. Canberra, Australia. December 8-13, 2003.

Pant, Millie, Radha Thangaraj, Ved Pal Singh and Alith Abraham. “Particle Swarm Optimization Using Sobol Mutation.” Pages 367 to 372 in, The Proceedings of the 2008 First International Conference on Emerging Trends in Engineering and Technology. Nagpur, India. 16-18 July 2008.

Pant, Millie, Radha Thangaraj and Alith Abraham. “Low Discrepancy Initialized Particle Swarm Optimization for Solving Constrained Optimization Problems.” *Fundamenta Informaticae* 95 No. 4 (December 2009):1-21.

Pant, Millie, Ved Pal Singh and Alith Abraham. “Differential Evolution using Quadratic Interpolation for Initializing the Population.” Pages 375 to 380 in, Proceedings of the 2009 Advance Computing Conference, IACC 2009. IEEE International. Delhi, India. 6-7 March 2009.

Park, Stephen K. and Keith W. Miller, "Random Number Generators: Good Ones Are Hard to Find." *Communications of the Association for Computing Machinery (ACM)*, Vol. 31 No. 10 (October 1988): 1192-1201.

Parsopoulos, Konstantinos E, and Michel N. Vrahatis. “Initializing the Particle Swarm Optimizer Using the Nonlinear Simplex Method” In, *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*. The Artificial Intelligence Series. Edited by A. Grmela and N.E. Mastorakis. World Scientific and Engineering Academy and Society (WSEAS) Press: Interlaken, Switzerland. 2002.

Pedersen, Magnus Erik Hvass. “Tuning & Simplifying Heuristical Optimization.” Unpublished Ph.D. Thesis. School of Engineering Sciences, Computational Engineering and Design Group. University of Southampton. England. January 2010.

Peltokangas Riikka and Aki Sorsa. “Real-coded Genetic Algorithms and Nonlinear Parameter Identification.” Report A No. 48. Control Engineering Laboratory, University of Oulu. April 2008.

Pham, D. T,. A. Ghanbarzadeh, E. Koc., S. Otri, S. Rahim and M. Zaidi. “The Bees Algorithm—A Novel Tool for Complex Optimization Problems.” Innovative Production Machines and Systems 2006 Virtual Conference. July 2006. Available from: http://conference.iproms.org/forums/iproms_2006/optimisation_techniques . Last accessed on 12/29/2009.

Pomeroy, Paul. “An Introduction to Particle Swarm Optimization.” March 2003. Online article accessed at: www.adaptiveview.com/articles.

Potter, Walter D., Eric Drucker, Pete Bettinger, Frederick Maier, Max Martin, D. Luper, M. Watkinson, G. Handy, C. Hayes. “Diagnosis, Configuration, Planning, and Pathfinding: Experiments in Nature-Inspired Optimization.” *in, Natural Intelligence for Scheduling, Planning and Packing Problems.* Studies in Computational Intelligence. Vol. 250. Berlin, Germany: Springer Verlag. 2009. pages 267-294.

Press, William H., Brian P. Flannery, Saul A. Teukolsky and William T. Vetterling. *Numerical Recipes in Pascal—The Art of Scientific Computing.* New York: Cambridge University Press. 1989.

Price, Kenneth V. and Rainer M. Storn. "Differential Evolution" *Dr. Dobb's Journal* Issue 264 (April 1997):18-24 and 78.

Price, Kenneth V., Rainer M. Storn and Jouni A. Lampinen. *Differential Evolution - A Practical Approach to Global Optimization.* Springer-Verlang: Belin, Germany. 2005. 538 pages.

Rahnamayan, Shahryar and G. Gary Wang. "Solving Large Scale Optimization Problems by Opposition-Based Differential Evolution (ODE). " *WSEAS Transactions on Computers* 10 Vol. 7 (October 2008):1792-1804.

Rahnamayan, Shahryar, Hamid R. Tizhoosh and Magdy M.A. Salama. "Opposition-Based Differential Evolution." Chapter 6 in, *Advances in Differential Evolution.* Studies in Computational Intelligence, Volume 143. Edited by Uday K. Chakraborty. Springer-Verlang: Belin, Germany. 2008.

Rajkumar, N. Timo Vekara and Jarmo T. Alander. "A Review of Genetic Algorithms in Power Engineering." in, *AI and Machine Consciousness—Proceedings of the 13th Finish Artificial Intelligence Conference.* Tapani Raiko, Penti Haikonen and Jaakko Vayrynen, editors. Esppo, Finland. August 20-22, 2008. pages 15-32.

Reeves, Collin R. "Genetic Algorithms." Chapter 3 in, *Handbook of Metaheuristics* 2nd Edition. edited by Michel Gendreau and Jean-Yves Potvin. Springer Business and Science: New York City, NY. 2010. 650 pages.

Rau, Narayan S. *Optimization Principles—Practical Applications to the Operation and Markets of the Electric Power Industry.* IEEE Press Series on Power Engineering. P.M. Anderson, Series Editor. New York, New York: John Wiley and Sons. 2003. 339 pages.

Richards, Mark and Dan Ventura. "Choosing a Starting Configuration for Particle Swarm Optimization." pages 2309–2312 in, *Proceedings of the Joint Conference on Neural Networks.* July 2004.

Robertson, Grant. "How Powerful was the Apollo 11 Computer?" *Download Squad*. Weblogs, Inc. RSS Feed. July 20, 2009 at 8:30 pm.

Shah-Hosseini, Hamedi, "The Intelligent Water Drops Algorithm: A Nature-Inspired Swarm-Based Optimization Algorithm." *International Journal of Bio-Inspired Computation,* Vol. 1, Nos. 1 and 2 (January 2009):71-79.

Simopoulos, Dimitris N. Stavroula D. Kavatza and Costas D. Voumas. "An Enhanced Peak Shaving Method for Short Term Hydrothermal Scheduling." *Energy Conversion and Management* Vol. 40 No. 1 (November 2007): 2018-3024.

Staschus, Konstantin, Andrew M. Bell, and Eileen Cashman. "Usable Hydro Capacity, Electric Utility Production Simulations, and Reliability Calculations."

Institute of Electrical and Electronics Engineers (IEEE), *Transactions on Power Systems* Vol 5 No. 2 (May 1990):531-538.

Storn, Rainer M. and Kenneth V. Price. *Differential Evolution—A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces.* Technical Report TR-95-12, International Computer Science Institute. March 1995. Available from: http://www.icsi.berkeley.edu .

Storn, Rainer M. and Kenneth V. Price. "Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces." *Journal of Global Optimization* 11 No. 4 (December 1997):341-359.

Tang, Jun and Xiaojuan Zhao. "A Hybrid Particle Swarm Optimization with Adaptive Local Search." *Journal of Networks* Vol. 5 No. 4 (April 2010):411-418.

Tang, K. X. Yao, P. N. Suganthan, C. MacNish, Y. P. Chen, C. M. Chen, and Z. Yang. *Benchmark Functions for the CEC'2008 Special Session and Competition on Large Scale Global Optimization*. Technical Report, Nature Inspired Computation and Applications Laboratory, USTC, China, 2007.

Uy, Nguyen Quang, Nguyen Xuan Hoai, RI McKay, and Pham Minh Tuan. "Initializing PSO with Randomised Low-Discrepancy Sequences: The Comparative Results." Pages 1985-1992 in, Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2007, 25-28 September 2007, Singapore. 2007.

Valle, Yamille del, Ganesh Kumar Venayagamoorthy, Salman Mohagheghi, Jean-Carlos Hernandez and Ronald G. Harley. "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems." *IEEE Transactions on Evolutionary Computation* Vol 12 No. 2 (April 2008):171-195.

Veselka, Thomas D., Leslie D. Poch, Clayton S. Palmer, Samuel Loftin and Brent Osiek *Financial Analysis of Experimental Releases Conducted at Glen Canyon Dam during Water Years 1997 through 2005* ANL/DIS-10-7 August 2010. [REVISED VERSION]

Veselka, Thomas D., Leslie D. Poch, Clayton S. Palmer, Samuel Loftin and Brent Osiek. *Ex Post Power Economic Analysis of Record of Decision Operational Restrictions at Glen Canyon Dam* ANL/DIS-10-6. Argonne National Laboratory. Argonne, Illinois. April 2010. 96 pages.

Veselka, Thomas D., O. Benjamin Schoepfle and Matthew Mahalik. *CRSP Basin-Wide Economic Methodology—Modeling the Aspinall Cascade*. Systems Science Group, Argonne National Laboratory. Argonne, Illinois: Argonne National Laboratory. July 2003.

Vicaria, Fernando. “Totally Random—Getting a Truly Random Number.” *Delphi Informant* 9. No. 10 (October 2003):8-14.

Wahde, Mattias. *Biologically Inspired Optimization Methods-- An Introduction* Southampton, MA: WIT Press. 2008. 225 pages.

Weber, Ernst Juerg. “Optimal Control Theory for Undergraduates Using the Microsoft Excel Solver Tool.” *Computers in Higher Education Economics Review (Cheer)* 19 No. 1 (2007):4-15.

Weise, Thomas. *Global Optimization—Theory and Practice* 2nd Edition. eBook. Version 2008-07-07. Available from: www.it-weise.de . 2008. 703 pages.

Wong, Tien-Tsin, Wai-Shing Luk and Pheng-Ann Heng. “Sampling with Hammersley and Halton Points.” *Journal of Graphics Tools* 2 No. 2 (November 1997): 9-24.

Wolpert, David H. and William G. Macready. “No Free Lunch Theorems for Optimization” *IEEE Transactions on Evolutionary Computation* Vol. 1 No. 1 (April 1997):67-82.

Wood, Allen J. and Bruce F. Wollenberg. *Power Generation, Operation and Control.* 2nd Edition. New York, New York: John Wiley and Sons, 1996.

Wright, Alden H. “Genetic Algorithms for Real Parameter Optimization.” In, *Foundations of Genetic Algorithms.* Gregory J.E. Rawlins (editor). San Mateo, CA: Morgan Kaufmann. 1991. pp. 205-218.

Yang, Xin-She. “Firefly Algorithm, Stochastic Test Functions and Design Optimization.” *International Journal of Bio-Inspired Computation* Vol 2 No. 2. (March 2010):78-84.

Yang, Xin-She and Suash Deb. “Engineering Optimization by Cuckoo Search.” *International Journal of Mathematical Modeling and Numerical Optimization.* Vol. 1 No.4 (April 2010):330-343.

Yang, Xin-She and Suash Deb. “Cuckoo Search via Levy Flights.” in, *Proceedings of the World Congress on Nature and Biologically Inspired Computing.* (NaBIC 2009, India). IEEE Publications, USA. pages 210-214.

Zielinski, Karin and Rainer Laur. “Stopping Criteria for Differential Evolution in Constrained Single-Objective Optimizations.” Chapter 5 in, *Advances in Differential Evolution.* Studies in Computational Intelligence, Volume 143. Edited by Uday K. Chakraborty. Springer-Verlang: Belin, Germany. 2008.

Zielinski, Karin, Petra Weitkemper, Rainer Laur and Karl-Dirk Kammeyer. "Examination of Stopping Criteria for Differential Evolution based on a Power Allocation Problem." In, volume 3, pages 149–156, Proceedings of the 10th International Conference on Optimization of Electrical and Electronic Equipment. Brasov, Romania, 18-19 May 2006.

Zielinski, Karin and Rainer Laur. "Stopping Criteria for Constrained Single-Objective Particle Swarm Algorithm." *Informatica* 31 No. 1 (March 2007):51-59.

Mission Statements

The mission of the Department of the Interior is to protect and provide access to our Nation's natural and cultural heritage and honor our trust responsibilities to Indian Tribes and our commitments to island communities.

The mission of the Bureau of Reclamation is to manage, develop, and protect water and related resources in an environmentally and economically sound manner in the interest of the American public.