

Benchmark Tests of Evolutionary Algorithms: Mathematic Evaluation and Application to Water Distribution Systems

B. S. Jung^{1*}, B. W. Karney² and M. F. Lambert³

¹MWH Soft, 300 N. Lake Avenue Pasadena, CA 91101, USA

²Department of Civil Engineering, Univ. of Toronto, Toronto, ON M5S 1A4, Canada

³Department of Civil and Environmental Engineering, University of Adelaide, Adelaide SA5005, Australia

ABSTRACT. Evolutionary Algorithms (EAs) are a set of probabilistic optimization algorithms based on an analogy between natural biological systems and engineered systems. In this paper, the computational performance a set of specific EAs (specifically, the Genetic Algorithm, Evolutionary Programming, Particle Swarm Optimization, Ant Colony Optimization and Shuffled Complex Evolution Algorithm) are compared using a set of four mathematical test objective functions. In addition, a hybridization of EAs with other local search methods is introduced to improve or fine-tune the performance the primary EA. As a case study, the EAs are applied to a calibration problem for a water distribution system and ably show their robust and global convergence characteristics.

Keywords: Benchmark, calibration, evolutionary algorithms, hybridization

1. Introduction

Although sometimes requiring great effort to do so, many engineering design problems can be casted as an optimization problem. The fundamental process of an optimization begins with some candidate solutions and iteratively refines and improves them through various procedures. The task is to find a design that is optimal in terms of a specific objective function where the solution is determined by the values of the decision variables that best achieves the specified purpose of the project.

Classical methods of optimization involve the use of gradients or higher-order derivatives of the objective function with respect to changes in the candidate solutions. Such methods can be shown to have exponential local convergence on certain well-conditioned functions, but could also result in suboptimal solutions on the responses that generate multiple local optima. Also, they are not well suited for processing inaccurate, noisy and complex data, although they do sometimes excel at dealing with complicated systems (Back et al., 1997). The real world problems always generate non-convex response surface so robust methods of optimization are often required to generate suitable results.

Evolutionary Algorithms (EAs) define a field where such difficult optimization problems are studied in depth. It complements the study of traditional computational systems. There are several types of EAs in use today. The four traditional algorithms are: the Genetic Algorithm (GA) created by Holland (1973) and made famous by Goldberg (1989), the Evolutionary Programming (EP) created by Fogel (1963) and

developed further by his son Fogel (1992), the Evolution Strategies (ES) strongly promoted by Back (1996), and the Genetic Programming (GP) recently developed by Koza (1992). More recently, new EAs have arisen, such as Particle Swarm Optimization (PSO) created by Kennedy and Eberhart (1995), Ant Colony Optimization (ACO) created by Dorigo and Gambardella (1997), and the Shuffle Complex Evolution (SCE) algorithm introduced by Duan et al. (1992). The field of evolutionary computation has grown up around these techniques, with its roots still firmly in evolutionary biology and computer science.

The purpose of this paper is to present EA and its specific algorithms of GA, EP, PSO, SCE and ACO. Also, a hybridization of EA with a local search method is suggested to improve the performance of the original EA. These four algorithms are tested with four nonlinear benchmark functions where two are unimodal and the other two are multimodal. Moreover, EA is applied to an optimization problem for a water distribution system calibration using so-called inverse transient analysis (Liggett and Chen, 1994; Jung and Karney, 2004a). The EAs help to solve the inverse problem in which system parameters (such as the lumped leak coefficients and friction factors) are determined using measured transient pressure head data, a transient simulation program and an optimization routine.

2. General Framework of Evolutionary Algorithms

EA is the study of computational systems that use ideas and get inspiration from the natural evolution and adaptation. This method aims to understand such computational systems and developing more robust and efficient ones for solving the complex real-world problems. Problems that are dealt with by

* Corresponding author: BongSeog.Jung@mwhsoft.com

such computational systems are usually highly nonlinear and contain inaccurate and noisy data.

All evolutionary algorithms have two prominent features which distinguish themselves from other search algorithms. First, they are all population-based. A certain number of individuals, grouped as a population, are used to explore the solution space and thus to find the optimum in the system. Although some of EAs (i.e., Evolution Strategies) may have much more offspring than that of the parent generation, the best-fitted offspring are selected to achieve the same number from one generation to the next. Second, there is communications and information exchange among individuals in the test population. Such communications and information exchange are represented as search operators (i.e., crossover, recombination and mutation) in Figure 1 which summarize a general framework of EA. By applying the search operators, offspring (new individuals) can be generated from parents (existing individuals).

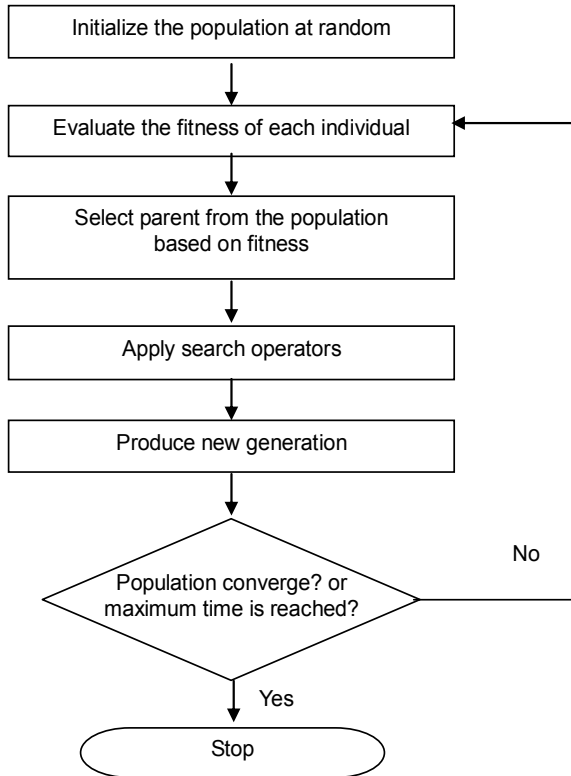


Figure 1. General framework of ECA.

3. Characters of Evolutionary Computation

EA is an emerging field which has grown rapidly in recent years. EA has been found to be a source of some of the most flexible, efficient and robust of all search algorithms known to the computer science. There are many beneficial characteristics of evolutionary computation.

The main advantage of EA is that it is conceptually sim-

ple. The traditional deterministic gradient-based methods can provide a rigid guarantee of success, but they normally do so at the expense of requiring that the function satisfy certain restrictive conditions such as meeting a number of specific gradient conditions. Even with these, gradient-based methods sometimes converge to local sub-optima. In addition, they are not robust for dynamic changes in the environment and often require a complete restart in order to provide a solution (e.g., dynamic programming). On the other hand, the simplicity of EA provides a superior adaptability for finding a solution in changing circumstance. Moreover, EA involves the evaluation of the function at a random sample of points in the feasible parameter space, followed by subsequent manipulations of the sample using a combination of deterministic and probabilistic rules. Even though the evolutionary algorithms can guarantee convergence only in a probabilistic sense, they are quite efficient in practice and have the major advantage that they do not usually impose restrictive conditions on the nature of the function. Of particular importance, evolutionary algorithms often possess the capability for the search process to climb out of a local minimum or to prevent the search from becoming prematurely localized. These characteristics give evolutionary computations a higher potential of locating the region of the global optimum solution.

EA can be applied to virtually any problem that can be formulated as a functional optimization task. EA produces a data structure to represent solutions, a performance index to evaluate solutions, and variation operation to generate new solutions from old solutions. The state space of possible solutions can be disjoint and can encompass infeasible regions. The formulation procedure is independent to functional characteristics, in contrast with other numerical techniques which might be applicable for only continuous values or other constrained sets. The representation of EA allow for variation operators that maintain a behavioural link between parent and offspring. A continuum of possible changes should be allowed such that the effective step size of the algorithm can be turned, perhaps online in a self-adaptive manner. This flexibility allows for applying essentially the same procedure to discrete combinatorial problems, the continuous-valued parameter optimization problems, the mixed-integer problems, and so forth.

Another useful feature is capability of self-optimization. Most of classic optimization techniques require appropriate setting of exogenous variables. This is true for evolutionary algorithms as well. However, there is a long history of using the evolutionary process itself to optimize these parameters as part of the search for optimal solution (Back et al., 1997).

EA, despite its excellent properties, is notorious for its computational burden due to its population-based characteristic. Especially, it has disadvantages of the well-noted slow final convergence and the ever present danger of still converging to a local minimum. However, EA offers a framework such that it is relatively easy to combine with local search optimization to improve its performance by exploiting their advantages. This may be as simple as the use of a gradi-

ent-based minimization used after primary search with an EA (Kapelan et al., 2000), or it may involve simultaneous application of algorithms (Kapelan et al., 2002). Also, EA can be used to optimize the performance of neural network, fuzzy systems, production systems, and other program structures (Back et al., 1997). In many cases, the limitations of conventional approaches can be avoided.

The other interesting characteristic to relieve the computational burden of EA is that it is intrinsically a highly parallel process. As the distributed processing computers are becoming more readily available, there is a corresponding increased potential for applying the evolutionary algorithms to more complex problems (Back et al., 1997). It is often the case that individual solutions can be evaluated independently of the evaluations that are assigned to the competing solutions. The evaluation of each solution can be handled in parallel and only selection requires some serial processing. Recently, Balla and Lingireddy (2000) reported results from an exploratory research that implemented a complicated optimization model based on a distributed genetic algorithm on a network of PCs.

4. Evolution Algorithms

There are many algorithms that are based or inspired by biological systems. Five fairly well-known ones and a new hybrid one are described and used here.

4.1. Genetic Algorithm

The best known EC paradigm is the Genetic Algorithm (GA), created by Holland and made popular at least for engineers by Goldberg (1989). The GA approach is used widely, especially in engineering and industrial applications. In the GA approach, the population is, mostly, binary encoded and genetic operators, inspired to mimic DNA evolutionary procedures, are applied to the population in order to stimulate the system to evolve, and thus to explore the search space efficiently. Numerical approaches using GA have been explored (Goldberg, 1989; Simpson et al., 1994; Balla and Lingireddy, 2000, Jung and Karney, 2006). Since this procedure is now well known, it is only briefly reviewed here.

4.2. Evolutionary Programming

Evolutionary Programming (EP) developed by Fogel (1963) is the one of traditional EA paradigms. EP is similar to GA in its use of a population of candidate solutions to evolve an answer to specific problems; however EP places emphasis on developing behavioural models, that is, models of the way observable system interact with the environment. EP is derived from the simulation of adaptive behaviour in evolution; GA is derived from the simulation of genetics. The difference is perhaps subtle, but important. GA works in the genotype space of the information with, in most cases, a binary coding, while the EP emphasizes the phenotype space of observable behaviours using real numbers. Therefore, EP is directed at evolving behaviour that solves the problem at hand with what

is called "phenotypic evolution". Another significant characteristic of EP is self-adaptation, which provides the capability of strategy parameters to evolve themselves, thus directing mutation into more promising parts of the search space.

EP is a more flexible approach to evolution than some of the other paradigms. The operators are freely adapted to fit the problem at hand. Generally the paradigm relies on mutation, not sexual recombination, to produce offspring. EP usually generates the same number of children as parents. The parents are selected to reproduce using a tournament method; their features are mutated to produce children who are added to the population. When the population doubled, the members of parents and offspring are ranked, and the best ranking half is kept for the next generation so the original population size is maintained. Recently, Soh and Dong (2001) incorporated EP algorithm with the finite-element method for solving various inverse problems in civil and structural engineering. They also introduced several improved techniques that were added to the conventional EP algorithm to increase the efficiency and yet to retain the versatility of the conventional EP algorithm.

4.3. Particle Swarm Optimization

Recently, a new EA approach has arisen, based on another population search procedure, called Swarm Intelligence (SI). SI argues that intelligent cognition derives from the interaction of individuals in a social environment and that the main ideas of sociocognition can be effectively applied to develop stable and efficient algorithms for selected optimization tasks (Kennedy and Eberhart, 2001). One of these SI techniques, called Particle Swarm Optimization (PSO), has been developed to simulate the movement of a flock of birds searching for food. PSO has been used mainly for Continuous Optimization tasks and was originally developed by Kennedy and Eberhart (1995). In this technique, the population of potential solutions is called a "swarm" and the approach is to explore the search space simulating the movement of a flock of birds searching for food. However, in the computer version of this search, the global exchange of information among all individuals, which are called "particles", takes place and each particle can profit from the discoveries of the rest of the swarm. PSO has proven to be an efficient algorithm for solving hard optimization problems and engineering applications, including neural network training and human tremor analysis (Kennedy and Eberhart, 2001).

There are many variants of the PSO technique developed thus far. In this paper, a version of the algorithm derived by adding an inertial weight to the original PSO dynamics has been used (Shi and Eberhart, 1989). Assuming that the search space is D -dimensional, we denote the current position by $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ of the i th particle of the swarm and by $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ the best position it ever had within the search space. Let g be the index of the best particle in the swarm and $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ the velocity (position change) of the i th particle. The swarm is manipulated according to the equations.

$$v_{id} = wv_{id} + c_1r_1(p_{id} - x_{id}) + c_2r_2(p_{id} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

where $d = 1, 2, \dots, D$; $i = 1, 2, \dots, N$ and N is the size of the population; w is the inertial weight; c_1 and c_2 are two positive constants; r_1 and r_2 are two random values in the range $[0, 1]$. Equation (1) is used to calculate the i th particle's new velocity, a determination that takes into consideration three main terms: the particle's previous velocity, the distance of the particle's current position from its own best position, and the distance of the particle's current position from the swarm's best experience (position of the best particle). Thus, each particle or potential solution moves to a new position according to (2). The performance of each particle is measured using a predefined fitness function. The inertial weight w plays an important role for the convergence behaviour of the technique. It is used to control the impact of the previous history of velocities to the current velocity of each particle, regulating this way the trade-off between the global and local exploration abilities of the swarm, since large values of w facilitate global exploration of the search space (visiting new regions) while small values facilitate local exploration, i.e., fine-tuning the current search area. The initialization of the swarm is done using a uniform distribution over the search space. By gradually decreasing the inertial weight from a relatively large value to a small value through the course of the PSO run, the PSO tends to exhibit a more global search ability at the beginning of the run while having more local search ability near the end of the run. Recently, Jung and Karney (2004b, 2006) apply PSO to calibrate system parameter in water distribution system and to obtain optimal pipe diameters in the pipeline system with allowance for water hammer conditions, respectively. This PSO applications are compared with GA and present fast convergences and slightly better optimization results than GA.

4.4. Shuffled Complex Evolution Algorithm

The shuffled complex evolution (SCE) algorithm introduced by Duan et al. (1992) is a robust, effective and efficient strategy for function minimisation. This algorithm combines four concepts that have each proved successful for global optimisation: a combination of probabilistic and deterministic approaches, the concept of clustering, systematic evolution of a "complex" of points across the parameter space, and the utilization of competitive evolution. Gan and Biftu (1996) noted that these four features represented the best features of several optimisation methods. This algorithm was constructed around the controlled random search (CRS) method described by Price (1983), using its best features such as global sampling and complex evolution, and incorporated the powerful concepts of competitive evolution and complex shuffling. These latter features allow the sample information to be thoroughly exploited in order to find global solutions.

The SCE algorithm is initiated by sampling a random set of parameter values from the feasible parameter space. This set of points, termed a "population", is then partitioned into a

number of smaller groups (or "complexes"), each of which is allowed to evolve independently according to a competitive complex evolution strategy, adapting the deterministic simplex method of Nelder and Mead (1965). This method of evolution allows each complex to search the parameter space in different directions. The complexes are periodically shuffled to enable information sharing, and at random locations new parameter values are introduced to the complexes to ensure the process of evolution does not get trapped by unpromising regions (Duan et al., 1992). This method is repeated until sufficient convergence is achieved, with the population moving towards the globally optimal values. By combining competitive evolution and complex shuffling, the SCE algorithm ensures that information about the parameter space obtained by each complex of samples is shared across the entire population, which allows an efficient search of the feasible parameter space.

The SCE algorithm has a number of parameters that require specification, including the number of complexes; the number of points in a complex; the number of points in a sub-complex; the number of consecutive offspring generated by each subcomplex; and the number of evolution steps taken by each complex. Duan et al. (1993, 1994) provide default values for all of these parameters, except for the number of complexes, which is highly dependent upon the complexity of the problem. In this study the number of complexes was set equal to the dimension of the problem. (i.e., 2, 6 and 20).

4.5. Ant Colony Optimization

Ant Colony Optimization (ACO) algorithms were first introduced by Dorigo et al. (1996) for solving combinatorial optimization problems. ACO mimics behaviors of a population of ants when determining the shortest path from its nest to a food source. Each ant deposits a chemical substance, pheromone, which provides an indirect form of communication with other colony members. Ants select a path proportional to the level of pheromone, thus over time creating a 'feedback mechanism' with shorter paths being reinforced with higher levels of pheromone. An ACO algorithm iteratively constructs solutions whereby each ant utilizes the probabilistic policy given in Equation (3) to select an option (path) for every decision variable. That is, the probability p of selecting option i for path j at some iteration t depends upon the pheromone intensity $\tau_{i,j}(t)$, that is representative of the learned information; and the desirability $\eta_{i,j}$, that acts as a bias against less desirable options. Two parameters α and β control the relative importance of pheromone intensity and desirability:

$$p_{i,j}(t) = \frac{\tau_{i,j}(t)^\alpha \eta_{i,j}^\beta}{\sum_i \tau_{i,j}(t)^\alpha \eta_{i,j}^\beta} \quad (3)$$

After the generation of solutions by the colony, a decay parameter, ρ , is used to reduce the existing pheromone levels on all paths and new pheromone is added to given paths

depending upon a specified updating policy. The decay parameter represents the degree to which information from previous iterations is retained, and can be used to control sub-optimal convergence. Numerous ACO algorithms exist, with most of the variation between them due to different pheromone updating policies. The standard ACO version is referred to as Ant System (AS), where each ant in the colony updates the pheromone for the paths it traversed using the policy given in Equation (4):

$$\tau_{i,j}(t) = \rho \tau_{i,j}(t-1) + \frac{Q}{f_k(\cdot)} \quad (4)$$

where $f_k(\cdot)$ is the overall solution quality determined by ant k (where $f_k(\cdot)$ is positive and has lower values representing better solutions) and Q is an arbitrary proportionality constant to ensure that pheromone additions are of a suitable magnitude. Other versions such as AS_{rank}, and AS_{elite} restrict the updating policy to a select number of ants and another popular algorithm, the Max-Min Ant System, uses dynamically evolving bounds for the pheromone levels on each path to encourage wider exploration.

ACO has been successfully applied to the Travelling Salesman Problem (Dorigo and Gambardella, 1997) and for the optimal design of water distribution systems, Maier et al. (2003) demonstrated the superior performance of an ACO when compared to a GA. This paper applies only the standard ACO, known as AS. The objective functions in this paper do not allow for the feature of path desirability to be implemented, thus $\eta_{i,j}$ was set equal to unity. The parameter alpha, was given a standard setting $\alpha = 1$, and the pheromone decay parameter was set to $\rho = 0.9$ after preliminary testing. The initial pheromone levels on all paths were arbitrarily set to 50.0, and a proportionality constant of 1000.0 was used in the updating procedure.

4.6. Hybrid Evolutionary Algorithm

As indicated in Section 3, it has several disadvantages including the well-noted slow final convergence and the ever present danger of still converging to a local minimum despite its population basis. However, one of the good characteristics of EA is that they are easy to hybridize with other kinds of optimization to improve its performance by exploiting their advantages. Such optimization methods range from exact algorithms which are studied in mathematical programming, such as integer programming, dynamic programming, branch and bound, polyhedral approaches, and linear and nonlinear programming, to heuristic (or approximate) algorithms that are tailored to the given problem domains, such as greedy methods, local search (or hill climbing) and other heuristic constructions of solutions (Back et al., 1997).

However, the most simple, but most efficient hybridization might be to combine EA with local search method. Quick convergence of a local search method would help to increase overall convergence speed of the EA. Therefore, hybridization

of evolutionary computation with a local search method is an attractive methodology to compensate for their disadvantages. The characteristics of two optimizations make it obvious that EA should be used first to find the region of global optimum solution. The local search method is then employed to find the optimum point to the required level of accuracy.

Kapelan et al. (2000) suggested a hybrid form of GA and Levenberg-Marquardt (LM) method for the calibration of water distribution system hydraulic models. A GA search is performed in the first stage until some termination criterion is met. Subsequently, the LM search is performed using the best solution found by the first-stage GA search as the starting point. However, the main problem associated with the two-staged method is how to decide when to stop the GA and start using the LM method. Therefore, they introduced another hybrid method to include local search operator with pre-specified probability rate in the genetic operators of GA (Kapelan et al., 2002). The hybridization with LM method in the both approaches efficiently improves the performance of the GAs but it requires the expense of gradient information of the system. Recently, Zyl et al. (2004) developed a hybrid method which combines the GA method with a hillclimber search strategy. Unlike the previous approaches, local search method using Hillclimber strategies complement GAs by being efficient in finding a local optimum, but require no gradient information. This characteristic reserves the robustness of the EA, as well as improves the convergence speed of EA.

In this paper, a new hybrid form to combine PSO with local search method is introduced. As a local search method, Powell's method is applied because it is quadratically convergent without requiring derivative evaluation (Press et al., 1992). PSO is used first to get the solution near the global optimum solution and then Powell's method is employed to find the best solution starting with the result of the PSO search. However, the switching from PSO to Powell's method is a tough problem and it should be carefully considered depending on the system structures. To decide when to terminate both methods, the relative error is defined as:

$$\varepsilon = \frac{|f_{\text{new}} - f_{\text{old}}|}{(|f_{\text{new}}| + |f_{\text{old}}|)/2} \quad (5)$$

where f_{new} and f_{old} are the function values for the present iteration and the previous one, respectively. When ε becomes less than a pre-specified stopping criterion, the computation is terminated. The fractional tolerances, the stopping criteria, are pre-defined for the both PSO ($F_{\text{tol, PSO}}$) and Powell's method ($F_{\text{tol, Powell}}$). It might be reasonable to choose the bigger number of $F_{\text{tol, PSO}}$ than that of $F_{\text{tol, Powell}}$ because the PSO's purpose is to find the approximate region where a global optimum is located while Powell's method could save computation time in finding the global optimum. Also, due to the stochastic character of PSO, it may be useful to wait more generations (N_{PSO}), thus allowing the search to reach a better solution region after the relative error ε becomes smaller than the fractional tolerance. This might prevent premature convergence to

a local optimum in a multimodal function and provide more chance to reach the global solution region by stretching out the PSO particles to other local optima. The proper selections of the two fractional tolerances ($F_{\text{tol, PSO}}$ and $F_{\text{tol, Powell}}$) and the consecutive generation number (N_{PSO}) are important for increasing the performance of the hybrid evolutionary algorithm and are strongly dependent upon the system characteristics.

5. Benchmark Test

In order to compare the computational performances of specific EAs, four benchmark functions are presented as follows:

$$\text{Sphere function: } f_1(x) = \sum_{i=1}^n x_i^2 \quad (6)$$

Rosenbrock function:

$$f_2(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right] \quad (7)$$

Generalized Rastrigin function:

$$f_3(x) = \sum_{i=1}^n \left[x_i^2 - 10 \cos(2\pi x_i) + 10 \right] \quad (8)$$

Generalized Griewank function:

$$f_4(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (9)$$

Function f_1 and f_2 are unimodal functions that have a single local maximum or minimum only. The sphere function f_1 , shown in Figure 2(a), is well-known simple function with the global minimum at $\forall x_i = 0$ but Rosenbrock function f_2 , shown in Figure 2(b), is widely known as the banana function, which has been notorious for its slow convergence because of the way the curvature bends around the origin. The global minima is at $\forall x_i = 1$, as is clearly visible from the formula, but the valley passing through the origin descends very slowly in comparison with the steepness of the valley walls, which create a difficulty in finding the minima.

In contrast, the generalized Rastrigin function f_3 and generalized Griewank function f_4 are typical examples of nonlinear multimodal functions that have many local optima as well as a global one. Both functions have the same global minimum at $\forall x_i = 0$ and their number of local minima increases exponentially with the problem dimension. These functions are fairly difficult problems due to its complicated search space and its large number of local minima.

For unimodal functions, the convergence rates of EAs are more interesting than the final results of optimization; for multimodal functions, the final results are much more important since they reflect an algorithm's ability to escape from poor local optima and thus to locate a good near-global opti-

um. The number of test functions might be insufficient to assess all characteristics and performance of each EA, but they provide the general trend and some guideline for each algorithm. Certainly the shape of these functions can offer some significant challenges to an optimization procedure.

5.1. Test of EAs

Five EAs - GA, EP, PSO, SCE and ACO are first explored with the four benchmark functions. Also, in order to compare the algorithm performances according to the different dimension sizes, three sizes (2, 6 and 20) are considered for all benchmark functions. Due to the different characters of the EAs, the parameters chosen for each algorithm differ, but the common parameters like population size, generation number and the search space of each dimension are fixed and set as 200, 200 (total 40000 evaluations) and -100 to 100, respectively.

More specifically for the GA, the probability of mutation is 0.02; the probability of (uniform) crossover is 0.5; tournament selection and elitism (in which the best individual is copied to the next generation) are selected. The chromosome length of each dimension is set as 15 so that the discrete searching space of the GA is about $0.006 (200/2^{15})$. In the EP, the algorithm has no recombination and relies on mutation only as the search operator. It also uses the self-adaptation principle to evolve the probability of mutation online during the search. After calculating the standard deviation proportional to the fitness value, mutation is applied to all individual with Gaussian noise of zero mean. As a selection method, a stochastic tournament selection is used. After mutating the parent generation, the algorithm compares each individual with other individuals and records the number of wins. Both original parent and mutated version are sorted in descending order of wins scored. Based on the empirical study of PSO (Shi and Eberhard, 1999), a linearly decreasing inertial weight is used which starts at 0.9 and ends at 0.4 and $c_1 = 2$ and $c_2 = 2$ in Equation (1). The maximum velocity of the PSO is set as 10 for all dimensions. The parameters used for SCE and ACO are described in the sections 4.5 and 4.6 before.

Figure 3 presents the evolution procedure to minimize the 6 dimensional Sphere function with GA, EP, PSO, SCE and ACO. After 10000 evaluations, the PSO, SCE, GA and EP solutions have converged to similar minimum values but the PSO and SCE approaches shows a more rapid initial convergence rate than GA and EP. Compared to other four algorithms, ACO presents the slowest convergence as well as the worst minimum value in the final result. Figure 4 shows the evolution procedure to minimize the 20 dimensional generalized Griewank function with the five algorithms.

Remarkably, SCE and PSO not only show the fastest convergence (like Figure 3), but also present the best minimum function value in the final result. This may indicate that the both optimizations have a strong ability to escape from poor local optima and to find a global or near-global minimum. Table 1 show the results of all 12 benchmark tests with the five optimization methods and four different functions.

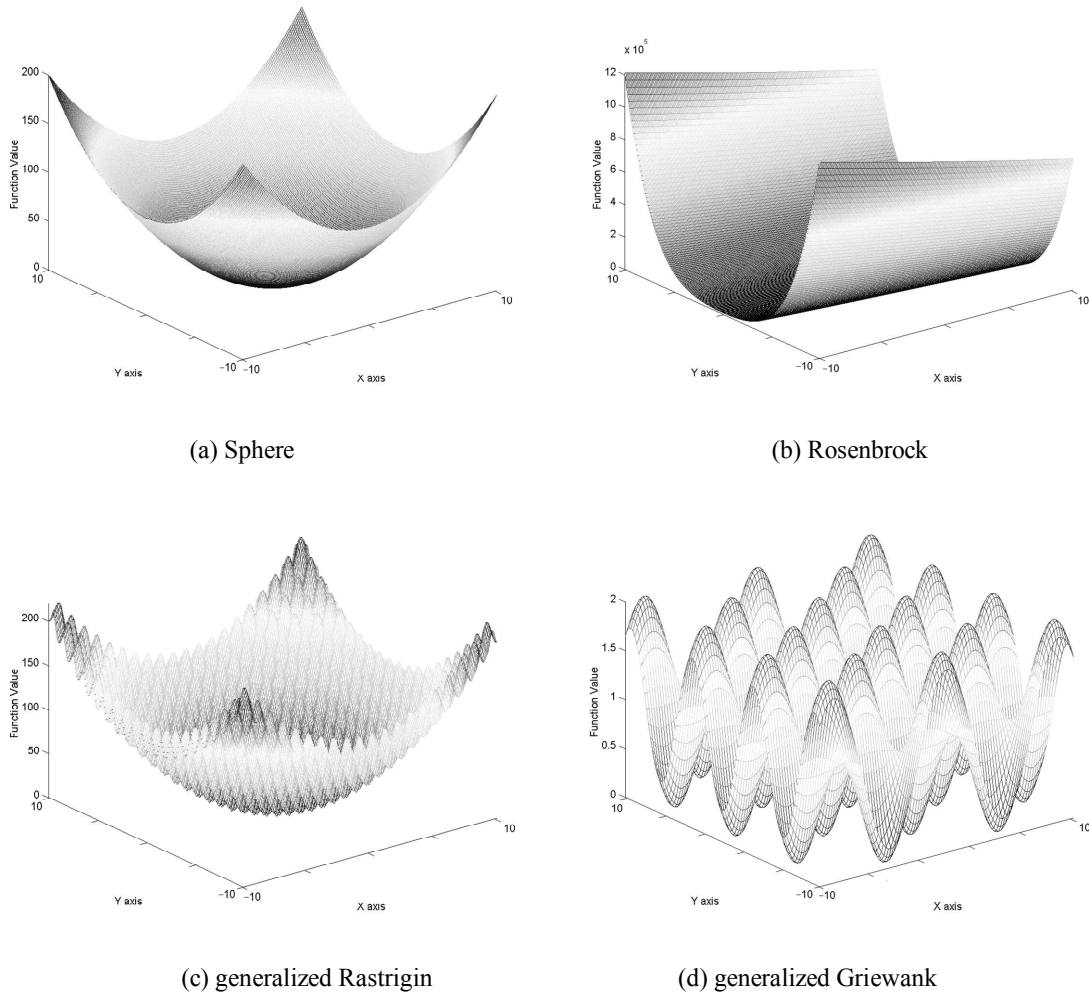


Figure 2. Topographies of the benchmark function with 2 dimensions.

Not surprisingly, the final results of the all optimization deteriorate as the dimension size increases but one noticeable feature is that that the all minimum results of SCE are better than those of the other algorithms except for the result of PSO with two dimensional generalized Griewank function. This excellent convergence may be partly due to the “hybrid-like” characteristic of SCE using the simplex algorithm, as a local search method, for generating offspring. However, even the SCE approach had difficulties finding a minimum in high dimensional problem of the Rosenbrock function and converged to a poor minimum (17.6); of course, the other procedures did even worse, but this shows that no single technique is perfect.

5.2. Test of Hybrid PSO

As shown in Figures 3 and 4, EA optimizations have a general trend of a rapid initial convergence followed by a

slower final one. Interestingly, the hybrid characteristic of SCE provides a faster convergence as well as the smaller minimum function value than the other classical EAs. Therefore, this benchmark test introduces a hybridization of EA with Powell’s method as a local search method. Due to its strongest performance of the four classical EA algorithms, PSO is selected and hybridized with Powell’s methods, called as Hybrid Particle Swarm Optimization (HPSO). For the convergence criteria, two fractional tolerances ($F_{tol, PSO}$ and $F_{tol, Powell}$) and the consecutive generation number after the tolerance (NPSO) are selected as 10^{-2} , 10^{-6} and 10, respectively.

Table 2 presents the optimization result of HPSO, especially indicating the number of generation to converge to optimum and its minimum function value. In most tests, HPSO converges within 100 generations, which clearly indicate the hybridization of EA improves the computation time of the original EA. In addition, the HPSO find better improved mini-

imum results than the PSO and SCE in Table 1, especially for high dimensional unimodal functions. The final results of HPSO in the multimodal functions are better than the original PSO but it still show the localized optimum in the multimodal functions. It might be reason that the Powell's method finds the local minimum in the valley that PSO assume a global minimum would be located, but this was the wrong location. Therefore, the results of multimodal functions present that the success of this hybrid optimization method crucially depends on the first-stage PSO search.

Table 1. Optimal Function Value of EAs

	Dim.	Benchmark function			
		Sphere	Rosenbrock	Rastrigin	Griewank
PSO	2	0	0	0	0
	6	0	0.31	5.76	0.03
	20	8.58	1230	147	0.29
GA	2	0	0.01	0	0.01
	6	0.35	165	16.1	0.11
	20	1600	1.47E+08	2150	1.40
EP	2	0.00	0.00	0.00	0.01
	6	0.00	11100	40.4	0.56
	20	850	8.71E+06	968	1.33
ACO	2	0	0.450	0.44	0.02
	6	7.83	807	58.02	0.48
	20	11000	3.01E+09	11400	2.9
SCE	2	0	0	0	0.01
	6	0	0	0.995	0.01
	20	0	17.6	10.07	0

6. Application to the Calibration of Water Distribution System

A water distribution system (WDS) is an engineered system that is used to convey water from the source (well, river, etc.) to the consumer. The main aim of a WDS is to deliver water to the consumer when it is necessary, in the correct quantity and in accordance with the relevant water quality standards. Obviously, a mathematical model that can accurately simulate WDS behaviour is of great importance for every WDS authority.

To allow for meaningful use, any WDS hydraulic model should first be calibrated. Calibration is the process in which a certain number of WDS model parameters are adjusted until the model closely mimics the behaviour of the real WDS. Traditionally, calibration was, and unfortunately often still is, treated as a manual task. However, it has been shown recently that much better results can be achieved if calibration of the analyzed WDS model is formulated. This is another class of optimization problems using an objective function with the difference measured transient pressures and predicted pressures which hydraulic simulation computer model yield with assumed system parameter, called as inverse transient analysis (Liggett and Chen, 1994; Jung and Karney, 2004a).

The example pipe network shown in Figure 5 comprises two reservoirs at nodes 1 and 16, forty-five pipes, as well as twenty-nine nodes. This is a gravity flow system that draws water from the higher reservoir to the downstream network. The elevations of two reservoirs at node 1 and 16 are 50 m and 45 m, respectively, and other nodes are zero. The length, diameter, wave speed and Darcy-Weisbach friction factor of all pipes are assumed known and are given in 500 m, 0.3 m, 1,000 m/s and 0.015, respectively. Ten 12 L/s demands at nodes 2, 3, 5, 14, 15, 25, 26, 27, 28 and 29, six 24 L/s demands at nodes 7, 9, 13, 17, 19 and 23, and five 36 L/s demands at nodes 10, 11, 12, 16 and 18 are considered for the specific case considered here.

In order to introduce transient conditions into the case study for inverse transient calculation, a valve closing at node 6 are chosen to characterize the performance of the system. In order to compare the inverse calculation responses according to the degree of transient severity, three kinds of valve closing time are introduced and initiate rapid transient (inst. shut-off), moderate transient (5.0 s) and mild transient (20 s). In this case study, the leak point is located at node 22; the leak area is 0.0005 m². In order to represent the global characteristic of pipe system, the measurement points for inverse transient calibration are selected at node 13 and 16. In inverse calculation, the possible leakage nodes are select with node 4, 8, 22, and 24 and the range of leak area in both optimization methods is 0 to 0.002 m². Unknown pipe friction factors are pipe 3, 15, 18, and 26 and the range of friction factor is 0.001 to 0.5.

In the test, the population size and generation number for the optimization programs are set as 50 and 200, so the total number of evaluations becomes 10000. Figure 6 shows the evolution procedures of GA, PSO and SCE to minimize the difference between the measure head trace and the predicted one during the inverse calculation and Table 3 indicates their calibration results. This calibration study shows GA and PSO are slightly better convergence speed than SCE but PSO and SCE show the excellent ability to find the fairly accurate leakage and friction factors, especially SCE for mild transient case.

Three optimization programs show poorer calibration results for the case of a mild transient (20 s) than the ones for rapid transients (inst. and 5.0 s). Perhaps, not surprisingly, this indicates that the rapid transient is much more efficient for determining the detailed characteristics of pipe system. There are obviously practical implications if this is a general result, since mild transients are certainly less intrusive to the system. Interestingly, the milder transient has the faster convergence speed in Figure 6. One reason might be that the mild transient does not uniquely characterize the system, so the 'mismatch' with real system doesn't make a significant difference to optimization process, which causes quick but faulty convergence.

7. Conclusions

Evolutionary Algorithms are robust and globally oriented,

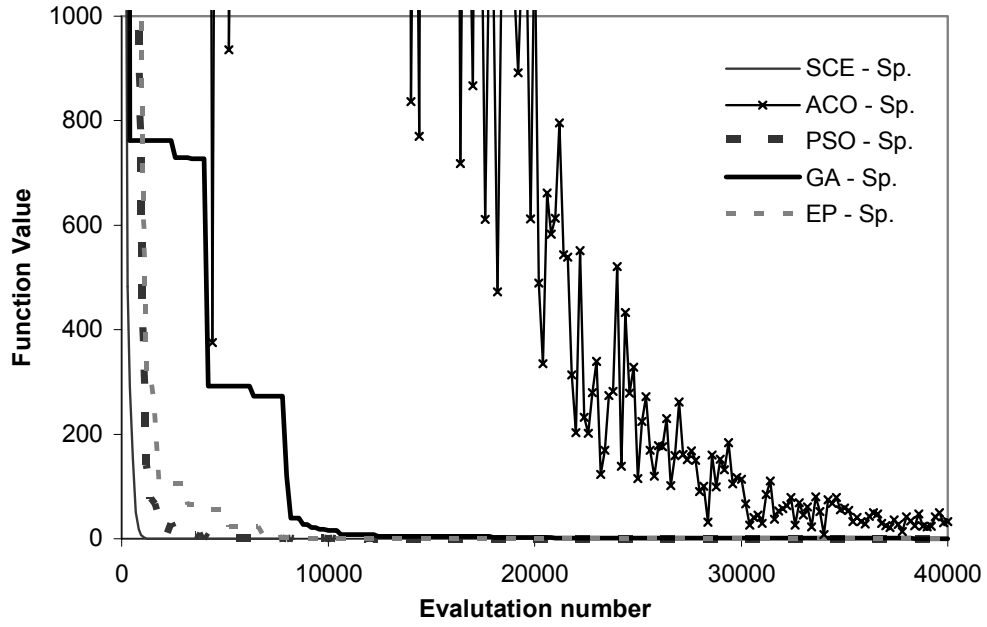


Figure 3. Evolution Procedure with Sphere function (6 dim.).

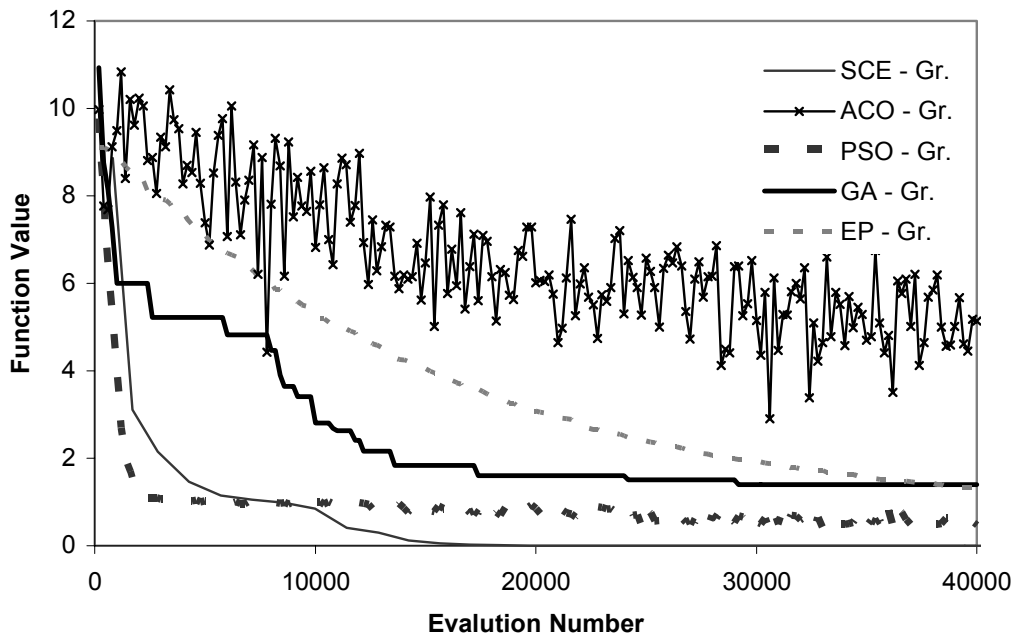


Figure 4. Evolution Procedure with generalized Griewank function (20 dim.).

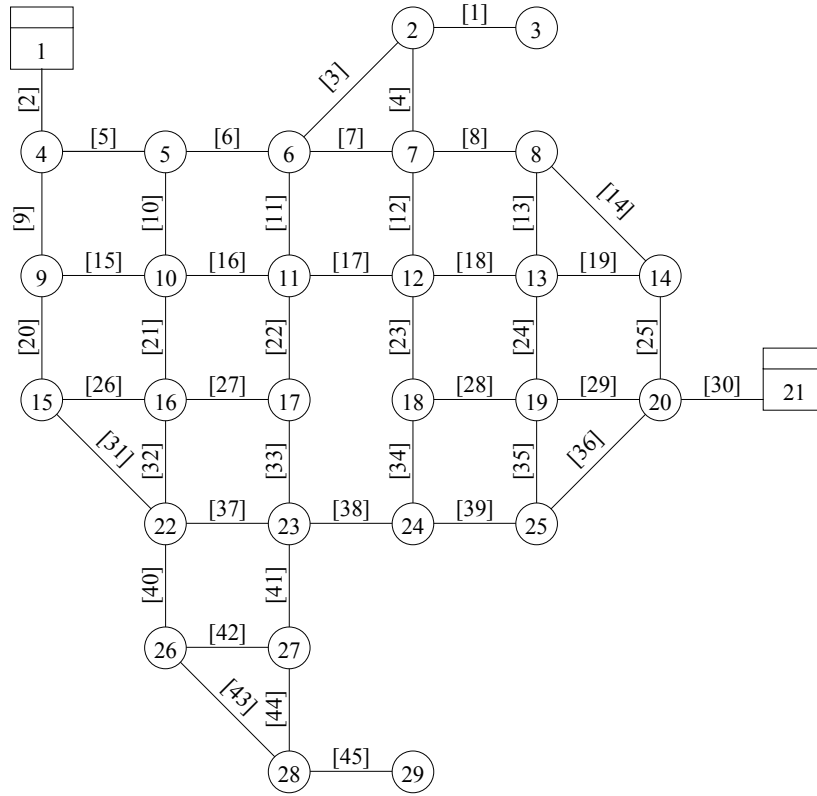


Figure 5. Example pipe network.

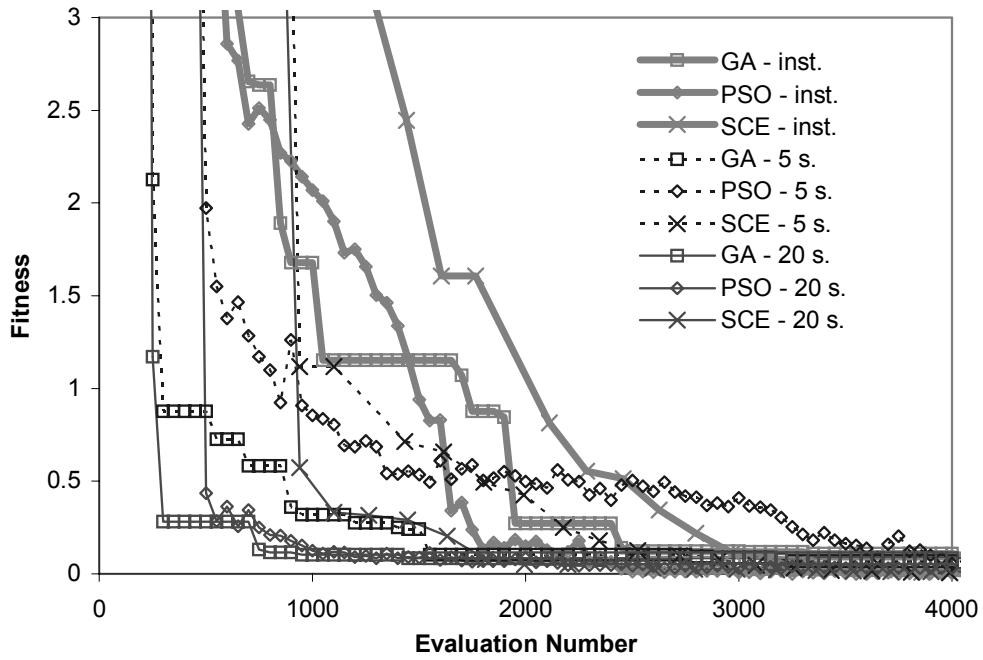


Figure 6. Evolution procedure of GA and PSO calibration.

Table 2. Optimization Result of HPSO

	Dimensions	Benchmark function			
		Sphere	Rosenbrock	Rastrigin	Griewank
Number of generation to converge	2	34	42	38	21
	6	106	41	51	17
	20	46	73	31	44
Optimal function value	2	0	0	0	0.01
	6	0	0	3.98	0.03
	20	0	0	67.66	0.01

Table 3. Leakage and friction factor calibration with GA and PSO

	Real	Valve closure time									
		GA			PSO			SCE			
		Inst.	5.0 s	20 s	Inst.	5.0 s	20 s	Inst.	5.0 s	20 s	
Possible leakage node (10^{-4} m^2)	4	0	0	0	2.5	0	0	0.1	0	0	0.03
	8	0	0	0.2	2.2	0	0	5.1	0	0	0.02
	22	5	4.9	4.7	0.5	5	5	0	5	5	4.94
	24	0	0	0	0	0	0	0	0	0	0.02
Unknown friction factor (10^{-2})	6	1.5	2.07	2.07	3.64	1.5	1.47	1.48	1.5	1.5	1.49
	18	1.5	1.67	2.46	6.39	1.5	1.46	9.78	1.5	1.5	1.51
	33	1.5	1.28	3.24	33.5	1.5	0.1	0.10	1.5	1.5	1.23
	45	1.5	1.67	5.6	18.6	1.5	2.65	23.1	1.5	1.5	1.62
Fitness (10^{-4})	-	-	91.3	222	431	0	0.9	274	0	0	0.05

and are generally more straightforward to apply in situations where there is little or no *a priori* knowledge about the problem to solve. Because EA requires no derivative information and it is stochastic in nature, EA is capable of searching the solution space with a greater likelihood of finding the global optimum.

This paper presents five EA optimizations (GA, EP, PSO, SCE and ACO) and compares the five algorithms with four benchmark functions. The benchmark tests show the SCE and PSO show not only fast convergence speed for the unimodal functions, but also has the strongest global convergence to escape from poor local optima for the multimodal functions. In order to improve the PSO performance, a hybridization of PSO with Powell's method as a local search method is suggested and compared with the four benchmark functions. The hybrid optimization clearly improves the convergence speed and the final optimal result but it still can become trapped in one of the local optima of the multimodal function.

Many optimization problems from the civil and environmental engineering world are complex in nature and difficult to solve by conventional optimization techniques. In particular for large pipeline systems, the optimization of construction and maintenance costs can save many millions of dollar every year. Also, a growing concern has arisen nowadays over water loss in the existing water supply infrastructure in aging systems, so the calibration of friction and leakage is an important problem in drinking water system. Among the many optimization methods, it would be possible to say EA is excellent one

to solve large and complex pipeline systems that have nonlinear structural characteristics and multiple optima. In addition, as the speed of new computers is increasing rapidly, it seems evolutionary computation is more attractive program to solve large pipeline system.

References

- Back, T. (1996). *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New York, USA.
- Back, T., Fogel, D.B., Michalewicz, Z. and Baeck, T. (1997). *Handbook of Evolutionary Computation*, Institute of Physics Publishing, New York, USA.
- Balla, M.C. and Lingireddy, S. (2000). Distributed genetic algorithm model on network of personal computers. *J. Comput. Civil Eng.*, 14(3), 199-205.
- Dorigo, M. and Gambardella, L.M. (1997). Ant colonies for the traveling salesman problem. *BioSyst.*, 43, 73-81.
- Dorigo, M., Maniezzo, V. and Colomi, A. (1996). The ant system: Optimization by a colony of cooperating ants. *IEEE Trans. Syst. Man Cybern.*, 26, 29-42.
- Duan, Q., Sorooshian, S. and Gupta, V. (1992). Effective and efficient global optimization for conceptual rainfall-runoff Models. *Water Resour. Res.*, 28(4), 1015-1031.
- Duan, Q., Sorooshian, S. and Gupta, V.K. (1994). Optimal use of the Sce-Ua global optimization method for calibrating watershed models. *J. Hydrol.*, 158(3-4), 265-284.
- Duan, Q., Gupta, V.K. and Sorooshian, S. (1993). Shuffled complex evolution approach for effective and efficient global minimization. *J. Optimiz. Theory Appl.*, 76(3), 501-521.
- Fogel, L.J. (1963). *Biotechnology: Concepts and Applications*, Pren-

- tice Hall, Englewood Cliffs, NJ, USA.
- Fogel, D.B. (1992). *Evolving Artificial Intelligence*, Ph.D. Dissertation, University of California, San Diego, CA, USA.
- Gan, T.Y. and Biftu, G.F. (1996). Automatic calibration of conceptual rainfall-runoff models: Optimization algorithms, catchment conditions and model structure. *Water Resour. Res.*, 32(12), 3513-3524.
- Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing, MA, USA.
- Holland, J.H. (1973). Genetic algorithms and the optimal allocations of trials. *SIAM J. Comput.*, 2(2), 88-105.
- Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.
- Jung, B.S. and Karney, B.W. (2004a). Particle Swarm Optimization Compared to Genetic Algorithm for Calibration of Water Distribution System, in *Proc. of Ninth International Conference Pressure Surges*, Chester, UK.
- Jung, B.S. and Karney, B.W. (2004b). Fluid transients and pipeline optimization using GA and PSO: The diameter connection. *Urban Water J.*, 1(2), 167-176.
- Jung, B.S. and Karney, B.W. (2006). Optimization of transient protection devices using GA and PSO approaches. *J. Water Resour. Plan. Manage.*, ASCE, 132(1), 44-52.
- Kapelan Z.S., Savic, D.A. and Walters, G.A. (2002). Hybrid GA for calibration of water distribution hydraulic models, in *Proc. of ASCE Environmental & Water Resource Institute Annual Conference*, ASCE, Roanoke, Virginia.
- Kapelan Z.S., Savic, D.A. and Walters, G.A. (2000). Inverse Transient Analysis in Pipe Networks for Leakage Detection and Roughness Calibration, in *Proc. of Water Network Modelling for Optimal Design and Management*, Exeter, UK, pp. 143-159.
- Kennedy, J. and Eberhart, R.C. (1995). Particle swarm optimization, in *Proc. of the IEEE International Conference on Neural Networks IV*, Piscataway, New Jersey, pp. 1942-1948.
- Kennedy, J. and Eberhart, R.C. (2001). *Swarm Intelligence*, Morgan Kaufmann Publishers, San Francisco, CA, USA.
- Koza, J. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, MA, USA.
- Liggett, J.A. and Chen, L.C. (1994). Inverse Transient Analysis in Pipe Networks. *J. Hydraul. Eng.*, ASCE, 120(8), 934-955.
- Maier, H.R., Simpson, A.R., Zecchin, A.C., Foong, W.K., Phang, K.Y., Seah, H.Y. and Tan, C.L. (2003). Ant colony optimization for design of water distribution systems, *J. Water Resour. Plan. Manage.*, ASCE, 129(3), 200-209.
- Nelder, J.A. and Mead, R. (1965). A Simplex method for function minimization. *Comput. J.*, 7, 308-313.
- Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (1992). *Numerical Recipes in Fortran*, Cambridge university press, Cambridge.
- Price, W.L. (1983). Global optimization by controlled random search. *J. Optimiz. Theory Appl.*, 40, 333-348.
- Shi, Y.H. and Eberhart, R.C. (1998). A modified particle swarm optimizer, *IEEE International Conference on Evolutionary Computation*, Piscataway, New York, pp. 69-73.
- Shi, Y.H. and Eberhart, R.C. (1999). Empirical study of particle swarm optimization, in *Proc. of the 1999 Congress on Evolutionary Computation*, Piscataway, New Jersey, pp. 1945-1950.
- Simpson, A.R., Dandy, G.C. and Murphy, L.J. (1994). Genetic Algorithms compared to other techniques for pipe optimization. *J. Water Resour. Plan. Manage.*, 120(4), 423-443.
- Soh, C.K. and Dong, Y.X. (2001). Evolutionary programming for inverse problems in civil engineering. *J. Comput. Civil Eng.*, 15(2), 144-150.
- Zyl, J.E., Savic, D.A. and Walters, G.A. (2004). Operational optimization of water distribution systems using a hybrid Genetic Algorithm. *J. Water Resour. Plan. Manage.*, 130(2), 160-170.