A Review of Genetic Algorithm Optimization: Operations and Applications to Water Pipeline Systems

I. Abuiziah, N. Shakarneh

Abstract—Genetic Algorithm (GA) is a powerful technique for solving optimization problems. It follows the idea of survival of the fittest - Better and better solutions evolve from previous generations until a near optimal solution is obtained. GA uses the main three operations, the selection, crossover and mutation to produce new generations from the old ones. GA has been widely used to solve optimization problems in many applications such as traveling salesman problem, airport traffic control, information retrieval (IR), reactive power optimization, job shop scheduling, and hydraulics systems such as water pipeline systems. In water pipeline systems we need to achieve some goals optimally such as minimum cost of construction, minimum length of pipes and diameters, and the place of protection devices. GA shows high performance over the other optimization techniques, moreover, it is easy to implement and use. Also, it searches a limited number of solutions.

Keywords—Genetic Algorithm, optimization, pipeline systems, selection, cross over.

I. INTRODUCTION

MANY optimization problems from the hydraulic engineering world, in particular for large pipeline systems, are complex in nature and difficult to solve by conventional optimization techniques.

There are three main avenues of this research: genetic algorithms (GAs), evolutionary programming (EP), and evolution strategies (ESs). Among them, genetic algorithms are perhaps the most widely known types of evolutionary algorithm today. Recently, genetic algorithms have received considerable attention regarding their potential as an optimization technique for complex problems and have been successfully applied in the area of pipeline system. Well-known applications include pipeline optimization, pump operating, system reliability design, and many others.

Genetic algorithms (GAs) are receiving increasing application in a variety of search and optimization problems. These efforts have been greatly aided by the existence of theory that explains what GAs are processing and how they are processing it, the theory largely rests on Holland's exposition of schemata [1].

Genetic algorithm is a search algorithm based on natural selection and the mechanisms of population genetics [1], [2].

Itissam Abuiziah is Ph.D student, with the Rural Engineering Department, Institute of Agronomy and Veterinary Hassan II, Rabat, Morocco (e-mail: itissam2002@yahoo.com).
Nidal, Shakarneh is with the Palestinian ministry of education, Bethlehem, Palestine.

The basic idea of the GA is borrowed from the biological process of survival and adaptation. The result is an efficient algorithm with the flexibility to search complex spaces such as the solution space for the design of a pipe network.

Genetic algorithm technique requires that the set of decision variables should be represented by a coded string of finite length [4]. To implement a GA, one codes the decision variable set describing a trial solution as a binary or dual string or "chromosome".

Genetic algorithms differ from conventional optimization and search procedures [3]. Genetic algorithms probabilistic and not deterministic, also it works with a coding of solution set, not the solutions themselves. Moreover, it searches from a population of solutions, not a single solution. Finally, genetic algorithm uses the cost function and doesn't need derivatives.

A. Encoding

The decision variables of a problem are normally encoded into a finite length string this could be a binary string or a list of integers.

For example: 0 1 1 0 1 1 0 1 0 or 2 3 4 1 1 4 5

B. Selection

Genetic Algorithms are optimization algorithms that maximize or minimize a given function. Selection operator deserves a special position in genetic algorithm since it is the one which mainly determines the evolutionary search spaces. It is used to improve the chances of the survival of the fittest individuals. There are many traditional selection mechanisms used and many user specified selection mechanisms specific to the problem definition [5].

The selection operator mainly works at the level of chromosomes. The goodness of each individual depends on its fitness. Fitness value may be determined by an objective function or by a subjective judgment specific to the problem. As the generations pass, the members of the population should get fitter and fitter (i.e. closer and closer to the solution). Selection is one of the important operations in the GA process. Different selection mechanisms work well under different situations. Appropriate method has to be chosen for the specific problem to increase the optimality of the solution. For example, the proportional roulette has been used in many problems [6] and it outperformed the other strategies in the salesman problem, achieving best solution quality with low computing times [7].

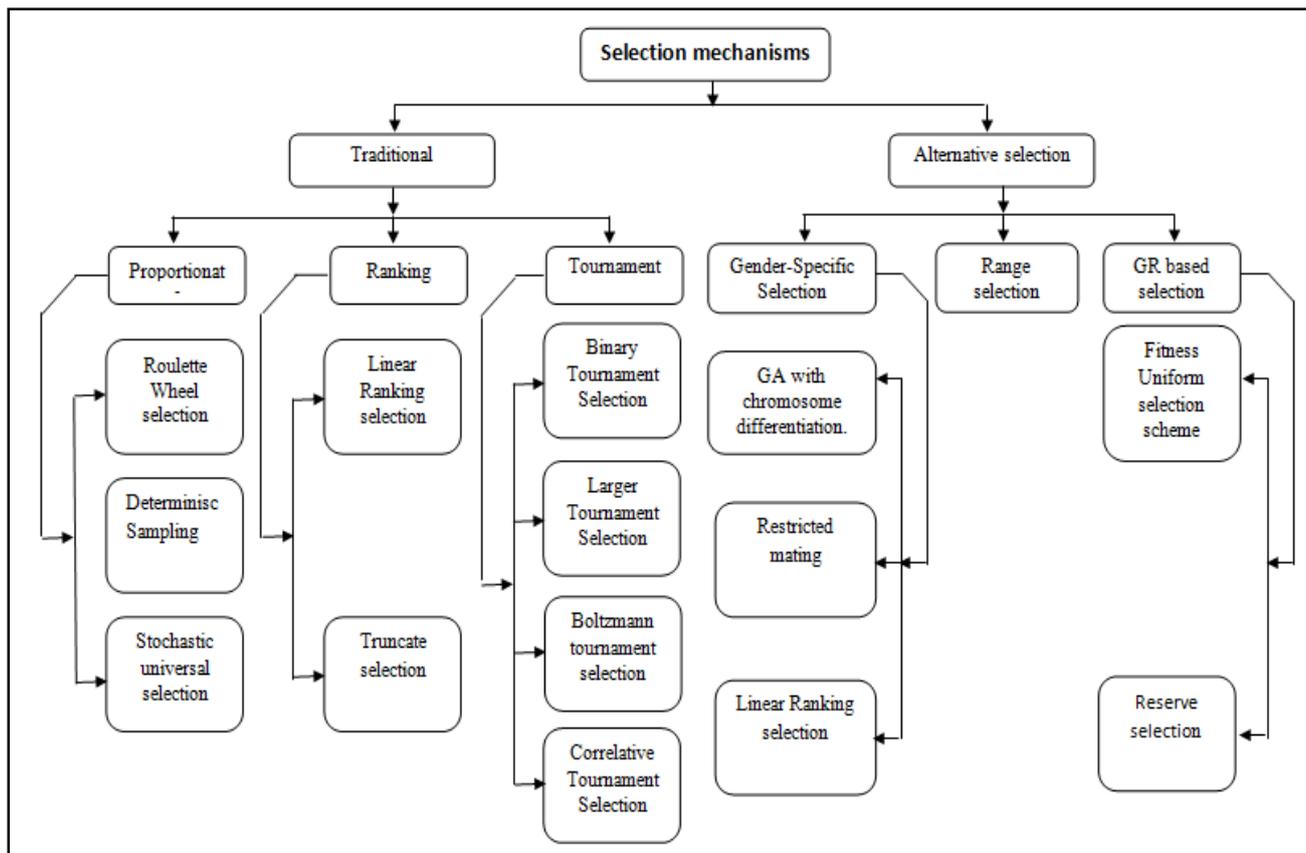The selection mechanisms are shown in Fig. 1.

World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:7, No:12, 2013

Fig. 1 Selection mechanisms

C. Cross over

Crossover operator plays an important role in producing a new generation. The crossover operator is a genetic operator that combines (mates) two chromosomes (parents) to produce a new chromosome (offspring). The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents. Crossover occurs during evolution according to a user definable crossover probability. There is number of cross over operators [8], [9] such as:
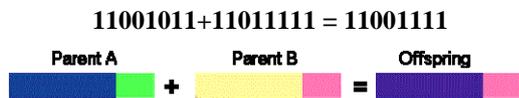
1. Single Point Crossover

$$11001011 + 11011111 = 11001111$$

Fig. 2 Single point crossover

2. Two Points Crossover

$$11001011 + 11011111 = 11011111$$

Fig. 3 Two Points Crossover

3. Intermediate (Uniform) Crossover

$$11001011 + 11011101 = 11011111$$

Fig. 4 Intermediate (uniform) crossover

4. Arithmetic Crossover

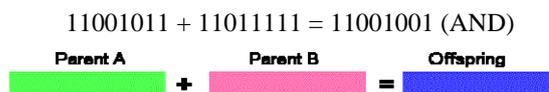$$11001011 + 11011111 = 11001001 \text{ (AND)}$$

Fig. 5 Arithmetic crossover

5. Heuristic Crossover

6. Ring Crossover [8].

A number of test functions with various levels of difficulty has been selected as a test polygon for determine the performance of crossover operators [8], A new crossover operator and probability selection technique is proposed by [10] based on the population diversity using a fuzzy logic controller, also the a new cross over operator introduced and used in the information retrieval (IR) [11].

D. Mutation

Mutation involves the modification of the value of each 'gene' of a solution with some probability pm, (the mutation

World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:7, No:12, 2013

probability). The role of mutation in genetic algorithm has been that of restoring lost or unexplored genetic material into the population to prevent premature convergence of the GA to suboptimal solution [12].

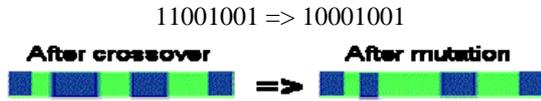Bit inversion - selected bits are inverted

$$11001001 => 10001001$$

Fig. 6 Mutation process

E. Implementation the Genetic Algorithm

1. Determine the initial population of creatures.
2. Determine the fitness of the population.
3. Reproduce the population using the fittest parents of the last generation.
4. Determine the crossover point, this can also be random.
5. Determine if mutation occurs and if so on which creature (s).
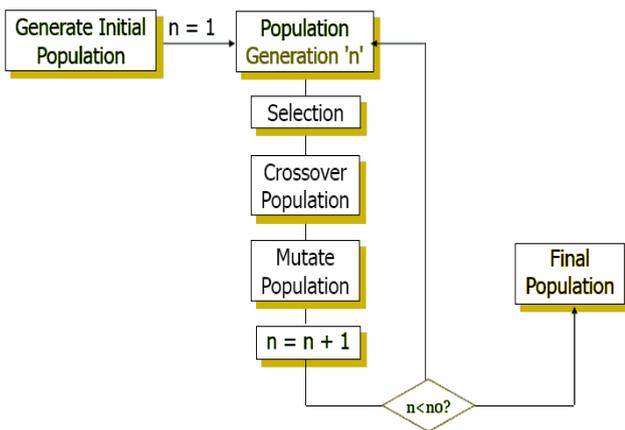6. Repeat from step 2 with the new population until condition (X) is true.

Fig. 7 Algorithm flowchart
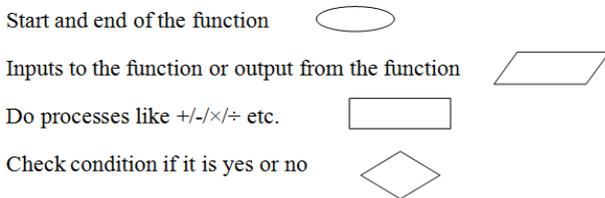
II. GA EXAMPLE

The meaning of the shapes:

Start and end of the function

Inputs to the function or output from the function

Do processes like +/-/×/÷ etc.

Check condition if it is yes or no

Fig. 8 The used shapes in the flowchart

A. Routlee Wheel Selection

- Let the current generation to be the parents have n individuals, each parent has l variables as follows:

$$parent = [prnt_1 \quad prnt_2 \quad \cdots \quad prnt_n] \quad size = l \times n$$

- compute the fitness (f_i) to each individual in the parents

$$F = [f_1 \quad f_2 \quad \cdots \quad f_n] size = 1 \times n$$

- compute the probability (p_i)) for each fitness

$$pi = \frac{fi}{\sum_{i=1}^{n} fi}$$

Then the probability vector is:

$$P = [p_1 \quad p_2 \quad \cdots \quad p_n] size = 1 \times n$$

The current generation and the probability are the input to the routtlee wheel selection function (rws) as shown in the flow chart. [individual] = rws (parent,P)
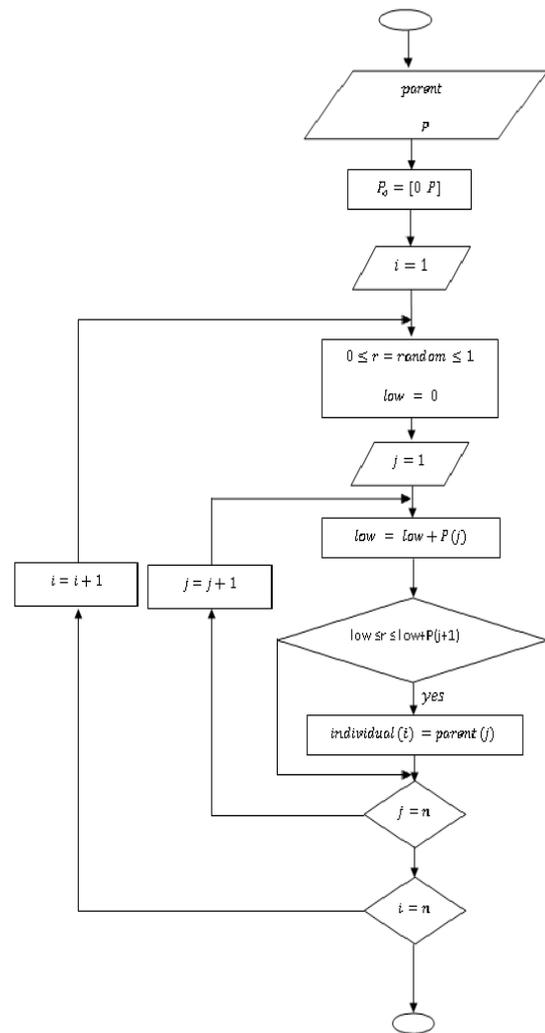

Fig. 9 Routlee wheel selection flow chart

B. Crossover

The crossover between two parents or two chromosomes ($prnt_1$ and $prnt_2$) is the operation of intersection between tow chromosomes to obtain new two children (offspring) (ofs_1 and ofs_2).To do the crossover we need to determine the two individuals ($prnt_1$ and $prnt_2$) to be used as input to the cross over function, also we need to determine the cross over probability which has to be high (pc > 0.7), and to determine the number of bits for each variable in the individual nb for

World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:7, No:12, 2013

coding.

$prnt_1$, $prnt_2$, ofs_1, ofs_2 and nb each has $size = l \times 1$

$r = round(pc \times nb(i)$ Compute the nearest integer to the values $pc \times nb(i)$ (i.e. the number of bits in each variable to crossed over)

r_t, of $size = l \times 1$ is a vector used to store the values of r.

$r\_t(i) = r$, to store the values of (r) at (i) in the (i) location in r_t

de2bi: is to change the values from decimal to binary (code the value) with the given number of bits

bi2de: is to change the values from binary to decimal (encode the value)

$temp = x1(1:r\_t(i))$: $temp$ equal the bits from 1 to the bit number $r\_t(i)$

$[ofs_1, ofs_2] = xo(prnt_1, prnt_2, nb, pc)$

Example of cross over:

Let: $prnt_1 = [11 \quad 105 \quad 50]^T$

$\quad prnt_2 = [22 \quad 63 \quad 35]^T$

$nb = [5 \quad 7 \quad 6]^T$, $pc = 0.7$

$r_1 = round(nb(1) \times pc) = round(5 \times 0.7) = 4$

$r_2 = round(nb(2) \times pc) = round(7 \times 0.7) = 5$

$r_3 = round(nb(3) \times pc) = round(6 \times 0.7) = 4$

$r\_t = [4 \quad 5 \quad 4]^T$

$de2bi(prnt_1(1),5) = de2bi(11,5) = 0 \quad 1 \quad 0 \quad 1 \quad 1$

$de2bi(prnt_2(1),5) = de2bi(22,5) = 1 \quad 0 \quad 1 \quad 1 \quad 0$

$ofs_1(1) = bi2de (0 \quad 1 \quad 0 \quad 1 \quad 0) = 10$

$ofs_2(1) = bi2de (1 \quad 0 \quad 1 \quad 1 \quad 1) = 23$

$de2bi(prnt_1(2),7) = de2bi(105,7)$

$\qquad\qquad\qquad = 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1$

$de2bi(prnt_2(2),7) = de2bi(63,7)$

$\qquad\qquad\qquad = 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1$

$ofs_1(2) = bi2de (1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1) = 107$

$ofs_2(2) = bi2de (0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1) = 61$

$de2bi(prnt_1(3),6) = de2bi(50,6)$

$\qquad\qquad\qquad = 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0$

$de2bi(prnt_2(3),6) = de2bi(35,6)$

$\qquad\qquad\qquad = 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1$

$ofs_1(3) = bi2de (1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1) = 51$

$ofs_2(3) = bi2de (1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0) = 34$

$prnt_1 = [11 \quad 105 \quad 50]^T$

$prnt_2 = [22 \quad 63 \quad 35]^T$

$ofs_1 = [10 \quad 107 \quad 51]^T$

$ofs_2 = [23 \quad 61 \quad 34]^T$

C. Mutation

The mutation is an operation occurred to the individual (chromosome) (prnt which change the value of one bit(ofs). To do the mutation in a certain generation, all individuals (indvt) are used as input to the mutation function; also the number of bits of each variable (nb) has to be known, then in the function the following are selected randomly:

1- The individual($prnt$).
2- The variable in the individual(var).
3- The bit in the variable(bi).

$[ofs, rn] = mu(indvt, nb)$

The above function returns the mutated individual and its location.

$indvt$: $size = l \times n$

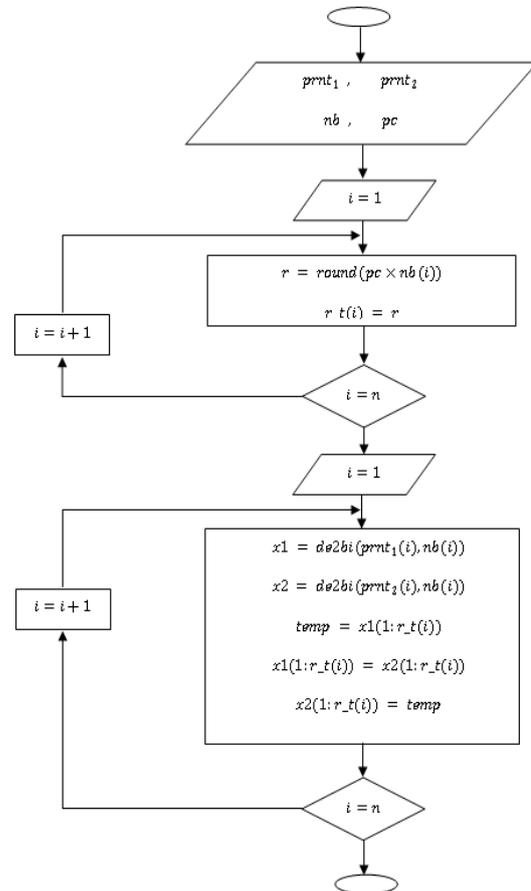nb: $size = l \times 1$

ofs: $size = l \times 1$

Fig. 10 Crossover flow chart

World Academy of Science, Engineering and Technology
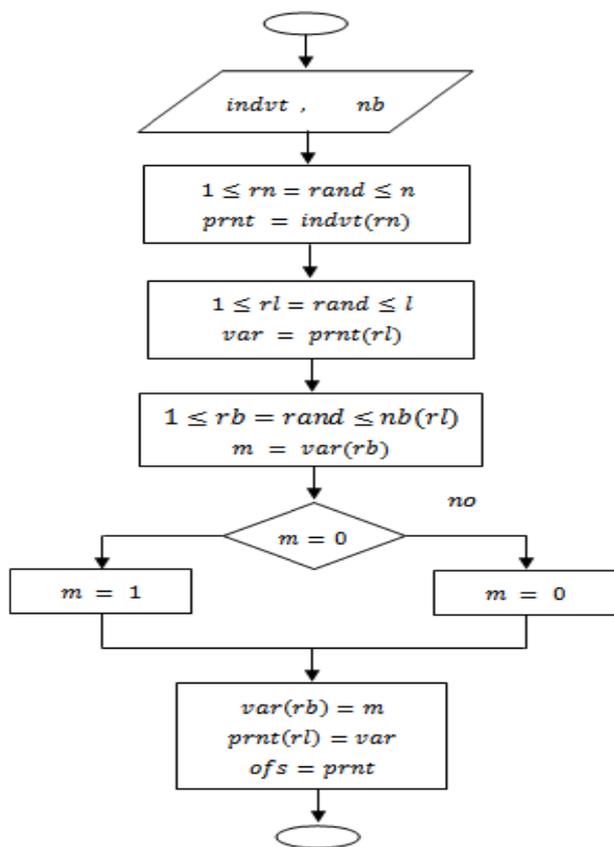International Journal of Mathematical and Computational Sciences
Vol:7, No:12, 2013

Fig. 11 The mutation flow chart

III. Method for Solving Optimization Problems

Ant Colony Optimization is based on the metaphor of ants seeking food. In order to do so, an ant will leave the anthill and begin to wander into a random direction. While the little insect paces around, it lays a trail of pheromone. Thus, after the ant has found some food, it can track its way back. By doing so, it distributes another layer of pheromone on the path. An ant that senses the pheromone will follow its trail with a certain probability. Each ant that finds the food will excrete some pheromone on the path. By time, the pheromone density of the path will increase and more and more ants will follow it to the food and back. The higher the pheromone density, the more likely will an ant stay on a trail. However, the pheromones vaporize after some time. If all the food is collected, they will no longer be renewed and the path will disappear after a while. Now, the ants will head to new, random locations [13], [14].

Particle Swarm Optimization1 (PSO), developed by Eberhart and Kennedy, is a form of swarm intelligence in which the behavior of a biological social system like a flock of birds or a school of fish is simulated. When a swarm looks for food, its individuals will spread in the environment and move around independently. Each individual has a degree of freedom or randomness in its movements which enables it to find food accumulations. So, sooner or later, one of them will find something digestible and, being social, announces this to its neighbors. These can then approach the source of food, too

[14].

Particle swarm optimization has roots in two main component methodologies. Perhaps more obvious are its ties to artificial life (A-life) in general, and to bird flocking, fish schooling, and swarming theory in particular. It is also related, however, to evolutionary computation, and has ties to both genetic algorithms and evolutionary programming [15].

Hill climbing is a very old and simple search and optimization algorithm for single objective functions f. In principle, hill climbing algorithms perform a loop in which the currently known best solution individual p ⋆ is used to produce one offspring pnew. If this new individual is better than its parent, it replaces it. Then, the cycle starts all over again. In this sense, it is similar to an evolutionary algorithm with a population size psof 1.

Also, many other methods have been used widely in many applications such as random optimization, simulated annealing and extremely optimization [14].

IV. Genetic Algorithm in Water Pipe Systems

Water distribution systems are usually designed to adequately satisfy the water requirements for a combination of domestic, commercial, public and fire fighting purposes [16].

Millions of dollars are spent each year on water distribution systems. Pipes optimization techniques provide an opportunity for potential savings in costs for water supply systems. These optimization techniques include linear programming, nonlinear programming, dynamic programming, enumerative approaches, and genetic algorithms. The former four techniques have been applied to pipe network optimization in the research literature over the last 30 years or so. Genetic algorithms provide a new approach to pipe network optimization [17].

A relatively comprehensive approach for the use of genetic algorithms for pipe networks optimization has been developed over the last ten years [4], [18]-[25].

Goldberg and Kuo applied GAs to the steady state optimization of a serial liquid pipeline. The system consisted of 10 pipes and 10 compressor stations each containing four pumps in series. The objective was to minimize the power requirements, while supplying a specified flow and maintaining allowable pressures. The three operators found near-optimal pump [4].

Murphy and Simpson used SGA (Structured Genetic Algorithm) to find the optimal solution of the network. The method chooses the optimal combination among the eight alternative decisions possible for each of the eight decision variables (pipes) [18].

Jung and Karney describe the optimal selection of pipe diameters in a network considering steady state and transient analysis in water distribution systems. Two evolutionary approaches, namely genetic algorithms (GA) and particle swarm optimization (PSO) are used as optimization methods to obtain pipe diameters. Both optimization programs, inspired by natural evolution and adaptation, show excellent performance for solving moderately complex real-world

World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:7, No:12, 2013

problems which are highly nonlinear and demanding [19].

Djebedjian describe the objective function that focused on the cost criteria of network components. While Berge et al. presented the water distribution systems optimization by selecting the optimal pipe diameters for water hammer transients. The optimization method used is the Genetic Algorithm (GA) [20].

Jung and Karney used both genetic algorithm (GA) and particle swarm optimization (PSO) approaches to optimize the network system including the transient. The aim is to optimize the preliminary selection, sizing and placement of hydraulic devices in a pipeline system in order to control its transient response. Three simple objective functions are considered: 1. to minimize the maximum head; 2. to maximize the minimum head; and 3. to minimize the difference between the maximum head and minimum head in the system. This study shows that the integration of a GA or PSO with a transient analysis technique can improve the search for hydraulic protection devices in a pipe network [21].

Berge presents the reliability-based water network optimization by selecting the optimal pipe diameters for steady state flow and water hammer under hydraulic reliability. He used the GA integrated with a hydraulic analysis solver, a Monte Carlo simulation program and a transient analysis program to improve the search for the optimal diameters under certain constraints. The application of GA optimization tool to the case study demonstrates the capability of the Monte Carlo method and the genetic algorithm to find the optimal pipe. The technique of the optimal pipe diameter selection is very economical as the network design can be achieved without using hydraulic devices for water hammer control [22].

Jung and Karney investigated the use of optimal design of a pipe network considering both steady and transient states. They used two global optimization methods, genetic algorithms (GA) and particle swarm optimization (PSO), to be employed to find the optimal pipe diameters in a system with allowance for water hammer conditions. In this application, both approaches exhibit similar evolution histories and optimal results [23].

Sarbu used the improved linear model for optimization of water distribution networks supplied from one or more node sources; according to demand variation has been studied [24].

Jung applied optimization methods to select the most economical set of pipe sizes that will produce the desired range of pressures in the network. The rationale behind an economical design is that by selecting the smallest possible diameter pipe set to minimize overall cost, pressures are marginally higher than an acceptable level for the specified design loading conditions [25].

V. Conclusion

Evolutionary Algorithms (EAs) are a set of probabilistic optimization algorithms based on an analogy between natural biological systems and engineered systems.

This paper reviews some works related to genetic algorithms operations and focusing on the application of

genetic algorithms to pipeline system. A simple GA consists of three basic operators: reproduction, crossover and mutation.

GA is globally oriented and generally straighter forward to apply in situations where there is little or no a priori knowledge about the problem to solve. Because GA requires no derivative information and it is stochastic in nature, GA is capable almost to find the global optimum.

GA is robust and has been proven theoretically and empirically to be able to efficiently search complex solution spaces

References

[1] J. H. Holland, "*Hierarchical descriptions of universal spaces and adaptive systems*", (1968).

[2] J. Holland, "*Adaptation in Natural and Artificial Systems*", University of Michigan Press, Ann Arbor. (Technical Report ORA Projects 01252 and *08226).*Ann Arbor: University of Michigan, Department of Computer and Communication Sciences, (1975).

[3] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning." Addison- Wesley Publishing Co., Inc., Reading, Mass, (1989).

[4] D. E Goldberg, and C. H. Kuo, "Genetic algorithms in pipeline optimization."*J.* Computing in *Cïv. Engrg.,*ASCE, 1(2), 128-141, (1 987*).*

[5] R. Sivaraj and T. Ravichandran, "Review of selection methods in genetic algorithm." *International Journal of Engineering Science and Technology (IJEST)*, 3(5), 3792-3797. (2011).

[6] A. Lipowski, and D. Lipowska, "Roulette-wheel selection via stochastic acceptance." (2011).

[7] N. M. Razali, and J. Geraghty, "Genetic Algorithm Performance with Different Selection Strategies in Solving TSP" *World Congress on Engineering*, London, U.K,. (2011).

[8] Y. Kaya, M. Uyar, and R. Tekdn, "A Novel Crossover Operator for Genetic Algorithms: Ring Crossover." (2011).

[9] A. Otman and A. Jaafar, "A Comparative Study of Adaptive Crossover Operators for Genetic Algorithms to Resolve the Traveling Salesman Problem."*International Journal of Computer Applications*, *31*(*11*), 9. (2011).

[10] M. J. Varnamkhasti, L. S. Lee, M. R. A. Bakar and W. J. Leong, "A Genetic Algorithm with Fuzzy Crossover Operator and Probability." *Hindawi Publishing Corporation, Advances in Operations Research*, 2012, 16. (2011).

[11] D. Vrajitoru, "Crossover improvement for the genetic algorithm in information retrieval." (1998).

[12] M. Srinivas and L. M. Patnaik, "Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms." *IEEE Transactions on systems, man and cybernetic*, 24(4), 656-667. (1994).

[13] H. R. Maier, A. R. Simpson, A. C. Zecchin, W. K. Foong, K. Y. Phang, H. Y. Seah, and C. L. Tan, "Ant Colony Optimization for Design of Water Distribution Systems."*Journal of water resources planning and management © ASCE*, 129(3), 10. (2003).

[14] T. Weise, *Global Optimization Algorithms – Theory and Application* Second Ed., Self-Published. (2009).

[15] J. Kennedy, and R. Eberhart, "Particle Swarm Optimization."*IEEE*, 7. (1995).

[16] A. N. El-Bahrawy and A. A. Smith, "Application of MINOS to water collection and distribution networks."*Civ** Engrg. *Systems,* Vol. 2, pp. 38-49. (1985).

[17] Z. Zhang, "Fluid transients and pipeline optimization using genetic algorithms", University of Toronto. (1999).

[18] L. J. Murphy and A. R. Simpson, "Pipe optimization using genetic algorithms."Research Report No- 93, Department of Civil Engineering, University of Adelaide, Australia, June, pp. 95. (1992).

[19] B. S. Jung and B. W. Karney, "Fluid transients and pipeline optimization using GA and PSO: the diameter connection." *Urban Water Journal*, 1(2), 167-176. (2004).

[20] Djebedjian. B., Mohamed, S. M. Mondy. A. and Abou Rayan. M., "Cost Optimization of Water Distribution Systems Subjected to Water Hammer," Proceedings of the Thirteenth International Water

World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:7, No:12, 2013

Technology Conference, IWTC13 2009, Hurghada, Egypt, 17-20 March, 491-513. (2009).

[21] B. Djebedjian, "Reliability-based water network optimization for steady state flow and water hammer."*6th International Pipeline Conference*, Calgary, Alberta, Canada, 11. (2006).

[22] B. S. Jung, W. Bryan and B. Karney, "Hydraulic Optimization of Transient Protection Devices Using GA and PSO Approaches." *Journal of water resources planning and management © ASCE* 132(1), 9. (2006).

[23] B. S. Jung and B. W. Karney, "Systematic Surge Protection for Worst-Case Transient Loadings in Water Distribution Systems."*journal of hydraulic engineering © ASCE*, 135(3), 218-223. (2009).

[24] I. Sarbu, "Design of optimal water distribution systems." *International journal of energy*, 3(4), 9. (2009).

[25] J. S.Jung, P. F. Boulos and T. Altman, "Optimal transient network design: A multi-objective approach" *American Water Works Association*103 (4), 118-127. (2011).