Canadian Journal of Civil Engineering

TESTING EVOLUTIONARY ALGORITHMS FOR OPTIMIZATION OF WATER DISTRIBUTION NETWORKS

Journal:	<i>Canadian Journal of Civil Engineering</i>
Manuscript ID	cjce-2018-0137.R1
Manuscript Type:	Article
Date Submitted by the Author:	28-Sep-2018
Complete List of Authors:	Moosavian, Naser; University of British Columbia, Applied Science Lence, Barbara; Dept. of Civil Engineering
Keyword:	Evolutionary algorithms, Water distribution networks, genetic algorithm, optimization
Is the invited manuscript for consideration in a Special Issue? :	Not applicable (regular submission)

SCHOLARONE™
Manuscripts

TESTING EVOLUTIONARY ALGORITHMS FOR OPTIMIZATION OF WATER DISTRIBUTION NETWORKS

Naser Moosavian

Department of Civil Engineering, University of British Columbia, Canada.

naser.moosavian@alumni.ubc.ca (corresponding author email)

Barbara Lence

Department of Civil Engineering, University of British Columbia, Canada

Abstract: Water distribution networks (WDNs) are one of the most important elements of urban infrastructure and require large investment for construction. Design of WDNs is classified as a large combinatorial discrete nonlinear optimization problem. The main concerns associated with the optimization of such networks are the nonlinearity of the discharge-head loss relationships for pipes and the discrete nature of pipe sizes. Due to these issues, this problem is widely considered to be a benchmark problem for testing and evaluating the performance of nonlinear and heuristic optimization algorithms. This paper compares different techniques, all based on evolutionary algorithms (EAs), which yield optimal solutions for least-cost design of WDNs. All of these algorithms search for the global optimum starting from populations of solutions, rather than from a single solution, as in Newton-based search methods. They use different operators to improve the performance of many solutions over repeated iterations. Ten EAs, four of them for the first time, are applied to the design of three networks their performance in terms of the least cost, under different stopping criteria, are evaluated. Statistical information for 20 executions of the ten algorithms are summarized, and Friedman tests are conducted. Results show that, for the Two-Loop benchmark network, the particle swarm optimization gravitational search (PSOGSA) and biology and bioinformatics global optimization (MEIGO) algorithms efficiently converge to the global optimum, but perform poorly for large networks. In contrast, given a sufficient number of function evaluations, the covariance matrix adaptation evolution strategy (CMAES)

and soccer league competition (SLC) algorithm consistently converge to the global optimum, for large networks.

Key words: Evolutionary algorithms, genetic algorithm, water distribution networks, optimization.

1. Introduction

Evolutionary algorithms (EAs) are becoming increasingly popular for their use in solving engineering decision problems and in practical applications such as calibration, because they: (i) are based on rather simple concepts and are easy to implement; (ii) do not require gradient information; (iii) can consider and bypass local optima; (iv) do not require continuous objective functions; and (v) can be utilized in a wide range of problems covering different disciplines. Generally, the main structure and searching process of different EAs are similar, however, their operators may vary. Being a nonlinear, nonconvex, nonpolynomial hard problems, the least-cost water distribution network (WDN) design problem, in its simplest form, is an attractive benchmark for evaluating innovations in the development of these algorithms. Over the years, the performance of various heuristic optimization techniques have been evaluated with this problem, and its variations, see, e.g., investigations of the efficiencies of genetic algorithms (Murphy and Simpson 1992); simulated annealing (Cunha and Sousa 2001); harmony search (Geem 2006); shuffled frog leaping (Eusuff and Lansey 2003); ant colony optimization (Maier et al. 2003); particle swarm optimization (Suribabu and Neelakantan 2006); cross entropy (Perelman and Ostfeld 2007); scatter search (Lin et al. 2007); differential evolution (Vasan and Simonovic 2010); self-adaptive differential evolution (Zheng et al. 2013); soccer league competition (Moosavian and Roodsari 2014); and improved genetic algorithms (Bi et al. 2015). Variations of some of these algorithms, e.g., those that include nondominated sorting, such as the

nondominated sorting genetic algorithm (NSGA), and nondominated sorting differential evolution (NSDE), have been tested in their application to multi-objective WDN design problems such as those that optimize cost as well as the network's reliability in the face of hydraulic and mechanical failure (see, e.g. Moosavian and Lence 2016). While these problems address the practical need for networks that are robust to uncertainty, by far, the most widely recognized benchmark is the single objective least-cost WDN design problem.

Though these heuristic algorithms show success for specific WDN design problems, there is no guarantee that the global optimum, or the complete Pareto optimal frontier, may be found. Furthermore, in the area of WDN design, only a few comparisons of EA algorithms have been undertaken (Zheng et al. 2013; Moosavian and Roodsari 2014; Bi et al. 2015), and these studies focus on a select few methods. This paper presents a rigorous comparison of ten EAs in application to the single objective least-cost design of WDNs, and evaluates the algorithms in terms of Freidman test results, the best solution obtained, and the distribution of the solutions.

The cost of a WDN design is a function of pipe size diameters and these pipe diameters in turn affect the network pressures. The main constraint of the problem is to satisfy the minimum required pressure at all network nodes. In applications of EAs, at each evaluation of the objective function, hydraulic analysis of the network must be conducted, which exacts a high computational burden, and is in fact the most computationally intensive portion of the optimization process. Therefore, the optimal solution, for a fixed number of function evaluations, is considered a suitable comparator for EAs. In this paper, ten meta-heuristic algorithms are evaluated in terms of the least cost of the WDN design, for three example networks, and under different stopping criteria based on the allowable number of function evaluations. To guard against selection of local optima, for each stopping criterion and each pipe network, these

algorithms are implemented 20 times. The algorithms include current versions of: genetic (GA); ant colony optimization (ACO); harmony search (HS); differential evolution (DE); particle swarm optimization gravitational search (PSOGSA); artificial bee colony (ABC); soccer league competition (SLC); covariance matrix adaptation evolution strategy (CMAES); biology and bioinformatics global optimization (MEIGO); and whale optimization (WOA) meta-heuristic algorithms. Several of these algorithms, i.e., GA, ACO, HS, DE, ABC, and SLC have either been commonly applied or are beginning to be applied in the analysis of WDNs. Others have shown high performance when applied to other complex, nonpolynomial hard optimization problems. In the following sections, the general WDN optimization model is introduced, the algorithms are briefly described, their application to three example networks are assessed, and conclusions are drawn from this comparison.

2. Problem formulation

A water distribution system is a collection of many components such as pipes, reservoirs, pumps and valves which are connected in order to provide water to consumers. The optimal design of such a system can be defined as the best combination of component sizes and component settings (e.g., pipe size diameters, pump types, pump locations and maximum power, and reservoir storage volumes) that exacts the minimum cost for a given network layout, such that hydraulic laws governing continuity of flow and energy are maintained and demands for water quantities and pressures at the consumer nodes are satisfied. In its simplest form, WDN design is formulated as a least-cost optimization problem with the selection of pipe sizes as the decision variables, while pipe layout and node connectivity, nodal demands, and minimum pressure head requirements, are imposed. The optimization problem is stated mathematically as:

$$[1] \quad \text{Min } C = \sum_{k=1}^{np} c_k(D_k) \times L_k$$

95 where $c_k(D_k) \times L_k$ = cost of pipe k per unit length with diameter D_k ; L_k = length of pipe k ; and np
 96 = number of pipes in the network. This objective function is minimized under the following
 97 constraints:

98 Flow continuity at nodes

99 For each node, flow continuity must be satisfied;

$$[2] \quad \sum Q_{in} - \sum Q_{out} = q_j, \quad \forall k \in nn$$

100 where q_j = water demand at node j (meters³/second); nn = number of nodes; and Q_{in} and Q_{out} =
 101 flow into and out of node j (meters³/second), respectively.

102 Energy conservation in loops

103 The total head loss around a closed pipe loop should be equal to zero, or the head loss along a
 104 loop between two fixed head reservoirs should be equal to the difference in water level of the
 105 reservoirs:

$$[3] \quad \sum_{k \in \text{loop } l} hf_k = \Delta H, \quad \forall l \in nl$$

106 where ΔH = difference between nodal pressures at both ends of a path (meters), and $\Delta H = 0$, if
 107 the path is closed; nl = number of loops; and hf_k = head loss due to friction in pipe k (meters)
 108 which is obtained from following equation:

$$[4] \quad hf_k = H_i - H_j = R_k Q_k^n$$

where H_i and H_j = nodal heads at the start, and at the end node of the pipe (meters), respectively;
 R_k = resistance coefficient of the k th pipe; Q_k = flow rate in the k th pipe (second/ meters²); and n
 = constant depending on the head loss equation, and is 1.852 for the most common expression
 for head loss, the Hazen-Williams head loss formulation.

Minimum pressure at nodes

For each junction node in the network, the pressure head should be greater than the prescribed
 minimum pressure head:

$$[5] \quad H_j \geq H_j^{min}, \forall j \in nn$$

where H_j = pressure head at node j (meters); nn = number of nodes; and H_j^{min} = minimum required
 pressure head (meters).

In the comparison of EA performance undertaken in this paper, the Global Gradient Algorithm,
 GGA (Todini and Pilati 1988), in a MATLAB environment, is applied to conduct the hydraulic
 analysis of the networks. GGA satisfies the continuity and energy conservation equations (Eq 2-
 4), while calculating the pressure head, H_j , at each junction node and the discharge, Q_k , in each
 pipe.

Pipe size availability

The diameter of the pipes should be selected from a set of commercially available sizes, and are
 thus discrete:

$$[6] \quad D_k = \{D(1), D(2), \dots, D(ns)\}, \quad \forall k \in np$$

where ns = number of candidate diameters.

3. Evolutionary Algorithms (EAs)

Generally, EAs are implemented in five main steps, including: (i) creation of a random initial population; (ii) evaluation of the objective function or fitness; (iii) selection of two or more solution vectors for the evolution process; (iv) implementation of an evolution strategy with different operators; and (v) selection of good solutions, and replacement of these in the population. Differences in EAs most often include variations in how they evolve new solution vectors, how they select good solutions from the population, and how they replace solutions in the updated population. With the exception of the ACO algorithm, which is designed for discrete problems, these algorithms are designed to solve continuous problems. Problems with discrete variables may be solved by obtaining solutions based on continuous variables, rounded to the nearest allowable integer value. In this section, a summary of the different EAs compared in this paper is presented. Links to the source code of the algorithms are provided below. Unless otherwise noted, all algorithms are Matlab File Exchange Codes obtained from Mathworks.com.

3.1. Genetic Algorithm (GA)

Implementation of the GA (Holland 1972) begins with identification of a random set, or population, of solutions (called chromosomes). The objective function value, or fitness, of each solution is determined and compared to determine those solutions, or population members, that are allowed to evolve. Typically, the selection process employs tournament selection, which involves a series of competitions, or "tournaments," among pairs of solutions chosen at random from the population. The winner of each tournament is the solution with the best fitness. These winners are selected to undergo evolution with crossover, mutation, and selection operations, probabilistic-based mechanisms through which new solutions are constructed. In crossover,

portions of the set of variable values in two solutions are exchanged, and in mutation, randomly selected variable values for one solution are changed. Under the selection operation, the new solutions are evaluated and may be used to further evolve the population, if they provide better fitness (i.e., improved objective function value), relative to other population members. The process is continued for a large number of generations or until no further improvement in the best objective function is obtained.

<https://www.mathworks.com/help/gads/examples/coding-and-minimizing-a-fitness-function-using-the-genetic-algorithm.html>

(2017 Version)

3.2. Ant Colony Optimization (ACO)

ACO is a probabilistic approach that mimics the normal behavior of ants that tend to find the shortest route while searching for food. Initially, agents that are akin to ants move randomly toward food, and their pathway is considered one solution. In moving along a pathway, they indicate the shortest path to follow (e.g., the best objective function value) probabilistically, in a manner that is akin to an ant depositing a pheromone trail.

For example, in WDN design, an ant's path is the set of pipe diameters selected for each pipe in the network. In the first iteration, each ant randomly selects a set of pipe diameters as a path or solution vector. The objective function (WDN cost) is evaluated for each solution vector and a pheromone value is assigned to that vector based on the value of the objective function. This process is repeated for all ants in the population. In subsequent iterations, the probability that an ant will select a given solution vector is a function of the total pheromone concentration assigned to that vector thus far, where paths with higher pheromone values are selected to be

followed. At each iteration the pheromone concentration of each solution vector is updated, and this process continues until a termination criterion is satisfied. Throughout this process, the shortest path will continuously be reinforced by an increasing number of pheromones, and the pheromone trails along less preferred paths will decrease in a manner akin to the evaporation process (for an application of ACO to WDNs, see Maier et al. 2003).

<https://www.mathworks.com/matlabcentral/fileexchange/52859-ant-colony-optimization-aco>

(2015 Version)

3.3. Harmony Search (HS)

The HS algorithm (Geem 2006) is conceptually based on the musical process of searching for a ‘perfect state’ of harmony, such as in jazz improvisation. Jazz improvisation seeks a best state (fantastic harmony) determined by aesthetic estimation, and analogously, HS seeks a best state (global optimum) determined by evaluating the objective function. In music, the aesthetic estimation is evaluated by the set of pitches played by each instrument, and in HS the objective function evaluation is performed for the set of values assigned to each decision variable. In music, the harmony quality is enhanced with practice, and in HS the solution quality is enhanced over many iterations.

Initially, the population comprises many randomly generated solution vectors, each possessing an objective function value, or harmony quality. In subsequent iterations, each new harmony, or solution vector, is generated based on three rules: memory consideration, pitch adjustment, and random selection. In memory consideration, the value of the first decision variable for the new vector is chosen from any value within its specified memory range, i.e., the range of variable values previously selected. Values of the other decision variables are chosen in the same

manner. In pitch adjustment, a randomly selected variable is changed slightly, relative to its current value. And in random selection, a randomly selected variable is changed to any value, regardless of its memory consideration or current value. If the harmony of any new vector is better than the existing worst harmony in memory, the new vector is included in memory and the worst harmony is excluded. This procedure is repeated until satisfaction of a termination criterion, and thus a fantastic harmony is found. This algorithm was obtained, with permission from the author, at:

<https://sites.google.com/a/hydrateq.com/www/>

(2006 Version)

3.4. Differential Evolution (DE)

DE (Price et al. 2005) is an improved version of the GA. Similar to the GA, there are three important operators involved in DE, including the mutation, crossover, and selection operators. The main difference between GA and DE is that GA relies on its crossover operator to exchange information among solutions, while DE primarily relies on its mutation operator to form new solution vectors.

Initially, the population comprises many randomly generated solution vectors, each possessing an objective function value. In the mutation process, the genetic material, or decision variable values, from three or more parent solutions are exchanged to produce one new solution. The crossover operator determines whether and where this new vector is allowed to breed with a vector from a previous generation, or iteration. Subsequently, the selection process compares the bred with the existing vector, and allows the best among these to survive to the next generation. By considering components of previous generations in the construction of new vectors, the

crossover operator efficiently shuffles information about successful solutions, enabling the search for an optimum to focus on the most promising area of the solution space. Over multiple iterations, DE refines the search by adapting the mutation increments (i.e., the step size of the allowable variable changes), based on the stage of the evolutionary process. At the beginning of the evolution process, the mutation operator favors exploration, i.e., larger incremental changes of variable values, and as evolution progresses, it favors exploitation, i.e., smaller incremental changes of variable values. This process continues until satisfaction of a termination criterion is achieved.

<https://www.mathworks.com/matlabcentral/fileexchange/18593-differential-evolution>

(2017 Version)

3.5. Particle Swarm Optimization Gravitational Search Algorithm (PSOGSA)

PSO (Kennedy and Eberhart 1995) originated as a simulation of a simplified social system, such as birds flocking or fish schooling. Similar to GA, PSO is also population-based and searches for optimal solutions by updating generations. However, unlike GA, PSO possesses no evolution operators. Instead, PSO relies on the exchange of information between individuals, or particles, of the population, or swarm. In order to select fitter particles for further populations, each particle adjusts its trajectory as a function of its current position (or solution), its previous best position (solution) over all previous generations, and the current best position attained among all other population members in the most previous generation. PSO presents the advantage of being conceptually simple, however, its main disadvantage is the risk of a premature search convergence due to the lack of diversity resulting from its simple operator.

PSOGSA (Mirjalili and Hashim 2010), a modification of PSO, employs the gravitational search algorithm (GSA) for exploring the fitness landscape (Rashedi et al. 2009), which is inspired by

the laws of motion, where particles are attracted proportional to the product of their masses and inversely proportional to the square of the distance between them. In GSA, for each solution in a generation of solutions, a directional move is evaluated as a function of two masses, and the distance between them. The masses are the objective function value of the solution divided by the sum of the objective function values for all solutions, and the difference between the objective function value and the worst objective function value in the given generation, divided by the range of objective function values for that generation. The distance between the masses is the difference between the solution vector, and the previous solution vector for this population member. PSOGSA uses PSO to exploit good solutions and GSA to explore a wider array of solutions.

<https://www.mathworks.com/matlabcentral/fileexchange/35939-hybrid-particle-swarm-optimization-and-gravitational-search-algorithm-psogsa>

(2018 Version)

3.6. Artificial Bee Colony (ABC)

In the ABC algorithm (Karaboga 2005) the position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. Search processes are based on three kinds of bees: employed bees; onlooker bees; and scout bees. Initially, the number of the employed and onlooker bees is equal to the number of solutions considered in the population, and there are no scout bees. Each cycle of the search procedure consists of the following actions: employed bees are sent onto their food sources (solutions) and the nectar amounts of the sources (performance) are evaluated; employed bees share the nectar information of food sources with the onlooker bees, and based on this information, the onlooker bees produce a new random food source

(solution) and evaluate the nectar amount of these food sources. This process continues iteratively, and if a solution represented by a food source is not improved by a pre-determined number of iterations, then the food source is abandoned by the onlooker and employed bees for that source, and the employed bee is converted to a scout bee, and is sent randomly onto possible new food sources. In this case, a new solution vector is randomly created and replaces the abandoned solution.

<https://www.mathworks.com/matlabcentral/fileexchange/52966-artificial-bee-colony-abc-in-matlab>

(2015 Version)

3.7. Soccer League Competition (SLC) Algorithm

The SLC algorithm (Moosavian and Roodsari 2014) is inspired from professional soccer leagues. It involves different teams, or collections of solutions, where each solution is a team member, and a number of operators act on the team members to perform an effective search for the near optimal solution. SLC mimics matches between teams and determines the winners based on their relative power, or the objective function value, where the winner of a match has a higher probability of increasing its power for future matches. After each match, all players (decision vectors) on the winning team undergo operations that change their values, and thus their power (i.e., strengthening or weakening each player), producing modified team members. Finally, the original strength of each player on this winning team is compared with its modified strength and the player, i.e., decision vector, with the greater strength, is allowed to play on the team in the next match.

An iteration, or round, is defined as a set of matches that allows each team, or modified team, to play with each other team in the league. Thereafter, the next round is played with these new teams. The user specifies the stopping criteria to be either based on a limit of the number of rounds undertaken, or the number of function evaluations made.

<https://www.mathworks.com/matlabcentral/fileexchange/56480-soccer-league-competition-slc-algorithm-for-discrete-problems>

(2016 Version)

3.8. Covariance Matrix Adaptation Evolution Strategy (CMAES)

CMAES generates new populations of solutions by sampling from a probability distribution that is constructed during the optimization process. It is derived from the concept of self-adaptation in evolution strategies, which adapts the covariance matrix of a multivariate normal search distribution. An initial population of solutions is generated, a high-performing fraction of this population is selected, and future populations are evolved based on the adaptation of the: (i) mean vector of the selected solutions; (ii) covariance matrices of the selected solutions and the total population of solutions; and (iii) an evolving global step size. The evolution path of the high-performing solutions is thus based on the vector difference between the best individuals in the current and previous generations, which accelerates the convergence of the algorithm. As the algorithm progresses, the global step size changes dynamically as a function of the distribution of the high-performing solutions, where larger increments favor exploration and smaller increments favor exploitation. While the algorithm outperforms other similar classes of learning algorithms on benchmark multimodal objective functions (Hansen 2004), there is no available

information about the performance of CMAES in discrete optimization problems, such as in WDN design.

This algorithm was one of five winners of the 2013 Congress on Evolutionary Computation (CEC) Competition, see:

http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2013/CEC2013.htm

<https://www.mathworks.com/matlabcentral/fileexchange/52898-cma-es-in-matlab>

(2015 Version)

3.9. MEtaheuristics for Systems Biology and Bioinformatics Global Optimization (MEIGO)

MEIGO is a black-box optimizer which incorporates the metaheuristic search algorithms of an Enhanced Scatter Search (ESS) and a Variable Neighbourhood Search (VNS). A diverse initial population of solutions is generated, VNS and ESS are applied to transform each solution into enhanced solutions, a subset of the enhanced solutions is further combined, and the best solutions among these are retained for iterative applications of the search algorithms. This process is repeated until a termination criterion is satisfied.

An ESS applies the same operator used in DE to explore the current population, or reference Set (RefSet). The ESS operator enhances diversity and prevents the search from premature stagnation by creating dissimilar solutions and retaining these in a revised RefSet. A VNS is a trajectory-based metaheuristic that performs a local search for each solution in the RefSet to move it toward the incumbent high-performing solutions, thus exploiting different local optima, among which the global optimum is expected to be found. One of the key operators of the algorithm is the strategy followed to change the degree of movement toward a given high-

performing solution, which perturbs the decision variables using a dynamically adapting distance criterion. The typical strategy consists of visiting the neighbourhood close to the current solution (i.e., small perturbations of the solution), until no further improvement is achieved.

MEIGO has been used to solve problems with real-valued and mixed-integer decision variables, and problems with integer and binary decision variables. It is viewed as efficient because it allows the user to apply parallel computation efforts (details of the algorithm are provided in Egea et al. 2014).

This algorithm was obtained, with permission from the author, at:

<http://nautilus.iim.csic.es/~gingproc/>

(2016 Version)

3.10. Whale Optimization Algorithm (WOA)

WOA mimics the social behavior of humpback whales and has recently been applied for optimization of engineering problems (Mirjalili and Lewis 2016). The WOA algorithm starts with a set of random solution vectors. At each iteration, the positions of solution vectors are updated with respect to either a randomly chosen solution vector or the best solution obtained thus far. The WOA employs three operators to simulate the search for prey, encircling prey and bubble-net foraging behavior of humpback whales. In the prey operation, new solutions in the current iteration are created by selecting a random solution from the previous iteration and subtracting the difference between a randomly changed version of it and a given solution vector from the previous iteration. The encircling prey operator generates a new solution in the current iteration by selecting the best solution from the previous iteration and subtracting the difference between a randomly changed version of it and a given solution vector from the previous

iteration. The bubble-net foraging operator generates a new solution vector in the current iteration by selecting the best solution from the previous iteration and adding a random logarithmic spiral of the difference between it and a given solution vector from the previous iteration.

<https://www.mathworks.com/matlabcentral/fileexchange/55667-the-whale-optimization-algorithm>

(2017 Version)

4. Applications to WDN design

The performance of the ten above-mentioned EAs is tested for three water distribution networks, the Two-Loop and Hanoi benchmark networks, and the WDN for the City of Farhadgerd, Iran. For consistent comparison, three stopping criteria are defined as: (i) the maximum number of function evaluations is set to 1000. Given this criteria, the initial speed of convergence and performance of the algorithms may be evaluated.; (ii) the maximum number of function evaluations is set to 10000. Here, the performance of the algorithms is evaluated for a more mature evolution, compared with (i).; and (iii) the maximum number of function evaluations is set to 40000, for which the evolution of the algorithms is considered to be fully mature. For each of these criteria, all algorithms are executed 20 times, i.e., 20 sets of initial populations of solutions are generated based on 20 different random number sequences.

Statistical analysis of results for all algorithms are performed and the mean, median, standard deviation, minimum, and maximum for the 20 executions of the algorithms are calculated. The mean shows the overall performance of the algorithm for the different runs. The standard deviation indicates the variation of the final solution with respect to mean value. Small standard

deviation values show that an algorithm has a similar behavior regarding its search process for different initial random populations. All of the computations are implemented in the MATLAB programming language environment with an Intel(R) Core(TM) 2Duo CPU P8700 @ 2.53GHz and 4.00 GB RAM.

4.1. Two-loop benchmark network

This network, shown in Figure 1, is a hypothetical benchmark which has seven nodes and eight pipes with two loops, and is fed by a reservoir with a 210 m fixed head (Alperovits and Shamir 1977). The pipes are all 1,000 m long with a Hazen-Williams coefficient of 130. The minimum pressure limitation is 30 m above ground level for all nodes. There are 14 available commercial pipe diameters. The degree of candidate diameters is an indicator of the size and complexity of the network problem, and is defined as the number of candidate diameters divided by the number of pipes. For this case, the degree of candidate diameters is equivalent to $14 / 8 = 1.75$. Geem (2006) determines the minimum cost of this network to be \$419,000 US.

The performance of the algorithms, in terms of the minimum total cost obtained, under the three stopping criteria, is shown in the Tables 1, 2, and 3, for the 1000-, 100000, and 40000-function evaluation stopping criteria, respectively. The results show that, for the first stopping criterion, the PSOGSA exhibits the best mean cost of \$433,000 US and the MEIGO algorithm second best mean of \$436,000 US, while the MEIGO algorithm exhibits the best minimum of \$419,000 US and PSOGSA the second best minimum of \$420,000 US. The PSOGSA also has the least standard deviation of 1467 which indicates that it exhibits the best overall performance among the remaining algorithms. The GA and HS algorithms result in the third and fourth lowest mean cost, respectively, and the ACO and WOA algorithms exhibit the worst performance among all algorithms.

As the stopping criterion is relaxed to 10000 function evaluations, the performance of the ABC, SLC, HS, CMAES and DE algorithms improves significantly, and each of these algorithms achieve the global optimal minimum cost. Here, the ABC, DE, and SLC algorithms have the least standard deviation, in this order. These small standard deviation values show that the algorithms perform the search process consistently, for the breadth of different initial random populations. In contrast, the PSOGSA algorithm also achieves the global optimum, but its average performance over 20 executions is relatively poor. Similar results are found with stopping criteria is relaxed to 40000 function evaluations.

4.2. Hanoi benchmark network

The Hanoi network, shown in Figure 2, consists of 32 nodes, 34 pipes with 3 loops, and is fed by gravity from a reservoir with a 100 m fixed head. All pipes have a Hazen-Williams coefficient of 130 and the minimum head limitation at all nodes is 30 m above ground level (Alperovits and Shamir 1977). There are six possible pipe diameters and 34 pipes in the system, thus the degree of candidate diameter is equivalent to $6 / 34 = 0.1765$. The global optimum solution has the total cost of \$6.1 million US (Alperovits and Shamir 1977).

Optimization results for this network are provided in Tables 4, 5, and 6 for the 1000-, 10000-, and 40000-function evaluation stopping criteria, respectively. For the first criterion, the SLC algorithm exhibits the highest performance with the minimum and mean least cost equal to \$7.0 and \$8.3 million, respectively. It also has the smallest standard deviation. The MEIGO and WOA algorithms obtain equal minimum costs of \$7.9 million US, and mean costs of \$8.5 and \$8.7 million US, respectively. For the 10000-function evaluation criterion, the SLC also exhibits the best performance, and has a minimum and mean least cost of \$6.1 and \$6.3 million US, respectively. The CMAES algorithm has the smallest standard deviation, but does not find the

lowest cost overall. The CMAES, GA and MEIGO algorithms are the second, third and forth best algorithms, respectively. A similar trend is exhibited for the 40000-function evaluation criterion, where the SLC and CMAES algorithms exhibit the highest performance, in this order, and WOA, PSOGSA and ACO exhibit the worst performance. This indicates that the convergence properties of the PSOGSA algorithm do not improve when the number of iterations increases. A comparison of the Hanoi and Two-Loop benchmark networks indicates that as the degree of candidate diameter decreases, the convergence potential of the SLC and CMAES algorithm improves significantly.

4.3. Farhadgerd network

The Farhadgerd WDN serves a population of approximately 8200 in the City of Farhadgerd, Iran, a residential community near the regional capital, Mashhad. The network, shown in Figure 3, comprises 68 pipes and 57 nodes, has one reservoir with a head of 510 m, and a minimum pressure requirement of 20 m at all nodes. There are nine possible pipe diameters and thus the degree of candidate diameter for this network is equivalent to $9 / 68 = 0.13$. Relevant data for this network is provided in Appendix A.

Optimization results for this network are provided in Tables 7, 8, and 9 for the 1000-, 10000-, and 40000-function evaluation stopping criteria, respectively. In this paper, the recorded optimal cost for Farhadgerd is \$1.78E7 US.

For the first first criterion, the PSOGSA exhibits the best mean cost of \$3.03E07 US and the MEIGO algorithm second best mean of \$3.2 E07 US, while the MEIGO algorithm exhibits the best minimum of \$2.2E07 US and PSOGSA the second best minimum of \$2.77E07 US. The ACO, DE and ABC algorithms exhibit the worst performance with respect to the other

algorithms, under this criterion. For the 10000-function evaluation criterion, the CMAES algorithm again out-performs all other algorithms, with a minimum and mean least cost of \$1.8E07 and \$1.9E07 US, respectively. It also exhibits the smallest standard deviation. The SLC algorithm performs second-best, under this criterion. The results of CMAES and SLC are similar for the cases with a stopping criterion of 40000 function evaluations. The minimum and maximum least cost solutions obtained by both CMAES and SLC algorithm are \$1.78E07 and \$1.8E07 US, respectively, which are less than all solutions obtained by other algorithms.

5. The Friedman test

The Friedman test (Friedman 1937; Friedman 1940), a nonparametric analysis of variance, implemented using IBM-SPSS Statistics 23 software, is used for comparing the performance of the algorithms. The test is a multiple comparisons test that may be used to detect significant differences between the responses of two or more subjects to various treatments. For the results presented herein, the response is the mean least cost values, the subjects are the example networks, and the treatments are the applications of the various algorithm. Three Friedman tests were conducted, one for each of the stopping criteria. The test determines the ranks of the responses, or mean least cost solutions, for each of the three networks, resulting under each of the algorithms. For each algorithm, the mean of these ranks is evaluated. The Friedman test statistic, based the number of subjects, the number of is treatments, and mean ranks of the responses for each treatment, is tested against a Chi-Square critical value, and indicates whether the various treatments, or algorithms, are significantly different.

Results of the Friedman test are provided in Figures 4a, 4b, and 4c, for the stopping criteria based on 1000, 10000, and 40000 function evaluations, respectively. In these figures, the mean rank of the algorithms is the mean of the ranks of the resulting mean least cost solutions over

each of the three networks, where 10 is the maximum possible mean rank, N is equal to 3 subjects, Chi-Square is the Chi-Square critical value for $N=3$, 10-1 degrees of freedom (df), the given mean ranks and a significance level of 0.05. Asymp. Sig., the asymptotic significance, is the p-value for the test and indicates similarity of algorithms if this value is greater than 0.05.

The Friedman test results show that for a stopping criteria of 1000 function evaluations, PSOGSA, MEIGO and GA have the lowest mean ranks, in increasing order, and as a result, higher performance. For larger stopping criteria, SLC, CMAES and DE have the lowest mean ranks, in increasing order, and thus better performance. Furthermore, rejection of the null hypothesis for all three Friedman tests (i.e., Asymp. Sig. or p-value < 0.05), in all cases, indicates that the performance of the algorithms differs significantly.

6. Conclusion

In this paper, ten EAs are evaluated for finding the optimal least-cost design of three water distribution networks, under three pre-defined stopping criteria: 1000, 10000, and 40000 function evaluations. The performance of the algorithms is evaluated based on statistical analysis of 20 different executions of the algorithms and Friedman tests of the mean of the resulting solutions. The algorithms are compared in terms of the mean, median, standard deviation, minimum, and maximum of all executions for each network, as well as the mean rank in the Friedman test. Small mean and standard deviation values indicate that an algorithm provides an attractive and consistent performance for different initial random populations. For the Two-loop network nearly all ten algorithms are able to identify the global optimum solution, though the small standard deviations of the ABC, DE, and SLC algorithms indicate high consistency. For the Hanoi network the SLC exhibits the best performance, and has the lowest minimum and mean cost. The CMAES, GA and MEIGO algorithms are the second, third and forth best algorithms,

respectively, in terms of mean cost. For the Farhadgerd network, the CMAES algorithm provides the lowest mean and standard deviation, and the CMAES and SLC algorithms provide the lowest minimum cost among all ten algorithms. Friedman test results show that the performance of the algorithms differ significantly, and that at early stages of the solution process, PSOGSA, MEIGO and GA perform well, but that as the algorithms proceed to undertake more function evaluations, the SLC, CMAES and DE out-perform the others.

Due to the requirement to satisfy the minimum required pressure throughout the network, a WDN design problem may have various local optima which may be spread over the feasible region. The minimum required pressure is a nonlinear function of the pipe size diameters and their configuration. For a given design, if any one pipe diameter were to be changed, a corresponding change in the direction of flow in the pipe may occur and the distribution of the pressures throughout the entire system may also change in response. Therefore, when the optimization algorithm is close to the global optimum, there is no guarantee of finding it. This hydraulic sensitivity makes the optimization of the network model difficult to solve.

To overcome this problem and effectively escape from local optima, EAs should be designed to provide diverse solutions. The solution approach should not concentrate solely on the evolution of the best set of solutions, but on a wide range of solutions, making it independent of the initial population, which has typically been used to determine the direction of search. The CMAES and SLC algorithms exhibit these properties and can successfully avoid being trapped at local optimum solutions. CMAES creates new generations of solutions based on the entire distribution of a wide range of solutions, thus enhancing the exploration process, and adapts the global step size dynamically to efficiently conduct the exploitation process. By applying a combination of coarse (exploration)- and fine (exploitation)-scale search processes, the SLC algorithm is less

dependent on a single initial population than most EAs. The coarse-scale search process of the SLC is conducted among several different populations (i.e., teams) that are distributed across the feasible region, while the fine-scale search process is undertaken within each of the populations (i.e., among players of the teams). Another advantage of the SLC is that it applies different evolutionary operators for different teams and different players based on their performance (i.e., power).

The CMAES and SLC algorithms show promise for applications to more complex WDNs, and other nonpolynomial hard problems. Because it uses the distribution of the population at a given generation to provide a wide and deep exploration of the feasible region, the CMAES may be integrated with other algorithms, such as the SLC, to enhance their exploration and exploitation processes. Furthermore, in the CMAES algorithm the step size of the search may shrink to the point of restricting exploration, and to guard against such cases, the mutation operators of DE or ABC may be adapted and integrated in CMAES. While the high performing algorithms evaluated in this paper were analyzed for the single objective least-cost WDN design problem, consideration of other objectives, such as network reliability and robustness, requires the advancement of multi-objective versions of the algorithms for evaluating the Pareto optimal frontier for the collection of objectives. One approach for doing so is to integrate nondominated sorting with existing algorithms. Given the performance of the CMAES and SLC algorithms for addressing the presented network problems, future work in the advancement of these algorithms should develop their multi-objective versions, and evaluate these against the performance of other widely used algorithms, such as NSGA, NSGA-II and NSDE.

527 7. References

- 528 Alperovitz, E. and Shamir, U., 1977. Design of Optimal Water Distribution Systems. *Water*
529 *Resour. Res.*, **13**(6): 885–900.
- 530 Bi, W., Dandy, G.C. and Maier, H.R., 2015. Improved Genetic Algorithm Optimization of Water
531 Distribution System Design by Incorporating Domain Knowledge, *Environmental Modelling &*
532 *Software*, **69**: 370–381.
- 533 Cunha, M.C. and Sousa, J., 2001. Hydraulic Infrastructures Design Using Simulated Annealing,
534 *J. Infrastruct. Syst.*, **7**(1): 32–38.
- 535 Egea, J.A., Martí, R., Banga, J.R. 2010: An evolutionary method for complex-process
536 optimization. *Comput Oper Res.* 2010, **37** (2): 315-324. 10.1016/j.cor.2009.05.003.
- 537 Egea, J.A., Henriques, D., Cokelaer, T., Villaverde, A.F., MacNamara, A., Danciu, D.P., Banga,
538 J.R. and Saez-Rodriguez, J. 2014. MEIGO: an open-source software suite based on
539 metaheuristics for global optimization in systems biology and bioinformatics. *BMC*
540 *Bioinformatics* **15**(136): 1-9.
- 541 Hansen, N. and Kern, S. 2004. Evaluating the CMA Evolution Strategy on Multimodal Test
542 Functions. In Eighth International Conference on Parallel Problem Solving from Nature PPSN
543 VIII, Proceedings, 282-291, Berlin: Springer.
- 544 Holger R. Maier, H.R., Simpson, A.R., Zecchin, A.C. and Foong, W.K. 2003. Ant Colony
545 Optimization for Design of Water Distribution Systems, *J. Water Resour. Plan. Manag.*, **129** (3):
546 200-209.

- 547 Holland, J. H. 1975. *Adaptation in Natural and Artificial Systems: An Introductory Analysis*
548 *with Applications to Biology, Control, and Artificial Intelligence*. Ann Arbor, MI: University of
549 Michigan Press, 1975.
- 550 Eusuff, M.M. and Lansey, K.E. 2003. Optimisation of Water Distribution Network Design Using
551 Shuffled Frog Leaping Algorithm, *J. Water Resour. Plan. Manag.*, **129**(3): 210–225.
- 552 Friedman, M. 1937. The use of ranks to avoid the assumption of normality implicit in the
553 analysis of variance, *Journal of the American Statistical Association*, **32** (1937): 674–701.
- 554 Friedman, M. 1940. A comparison of alternative tests of significance for the problem of m
555 rankings, *Annals of Mathematical Statistics*, **11** (1940): 86–92.
- 556 Geem, Z.W. 2006. Optimal Cost Design of Water Distribution Networks Using Harmony
557 Search, *Eng. Optim.*, **38**(3): 259–280.
- 558 Karaboga, D. 2005. An Idea Based On Honey Bee Swarm For Numerical Optimization,
559 Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering
560 Department, 2005.
- 561 Kennedy, J. and Eberhart, R.C. 1995. Particle Swarm Optimization, in: *Proceedings of the 1995*
562 *IEEE International Conference on Neural Networks*, **4**: 1942–1948.
- 563 Lin, M.D., Liu, Y.H., Liu, G.F. and Chu, C.W. 2007. Scatter Search Heuristic For Least-Cost
564 Design of Water Distribution Networks, *Eng. Optim.*, **39** (7): 857–876.
- 565 Maier, H.R., Simpson, A.R., Zecchin, A.C., Foong, W.K., Phang, K.Y., Seah, H.Y. and Tan,
566 C.L. 2003. Ant Colony Optimization For The Design of Water Distribution Systems, *J. Water*
567 *Resour. Plan. Manag.*, **129** (3): 200–209.

- 568 Mirjalili, S. and Lewis, A. 2016. The Whale Optimization Algorithm, *Advances in Engineering*
569 *Software*, **95**, 51-67. <http://dx.doi.org/10.1016/j.advengsoft.2016.01.008>.
- 570 Mirjalili, S., and Hashim, S. Z. M. 2010. A New Hybrid PSO-GSA Algorithm for Function
571 Optimization, *Computer and Information Application (ICCIA), 2010 International Conference*
572 *on. IEEE*, 374-377.
- 573 Mladenović N, Hansen P: Variable neighborhood search. *Comput Oper Res.* 1997, 24: 1097-
574 1100.
- 575 Moosavian, N., and Lence, B. 2016. Non-dominated Sorting Differential Algorithms for Multi-
576 objective Optimization of Water Distribution Systems. *Journal of Water Resources Planning &*
577 *Management, ASCE*, DOI:10.1061/(ASCE)WR.1943-5452.0000741.
- 578 Moosavian, N and Roodsari, B.K. 2014. Soccer League Competition Algorithm: A Novel Meta-
579 Heuristic Algorithm For Optimal Design of Water Distribution Networks, *Swarm and*
580 *Evolutionary Computation*, **17**: 14-24.
- 581 Murphy, L.J. and Simpson, A.R. 1992. Pipe Optimization Using Genetic Algorithms, Research
582 Rep. No. **R93**, Dept. of Civil Engineering, Univ. of Adelaide, Adelaide, Australia.
- 583 Perelman, L. and Ostfeld, A. 2007. An Adaptive Heuristic Cross-Entropy Algorithm For Optimal
584 Design of Water Distribution Systems, *Eng. Optim.*, **39** (4): 413–428.
- 585 Price, K.V., Storn, R.M. and Lampinen, J.A. 2005. (Eds.), *Differential Evolution: A Practical*
586 *Approach to Global Optimization*, Springer Natural Computing Series.
- 587 Rashedi, E., Nezamabadi, S., and Saryazdi, S. 2009. GSA: A Gravitational Search Algorithm,
588 *Information Sciences*, 179(13), 2232- 2248.

- 589 Suribabu, C.R. and Neelakantan, T.R. 2006. Design of Water Distribution Networks Using
590 Particle Swarm Optimization, *Urban Water J.*, **3** (2): 111–120.
- 591 Todini, E., and Pilati, S.1988. A Gradient Algorithm for the Analysis of Pipe Network,
592 *Computer Applications in Water Supply, 1, System Analysis and Simulation*, B. Coulbeck and O.
593 Chu-Hou, eds., Wiley, London, 1–20.
- 594 Vasan, A. and Simonovic, S. 2010. Optimization of Water Distribution Network Design Using
595 Differential Evolution, *J. Water Resour. Plan. Manag.*, **136** (2): 279–287.
- 596 Zheng, F., Zecchin, A.C. and Simpson, A.R. 2013. Self-Adaptive Differential Evolution
597 Algorithm Applied to Water Distribution System Optimization, *J. Comput. Civil Eng.*, ASCE,
598 **27**(2): 148–158.

Draft

Appendix A: Pipe and nodal information for Farhadgerd network

Table A1. Candidate pipe size diameters and corresponding costs (\$/meter length) for Farhadgerd network

Diameter (mm)	Cost (\$/meter length)
63.8	638
79.2	792
96.8	968
150	1500
200	2000
250	2500
300	3000
350	3500
400	4000

Draft

Table A2. Nodal information for Farhadgerd network

ID	Elevation (m)	Demand (L/s)	ID	Elevation (m)	Demand (L/s)
R-1	510	-	J-28	454.5	0.9242
J-1	479	2.1037	J-29	458	1.0336
J-2	468.7	3.3318	J-30	455.8	1.2403
J-3	466.5	1.4227	J-31	437	1.4462
J-4	467.1	0.5594	J-32	434.6	0.9990
J-5	469	0.1581	J-33	430	1.3035
J-6	467.3	1.2282	J-34	439.4	0.3996
J-7	469.2	2.4685	J-35	433.3	16.3083
J-8	460.5	0.3891	J-36	437	1.1037
J-9	461.3	4.5357	J-37	455.6	0.5594
J-10	465.4	0.9850	J-38	468.5	10.3238
J-11	476.3	0.4621	J-39	472.7	0.9850
J-12	477	0.6931	J-40	478.7	0.9485
J-14	472.5	0.1581	J-41	486	0.9971
J-15	478.7	0.6688	J-42	477.5	0.3770
J-16	478.5	0.5594	J-43	473.3	0.7053
J-17	475.5	4.7546	J-44	471.4	2.1037
J-18	475	0.2067	J-49	431.2	0.3140
J-19	480	0.3162	J-50	431.8	0.1618
J-20	472	1.3133	J-51	462.8	3.5872
J-21	456.2	0.6931	J-52	440	0.0000
J-22	458.3	1.2890	J-53	454.5	0.1000
J-23	446.3	0.8147	J-13	455.8	0.0000
J-24	440	0.5715	J-45	455.8	0.0000
J-25	445.4	1.5200	J-47	440	0.0000
J-26	438	2.7688	J-48	440	0.0000
J-27	441	2.0076	J-46	455.8	0.0000

Table A3. Pipes characteristics for Farhadgerd network

ID	J-in	J-out	Length (m)	C	ID	J-in	J-out	Length (m)	Hazen- Williams factor, C
P-1	R-1	J-1	885.3851	120	P-41	J-35	J-36	437.3994	130
P-2	J-1	J-2	328.5421	120	P-42	J-36	J-26	359.4174	130
P-3	J-2	J-3	86.0021	120	P-43	J-30	J-37	141.5928	130
P-4	J-3	J-4	195.1899	120	P-44	J-37	J-38	667.3086	130
P-5	J-4	J-5	30.9500	120	P-45	J-38	J-39	242.5379	120
P-6	J-5	J-6	68.4875	130	P-46	J-39	J-40	229.1806	130
P-7	J-6	J-7	563.0813	130	P-48	J-41	J-42	203.4631	130
P-8	J-7	J-8	417.1908	130	P-49	J-42	J-43	100.3686	130
P-9	J-8	J-9	416.8621	120	P-50	J-43	J-29	503.9709	120
P-10	J-9	J-4	294.4953	120	P-51	J-38	J-44	385.2943	130
P-11	J-7	J-10	367.3761	130	P-57	J-32	J-49	190.6523	130
P-12	J-10	J-11	372.3605	130	P-58	J-49	J-50	208.6375	130
P-13	J-11	J-12	147.7453	130	P-59	J-50	J-33	227.8588	130
P-16	J-14	J-15	213.1492	130	P-60	J-34	J-31	278.1212	120
P-17	J-15	J-16	233.7499	130	P-61	J-7	J-14	322.7427	130
P-18	J-14	J-17	315.1370	130	P-62	J-12	J-16	193.8307	130
P-19	J-17	J-18	247.3020	130	P-63	J-27	J-35	354.9940	130
P-20	J-18	J-19	155.8379	130	P-64	J-2	J-51	342.1014	130
P-21	J-19	J-20	138.7134	130	P-56	J-51	J-28	299.1446	130
P-22	J-20	J-5	302.7770	130	P-55	J-1	J-43	135.5139	120
P-23	J-9	J-21	202.5051	130	P-54	J-43	J-39	444.0197	120
P-25	J-22	J-3	421.6994	130	P-53	J-45	J-31	505.6546	120
P-26	J-21	J-23	451.5464	130	P-52	J-47	J-26	247.9111	130
P-27	J-23	J-24	94.9841	130.013	P-47	J-53	J-27	445.0974	130
P-28	J-24	J-25	138.8879	130	P-38	J-52	J-53	504.6454	130
P-29	J-25	J-22	381.8934	130.013	P-36	J-53	J-13	468.0775	130
P-33	J-28	J-25	388.1121	130	P-32	J-48	J-52	16.5572	130
P-34	J-28	J-29	348.5208	130	P-31	J-46	J-13	16.3047	120
P-35	J-29	J-30	125.7833	120	P-30	J-45	J-13	52.5861	120
P-37	J-31	J-32	333.0668	130	P-24	J-46	J-30	14.3932	120
P-39	J-33	J-34	286.3284	130	P-15	J-47	J-52	74.3145	130
P-40	J-34	J-35	436.9019	130	P-14	J-48	J-24	13.5482	130
P-65	J-46	J-30	14.3932	120	P-66	J-46	J-13	16.3047	120
P-67	J-48	J-24	13.5482	130	P-68	J-48	J-52	16.5572	130

Figure 1: Two-Loop benchmark network

Draft

Figure 2: Hanoi benchmark network

Figure 3: Farhadgerd network

Figure 4: Results of Friedman test for: a) 1,000, b) 10,000, and c) 40,000 function evaluations of three networks

Table 1: Least cost solutions for the Two-Loop benchmark network: 1000 function evaluations

	GA	ACO	HS	DE	PSOGSA	ABC	CMAES	SLC	MEIGO	WOA
Mean	4.57E05	5.84E05	4.73E05	5.38E05	4.33E05	4.86E05	5.23E05	4.98E05	4.36E05	5.49E05
Median	4.54E05	5.73E05	4.72E05	5.34E05	4.30E05	4.79E05	5.21E05	4.90E05	4.39E05	5.28E05
Std	1.77E04	6.09E04	1.76E04	3.19E04	14.67E3	3.10E04	2.32E04	3.42E04	1.11E04	6.78E04
Minimum	4.29E05	4.61E05	4.41E05	4.82E05	4.20E05	4.37E05	4.82E05	4.42E05	4.19E05	4.43E05
Maximum	5.02E05	7.19E05	5.02E05	6.00E05	4.78E05	5.58E05	5.59E05	5.81E05	4.60E05	7.23E05

Table 2: Least cost solutions for the Two-Loop benchmark network: 10000 function evaluations

	GA	ACO	HS	DE	PSOGSA	ABC	CMAES	SLC	MEIGO	WOA
Mean	4.47E05	4.54E05	4.21E05	4.21E05	4.30E05	4.20E05	4.21E05	4.20E05	4.25E05	5.03E05
Median	4.48.E05	4.55.E05	4.20.E05	4.20.E05	4.29.E05	4.20.E05	4.20.E05	4.20.E05	4.20.E05	4.98.E05
Std	1.77.E04	8.50.E03	3.62.E03	1.11.E03	9.53.E03	4.53.E02	6.98.E03	1.28.E03	8.91.E03	3.91.E04
Minimum	4.20.E05	4.34.E05	4.19.E05	4.19.E05	4.19.E05	4.19.E05	4.19.E05	4.19.E05	4.19.E05	4.43.E05
Maximum	4.95.E05	4.69.E05	4.32.E05	4.22.E05	4.46.E05	4.20.E05	4.27.E05	4.24.E05	4.43.E05	5.81.E05

Table 3: Least cost solutions for the Two-Loop benchmark network: 40000 function evaluation

	GA	ACO	HS	DE	PSOGSA	ABC	CMAES	SLC	MEIGO	WOA
Mean	4.31E05	4.37E05	4.20E05	4.19E05	4.24E05	4.19E05	4.22E05	4.19E05	4.22E05	4.67.E05
Median	4.27E05	4.38E05	4.19E05	4.19E05	4.20E05	4.19E05	4.20E05	4.19E05	4.20E05	4.53.E05
Std	1.06E04	6.13E03	1.90E03	3.48E02	8.81E03	1.87E02	2.29E03	5.10E02	6.93E03	4.85.E04
Minimum	4.20E05	4.23E05	4.19E05	4.19E05	4.19E05	4.19E05	4.19E05	4.19E05	4.19E05	4.22.E05
Maximum	4.51E05	4.46E05	4.27E05	4.20E05	4.45E05	4.20E05	4.43E05	4.20E05	4.43E05	6.48.E05

Table 4: Least cost solutions for the Hanoi benchmark network: 1000 function evaluations

	GA	ACO	HS	DE	PSOGSA	ABC	CMAES	SLC	MEIGO	WOA
Mean	1.7E07	2.2E08	4.5E07	8.2E07	1.34E07	3.2E08	6.5E07	8.3E06	8.5E06	8.7E06
Median	1.4E07	2.2E08	4.2E07	7.9E07	9.88E06	3.5E08	5.6E07	7.3E06	8.4E06	8.6E06
Std	8.6E06	5.9E07	1.6E07	4.3E07	7.35E06	1.4E08	4.4E07	3.3E06	4.0E05	5.0E05
Minimum	8.4E06	1.3E08	2.7E07	2.9E07	7.35E06	1.1E08	9.7E06	7.0E06	7.9E06	7.9E06
Maximum	3.5E07	3.5E08	8.3E07	2.1E08	3.01E07	5.4E08	2.2E08	2.0E07	9.3E06	9.7E06

Table 5: Least cost solutions for the Hanoi benchmark network: 10000 function evaluations

	GA	ACO	HS	DE	PSOGSA	ABC	CMAES	SLC	MEIGO	WOA
Mean	6.6E06	8.3E06	7.1E06	6.7E06	7.74E06	7.0E06	6.4E06	6.3E06	6.6E06	8.4E06
Median	6.6E06	7.9E06	7.1E06	6.7E06	7.49E06	7.0E06	6.4E06	6.3E06	6.6E06	8.5E06
Std	1.4E05	1.5E06	1.1E05	7.7E04	8.49E05	1.9E05	3.6E04	1.5E05	1.7E05	3.5E05
Minimum	6.4E06	7.3E06	6.7E06	6.5E06	7.08E06	6.6E06	6.3E06	6.1E06	6.4E06	7.7E06
Maximum	6.9E06	1.4E07	7.2E06	6.8E06	1.1E07	7.4E06	6.4E06	6.6E06	7.0E06	9.3E06

Table 6: Least cost solutions for the Hanoi benchmark network: 40000 function evaluations

	GA	ACO	HS	DE	PSOGSA	ABC	CMAES	SLC	MEIGO	WOA
Mean	6.5E06	7.0E06	6.7E06	6.4E06	6.72E06	6.7E06	6.2E06	6.2E06	6.4E06	8.0E06
Median	6.5E06	7.0E06	6.7E06	6.4E06	6.70E06	6.7E06	6.2E06	6.3E06	6.4E06	7.9E06
Standard Deviation	1.8E05	7.8E04	7.8E04	2.8E04	1.65E05	8.6E04	1.3E04	1.4E05	1.2E05	3.8E05
Minimum	6.1E06	6.9E06	6.5E06	6.3E06	6.52E06	6.5E06	6.2E06	6.1E06	6.3E06	7.3E06
Maximum	6.9E06	7.2E06	6.8E06	6.4E06	7.10E06	6.8E06	6.3E06	6.6E06	6.7E06	8.7E06

Table 7: Least cost solutions for the Farhadgerd network: 1000 function evaluations

	GA	ACO	HS	DE	PSOGSA	ABC	CMAES	SLC	MEIGO	WOA
Mean	3.0E07	3.5E07	3.2E07	3.5E07	3.03E07	3.5E07	3.1E08	3.2E07	3.2E07	3.5E07
Median	2.9E07	3.5E07	3.2E07	3.5E07	3.03E07	3.5E07	5.2E07	3.24E07	3.4E07	3.5E07
Standard Deviation	1.1E06	9.1E05	1.4E06	9.6E05	2.11E06	1.6E06	6.7E08	1.67E06	4.7E06	3.4E06
Minimum	2.8E07	3.4E07	2.9E07	3.3E07	2.77E07	3.2E07	4.7E07	2.87E07	2.2E07	2.9E07
Maximum	3.2E07	3.7E07	3.4E07	3.6E07	3.58E07	3.8E07	2.5.E+9	3.43E07	3.6E07	4.1E07

Table 8: Least cost solutions for the Farhadgerd network: 10000 function evaluations

	GA	ACO	HS	DE	PSOGSA	ABC	CMAES	SLC	MEIGO	WOA
Mean	2.3E07	3.1E07	2.4E07	2.3E07	2.65E07	2.5E07	1.9E07	2.09E07	2.3E07	3.5E07
Median	2.3E07	3.1E07	2.4E07	2.3E07	2.63E07	2.5E07	1.9E07	2.09E07	2.3E07	3.5E07
Standard Deviation	1.0E06	8.5E05	4.7E05	4.4E05	1.29E06	1.1E06	2.1E05	5.65E05	7.9E05	2.9E06
Minimum	2.1E07	2.8E07	2.3E07	2.2E07	2.44E07	2.3E07	1.8E07	2.01E07	2.1E07	2.9E07
Maximum	2.6E07	3.3E07	2.5E07	2.4E07	2.94E07	2.7E07	1.9E07	2.23E07	2.4E07	4.0E07

Table 9: Least cost solutions for the Farhadgerd network: 40000 function evaluations

	GA	ACO	HS	DE	PSOGSA	ABC	CMAES	SLC	MEIGO	WOA
Mean	2.1E07	2.8E07	2.2E07	2.0E07	2.28E07	2.3E07	1.8E07	1.8E07	2.0E07	3.3E07
Median	2.1E07	2.8E07	2.2E07	2.0E07	2.22E07	2.3E07	1.8E07	1.8E07	2.0E07	3.3E07
Standard Deviation	8.9E05	9.2E05	3.3E05	2.7E05	1.8E06	8.3E05	9.2E04	3.0E05	3.4E05	3.2E06
Minimum	1.9E07	2.6E07	2.1E07	2.0E07	2.05E07	2.1E07	1.78E07	1.78E07	1.9E07	2.8E07
Maximum	2.3E07	2.9E07	2.3E07	2.1E07	2.70E07	2.4E07	1.8E07	1.87E07	2.0E07	3.9E07