Feifei Zheng, Angus R. Simpson and Aaron C. Zecchin

A method for assessing the performance of genetic algorithm optimization for water distribution design

Water Distribution Systems Analysis 2010 - Proceedings of the 12th International Conference, WDSA 2010, Tucson, Arizona, United States, 12 Sep-15 Sep 2010 /Kevin E. Lansey, Christopher Y. Choi, Avi Ostfeld and Ian L. Pepper (eds.) pp. 771-785

10.1061/41203(425)72

A method for assessing the performance of genetic algorithm optimization for water distribution system design

by

Zheng, F., Simpson, A.R. and Zecchin, A.C.

Water Distribution System Analysis (WDSA) 2010

A METHOD FOR ASSESSING THE PERFORMANCE OF GENETIC ALGORITHM OPTIMIZATION FOR WATER DISTRIBUTION DESIGN

Feifei Zheng, Angus R. Simpson and Aaron C. Zecchin

School of Civil, Environment and Mining Engineering, University of Adelaide
Adelaide, Australia
fzheng@civeng.adelaide.edu.au

Abstract

The paper proposes a new methodology for assessing the effectiveness of GA parameters in consistently finding similar low cost solutions over a broad range of different starting random number seeds. The method involves testing the parameters of probabilities of crossover (for various crossover types) and probabilities of bitwise mutation with 1000 different random number seeds. In addition, this methodology allows different varieties of GA to be compared so that the methodology with the best performance can be determined. The proposed methodology has been verified for its effectiveness on the previously published benchmark network New York Tunnels Problem.

Keywords
Genetic algorithm, water distribution systems, crossover scheme, probability of crossover, probability of mutation

1. INTRODUCTION

The cost of construction and operation for a water distribution system (WDS) is a significant part of a municipal budget. As a result, the optimization of WDSs has attracted the interest of many researchers and has frequently been considered in the literature over the years. The usual objective of optimization for WDSs is to minimize the capital cost (including construction costs) plus the operating costs subject to a set of constraints which are mathematically nonconvex and nonlinear. Traditional techniques, such as linear programming and nonlinear programming, have been used previously to optimize the design of WDSs (Alperovits and Shamir 1977; Quindry et al. 1981; Murtagh and Saunders 1987). However, it is problematic to find optimal designs with these methods.

Evolutionary algorithms have been introduced over the last 15 years to determine the least-cost design of water distribution systems. Among them, genetic algorithm (GA) optimization has gained popularity in terms of optimal design of the water distribution systems because of its robustness and search ability (Simpson et al. 1994; Savic and Walters 1997). Genetic algorithms mimic evolutionary principles to create an optimisation search (Holland 1975). In terms of optimal design of WDSs, GA optimization has an advantage over traditional optimization techniques in that it deals more easily with discrete commercially available pipe sizes and the GA search has been proven to perform efficiently compared with traditional optimisation techniques (Simpson et. al 1994). Based on the standard genetic algorithm (SGA), a number of improved GA optimisation techniques have been proposed by researchers. A GA incorporating a creeping mutation operator has been shown to more efficient (Dandy et al. 1996). Wu et. al (2001) introduced a fast messy genetic algorithm to deal with the optimization of the water network. Vairavamoorthy and Ali (2005) used a pipe index method to modify the GA-based pipe optimization.

The objective of this paper is to present a new methodology for assessing the impacts of different GA parameters on the performance of GAs for the optimisation of water distribution systems. The method involves testing the parameters of probabilities of crossover for various crossover types and probabilities of bitwise mutation with 1000 different random number seeds. One pipe network design case study is presented in the paper in order to demonstrate the new methodology. The obtained data is plotted and tabulated to enable the assessments of the effectiveness of the GA parameters combinations.

2. GENETIC ALGORITHM REPRESENTATION AND OPERATORS

2.1 Coding

The decision variables describing trial solutions of a pipe network design are represented by a uniquely coded string in GA optimisation of water distribution system design. Coding schemes include binary coding, Gray coding and integer coding. Binary coding has been extensively used. A typical binary coding mapping is shown in the second column of Table 1. Pipe sizes are coded by different substrings, such as substring 000 representing pipe diameter size of 12 inches. A Hamming cliff effect is produced when the binary coding of two adjacent values differs in each of their bits, such as substrings 011 and 100 in the second column of Table 1. The Hamming cliff may lead the GA to a non-global optima (Goldberg 1991). A Gray coding scheme was introduced by Dandy et al. (1996) to avoid the Hamming cliff. The Gray codings for the decision variables are given in the third column of Table 1. It is noticed that the Gray coding representation is such that adjacent decision variables differ only 1 bit in their corresponding coded substrings. Another issue that needs to be addressed is the possible redundant states of binary coding and Gray coding. For example, there is no corresponding pipe size for the substring of 111 in binary coding and Grady coding in Table 1. However, the substring of 111 will be inevitably generated while applying the crossover and mutation operators. Vairavamoorthy and Ali (2000) applied an integer coding method in genetic algorithms to avoid the problem of redundant states often found when using binary coding or Gray coding. A typical example of integer coding is given in the fourth column of Table 1. The integer values from 1 to 6 are used to represent the 6 diameters.

Table 1. A typical example of different coding schemes

Diameters (inches)	Corresponding Coded substrings		
	Binary	Gray	Integer
12	000	000	1
16	001	001	2
20	010	011	3
24	011	010	4
30	100	110	5
40	101	100	6

2.2 Selection

The primary objective of the selection operator is to replicate the good solutions and eliminate bad solutions in a population while keeping the population size constant. The selection operator makes more copies of the good solutions at the expense of bad solutions, but no new solutions are created. A number of selection methods may be used in a GA including tournament selection, proportionate selection and ranking selection. It has been demonstrated that the tournament selection has better or equivalent convergence and computational overhead when compared to proportionate selection and ranking selection (Goldberg and Deb 1991).

2.3 Crossover

In the crossover operator, two strings are randomly chosen to exchange some portion of their strings in order to create two new strings. There exists a number of crossover schemes including one point crossover, two point crossover and uniform crossover (Deb 2001). Two integer coded strings are used to illustrate the different crossover schemes as shown in Figure. 1.

Parents										Offspring								
1	2	4	1	3	1	5	3	6	→	1	2	4	1	1	6	4	5	2
3	5	1	3	1	6	4	5	2		3	5	1	3	3	1	5	3	6

(a) One-point crossover

Parents										Offspring								
1	2	4	1	3	1	5	3	6	→	1	2	1	3	1	6	4	3	6
3	5	1	3	1	6	4	5	2		3	5	4	1	3	1	5	5	2

(b) Two-point crossover

Parents										Offspring								
1	2	4	1	3	1	5	3	6	→	1	5	4	1	1	6	5	3	2
3	5	1	3	1	6	4	5	2		3	2	1	3	3	1	4	5	6

(c) Uniform crossover (Each pair of bits is considered for crossover)

Figure 1. An example of different crossover schemes

In one-point crossover, two strings are picked from the population and denoted as parents. A site along the length of the string is chosen randomly and all the bits on the right side of cross site are exchanged between the two strings. Two new strings are then created called offspring. As shown in Figure 1(a), the fifth bit is chosen to be the crossover site and all bits on the right side of the cross site (given in light gray) are exchanged. If two sites are randomly selected along the string and all the bits between the two sites are exchanged, this method is called two-point crossover. From the Fig. 1(b), the bits between the third and the seventh (given in light gray) are exchanged to create two new solutions. In uniform crossover, each pair of bits is considered in turn in the parent strings. Exchange between the strings occurs with a probability, which is usually 0.5. As shown in Fig. 1(c), the second, fifth, sixth and ninth bits (given in light gray) are exchanged between the parent strings. In addition to the crossover schemes, the probability of crossover (P_c) has a significant effect on the performance of GAs and a relatively high probability of crossover (P_c=0.6 to 1.0) has been recommended (Goldberg 1989).

2.4 Mutation

Mutation is needed to maintain the diversity in the population and alters a string locally to hopefully create a better string. Bitwise mutation has been extensively used in a GA, where each bit has a specified probability of mutation (P_m). The mutation may result in a pipe size that considerably different compared to the original one. A relatively low probability of bitwise mutation (P_m=0.01) is employed for most GA runs. Dandy et. al (1996) implemented a GA with probability of mutation being 0.01 to optimize the New York Tunnels Problem (NYTP). Vairavamoorthy and Ali (2000) used a GA with probability of mutation being 0.01 to optimize the New York Tunnels Problem (NYTP) and the Hanoi Problem (HP). Goldberg and Koza (1990) proposed a probability of mutation P_m could be defined as $p_m=1/str$ where p_m is the mutation parameter and *str* is the length of the string.

3. CASE STUDY

The various types of GA implementation considered in this paper have been coded in C++ combined with the EPANET2 network solver to computer the hydraulic balance. The New York Tunnels Problem (NYTP) is used to test the effectiveness of the new methodology.

A schematic of the NYTP system is given in Fig. 2. The network is composed of 21 existing tunnels and 20 nodes fed by the fixed-head reservoir. All the details of this network including the head constraints, pipe costs, and water demands can be found in the literature (Dandy et al.1996). The objective is to determine which pipes in parallel with the existing system give the least-cost solution while satisfying the minimum head requirement at all nodes. There are 15 pipe diameters that can be selected for NYTP and these are {36, 48, 60, 72, 84, 96, 108, 120, 132, 144, 156, 168, 180, 192, 204} inches. In addition, a zero pipe size provides a total of 16 options (15 actual pipe diameters plus a zero pipe size) for each link.

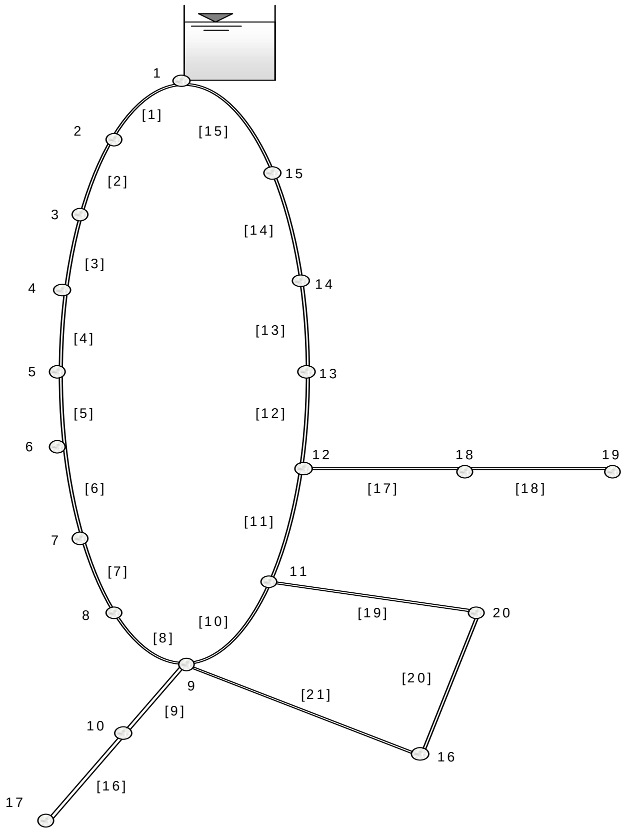


Figure 2. Layout of the NYTP network

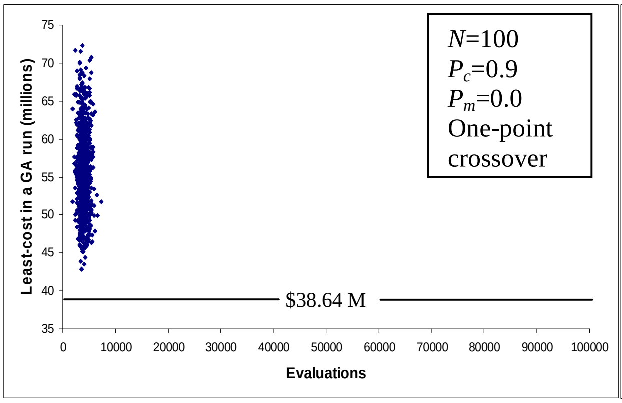
3.1 Test 1: crossover schemes

The NYTP case study has been used to test the impacts of three different crossover schemes on the performance of the GAs. The three schemes included one-point crossover, two-point crossover and uniform crossover. As can be seen from Table 2, for the case study, two groups have been designed to test the performance of GAs with three different crossover schemes. The first group consisting of GA1, GA2 and GA3 was used to test the performance of non-mutation GAs (P_m=0.0) with different crossover schemes. The second group consisting of GA4, GA5 and GA6 was designed to test the performance of GAs with three different crossover schemes, but with mutation (P_m=0.03).

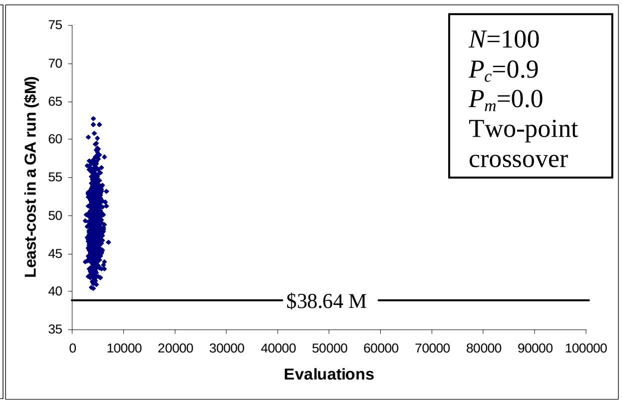
Table 2. GA Parameters

GA Parameters	GA parameters applied to the NYTP					
	Group 1			Group 2		
	GA1	GA2	GA3	GA4	GA5	GA6
Crossover scheme	One-point	Two-point	Uniform	One-point	Two-point	Uniform
Probability of crossover (P_c)	0.9	0.9	0.9	0.5	0.5	0.5
Probability of mutation (P_m)	0	0	0	0.03	0.03	0.03

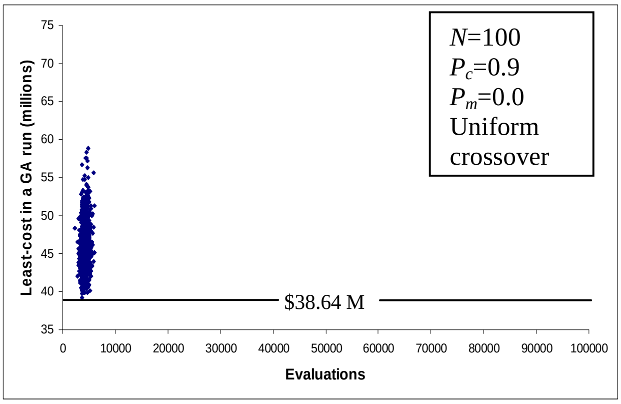
In order to compare the performance of GAs with different crossover schemes in each group, all the parameters of GAs in the same group were set the same except the crossover scheme. For example, all 6 GAs used the integer coding and tournament selection. The population size and maximum number of evaluations for each GA were identical, with values of 100 and 100,000, respectively. For each GA application, 1000 runs with different random number seeds were implemented. The results of different GAs runs applied to the NYTP are shown in Fig. 3.



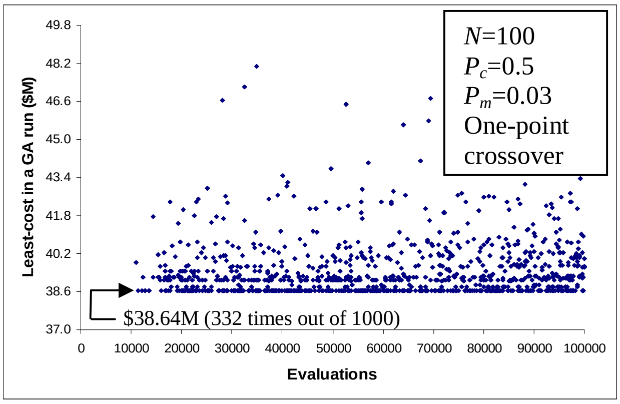
(a) Results of GA1 runs applied to the NYTP



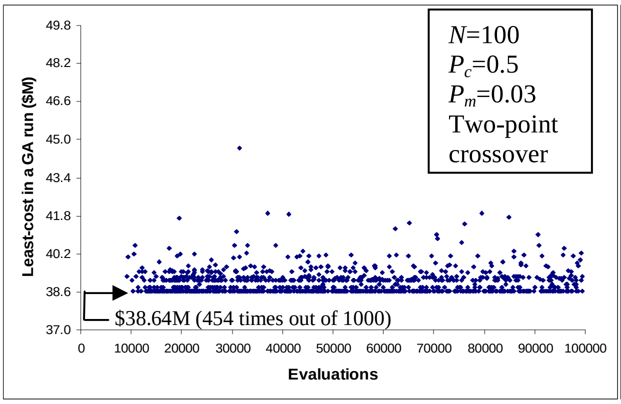
(b) Results of GA2 runs applied to the NYTP



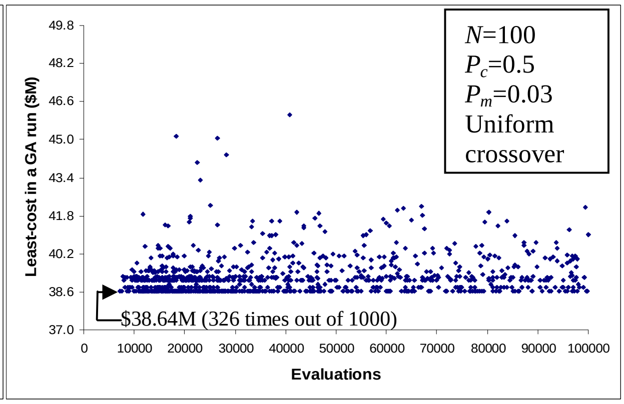
(c) Results of GA3 runs applied to the NYTP



(d) Results of GA4 runs applied to the NYTP



(e) Results of GA5 runs applied to the NYTP



(f) Results of GA6 runs applied to the NYTP

Figure 3**.** Results of different GAs applied to the NYTP

As can be seen from Table 3, the known-least-cost solution for NYTP is $38.64 million found first by Maier et al. (2003) using Ant Colony Optimization technique, compared with $38.80 million found by Dandy et al. (1996) and Cunha and Sousa (2001), and $39.20 million found by Morgan and Goulter (1985). The solution found by Savic and Walter (1997) with a value of $37.13 is infeasible when determined by EPANET2. The current best solution has also been obtained in this study.

Table 3 Solutions of the NYTP

Links/ Nodes	Network design (diameters in inches)			
	Savic and Walters(1997)	*Morgan and Goulter(1985)*	*Dandy et al.(1996)* and *Cunha and Sousa (2001)*	*Maier et al. (2003)* and *Present work*
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	108	144	0	144
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0
11	0	0	0	0
12	0	0	0	0
13	0	0	0	0
14	0	0	0	0
15	0	0	120	0
16	96	96	96	96
17	96	96	96	96
18	84	84	84	84
19	72	60	72	72
20	0	0	0	0
21	72	84	72	72
Total Cost ($M)	37.13*	39.2	38.80	38.64

*=an infeasible solution determined by EPANET2.

Note: The hydraulic head is calculated by EPANET2 with an accuracy of 0.000001.

Fig. 3(a), Fig. 3(b) and Fig. 3(c) show scatter plots of solutions of 1000 different starting random number seeds including GA1 (N=100, P_c=0.9, P_m=0.0, one-point crossover), GA2 (N=100, P_c=0.9, P_m=0.0, two-point crossover) and GA3 (N=100, P_c=0.9, P_m=0.0, uniform crossover) runs for the NYTP, respectively. It can be seen from the three figures, that the solutions have converged on a high cost value and there were no improvements on the solutions after relatively few evaluations. It has been demonstrated that the GA with a mutation rate of zero easily gets stuck at a local solution. The solutions of GA3 are closer to known-least-cost solution value of $38.64 million compared to that of GA1 and GA2.

Table 4 shows a comparison of the performance of different GAs applied to the NYTP. The GA parameters are outlined in the first column. The total number of solutions in different cost ranges is given in the second to the ninth columns, respectively. For example, the number in the second column represents the number of times out of 1000 runs the solution with a cost of $38.64 million was found. The average cost and mean number of evaluations for 1000 GA runs are given tenth and eleventh columns of Table 4, respectively.

As indicated in Table 4, the average cost of the 1000 GA3 runs was $45.952 million compared with $55.573 million of the 1000 GA1 runs and $48.471 million of the 1000 GA2 runs. The GA3 (with uniform crossover) was able to find lower cost solutions than GA1 and GA2. For example, solutions between $40.00 and $42.00 million were found 72 times out of 1000 runs for GA3, compared with zero for GA1 and 19 for GA2. Thus, for a mutation rate of zero, the GA with uniform crossover (GA3) performed the best and the GA with one-point crossover (GA1) performed the worst in terms of being able to find lower cost solutions for the NYTP. The mean number of evaluations for 1000 GA1, GA2 and GA3 runs were similar with a value of 3914, 4360 and 4228 evaluations respectively.

Table 4. Results of different GAs (N=100) for optimizing the NYTP

GAs	Total number of solutions in different cost ranges ($M)							Average cost for 1000 runs($M)	Average No. of evaluations for 1000 runs
	38.64	38.80	38.80-39.50	39.50-40.00	40.00-42.00	42.00-44.00	>44.00		
GA1(P_c=0.9, P_m=0.0, One-point crossover)	0	0	0	0	0	4	996	55.573	3914
GA2(P_c=0.9, P_m=0.0, Two-point crossover)	0	0	0	0	19	67	914	48.471	4360
GA3(P_c=0.9, P_m=0.0, Uniform crossover)	0	0	1	5	72	214	708	45.925	4228
GA4(P_c=0.5, P_m=0.03, One-point crossover)	332	63	285	85	171	48	16	39.534	60039
GA5(P_c=0.5, P_m=0.03, Two-point crossover)	454	103	332	61	49	1	0	38.998	49950
GA6(P_c=0.5, P_m=0.03, Uniform crossover)	326	110	344	83	126	10	1	39.234	40467

Fig. 3(d), Fig. 3(e) and Fig. 3(f) show scatter plots of solutions of 1000 starting random number seeds including GA4 (N=100, P_c=0.5, P_m=0.03, one-point crossover), GA5 (N=100, P_c=0.5, P_m=0.03, two-point crossover) and GA6 (N=100, P_c=0.5, P_m=0.03, uniform crossover) runs for the NYTP, respectively. The solutions of these three GAs are horizontally distributed when compared with that of GA1, GA2 and GA3. As can be seen from Table 4, 332 times out of 1000 GA4 runs, 454 times out of 1000 GA5 runs and 326 times out of 1000 GA6 runs were able to find known-least-cost solution with a cost of $38.64 million. The GA4 performed the worst as its average cost and evaluations for 1000 runs were both larger than the GA5 and GA6. The average cost of 1000 GA5 was $38.998 million which was slightly less than $39.234 million obtained by 1000 GA6 runs. However, The GA6 was able to converge more quickly with average evaluations of 1000 runs being 40,467 compared with 49,950 of 1000 GA5 runs.

3.2 Conclusion of Test 1

The impact of different crossover schemes on the performance of GAs has been assessed by a new methodology with a broad range of different random number seeds. The results of the case study have shown that, with a zero mutation rate (P_m=0.0), the GA with uniform crossover performed the best and the GA with one-point crossover performed the worst.

With a mutation operator (P_m>0.0), the results have also shown that the GA with one-point crossover performed the worst. The GA with two-point crossover GA slightly outperformed the GA with uniform crossover in terms of being able to find the known-least-cost solutions. However, the GA the uniform crossover had the advantage in efficiency, as shown in Table 3.

In a conclusion, the GA with one-point crossover has been demonstrated, in this study, to perform the worst and the GA with uniform crossover is preferable in terms of optimal design of water distribution systems.

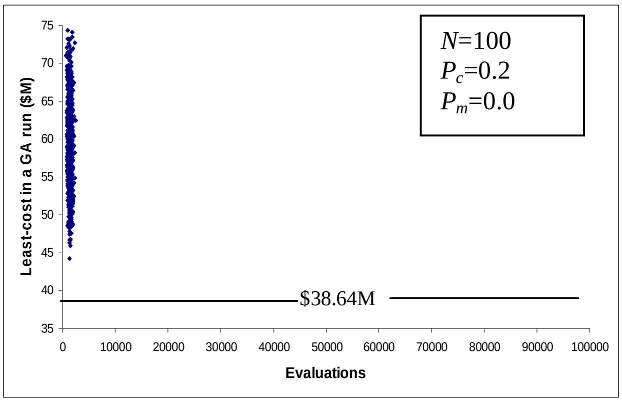
3.3 Test 2: probability of crossover

Commonly, a relatively high probability of crossover (P_c=0.6 to 1.0) has previously been recommended for use in a GA (Goldberg 1989). In this study, the new methodology has been used to analyse the impacts of different probabilities of crossover on the performance of GAs. The proposed methodology has been performed on the NYTP case study. All the GAs in this test used the integer coding, tournament selection, two-point crossover and worked with a population size of 100. The maximum allowable number of evaluations for all the GA applications in this study was 100,000. The other parameters for the NYTP case study are given in Table 5.

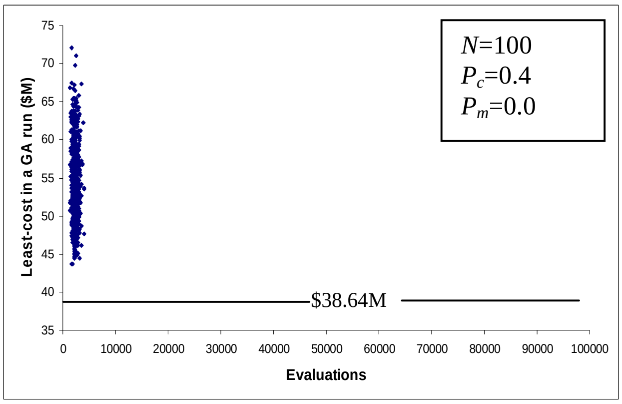
As seen in Table 5, the GA7 to 11 have been used to test performance of zero mutation rate (P_m=0) GAs applied to the NYTP with probability of crossovers of 0.2, 0.4, 0.6, 0.8 and 1.0 respectively. The GA12 to 16 have been used to test the performance of GAs applied to the NYTP with probability of crossover being 0.2, 0.4, 0.6, 0.8 and 1.0, and with the same probability of mutation being 0.03. The results of different GA applications for the NYTP are shown in Fig. 4.

Table 5 GA parameters applied to the NYTP

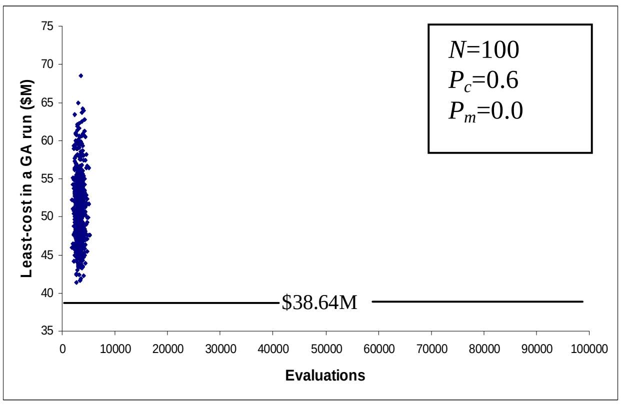
GA Parameters	GA7	GA8	GA9	GA10	GA11	GA12	GA13	GA14	GA15	GA16
Probability of crossover (p_c)	0.2	0.4	0.6	0.8	1.0	0.2	0.4	0.6	0.8	1.0
Probability of mutation (p_m)	0.0	0.0	0.0	0.0	0.0	0.03	0.03	0.03	0.03	0.03



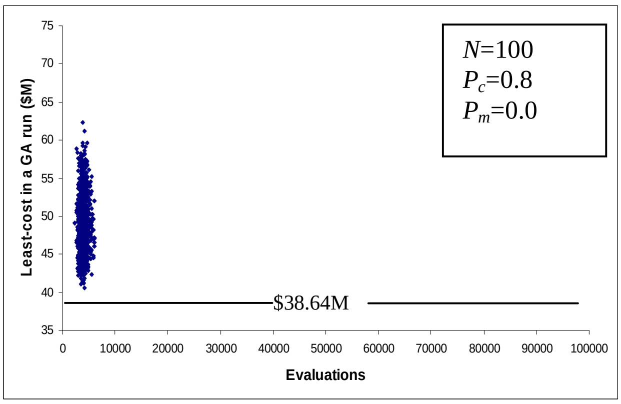
(a) Results of GA7 runs applied to the NYTP



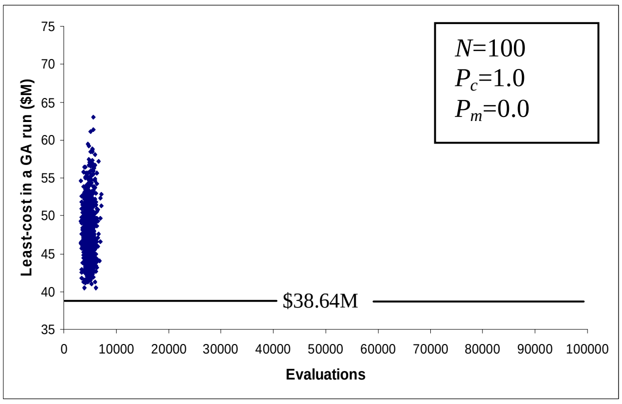
(b) Results of GA8 runs applied to the NYTP



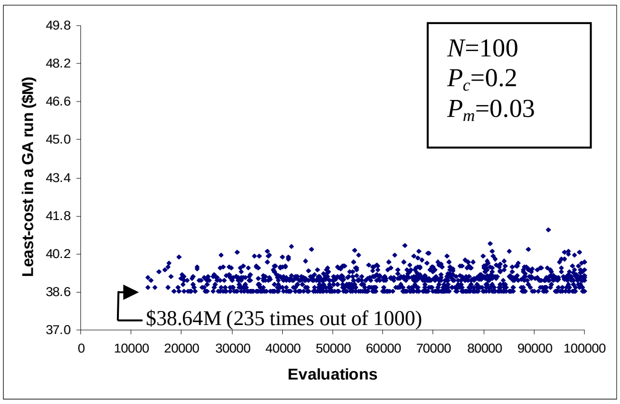
(c) Results of GA9 runs applied to the NYTP



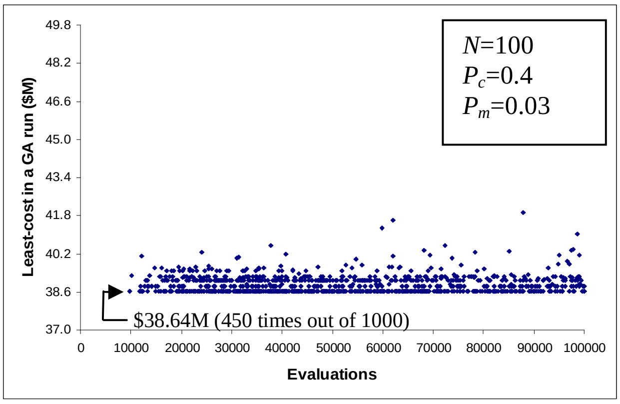
(d) Results of GA10 runs applied to the NYTP



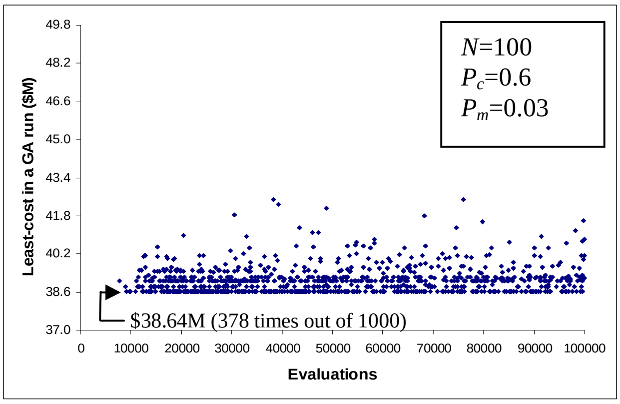
(e) Results of GA11 runs applied to the NYTP



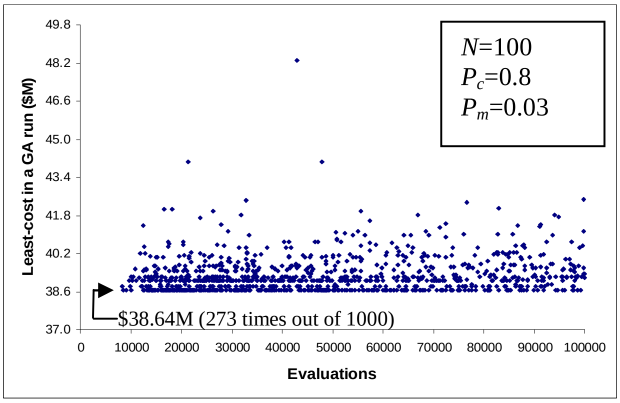
(f) Results of GA12 runs applied to the NYTP



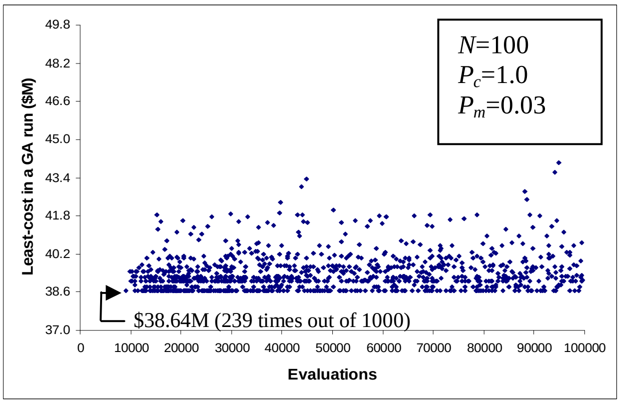
(g) Results of GA13 runs applied to the NYTP



(h) Results of GA14 runs applied to the NYTP



(i) Results of GA15 runs applied to the NYTP



(j) Results of GA16 runs applied to the NYTP

Figure 4. Results of GAs with different probabilities of crossover for optimizing the NYTP

The results of scatter plots for 1000 different starting random number seeds including GA7 (N=100, P_c=0.2, P_m=0.0), GA8 (N=100, P_c=0.4, P_m=0.0), GA9 (N=100, P_c=0.6, P_m=0.0), GA10 (N=100, P_c=0.8, P_m=0.0) and GA11 (N=100, P_c=1.0, P_m=0.0) runs applied to the NYTP are shown in Fig. 4(a) to (e), respectively. As can be seen from these figures, the solutions of GA7, GA8, GA9, GA10 and GA11 were clustered and the solutions of 1000 GA runs as the probability of crossover increased from rate 0.2 to 1.0 moved towards lower cost and higher evaluations. As shown in Table 9, with a zero mutation rate

(P_m=0.0), the GA with higher probability of crossover outperformed the GA with lower probability of crossover in terms of being able to find lower cost solutions for the NYTP. For example, solutions below $44.00 million were found 110 times out of 1000 runs for GA11, compared with zero for GA7, twice for GA8, 23 times for GA9 and 64 times for GA10. It is observed from Table 6, for a mutation rate of zero (p_m=0.0), the average cost of 1000 GA runs was reduced as the probability of crossover increased from 0.2 to 1.0. This demonstrated that, with a mutation rate of zero (P_m=0.0), the GA with a higher crossover probability such as P_c=1.0 outperformed the GA with a lower crossover probability. As can be seen from the last column of Table 6, the GA with a lower crossover probability converged with fewer evaluations. As an example, for GA7 the number of average evaluations required by the 1000 runs was only 1426, indicating its premature convergence.

Table 6. Results of different GAs (N=100) applied to the NYTP

GAs	Total number of solutions in different cost ranges ($M)							Average cost for 1000 runs($M)	Average No. of evaluations for 1000 runs
	38.64	38.80	38.80-39.50	39.50-40.00	40.00-42.00	42.00-44.00	>44.00		
GA7(P_c=0.2, P_m=0.0)	0	0	0	0	0	0	1000	59.245	1426
GA8(P_c=0.4, P_m=0.0)	0	0	0	0	0	2	998	53.961	2356
GA9(P_c=0.6, P_m=0.0)	0	0	0	0	3	20	977	50.810	3203
GA10(P_c=0.8, P_m=0.0)	0	0	0	0	10	54	936	49.180	4006
GA11(P_c=1.0, P_m=0.0)	0	0	0	0	24	86	890	47.970	4739
GA12(P_c=0.2, P_m=0.03)	235	141	459	144	1	0	0	39.095	63006
GA13(P_c=0.4, P_m=0.03)	450	150	343	36	21	0	0	38.916	51925
GA14(P_c=0.6, P_m=0.03)	378	119	353	77	69	4	0	39.064	49601
GA15(P_c=0.8, P_m=0.03)	273	99	364	120	135	6	3	39.272	48277
GA16(P_c=1.0, P_m=0.03)	239	69	364	162	129	36	1	39.368	47762

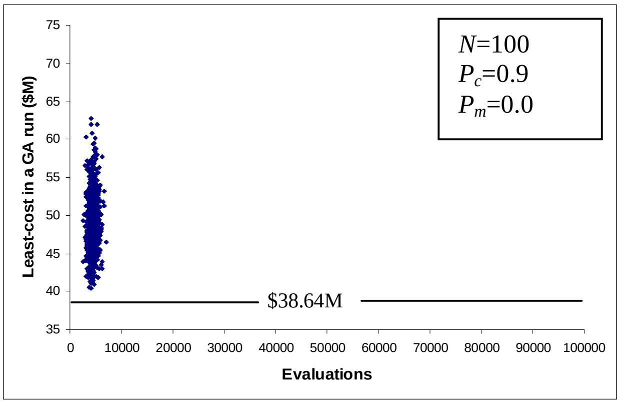
Fig. 4(f) to (j) show the scatter plots of solutions of 1000 different starting random number seeds including GA12 (N=100, P_c=0.2, P_m=0.03), GA13 (N=100, P_c=0.4, P_m=0.03), GA14 (N=100, P_c=0.6, P_m=0.03), GA15 (N=100, P_c=0.8, P_m=0.03) and GA16 (N=100, P_c=1.0, P_m=0.03) runs for the NYTP, respectively. The solutions tend to be horizontally distributed in these figures. It is seen from Table 6, comparing the average cost for 1000 runs, the GA13 run was the lowest. In addition, GA13 was able to find the known-least-cost solution 450 times out of 1000 runs, which was a higher number of times than GA12, GA14, GA15 and GA16. The average evaluations of 1000 runs for GA 12 to GA22 were similar as shown in Table 9. Thus, this experimentation indicated that with a mutation rate such as P_m=0.03, the GA with a crossover probability of 0.4 performed the best when applied to the NYTP.

3.4 Conclusion of Test 2

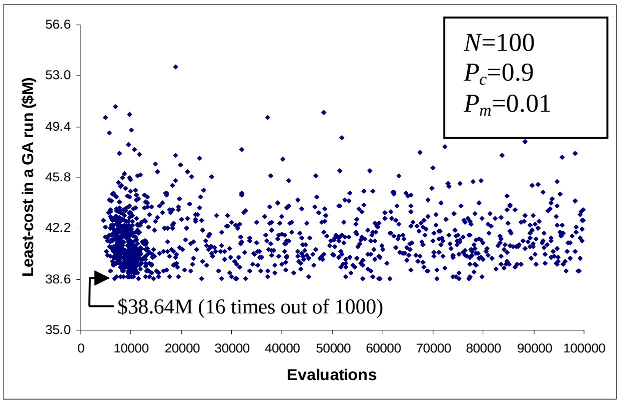
The impact of different probabilities of crossover on the performance of GAs has been assessed based on a new methodology using a broad range of different random number seeds. From the results of the case study, it has revealed that, for a zero mutation rate (P_m=0.0), a GA with a relatively high probability of crossover was preferable. However, with mutation, the GA with a relatively lower crossover probability performed better than the GA with a higher probability of crossover. In this study, among the GA with crossover probabilities of 0.2, 0.4, 0.6, 0.8, 1.0, the GA with a crossover probability of 0.4 performed the best when applied to the NYTP.

3.5 Test 3: Probability of bitwise mutation

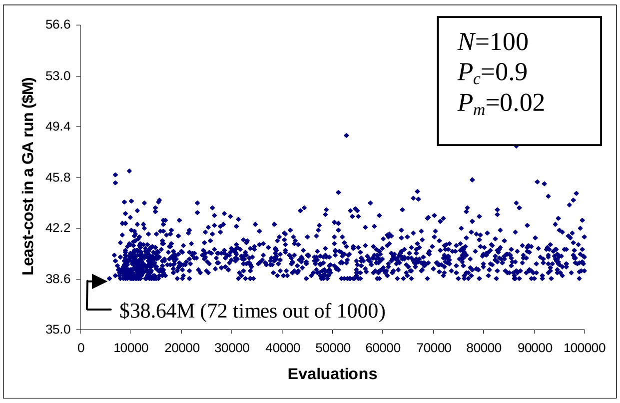
The proposed methodology has also been used to test the impacts of different probabilities of mutation on the performance of GAs applied to the NYTP case study. All the GAs in this test used the integer coding, tournament selection and two-point crossover (P_c=0.9). The population size was 100 and the maximum number of evaluations was 100,000 for each GA application in this study. Mutation probabilities of 0.00, 0.01, 0.02, 0.03, 0.04 and 0.05 employed in GA17, GA18, GA19, GA20, GA21 and GA 22 respectively have been used to test the impacts of different bitwise mutation probabilities on the performance of GAs applied to the NYTP. The results of GAs applied to the NYTP case study are presented in Fig. 5.



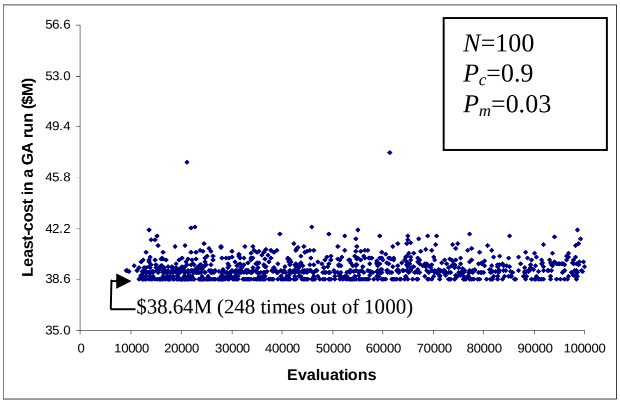
(a) Results of GA17 runs applied to the NYTP



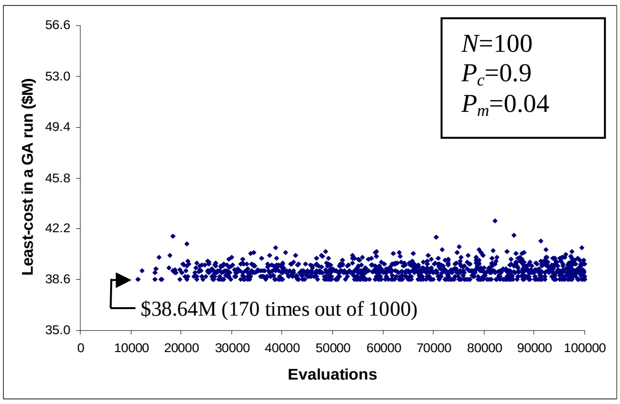
(b) Results of GA18 runs applied to the NYTP



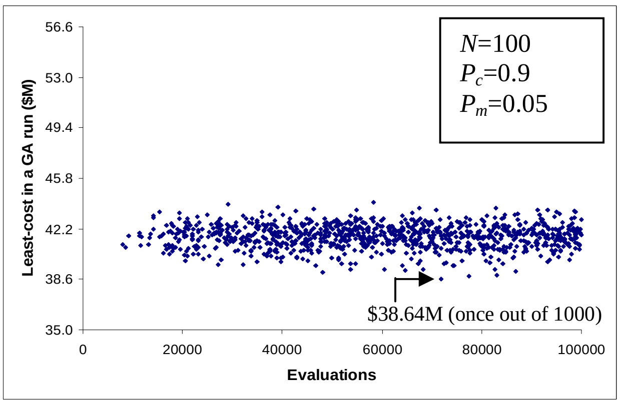
(c) Results of GA19 runs applied to the NYTP



(d) Results of GA20 runs applied to the NYTP



(e) Results of GA21 runs applied to the NYTP



(f) Results of GA22 runs applied to the NYTP

Figure 5. Results of GAs with different probabilities of mutation for the NYTP

Table 7. Results of different GAs (N=100, P_c=0.9) for optimizing the NYTP

GAs	Total number of solutions in different cost ranges ($M)							Average cost for 1000 runs($M)	Average No. of evaluations for the 1000 runs
	38.64	38.80	38.80-39.50	39.50-40.00	40.00-42.00	42.00-44.00	>44.00		
GA17(P_m=0.0)	0	0	0	0	19	67	914	48.470	4360
GA18(P_m=0.01)	16	22	70	93	473	224	102	41.155	32904
GA19(P_m=0.02)	72	53	181	185	413	78	18	40.220	44697
GA20(P_m=0.03)	248	75	349	152	169	5	2	39.362	48560
GA21(P_m=0.04)	170	100	453	195	81	1	0	39.256	64904
GA22(P_m=0.05)	1	1	9	26	623	339	1	41.656	59620

Fig. 5(a) to (f) show the scatter plots of solutions of 1000 different starting random number seeds including GA17 (N=100, P_c=0.9, P_m=0.0), GA18 (N=100, P_c=0.9, P_m=0.01), GA19 (N=100, P_c=0.9, P_m=0.02), GA20 (N=100, P_c=0.9, P_m=0.03), GA21 (N=100, P_c=0.9, P_m=0.04) and GA22 (N=100, P_c=0.9, P_m=0.05) runs for the NYTP, respectively. As can be seen from these figures, the solutions of GA17 were well above the known-least-cost solution with a value of $38.64 million, compared with other GAs. From Fig. 5 (b) and (c), it is observed that a cluster of solutions converged with few evaluations, indicating the premature convergence of GA 18 and GA19. The solutions of 1000 GA22 run were much further away from the known-least-cost solution as can be seen from Fig. 5(f).

As shown in Table 7, run GA17 (P_m=0.0) performed the worst with its average cost of 1000 runs being $48.470 million. The performance of the GA improved significantly as the mutation rate was increased from 0.01 to 0.05. For run GA17 (P_m=0.0), it was noted that the number of average evaluations of 1000 runs was only 4360 which was considerately less than other GAs. This demonstrates that a GA with a mutation rate of zero is highly likely to converge on a local optimal solution. On average, the cost of 1000 runs found by GA20 with a value of $39.362 million was slightly higher than GA21 with a value of $39.256 million, but significantly lower than other GAs. It is found that in Table 7, GA20 was able to converge faster than GA21 when comparing the average evaluations for 1000 GA runs. Run GA20 was able to find the known-least-cost solution 248 times out of 1000 runs, compared with 16 times out of 1000 GA18 runs, 72 times out of 1000 GA19 runs, 170 times out of 1000 GA21 runs and only once out of 1000 GA22 (P_m=0.05) runs. Thus, the GA with a mutation probability of 0.03 (GA20) applied to the NYTP performed the best in this study.

3.6 Conclusion of Test 3

The impact of different probabilities of mutation on the performance of GAs has been assessed by the new methodology for a broad range of different random number seeds. From the results of the case study, it has been shown that a GA with probability of mutation being 0.03 (P_m=0.03) was preferable for the NYTP case study. A GA with a relatively high probability of mutation such as a GA with a mutation rate of 0.05 for the NYTP was not effective. While a GA with too low probability of mutation such as a GA with a mutation rate of 0.01 for the NYTP has exhibited premature convergence. A GA with an appropriately selected mutation probability improved the performance considerably when compared with a GA using a mutation rate of zero, showing the importance of the mutation operator.

4. SUMMARY AND CONCLUSIONS

A proposed new methodology has been used to assess the performance of genetic algorithm optimisation for water distributions systems with 1000 different random number seeds. All the results have been plotted and tabulated to enable the comparison. The new method has involved testing the parameters of crossover schemes, probability of crossover and probability of bitwise mutation on performance of GAs for optimizing a case study, the New York Tunnels Problem. The conclusions of this study are given as following:

1. The test on the performance of different crossover schemes has been implemented on the GAs with a mutation of zero and also for GAs with a mutation rate of 0.03. Both results have indicated that a GA with one-point crossover was least effective and uniform crossover was preferable compared with one-point crossover or two-point crossover.

2. Experimentation on the performance of various probabilities of crossover involved in GAs has revealed that a relatively high probability of crossover (P_c=0.8 to 1.0) was effective in a GA with a mutation rate of zero. However, a relatively low probability of crossover (P_c=0.4 to 0.5) has been demonstrated to be more effective in a GA with mutation. For example, with the same mutation probability of 0.03, a GA with crossover probability of 0.4 has been shown to be more effective than GAs with other crossover probabilities such as P_c=0.8 or 1.0 in terms of optimizing the NYTP. This result differs with that found by Goldberg (1989), in which a relatively high crossover rate (P_c=0.6 to 1.0) was recommended

3. The performance of different bitwise mutation probabilities employed in GAs has been investigated in this paper. It was found from results that a GA with a mutation rate of zero performed the worst, as its solutions were all well higher in cost than the known-least-cost solution of the NYTP case study. The performance of GAs with mutation probabilities of 0.01, 0.02, 0.03, 0.04 and 0.05 improved significantly compared with a GA with a mutation rate of zero. This demonstrated that the GA search was dominated by the mutation operator. A GA with a mutation rate of 0.03 has been demonstrated to be preferable as it was able to find the known-least-cost solution for the NYTP case study more frequently. A GA with a too low a probability of mutation such as P_m=0.01 or with a too high probability of mutation such as P_m=0.05 have been both shown to be less effective. In terms of determining an appropriate mutation rate for the NYTP case study, the mutation rate of approximately 0.05 (1/21) was suggested by Goldberg and Koza (1990) as the length of string was 21 for the case study, and a mutation rate of 0.01 was used in most GA-based optimisation model (Dandy et. al 1996; Vairavamoorthy and Ali 2000). However, in this study a moderate probability of mutation with value of 0.03 has been clearly demonstrated to be more effective for the NYTP case study.

Further work is required to determine if the results presented in this study extend to general case of WDS optimisation.

References

Alperovits, E., and Shamir, U. (1977). "Design of Water Distribution Systems." *Water Resources Research*, 13(6), 885-900.

Cunha, M. C., and Sousa, J (2001). "Hydraulic Infrastructures Design Using Simulated Annealing." *Journal of Infrastructure Systems*, 7(1), 32-38.

Dandy, G. C., Simpson, A. R., and Murphy, L. J. (1996). "An improved genetic algorithm for pipe network optimization." *Water Resources Research*, 32(2), 449-457.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithm*, Wiley, London.

Eusuff. M. M., and Lansey. K. E. (2003). "Optimisation of water distribution network design using shuffled frog leaping algorithm." *J. Water Resources Planning and Management*, 129(3), 210-225.

Fujiwara, O., and Khang, D. B. (1990). "A two-phase decomposition method for optimal design of looped water distribution networks." *Water Resource Res*., 23(6), 977-982.

Goldberg, D, E., and Kuo, C. H. (1987). "Genetic algorithm in pipeline optimization." *J. Comput. Civ. Eng*., 1(2), 128-141.

Goldberg, D, E., and Koza, J. R. (1990). "Genetic algorithm in search, optimization and machine learning." *Workshop notes, Computer Science Department, Stanford University,* August 6-10.

Goldberg D. E. (1991). “Real-coded genetic algorithms, virtual alphabets, and blocking.” *Complex Systems*, 5(2), 139-167.

Goldberg D. E., and Deb, K (1991). “A comparison of selection schemes used in genetic algorithms.” In *foundations of genetic algorithm 1 (FOGA-1),* pp. 69-93.

Goldberg, D. E (1989). Genetic algorithm in search, optimization and machine learning. Reading, MA: Addison-Wesley.

Maier, H. R., et al. (2003). “Ant colony optimization for the design of water distribution systems.” *J. Water Resources Planning and Management*, 129(3), 200-209.

Murtagh, B. A., and Saunders, M. A. (1987). MINOS 5.1 *Users guide*. Systems Optimization Lab, Dept of Operational Research, Stanford University, Stanford, Calif.

Quindry, G. E., and Liebman, J. C. (1981). “Optimization of Looped Water Distribution Systems.” *J. Environmental Engineering, Div. Am. Soc.Civ. Eng.*, 107 (EE4).

Simpson, A. R., Dandy, G. C., and Murphy, L. J. (1994). “Genetic algorithms compared to other techniques for pipe optimization.” *J. Water Resources Planning and Management*, 120(4), 423-443.

Savic, D. A., and Waters G.A. (1997). “Genetic Algorithms for least-cost design of water distribution networks.” *J. Water Resource Planning and Management*, 123(2), 67-77.

Vairavamoorthy, K., and Ali, M. E. (2000). “Optimal Design of water distribution systems using genetic algorithms.” *Computer-aided Civil and Infrastructure Engineering*, 15(2), 374-382.

Vairavamoorthy, K., and Ali, M. E. (2005). “Pipe index vector: A method to improve genetic-algorithm-based pipe optimization.” *J. Hydraul. Eng.*, 131(12), 1117-1125.

Wu, Z. Y., Boulos, P. F., Orr, C. H., and Ro, J.J. (2001). “Using genetic algorithms to rehabilitate distribution systems.” *J.,Am.Water Works Assoc.*, 93(11),74-85.

Zecchin, A. C., Simpson, A. R., and Maier, H. R. (2005). “Application of two ant colony optimization algorithms to water distribution system optimization.” *Mathematical and Computer Modelling*, 44,451-668.