IMPROVEMENT OF THE YAZD WATER CONVEYANCE CONTROL SYSTEM BY GA OPTIMIZATION

A.Malekpour[1], B.W.Karney[2], B.J.Adams[3]
[1]Graduate Student, Univ of Toronto, 35 St. George St., Toronto, On, M5S 1A4, Canada, Email: homalekpour@yahoo.com
[2]Professor, Univ of Toronto, 35 St. George St., Toronto, On, M5S 1A4, Canada, Email: Karney@ecf.utoronto.ca
[3]Professor, Univ of Toronto, 35 St. George St., Toronto, On, M5S 1A4, Canada, Email: adams@ecf.utoronto.ca

ABSTRACT

In this paper a simple strategy involving pump on-off control is proposed to improve the performance of the Yazd water conveyance system in Iran. To evaluate the efficiency of the proposed logic, an unsteady hydraulic model is developed on the basis of quasi-steady theory and the flow control logic is introduced into the model through the boundary conditions. The model is used to evaluate the hydraulic behavior of all components of the system for a possible range of operation flow condition over a 24 hour period. Although the control logic can be effectively modeled, the number of on-off cycles of the pumps in low operating flow appears to be high for such a long pipeline. This cycling problem can be improved by increasing the volume of the reservoirs fed by each pump station. A GA optimization model is employed to find the optimum expansion for each reservoir. Since the hydraulic constraints could not be defined by analytical functions, the flow model is embedded in the optimization routines to record the number of on-off cycles of each pump and to check whether any constraints are violated during the operation. Assuming the number of on-off cycle of each pump is limited to just one per day, the optimal solution is determined. The results show that the convergence is quickly achieved and the optimal solution is reached during the first 30 generations. Comparing GA results with results obtained from an enumeration technique shows that a global solution is captured by GA but with considerably reduced computational expense.

KEY WORDS

Genetic algorithm, simulation, flow control, pumping.

1. Introduction

Constructed in 1999, the Yazd water conveyance pipe line is one of the longest and most important water conveyance systems in Iran. With a length of 333 km, the system was designed to provide a portion of the drinking water supply of the city of Yazd and three small cities in Yazd province. Like most long lines over relatively uneven terrain, the line is divided into several gravity and several pumping segments that are connected through a series of service reservoirs. The essential strategy is to pump water to the top of significant rises, and to let it descend from there by gravity. The central role of the service reservoirs is to better balance inflows and outflows into the system overall without inducing abrupt transient events.

Yet, despite best intentions, during the first year of operation, the Yazd Regional Water Board, hereinafter called the client, realized that the operation of such a long pipe line was highly challenging and that pressure and flow control was problematic within the system. Thus, the client called consulting firms in Iran to find an optimal and practical solution.

This study shows that an on-off strategy for operating pumps is the most economical one in terms of simplicity and energy consumption; however, to guarantee a smooth and stable control system, some of the reservoirs were found to be too small and require expansion. To evaluate the expansion for each reservoir, a mathematical model was developed to simulate the behaviour of the system under different control circumstances. The model was developed on the basis of a quasi-steady theory to show the response of different components of the system during its operation. Exercising the model showed that it is possible to limit the number of on-off cycles of the pumps in an acceptable range by increasing the volume of the reservoirs associated with each pump station.

However, rather than the expensive approach of arbitrarily increasing the size of the reservoirs, an optimization was sought to find the minimum volume of each reservoir that meets the performance constraints. To do this, a genetic algorithm is employed, and the results from which are corroborated by exhaustive enumeration techniques.

This paper first describes the components of the system and its operational problem. The mathematical model is next presented and an on-and-off strategy for controlling the pump stations is proposed. The behaviour of the system is then evaluated using this model. Both an enumeration and a Genetic Algorithm (GA) method for finding the optimal reservoir expansion are also described. Finally, the obtained results are discussed and conclusions are drawn.

2. System Description

The Yazd water conveyance pipe line has a maximum capacity of 3 m^3/s and a total length of 333 km and was designed to provide a portion of drinking water supply of

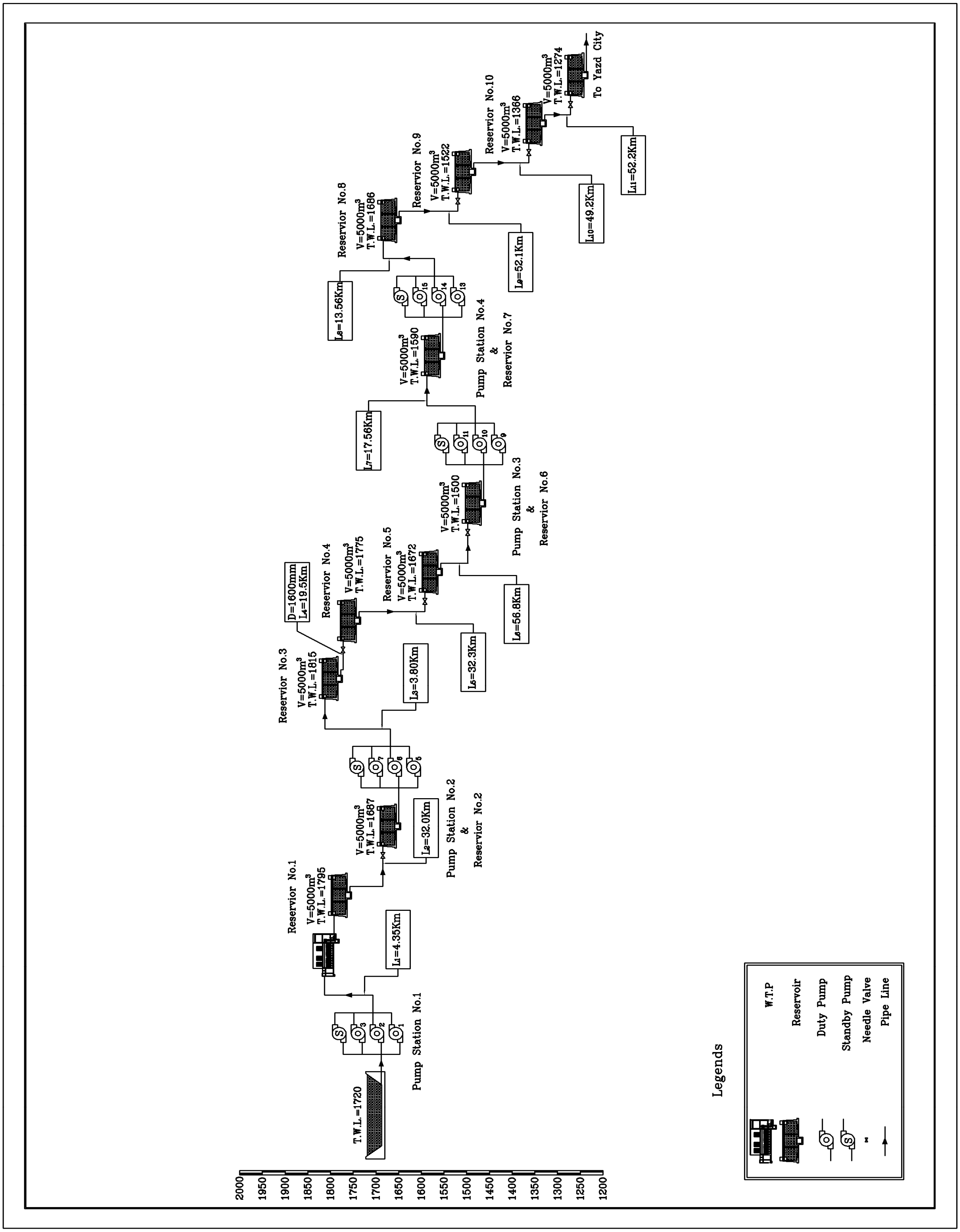

Figure 1. Yazd water conveyance system

the city of Yazd and two small cities (Meybod and Ardakan) in Yazd province. As can be seen in Figure 1, this line consists of 4 pumping and 7 gravity steel-pipeline segments (generally 1.4 m in diameter) that are connected through 11 equally sized reservoirs, each one with a volume 5000 m^3. The information on the pipeline is presented in Table 1.

Segment No.	Diameter (mm)	Length (m)	Head Source
1	1400	4350	Pumps
2	1400	32000	Gravity
3	1400	3800	Pumps
4	1600	19500	Gravity
5	1400	32300	Gravity
6	1400	56800	Gravity
7	1400	17560	Pumps
8	1400	13560	Pumps
9	1400	52100	Gravity
10	1400	49200	Gravity
11	1400	52200	Gravity
Total Length=333.370 (Km)			

Table 1. Pipeline information

In order to control the flow, two needle valves, each with a diameter of 1000 mm, are installed in parallel at the downstream end of each gravity pipeline segment. Except for the last gravity pipeline segments in which the needle valve openings are adjusted by electrical actuators, the other valves are adjusted mechanically. As shown in the Figure 2, each needle valve is connected to a floater placed in the associated reservoir through a cable mechanism. As the water head in the reservoir changes, the floater adjusts the associated needle valves accordingly.

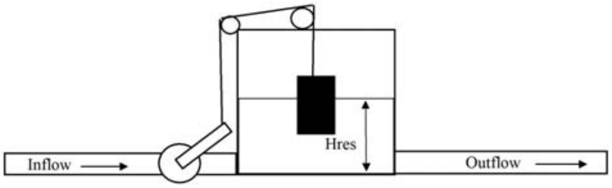

Figure 2. Floater-valve-cable mechanism

There are also four pump stations in the system. Each pump stations consist of 4 similar pumps, three duty pumps and one standby. Pump station information including static and maximum dynamic head (at the design flow), power and pipeline lengths are summarized in Table 2.

All of the information from the pump stations and their associate reservoirs (including discharge and heads of pumps, the number of active pumps, water height and inflow and outflow of the reservoirs etc.) is transmitted to the control room located at the end of line (Yazd city) via a radio telemetry system. A large panel at the control room mimics the status of the pump stations and reservoir data. All pumps can be remotely controlled from the operating room where the issued commands are transmitted via the radio telemetry system.

Pump Station No.	Pipe Length	Static Head	Dynamic Head	Power
	(m)	(m)	(m)	(MW)
1	4350	75	94	3×1.4
2	3800	128	142	3×2
3	17560	90	147	3×2
4	13560	96	141	3×2

Table 2. Pump station information

All of the information from the pump stations and their associate reservoirs (including discharge and heads of pumps, the number of active pumps, water height and inflow and outflow of the reservoirs etc.) is transmitted to the control room located at the end of line (Yazd city) via a radio telemetry system. A large panel at the control room mimics the status of the pump stations and reservoir data. All pumps can be remotely controlled from the operating room where the issued commands are transmitted via the radio telemetry system.

Unfortunately, the lack of a decision engine in the system results in the sophisticated control equipment being not optimally used, and flow control is performed poorly by the operators. In fact, decisions are made completely by the operators on the basis of available data and all commands are issued accordingly. Although such a control strategy can be used in simple systems, this is almost impossible for efficient control such complex systems as that of Yazd. This explains why just after a few months of operation, the client became convinced that efficient operation of the system was impossible, and there was a vital demand for improvement of the control system.

2.1 Flow Control System, Description and Difficulties

The challenge of this control system is illustrated by the hydraulic behavior of the system during a flow control procedure. To adjust the flow of the system to a demand value, the needle valve's opening at the end of the line is slowly changed; as is naturally expected, closing (opening) the valve reduces (increases) the flow. This changes the discharge and, accordingly, the pressure behind the valve. The pressure changes are then propagated to the upstream at the elastic wave velocity. As soon as this information reaches the closest upstream reservoir (reservoir No. 11) a local unbalanced condition is setup. In this condition, the difference between inflow and outflow causes a smooth change in the water level of the reservoir which in turn results in changing the opening of the associate needle valves through the cable-floater mechanism. The pressure changes behind the needle valves are then propagated to the next upstream reservoir and this procedure is repeated until the information reaches the first reservoir fed by pump station No. 4. At this moment, an unbalanced condition in the reservoir causes the water level to be changed (increased or decreased), and the reservoir may either empty or overflow, if correct number of pumps are not activated in the associated pump station. According to the water height in the reservoir the operator thus would operate the pumps manually. Unfortunately, this is not a simple task; indeed, making a proper decision depends on the demand flow, the hydraulics of the pumping line, total volume of the reservoir, the capacity of the pumps and many other factors. This is particularly the case because the capacities of pumping units are large compared to the volume of reservoir. This causes the water level in the reservoir to change quickly whenever a pumping unit is either activated or deactivated. Decision making at this pump station affects the upstream reservoir fed by pump station No. 3 and this demands a similar operational strategy at pump station No. 3 as well. The generated signal is then propagated to the upstream side of the pipeline through the gravity and pumping line segments sequentially.

This brief sketch illustrates the complexity of the flow control system and demonstrates how manual operation is almost impossible in this system. Indeed, several months of operational experience well justifies this claim. During this period, reports show that reservoirs fed by the pump stations were frequently empty or overflowing. The empty reservoirs caused large amount of air to enter the system setting up two phase flows in parts of the gravity lines. Contraction and expansion of entrapped air may cause pulsating flow in parts of the system, and depending on the ventilation of the system, this may result in harmful water hammer pressures [1]. Although little mechanical damage has yet been reported, this state of affairs cannot be regarded as a safe or sustainable operation of the system. Indeed, the lack of more serious problems might arise from the simple fact that the high ultimate strength of the steel pipeline is almost twice as much as its working strength.

3. Control System Improvement

In order to improve the control system, a simple logic algorithm is proposed for turning the pumps on and off. A flow chart of this algorithm is presented in Figure 3. As can be seen this algorithm continually processes the information received at the control room and makes a proper decision accordingly. The proposed logic is designed to be implemented on a programmable logic controller (PLC) which already exists in the control room. If this logic is implemented in an infinite loop, it can effectively process the information and issue online commands.

As shown in Figure 3, for a particular pump station, all decisions are made according to the status of its immediate downstream reservoir. The logic compares the height of water in the reservoir with both a predefined maximum elevation and the rate of change in water level to check if there is a requirement for turning off the pumps. A similar procedure is used for deciding if one more pump needs to be turned on. In this case, the control logic compares the height of water with both a predefined minimum level and the rate of change in the water level. To ensure that the reservoir neither empties nor overflows, both the maximum and minimum water level set points should be determined carefully.

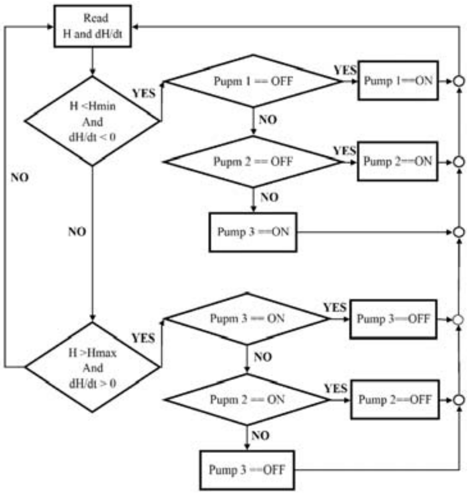

Figure 3. Flow control logic algorithm

To evaluate whether the proposed control logic is able to provide an efficient and stable condition in the system, a mathematical model could be employed to check unsteady hydraulic behavior of the system under the proposed control logic. Three 1D-type models are available for transient modeling of pressurized systems: full elastic (or water hammer) models, rigid water column models or quasi-steady models [2], [3]. Although the full elastic model is the most complete and can be used effectively in a wide range of flow conditions in the system, it is computationally expensive and its refinements are probably unnecessary; if the system can be operated within its constraints, rapid unsteady conditions should be avoided. Moreover, the computational issue is still important when a model is going to be used as a computational engine in an optimization model which is usually called many thousands of times. Of course, using such a model is inevitable if the pace of transients is quick and the compressibility of the water assists to generate local high and low pressures in the system.

Both the rigid column model and quasi-steady models, which are simplified forms of the full elastic model, can then be used herein. In the rigid column model, it is assumed that water flows as a solid sludge column and all points of the column have similar velocities at a particular time (if the diameter is constant along the length of pipe) [2]. This model, like the full elastic model, is able to simulate the real acceleration and deceleration of flow. However, in the quasi-steady model which is some what simpler than rigid column model, the acceleration term in the dynamic equation is neglected and assumed that the response of the system is completely simultaneous. In terms of simplicity and computational cost, the quasi-steady model is of course preferable as long as it is technically acceptable.

To verify if this simple model is suitable for present study, the hydraulic behavior of the system during transients is considered. In gravity segments of system, the pace of transients is controlled by the needle valve-floater mechanism. Since the large volume of the reservoir prevents the floater from a quick replacement, changes in opening of the valves and consequently changes in flow velocities occur very slowly. This means that the acceleration head which is equal to $\frac{L}{g}\frac{dv}{dt}$ (L=length of line segment) can be neglected compared with the friction loss of the line. However, in the pumping line segments where the practical issues demand a fairly quick turning on and off of the pumps, the acceleration time of flow may be an important issue. In this case, there is always a time lag between flow changes in the pump station and inflow changes in the associated reservoir. The longer the pumping line, the more lag time will be. Considering this fact, it is obvious that the quasi-steady model fails to simulate pumping lines accurately because in this model, flow changes in the pumping station are simultaneously detectable in the reservoir. Fortunately, in the discrete world a good selection of time step can very well compensate for such a lag time. A rough calculation shows that the maximum acceleration time in the pump stations never exceed 200 s under different flow conditions. The above discussion shows that a quasi-steady model with a time step of 200 s can be reasonably used to trace the transient behavior of the system.

It remains to select appropriate software to implement the algorithm on the hydraulic system. At first glance it might appear that the EPANET model supporting extended period hydraulic analysis [4] is well suited for the present study. However, this model cannot be directly used herein because of some technical reasons. First, there is no option available in EPANET for modeling of a valve-reservoir-floater boundary condition [4] and secondly this model cannot be used as an internal engine in the optimization model. Fortunately, EPANET also has a dll-based engine in which there are several functions for a complete extended period analysis and also for direct communication with all components of the system. A computer program is thus developed in Microsoft Visual Basic 6 to implement both the reservoir-valve-floater boundary condition and the proposed control logic. This computer program uses the EPANET engine internally to perform hydraulic calculations and to communicate with all components of the hydraulic system.

Since the proposed control logic should be able work well under different operational circumstances, the model was run for different possible discharges ranging between 1000 and 3000 L/S. A 24 hour simulation with a time step of 200 s appears to be long enough for the evaluation of the system and was therefore selected for modeling. Among a group of data obtained from the model, the number of times by which each pump is turned on and off during the simulation period and also the transient response of reservoirs fed by pump stations appeared to be sufficient for evaluating the control system. The results show that in low operating flow, the number of on-off cycles of the pumps reach the highest value (10 to 11 times) whereas with high operating flows, they are limited to 2 to 3 times. The results also revealed that in pump stations Nos. 1 and 2, three pumps should always be operated regardless of how much the operating flow is, but in pumping station Nos. 3 and 4 the numbers of operated pumps are proportional to the operating flow. As examples, the water level history of reservoir No. 1 for two operating flows of 1000 and 3000 L/S are presented in Figures 4 and 5, respectively. As can be seen, the proposed control logic perfectly control water level in the reservoirs.

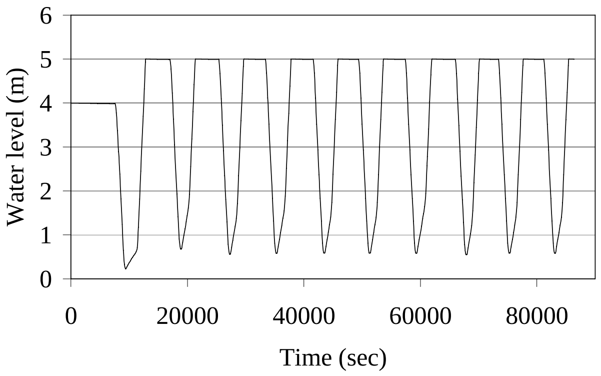

Figure 4. Water level history of reservoir 1 (1000 L/S)

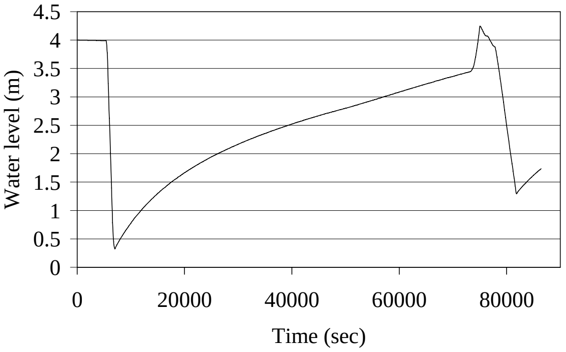

Figure 5. Water level history of reservoir 1 (3000 L/S)

Although there was no technical limitation for this kind of operation, the client required a solution by which the number of on-off cycles of pumps were limited to 1 to 2 times. By experimenting with the model, it is seen that it is practically possible by extending the volume of the reservoirs Nos. 1, 3, 7, and 8, although determining the optimal solution of the reservoir volumes required further study.

4. Optimal Solution for Reservoir Volumes

4.1 Exhaustive Enumeration

Assuming the maximum additional reservoir volume to be 50,000 m^3 and considering an incremental volume of 2000 m^3, there are 25 possible levels of extension for each reservoir. So the number of possible combinations of reservoirs is $(25)^4$=390,625.

A 24 hour hydraulic analysis was performed for all possible combinations and under different flow conditions (between 1000 and 3000 L/S with an increment of 200 L/S). Then all combinations were checked to determine if the number of on-off cycles of each pump violated the feasible number of on-off cycles for each pump which is 1 in this study. In the next step, the total cost of each feasible solution is calculated according to the unit costs presented in Table 3, and the optimal combination is determined. The optimal solution is also presented in Table 4.

Volume Range (m^3)	**Total Unit Cost ($/unit volume)**
0< V <=5000	100
5000< V <=20000	90
V > 20000	85

Table 3. Unit costs for different ranges of reservoir volume

Reservoir No.	**Extension (m^3)**	**Costs ($)**
1	9713	874,170
3	16522	1,486,980
7	41162	3,498,770
8	38610	3,281,850
Total=	**106,007**	**9,141,770**

Table 4. Optimal combination of reservoirs

4.2 Genetic Algorithm

Overview of Genetic Algorithm: The genetic algorithm approach is a search algorithm based on natural selection and the mechanisms of population genetics [5]. The simple idea of the GA search is based on the biological processes of survival and adaptation. Although now well known, the GA process is briefly described.

The first step in the GA implementation is to define decision variable set as a numerical string; this is the equivalent of a genetic chromosome. This encoding can be done in several ways, but a binary alphabet is probably the most common. The genetic algorithm evaluates each solution coded in a chromosome and determines the associated level of worth or “fitness” of the associated system model. Next, a collection of the better ranked system chromosomes is selected from an overall “population” of solutions. The GA applies three operators (reproduction, crossover, and mutation) to produce the next trial population. Reproduction is a process through which the fittest solutions are preserved and transmitted to the new population. Crossover produces two offspring strings by partial exchanging of corresponding segments of bits between two parent strings. Mutation prevents the loss of a potentially useful genetic trait by random changing of bit values. The GA is explained in the context of the present study below.

Implementation of Genetic Algorithm for Present Study: The genetic algorithm technique requires that a set of decision variables to be represented by a coded string of finite length. In this case, the four decision variables are the volume extension of reservoir Nos. 1, 3, 7, and 8, respectively. Assuming the optimal volume for each reservoir is between 0 and 45,000 m^3, a binary string of 12 bits can be employed to represents 4096 digits between 0 and 4095. If an incremental volume of 11 is chosen, 4096 volume options between 0 and 45,045 m^3 could be simply addressed by multiplying of digit number by 11. A typical chromosome employed in this research is shown in Figure 6.

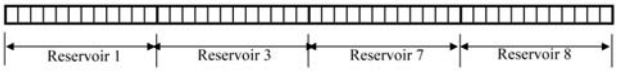

Figure 6. A typical chromosome

Generation of Initial Populations: The GA generates the initial population of solutions (of, say, size n=100) using a random number generator. Each bit position in 48-bit string takes on a value of either 1 or 0. Each successive 12 bits represent a specific option for the four reservoirs under study. For example if the 48 bits were “110111101111 000011110111 110000011110 101010101110” , the extended volume of reservoir 1, 3, 7,and 8 would be 39237, 2717, 34122, and 30074 m^3, respectively. This can be easily achieved by decoding the four binary numbers to digit numbers which are 3567, 247, 3102, and 2734 respectively and multiply them by 11 to produced appropriate scaling.

Computation of reservoir cost: The GA considers each of the 100 string chromosome in the population in turn. It decodes each substring into the corresponding reservoir volume and computes the total cost of material and construction cost according to Table 3.

Hydraulic analysis of each chromosome: As previously mentioned, a computer model was developed for transient analysis of the system for a period of 24 hours. This model is used as an embedded engine to evaluate the hydraulic performance of each chromosome under different operating flow, between 1000 and 3000 L/S. To keep the computational time in a reasonable range, a flow increment of 200 L/S is chosen herein. Therefore, for each chromosome, the hydraulic model should be run 11 times. Assuming a time step of 200 s for the model, 11×433=4763 (86400/200+1=433) hydraulic analysis steps are then needed for each chromosome. During the analysis, the number of on-off cycles for each pump is recorded for evaluating the hydraulic efficiency of each chromosome.

Computation of penalty and total cost: The GA assigns a penalty cost for each chromosome if the number of on-ff cycles of each pump exceeds the maximum allowable. The total number of extra cycles of all pumps is multiplied by a penalty factor say $X. The penalty cost should be such that near-optimal infeasible solutions are highly fit so that the optimum solution will be approached from both above and below. The optimum solution lies on the boundary between feasible and infeasible solutions [6]. The total computational cost for each chromosome is then taken as the sum of the reservoirs cost plus penalty cost.

$$TC = \sum_{i=1,3,7,8} C_i V_i + Penalty \times \sum_{m=1}^{4} \sum_{n=1}^{3} Max(TP_{mn} - TP_{Allowable}) \quad (1)$$

where m is the pump station index, n is the pump number index, i is reservoir index, and C and V are the unit price and volume of the reservoirs, respectively.

Computation of fitness: Each chromosome has a fitness that could be taken as some function of objective function. Using equations 1 and 2, the GA computes the fitness for each chromosome in the population. Since the GA searches for the minimum cost solution, objective function must be minimized. To ensure the surviving of the lowest cost chromosomes, a proper fitness function should be selected. A form of fitness function, as shown in the follow, could be the inverse of the total cost of reservoirs.

$$f_i = Fitness_i = \frac{1}{Total\,\mathrm{Re}servoir\ Cost_i} \quad for\ Chromosomes\ i = 1,2,...,k \quad (2)$$

Generation of new population using reproduction operator: The GA needs a selection scheme to generate members of the new generation. Thus, weighted roulette wheel method is used herein. The slot sizes on the weighted roulette wheel are selected according to the fitness of each member in the population. The selection operator assigns each string in the population to a segment of the roulette wheel. The size of the segment is proportional to the fitness f_i of the chromosome. The probability of selection of a particular chromosome is

$$P_i = \frac{f_i}{\sum_{j=1}^{k} f_j} \quad (3)$$

Chromosomes with higher fitness (i.e., lower costs) have a higher probability of being selected.

Crossover operator: Crossover creates two offspring chromosomes from two parents by partially exchange of bits between them. The GA randomly picks two chromosomes from the new population and generates a uniformly distributed random number between 0.0 and 1.0. If the random number is less than a predefined value, say p_i, the GA applies the operator otherwise, the GA does not apply crossover for these two particular chromosomes.

Mutation operator: Each chromosome formed as a result of reproduction and crossover is considered by the GA bit by bit. A uniformly distributed random number in the range 0.0 to 1.0 is then generated. If the random number is less than a predefined mutation probability, say p_m, the GA applies mutation by changing the bit value to the opposite value, otherwise, the GA does not apply mutation to that particular bit. Mutation probability is usually chosen between 0.0 and 0.05.

Production of successive generations: A GA with three operators of reproduction, crossover, and mutation are usually known as a standard genetic algorithm. Applying the three operators results in forming a new generation. The least cost solution is then recorded in each generation and process is repeated until no improvement is speculated. Typically a GA will evaluate 100-1000 generations.

5. Results

Numerical experience shows that the GA is very sensitive to penalty cost [7]. In this case, a penalty value of 2000 yields stable results. Figure 7 shows the least cost solutions obtained during 500 generations. As can be seen, the optimal solution is reached at both generation of 24 and 210, respectively.

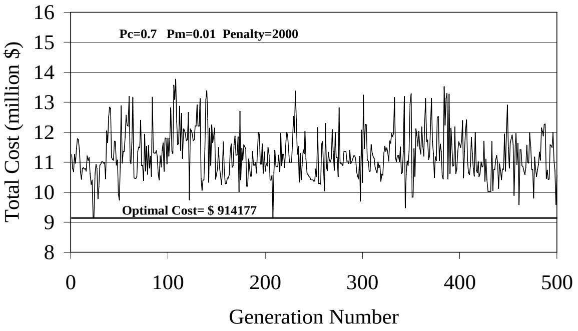

Figure 7. The least cost solution in 500 generations

In this case, the optimal cost is $914,177 which is very close to the global optimal solution ($914,000) achieved by enumeration. Table 5 compares the results obtained from enumeration and the genetic algorithm. The optimal volume arrangements obtained by the GA is somewhat different from that obtained by enumeration. This is likely because two different volume resolutions were used for searching: 2000 m^3 for enumeration and 11 m^3 for the GA.

	Enumeration		GA	
Rservoir No.	**Extension (m^3)**	**Costs ($)**	**Extension (m^3)**	**Costs ($)**
1	10,000	900,000	9713	874,170
3	16000	1,440,000	16522	1,486,980
7	36000	3,060,000	41162	3,498,770
8	44000	3,740,000	38610	3,281,850
Total=	**106,000**	**9,140,000**	**106,007**	**9,141,770**

Table 5. Optimal solution obtained by the GA and the enumeration

According to the information presented in Figure 7, a convergence curve is drawn in Figure 8. As can be seen, convergence is reached quickly in the early generations and no further improvement is achieved in the subsequent generations.

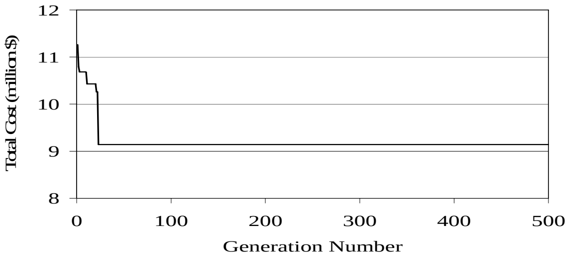

Figure 8. Convergence curve

A final point for debate is why the GA should be used when it is possible to solve by enumeration. Indeed, enumeration is only possible in only limited cases. For example, if the number of decision variables in this case increased by one, the number of required enumeration grows exponentially. Since in this case the most time consuming part of searching is running the hydraulic model, the number of run of the hydraulic model could be regarded as a good indicator for comparing the computational time required for both the enumeration and the GA methods. The number of the hydraulic run in the GA is independent of the number of decision variables and in this case is 550,000 times. Table 6 compares the number of hydraulic run needed for enumeration and the GA.

As can be seen, the number of hydraulic model runs needed by the GA model is only 12.5% of that in the enumeration model. It is also obvious that as the number of decision variables increases, the number of hydraulic model run explosively increases in the enumeration model while it remains constant in the GA model. Although the computational time required by both models was not recorded accurately, the enumeration models required a 23 hour computational time. This simply shows that if the number of decision variables increases by two, it will take around two years for a computer enumeration.

Number of Decision Variables	Enumeration Combinations	Percent Difference
4	4.30E+06	12.8
5	1.07E+08	0.512
6	2.69E+09	0.0205
7	6.71E+10	0.0008
8	1.68E+12	0.00003
9	4.20E+13	1.31E-06
10	1.05E+15	5.24E-08

Table 6. Number of hydraulic model run required by the enumeration compared with GA

6. Conclusion

According to the results obtained in this research it can be concluded that GA is an effective searching tool for the cases that do not have analytically-well defined mathematical constraints. Although GA does not guarantee a global-optimal solution, experiences shows that in many cases the global-optimal solution has been reached. The present study also shows that the proposed methodology can be also used in design where there are many decisions that have to be made. Not only does the optimization allow a more economical modification of the system, but the solution actually achieves what human operation could not – a trouble free operation that avoids the serious difficulties associated with reservoir over-topping or emptying, and the associated serious complications of these states.

References

[1] H.T. Falvey, *Air-water flow in hydraulic structures (engineering monograph No. 41)* (USA: United state bureau of reclamation, 1980).

[2] E.B. Wylie, and V.L. Streeter, *Fluid Transient in systems* (New Jersey: Prentice Hall, 1993).

[3] M.H. Chaudhry, *Applied hydraulic transients* (New York: Van Nostrand Reinhold, 1987).

[4] L.A. Rossman, *EPANET 2 Users, Manual* (Cincinnati: Environmental Protection Agency, 2000).

[5] J.H. Holand, *Adaptation in Natural and Artificial Systems* (Ann Arbor, Michigan: University of Michigan Press, 1975).

[6] A.R. Simpson, G.C. Dandy, & L.J. Murphy, Genetic algorithm compared to other techniques for pipe optimization, *Journal of Water Resources Planning and Management*, 120(4), 1994, 423-443.

[7] B.S. Jung, *Optimization and Calibration of Water Distribution Systems: Exploring Evolutionary Approaches while Accounting for Fluid Transients* (Toronto: PhD thesis, University of Toronto, 2005).