School of Post Graduate Studies
Addis Ababa Institute of Technology

OPTIMIZATION ANALYSIS OF WATER SUPPLY DISTRIBUTION SYSTEM

THE CASE STUDY OF AWASH SEBAT KILLO TOWN DISTRIBUTION SYSTEM

BY GEZAHEGN HAILEMARIAM

Thesis Submitted to Addis Ababa Institute of Technology, School of Post graduate Studies in partial fulfillment of the requirements for the Degree of Master of Science in Civil Engineering (Hydraulic Engineering).

Thesis Advisor
Dr. MEBRATE TAFESSE
Addis Ababa, Ethiopia

OCTOBER 2018

ADDIS ABABA UNIVERSITY

INSTITUTE OF TECHNOLOGY

SCHOOL OF CIVIL AND ENVIRONMENTAL ENGINEERING

HYDRAULIC ENGINEERING STREAM

OPTIMIZATION ANALYSIS OF WATER SUPPLY DISTRIBUTION SYSTEM

THE CASE STUDY OF AWASH SEBAT KILLO TOWN DISTRIBUTION SYSTEM

BY GEZAHEGN HAILEMARIAM

Thesis Submitted to Addis Ababa Institute of Technology, School of Post graduate Studies in partial fulfillment of the requirements for the Degree of Master of Science in Civil Engineering (Hydraulic Engineering).

Thesis Advisor
Dr. MEBRATE TAFESSE

Addis Ababa, Ethiopia

OCTOBER 2018

School of Graduate Studies

Addis Ababa University
School of Post Graduate Studies
Addis Ababa Institute of Technology

Thesis Submitted to Addis Ababa Institute of Technology, School of Graduate Studies in partial fulfillment of the requirements for the Degree of Masters of Science in Civil Engineering with Hydraulic Engineering.

Members of Examining Board

-------------------------------------- --------------------------

(Chairman) (Signature)

DR. MEBRATE TAFFESSE --------------------------

(Advisor) (Signature)

DR. DANIEL F/SELASSIE --------------------------

(Internal examiner) (Signature)

DR. GEREMEW SAHILU --------------------------

(External examiner) (Signature)

Certification

The undersigned certify that he has read the thesis entitled: Optimization Analysis of Water supply distribution system; The case of Awash Sebat killo town water supply system and hereby recommend for acceptance by the Addis Ababa University in partial fulfillment of the requirements for the degree of Master of Science.

DR. MEBRATE TAFFESSE

(Advisor)

Declaration and Copyright

I hereby declare to the Senate of Addis Ababa University that this thesis is entirely original work and all other materials are duly acknowledged. This work has not been submitted for academic degree award at any University.

Declaration

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

ACO = Ant Colony Optimization

CSA = Central Static Agency

ga = Genetic Algorithm

HH = House hold

i.e. = Remember

LP = Linear Programming

m = meter

m^3 = meter cube

m^3/sec = meter cube per second

NLP = Nonlinear Programming

Pcd = percapita demand

PSO = Particle Swarm Optimization

PVC pipe= Polyvinyl Chloride pipe

WDN = Water supply Distribution Network

ABSTRACT

Water supply distribution systems estimated to account of 80 to 85% of the total cost for the whole Water supply project (International Journal of Mechanical Engineering and Technology, 2014), that include source development, pump and motor house (including Pump and Generator) construction of treatment plant and reservoirs. Unless the cost for distribution system made to be decreased or optimized, the increase to the total project cost of the water supply system is inevitable. Thus, the problem is how to implement the water supply distribution network with optimum cost.

Currently, pipe sizes in distribution networks are assigned by trial and error method. Then pressure and velocity values are checked whether they are within permissible values or not.

This study is prepared to show how the formulated methodology using Matlab bring about optimized pipe size and hence reduction in the pipe network cost. The distribution system taken as a case study for analysis is Awash Sebat killo town water supply distribution system.

Eight loops are selected from the total distribution network based on the size of pipes found in the loops. The pipes in the selected loops have larger pipe sizes than other loops. Therefore, it is found reasonable to optimize these loops to get considerable cost reduction.

The optimization process was carried out taking the pressure values, pipe lengths, Nodal demands from the output of Epanet. The main constraint, pressure output, is made to pass through calibration test before it is being used for optimization by Matlab.

Objective and constraint functions are written in Matlab-file. The Objective function is the sum of the product of cost of each type of pipe per unit meter length. The Constraints are: the sums of the head loss around a loop need to be equal to zero, and the pressures at nodes in the loop must be greater than 15m, and flow velocity in the pipes need to be greater than 0.6m/sec.

After optimization process was completed changes for pipe sizes in the selected loops are noted. And these new optimized pipe diameters are entered in the design software (Epanet) to check whether the design criteria were preserved or not. The result shows that the pressure values at the nodes of the considered loops are greater than 15m. The analysis shows that the new optimized pipe diameters satisfy the required demand at the nodes before optimization is carried out.

The total cost of the pipes before optimization is compared with that after optimization. As a result, the total cost reduction is found to be 15.87% of the total cost for the pipes and fittings in the whole network.

Finally, recommendation is suggested to use the methodology developed to attain economical pipe sizes by reviewing any designed water supply distribution system.

ACKNOWLEDGEMENTS

I would like to thank Dr. Mebrate Tafesse, my adviser, for his support to make the thesis to have better quality and Dr. Agizew for his support to submit the final thesis on time. This thesis would not have been possible from the beginning without the help of Mr. Fikru Gemtecha, from WAD consulting Share Company, the consulting firm that made the design of the expansion of the distribution system of Awash Sebat killo town.

I would like to thank my family for their patience while I am doing this work at home.

1 INTRODUCTION

Safe drinking water is one of the basic necessities for human beings. The government of Ethiopia has carried out different program to meet the need of the public demand towards the provision of safe drinking water. For example in 2005 it has endorsed the Universal Access Program (UAP) that enables to provide safe water to all citizens of the nation. In addition, the first Growth and Transformation Plan (GTP-1) covering the period from 2011 to 2015 was also among these different program. This plan was to increase the rural, urban and total access to water supply coverage to 98%, 100% and 98.5% respectively (Ministry of Water and Energy, 2010 GTPI, water sector).

According to the base line survey conducted in nationwide, the Water supply coverage at the end of the GTP-I period in rural and urban area were 75.5% and 84.1% respectively. The figure shows that the achievement of the water supply coverage in rural and urban area left behind against the planned by 75.5% and 84.1% respectively (Ministry of Water and Energy, 2015).

The following are the major reasons GTP-I for not to attain its achievement (Ministry of Water, irrigation and Electricity, water sector)

- In capacity of Water supply provision equivalent to the existing economic growth both in Rural and Town.
- The huge increase of water demand due to rapid growth of life style in urban areas.
- The current capacity of Privet sectors to study, design, contract administration, operation and maintenance of Water supply infrastructures are low
- The constructed water supply structures are not sustainable for the reason that there is no adequate capacity built to run and administer the system.

- The Unaccounted for Water (UFW) in urban area is too high.
- Climate change brought bare lands that contribute to low surface and ground water recharge.

Currently, the second growth and transformation plan for water sector is being implemented. The main goals of the program are the following.

- To make 85 % achievement through improving accessibility of safe water for rural areas, as per 25 liter per head at a distance of minimum 1km.
- To increase the coverage of safe water for all towns 100 liter per head for level one town (Addis Ababa), 80lpcd for level 2 towns, 60 lpcd for level3 towns, 50lit for level 4 towns and 40lit pcd for level5 towns.
- To make the availability of water in all urban areas (973 towns), without interruption at least for 12hrs per day
- At least to lower and sustain Unaccounted for Water (UFW) to 20% in all urban areas.
- To lower the levels of damages on Water supply infrastructures at least to 7%.

Challenges in developing Ethiopian Water supply sector

Developing water supply infrastructures for rural and urban areas have many challenges. The following are among the lists that are applicable for safe Water supply (Abiy Girma, 2013).

- Low level of infrastructure Development
- Uneven spatial distribution of water resource potential
- Low level of Implementation Capacity at different level
- **Shortage of financial resource and huge investment requirement for water projects**

Financial Constraints

The need for Finance to implement water supply projects, in nationwide is huge. For example to implement GTP-II for water supply sector require Birr 82.8 Billion of which Birr 77.8 Billion is capital and Birr 5 billion recurrent budget (Ministry of Water, Irrigation and Electricity, 2015)

In line with this, this thesis tries to show how minimization of network cost is to be achieved taking an existing distribution network as a case.

1.1 PROBLEM STATEMENT

In Ethiopia, the water supply distribution system for a particular town is often designed using Epanet or Water cad software. In the process, the pipe sizes that are available in market selected and assigned by trial-and-error method to meet the design criteria. This method is very tedious and consumes much time and results in un optimized pipe size. In addition it cannot be able to obtain economical pipe size that simultaneously satisfies the design criteria (constraints).

The cost implication of assigning pipe diameter in this way has not been given much consideration as far as the system met the design criteria. In addition, uneconomical pipe network makes the distribution system more costly.

1.2 OBJECTIVE

The general objective of this study is to show how optimization of some selected loops which have larger pipe sizes (have high cost) in the distribution network can contribute to cost reduction taking an existing water supply distribution system as a case study.

The specific objectives of the thesis are:

- To review consultant's document whether the Water supply distribution network is made as per the design guide line of the country (Ministry of Water resources, 2006) such as Reservoir size, Peak hour demand, Distribution pipe sizes, etc.
- To enable the pipe network sizes more economical but to accommodate the peak hour demand.
- To reduce costs of accessories/fittings and gate valves associated with distribution pipes

1.3 Scope

The Water supply network analysis is done giving more consideration for loops which have larger pipe diameters such as DN 200mm, 100mm and 80mm. The minimum resulting optimized pipe diameter must not be less than DN25mm as per the design guide line for the distribution system (Ministry of Water Resources, 2006).The new optimized pipe size is made to discharge the peak hour demand.

2 LITERATURE REVIEW

The first law of thermodynamics states that for any given system, the change in energy (ΔE) is equal to the difference between the heat transferred to the system (Q) and the work done by the system on its surroundings (W) during a given time interval.

The energy referred to in this principle represents the total energy of the system, which is the sum of the potential energy, kinetic energy, and internal (molecular) forms of energy such as electrical and chemical energy. Although internal energy may be significant for thermodynamic analyses, it is commonly neglected in hydraulic analyses because of its relatively small magnitude.

In hydraulic applications, energy values are often converted into units of energy per unit weight, resulting in units of length. Using these length equivalents gives engineers a better “feel” for the resulting behavior of the system. When using these length equivalents, the engineer is expressing the energy of the system in terms of “head.” The energy at any point within a hydraulic system is often expressed in three parts.

Pressure head, P/γ

Elevation head, Z

Velocity head, $V^2/2g$

Where p = pressure (N/m^2) γ = specific weight (N/m3) z = elevation (m) V = velocity (m/s)

In addition to pressure head, elevation head, and velocity head, energy may be added to a system by a pump (for example), and removed from the system by friction or other disturbances. These changes in energy are referred to as head gains and head losses, respectively. Because energy is conserved, the energy across any two points in the system must balance. This concept is demonstrated by the energy equation:

$$P_1/\gamma + Z_1 + V_1^2/2g = P_2/\gamma + Z_2 + V_2{}^2/2g \text{ ……….. (1)}$$

Where, p = pressure (N/m2), γ = specific weight of the fluid (N/m3), z = elevation above a datum (m) V = fluid velocity (m/s), g = gravitational acceleration (m/s2), HG = head gain, such as from a pump (m) HL = combined head loss in m (Garg S.Kumar, 1996).In this thesis the above equation was used for to set up an optimization model. An optimization model is a composition of mathematical equations embodied in the general theory of mathematical programming. It includes a formal search procedure to determine the values of decision variables, for optimizing the Objective function.

Any violation to the values that can be assigned to decision variables can be expressed by mathematically through by means of inequalities or equations. Such mathematical expressions for the restrictions are often called constraints.

The constants in the Constraints and the objective function are the parameters of the model (Slobodan, 2007).

There may be a number of decision variables that can satisfy an objective function in mathematical optimization model. Though these values satisfy the objective function, they may not be the optimum solutions to satisfy the objective function in the given domain. Currently, numerous optimization techniques are being used in solving optimization problems. The principal optimization techniques are Linear Programming (LP), Dynamic Programming (DP), and Nonlinear programming (Vedula and P. Mujumuda 2005).

Linear programming

Linear Programming is one of the most widely used techniques in water resources systems management. It is applied only in problems where both the objective function and the system constraints are linear functions of the decision variables. Linear programming has proven to be an extremely powerful tool, both in modeling real-world problems and as a widely applicable mathematical theory; however, many interesting optimization problems are nonlinear. The general LP problem maximizes or minimizes a linear objective function

Z can be expressed mathematically as:

$Z = C_1X_1 + C_2X_2 + \ldots\ldots\ldots\ldots C_nX_n$ ………………………………………………… (2)

Subjected to the constraints;

$a_{11}\, x_1 + a_{12}x_2 + \ldots\ldots\ldots a_{13}x_3 \leq, =, \geq b_1$

$a_{21}\, x1 + a_{22}x_2 + \ldots\ldots\ldots a_{2n}x_n \leq, =, \geq b_2$

$a_{m1}\, x1 + a_{m2}x2 + \ldots\ldots\ldots a_{mn}x_n \leq, =, \geq b_n$……………………. (3)

And the non-negativity constraint:

$x_1 \geq 0, \ldots\ x_n \geq 0$…………………………………………………………………(4)

Where x_j are the decision variables, c_j, a_{ij} and bi are constants.

Dynamic programming

Dynamic Programming (DP) gives solution procedure by transforming to a sequential or multistage decision problem. Linear and/or nonlinear problems can be solved using Dynamic Programming. DP technique decomposes an N decision problem in to a sequence of N separate, but interrelated, single-decision sub problem. In contrast to linear programming (Svedula pp Mujumda, 2006)

Nonlinear programming

In nonlinear programming, the objective function and/or any of the constraints have nonlinear terms. There is no general solution procedure exist. In some applications, quadratic programmings are available. Linearization of nonlinear function or approximation by piecewise linear segments are some methods used for engineering applications.

The choice of techniques for a given problem depends on the configuration of the system being analyzed, the nature of the objective function and the constraints, the availability and reliability of data, and the depth of detail needed in the investigation. Linear programming and Dynamic programming are the most common mathematical programming models used in Water resource systems analysis (Vedula and P. Mujumuda 2005).

In Optimization analysis of water supply distribution system, nonlinear constraints could be involved in setting the optimization procedure. One of the nonlinear constraints in the WDN is for the criteria that the equation for the sum of head loss around a closed loop is to be zero. The other nonlinear constraint is the pressure at any given node in the distribution system should be in between the design criteria. Therefore, solving an optimization problem for at least for the above two nonlinear constraints require effective techniques to reach at decisions. Meta-heuristic approaches such as Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Genetic Algorithm (GA) can execute nonlinear Objective and Constraints types (International Journal of Mechanical Engineering and Technology, 2014).

2.1 Optimization Techniques of Water supply Distribution system

Over the past years Genetic algorithm (GA) methodology has been developed for pipe-network optimization (Murphy and Simpson, 1992; Dandy et al., 1993; Simpson et al., 1994; Murphy et al., 1993; Dandy et al., 1996; Liberatore et al., 1998).An extensive analysis of GA application to water system design and comparison of commercial optimizers has been reported in a recent work by Lippai et al. (1999). There are some methods similar to Genetic Algorithm which can perform nonlinear objective and Constraint functions. They are reviewed as follows.

2.1.1 Particles Swarm Optimization (PSO)

The method of PSO with weighted parameters was first used by Yuhui Shi et al in 1998.The modified PSO gives better solution for weighted parameter ranges from 0.9 to 1.2. In2008, Joaquin Izquierdo et al have applied PSO in existing problems and concluded that PSO gives better results as compared to other classical methods like dynamic programming. In 2009, Bansal Jagdish Chand et al have optimized both serial and branched networks using particle swarm optimization. They considered the cost per meter length of pipe as continuous function of pipe diameter. They have demonstrated their work with two different networks [series and branched network] and compared the results obtained with conventional methods i.e. Lagrange's Approach and Random Search Technique (RST). It was found that PSO either comparable or better than the conventional method.

In 2010, Yu song et al developed hybrid PSO by combining space transformation search with modified velocity model to overcome the drawback of simple PSO and also to get global optimal results. They applied this model in 8 bench mark problem and shown that this method has good performance for solving both unimodal and multimodal optimization problem. Babu Jinesh et al proposed a hybrid PSO model for water distribution network in 2013.

They have validated the performance of this model with two benchmark problems and found this model to be very efficient for exploring global optimal results with less computational effort. The use of other optimization techniques such as Ant Colony Optimization was also used in 21st century for optimizing water supply networks.

2.1.2 Ant Colony Optimization

In the Ant Colony Optimization (ACO) mathematical background is outlined and possible implementation strategy was suggested for identifying “shortest paths” in water pipe networks. Such shortest paths could be, not only the minimum pipe lengths between nodes of interest, but also the minimum number of valve operations required to keep a flow path active, the minimum number of customers affected during a flow reroute either because of planned maintenance or unplanned water leak conditions, and the minimum pressure drop along a path during adverse conditions. A case study of a specific urban water distribution network is described for the ACO below.

The optimization of water distribution network by Ant Colony Optimization technique was carried out in 2003 by Holger R. Maier et al and the results were compared with genetic algorithm. In one problem, it was found that results from both approaches were closely matching. M.H. Afshar used a different approach for pipe network optimization by using ant colony optimization with pheromone trail. An Ant algorithm with a minimum number of controlling parameters is introduced for pipe network optimization problems. This method uses the interrelation between pheromone change and initial pheromone strength to initialize the pheromone trail strength.

At the start of the computation, this method is shown to be capable of locating the best ever solutions obtained for these problems.

Symeon E. et al applied Ant Colony Optimization in two case studies of pipe routing problem. They observed that this method has quick convergence and hence takes less computational time.

2.1.3 Genetic algorithm

Genetic algorithm (G.A.) was invented by John Holland in 1975. It is a population based heuristic method based on “Survival of the Fittest”. Genetic algorithm is very useful tool for search and optimization problem. In genetic algorithm, a set of population is generated randomly, and then fittest population is selected for reproduction, using cross over and mutation operator. New chromosome is again evaluated for the fitness generation; this process is repeated until the termination condition is reached. In 1994, L. J. Murphy and A.R. Sympson, have applied this approach on pipe network optimization of a given flow requirement at nodal points by satisfying the maximum and minimum pressure constraints. The various options such as lining, duplication or deleting the existing pipe are proposed to get the best optimized solution. Genetic Algorithm uses the objective function value of the proposed network. It selects, combines and manipulates the possible solutions to the pipe network to get the minimum cost of network. It is a robust search method and very useful particularly for pipe network optimization. In 1996, Graeme C. Dandy et al introduced the power gradient and fitness function, creeping mutation operator and grey codes in place of bringing code in simple G.A. This improved G.A. was used for optimization of pipe network and results obtained were found to be significantly improved In 1998 Indrani Gupta et al suggested a methodology based on genetic algorithm for low cost design of new and existing water distribution network. The results obtained from the new method were compared with nonlinear programming method (NLP). The cost obtained by the method based on GA was lower than NLP method.

In 1998, Abebe et al optimized a network using global optimization tool with various random search algorithm, this model can handle both static and dynamic loading condition. They found two algorithms for giving promising solution which are adaptive cluster covering and genetic algorithm.

In 1999, Guoling Xue et al , had worked on the problem for computing the minimum cost of the pipe network, in which wells and treatment sites of given capacity were interconnected with powerful interior point algorithm to find the minimum cost.

In 2001, Z.Y. Wu et al used genetic algorithm for optimization of the large water distribution system using messy genetic algorithm to enhance the efficiency of optimization procedure. They concluded from results that convergence in messy genetic algorithm is faster than genetic algorithm.

In 2004, Keedwell et al worked on CANDA-GA (Cellular Automata for Network Design Algorithm Combined with genetic algorithm).In this approach the initial best solution was found by CANDA and then GA is applied to refine the solution. It is found from 3 different case studies that better optimized solution is obtained in less time.

In July 2007, Apple L.S. Chan et al optimized the pipe network for district cooling system with the use of genetic algorithm combined with the local search technique. They worked to get the optimum configuration of District cooling system with minimum pipe work and pumping cost. The results obtained from genetic algorithm are a global solution. It does not refine the solution efficiently. That's why a local search technique is associated with the G.A., so that best results can be obtained.

In 2011, S. Chandramouli and P. Malleswararao worked on the optimization of pipe network based on the reliability. Their focus was to improve the reliability based on the excess pressure available at demand nodes, using fuzzy logic concept and to incorporate it in the optimal design. For this two objective optimization, they used Genetic algorithm and EPANET tool kit in the Matlab.

Optimization of pipe network by using GA has some limitations like giving global solution and poor convergence. James Kennedy and Russell Eberhart developed the concept of Particle Swarm Optimization (PSO) in 1995 which has overcome the limitations of GA. Particle swarm optimization is a robust stochastic technique for optimization. In this method, the co-ordinates of each particle represent the possible solution and after each iteration, the particle moves towards optimal solution.

The basic algorithm for a GA is shown below.

(http://www.edc.ncl.ac.uk/highlight/rhjanuary2007g01.php)

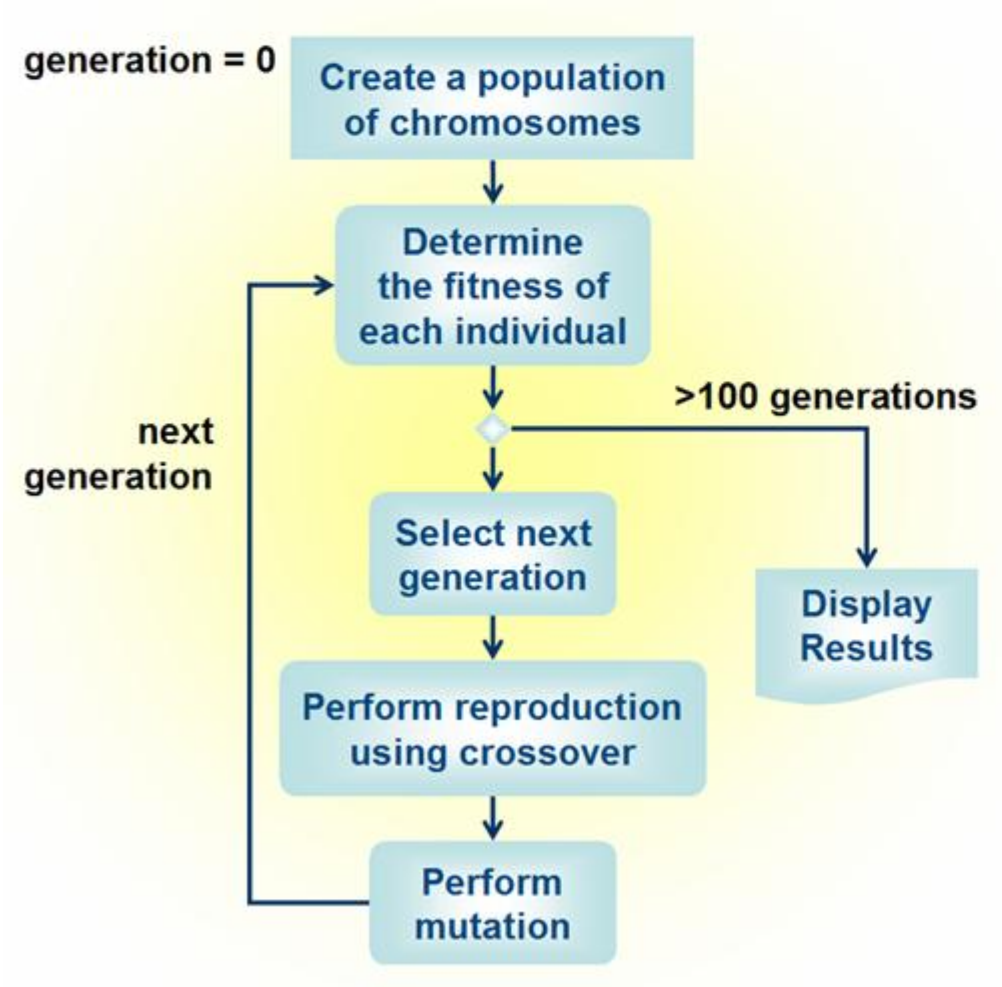


Figure-1 Basic Genetic Algorithm

The Algorithm

The basic algorithm by which GAs operate is reasonably well established. The above figure shows the generic one used by most, although some variations may occur depending on the application. In general the particular methods employed in each of the above stages do not matter. What is important is that the general algorithm is followed and the evolutionary techniques that underlie it are understood. The following sections discuss the need for each of these steps in terms of their relevance to evolutionary processes.

Population

A population is initiated of legal solutions, selected by choosing random input values. There are no fixed rules for how large the population should be. The answer is dependent upon the type of problem. For a simple problem with a regular search space a small population of 40 to 100 will probably be sufficient. For larger and more complex problems and especially those with an irregular search space, larger populations of 400 or more are recommended. The clue is a diverse population.

Fitness

The fitness of individual chromosomes is a relative matter. For example when maximizing a function; if one individual has a higher value, once processed by the function, than another then that individual is considered fitter. Things get a little more involved with multi-criteria problems. In these cases comparisons can be carried out to see if an individual dominates other members of a population by taking all criteria into consideration. If they do they are considered fitter. The most dominant, i.e. those who dominate all others are referred to as Pareto solutions. These are considered as candidate solutions to whatever problem you are trying to solve.

Selection of the Fittest

GAs operates over a number of generations. Following the evolutionary theme of this method, this means fitter solutions will tend to survive to the next generation. The selection method employed by many approaches is the roulette wheel selection process.

In nature all individuals have a chance of surviving from one generation to the next – fitter solutions (i.e. those most dominant) have a better chance. Weaker more dominated individuals have a smaller chance (still an opportunity) of surviving.

Crossover

Nature generates the next generation using a mating process. As a result two parents create offspring, who consist of the genetic material of both parents. These offspring can be weaker or fitter than their parents (or similar). If they are weaker they will tend to die out – if they are stronger their chances of survival are better. Gas tries to replicate this using a crossover operator. This emulates the mating process by exchanging chromosome patterns between individuals to create offspring for the next generation.

Mutation

Mutation exists in nature and causes an unanticipated change in a chromosome pattern. This can result in a much weakened individual and occasionally a much stronger one. Either way the principle behind mutation from an evolutionary point of view is that it occurs rarely, spontaneously and without reference to any other individual in the population. If the change is beneficial to the general population then that individual will tend to survive and will pass this trait on in future replication processes. Because of the way that GAs represent individuals this process is a very simple one and a typical mutation operator is relatively easy to implement. It is important to remember that these processes occur very infrequently otherwise they would have a disruptive effect on the overall population.

Mutation operator

Given that the previous parts of the general algorithm have measured fitness, performed selection based on fitness and then emulated the mating process, it may be assumed that enough distribution of the chromosome gene patterns has taken place to fully investigate all possible solutions. This is not so. If we consider the spread of chromosome patterns that existed with the initial random population it is possible to see that the processes carried out so far have not actually introduced any new variation in patterns.

3 REVIEW OF THE CONSULTANT DESIGN DOCUMENT

The average daily water demand of the town calculated by the consultant in the design document is made taking the following factors;

Daily peak demand factor = 1.15

Factor for seasonal variation = not taken

Peak hour demand = 1.8

The consultant has calculated the peak hour demand taking only a multiple of peak hour demand factor (1.15) and average day demand. Rather the peak hour demand should be the product of peak hour factor, Peak day demand (which is 1.3*average day demand), and factor for Seasonal variation (1.2 for dry area such as Awash Sebat killo town), according to the design guide line (Federal Democratic republic of Ethiopia, Ministry of Water resource, 2006).

The value of factors for Peak day demand, Peak hour and seasonal variations and how to apply on to the average demand to calculate Peak hour demand is stated in the design guide line and are listed below.

Daily peak demand factor = 1.3

Factor for seasonal variation = 1.1 to 1.2 (depends on the climatic condition of the area)

Peak hour demand = 2 (for population less than 20,000)

To correct the value of peak hour demand calculated in the design document, first the average day demand taken by the consultant has to be known. Thus, dividing the peak hour demand (of the consultant) by 1.8(Factor of peak hour used by the consultant),

Average day demand = 26.4 l/s/1.8 = 14.67 l/s

Corrected peak day demand = 14.67*1.3(factor of peak day demand) = 19.07 l/s.

To prove the validity of the corrected peak day demand, the existing reservoir volume has to be checked whether it is **one third of the corrected peak day demand** (Federal Democratic republic of Ethiopia Ministry of Water resource, 2006). Therefore,

Reservoir capacity = 1/3* peak daily demand.

= 1/3*19.07l/s (corrected peak day demand)

= 1647.648 lit/sec or 543m^3/day, which approximates the existing reservoir volume, 500m^3.

Finally, the corrected peak hour demand by which the distribution network had to be designed is calculated as follows;

Peak hour demand = 2(peak hour factor)*19.07(peak day demand)*1.2(Seasonal variation factor for dry area)

Peak hour demand = 45.7 l/s.

Compared to the peak hour demand of the consultant, 26.4 l/s, the difference is big. As a result the velocity output of the Epanet became too less.

Taking the corrected peak hour demand, optimizations of the selected loops are analyzed using the methodology formulated here after.

4 MATERIAL AND METHOD

4.1 Description of the study area

Awash Sebat killo is located in Eastern part of Ethiopia, in Afar national regional state, located at 223km North-East of Addis Ababa. Awash Sebat killo is found at 986m above sea level. It has hot and dry climatic condition. The mean annual temperature of the town is 26.76^0C while the mean annual rainfall is 568mm.

The projected population of the town was 18034, by the year the design was made (WAD Consulting Share Company, 2012). All components of the water supply project has been designed using the demand for this population size (WAD Consulting Share Company, 2012).

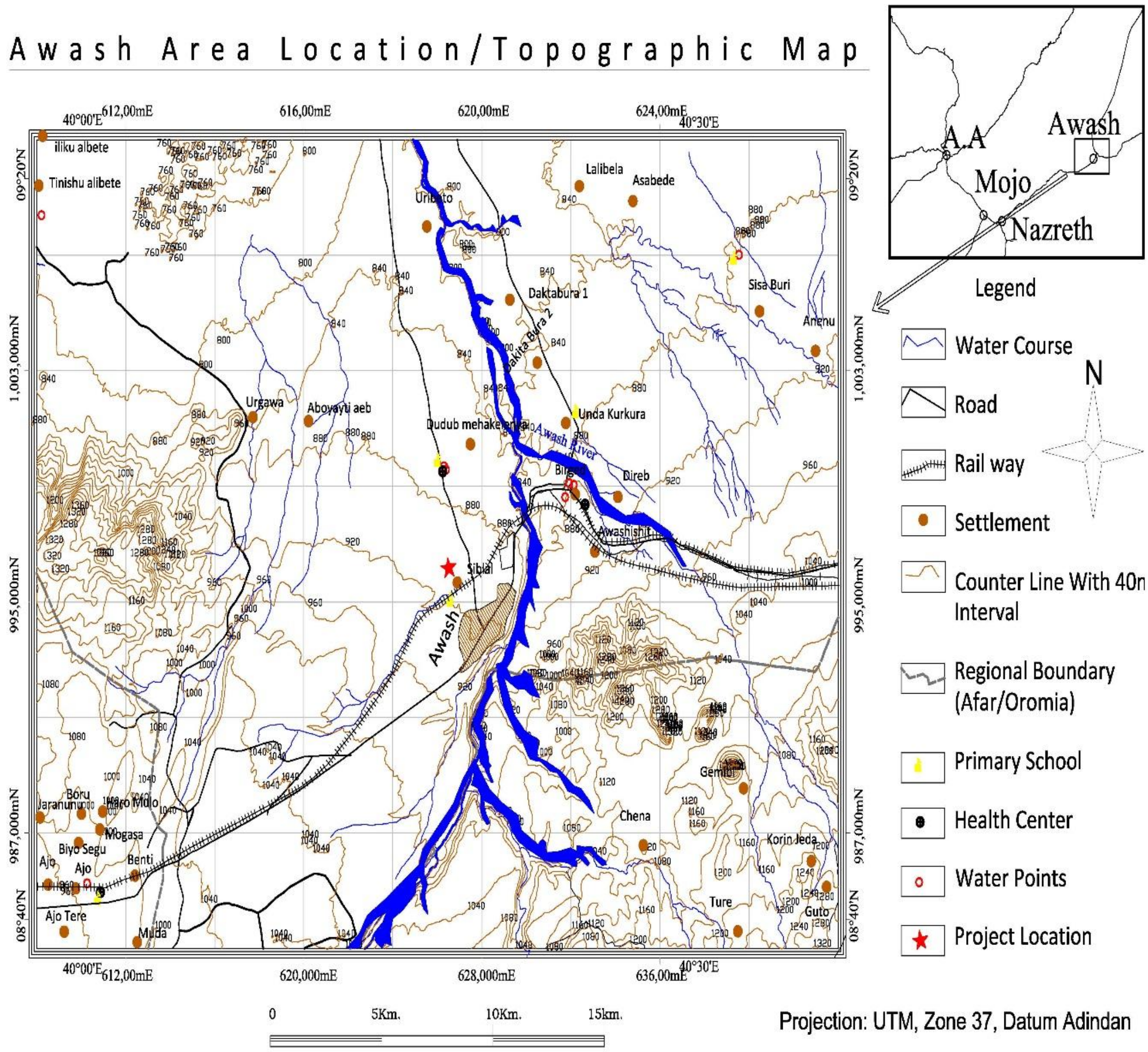


Figure-2 Awash and Awash area location

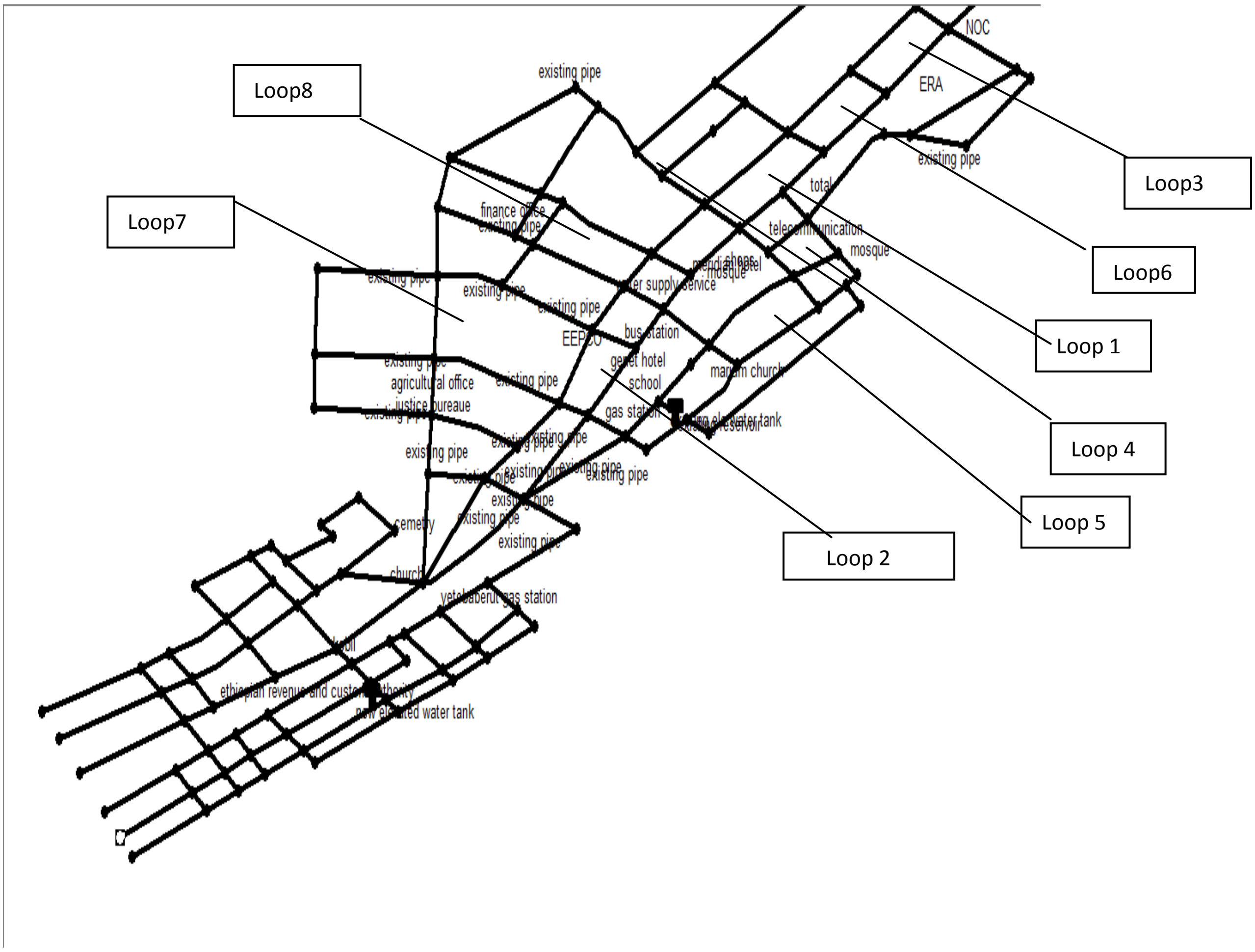


Figure-3Awash Water Supply distribution network layout

4.2 Awash Sebat killo town Water supply system

The Water source is Surface water (Awash River). The existing water supply system has units like raw water intake, treatment plant, reservoirs, pump and generator houses and Public taps.

The existing Service Reservoir 500m3 ground level and 250m3 elevated service reservoirs with total capacities of 750m3 is being used in the water supply distribution system.

The distribution network comprises of 400mm to 150mm DCI, 100mm to 50mm GS and 200mm to 50mm diameter UPVC pipe with a design working pressure of 10 bars. The total length of the existing distribution pipe network is 24,527.5m

As per the monthly water consumption records of the town water supply service, the average daily water consumption is estimated at 385m3/day or 25.9 l/c/d (WAD consulting Share Company, 2012)

4.3 Software and tools

In this study, Matlab is chosen to carry out the optimization of pipe size. Matlab (matrix laboratory) is a high level programming language and interactive environment for numerical computation, visualization and programming. It provides vast library of mathematical functions for linear algebra, statistics, Fourier analysis, filtering, **Optimization,** numerical integration and solving ordinary differential equations. It allows matrix manipulations; plotting functions and Data; implementation of algorithms; creation of user interfaces; including C, C++, and performing numerical methods**. Matlab provides simple and readily available structured statements to compute nonlinear objective and Constraint functions. Unlike other methods which requires preparing a program** such as C++, FORTRAN or other computer language, Accordingly, MATLAB version of 2016 (R2016a) is used in this thesis. The Epanet 2.0 output of the existing water supply system such as pressure and length of pipes are used in the study in order to use them for optimization being processed by Matlab after calibration test for Epanet is made.

4.4 Model Calibration and Validation (Epanet)

It is explained that the study is to analyze the existing water supply distribution system which its layout was modeled using Epanet2.0. The model needs to be verified and calibrated before its output is being used for this study. Accordingly, the model calibration and validation is performed and presented as follows.

Junctions of the main and secondary lines, which have Gate valves to water point branches, were selected to get values of pressures. There are 22 Water points in the distribution system, among them seven was permitted by the Water supply service of the town, to access them and take the measurements.

Table 1-Time series of pressure calibration (Measured vs. Epanet)

Time	Junction to Water points at	Measured Pressure on the site(x)	Model (Epanet) Pressure(y)
8:00Hrs	**Mariam church(J-101)**	27.00	26.94
9:00Hrs	**EEPCO(J-76)**	30.6	30.41
10:00Hrs	**Genet hotel (J-93)**	25.85	25.74
11:00Hrs	**Tele. Office (J-108)**	32	32.79
12:00Hrs	**Back, Total Gas station(J-90)**	33.2	36.25
13:00Hrs	**Near to finance office(J-78)**	30	33.93
14:00Hrs	**Near to water supply service(J-105)**	32	33.34

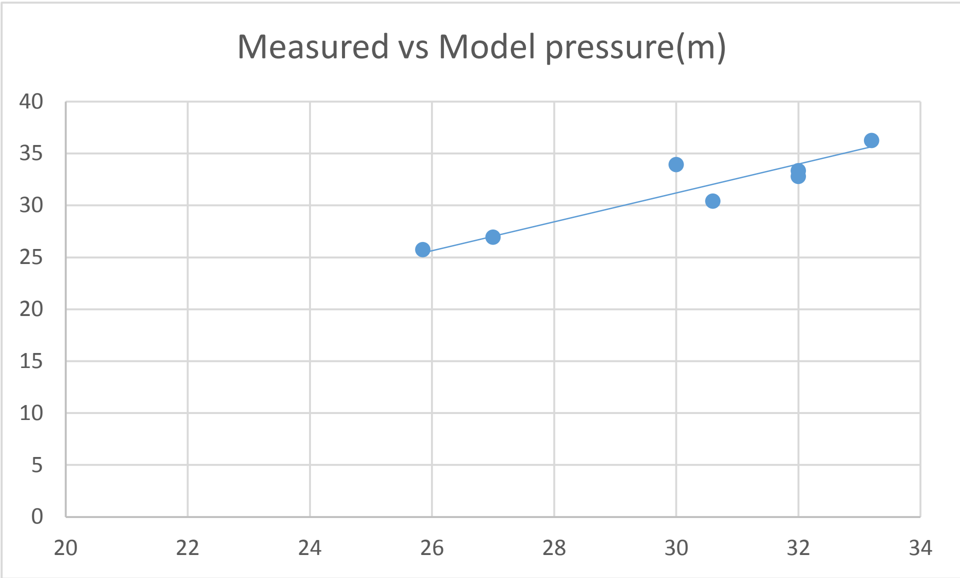


Figure 4- Pressure measured on ground Vs Pressure by Model (Epanet)

$$R^2 = \frac{\sum(X-\overline{X})(Y-\overline{Y})}{\sqrt{\sum(X-\overline{X})^2\,\sum(Y-\overline{Y})^2}}$$

Where,

- R is Person correlation
- X: Measured pressure on the field
- Y: Computed pressure by the model
- $\bar{X}$: Mean value of X
- $\bar{Y}$: Mean value of Y

Hence,

$$R^2 = 0.9293$$

The result shows that pressure value of the Epanet has strong correlation with the measured value obtained from the ground.

4.5 Optimization procedure

Pipe diameters in the loops are pipes of DN200mm, DN100mm, DN80mm, DN65mm and DN50mm. **Most of the Loops selected for optimization are those which comprise**

Pipes with DN200mm, DN100mm and DN80mm, considering they are more costly than others.

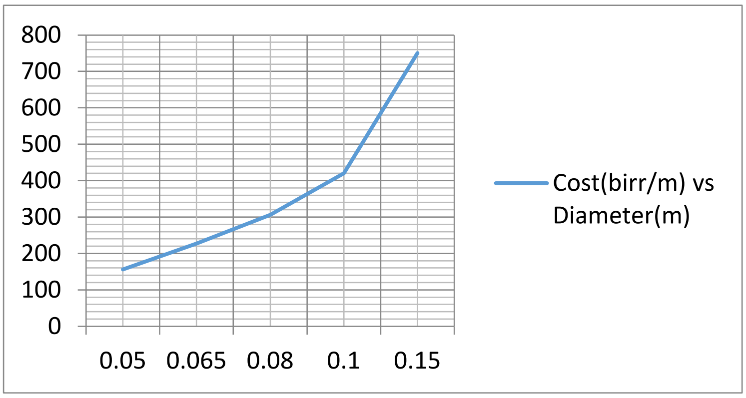
The objective function to be minimized is the sum of the product of cost of the pipes per meter length and total length of the pipes found in each loop. There are three Constraints. The first is the sum of head loss around a loop must be equal to zero and the second is the minimum pressure at nodes with in a loop must not be less than 15m. The third one is velocity in each pipe to be greater than 0.6m/sec (as it was stated in the design criteria). Detail explanation is given below.

The Objective function

The objective function is the total cost 'C' of pipes in the loop.

$$\mathbf{C} = \sum_{j=1}^{i=N} ci\,(Di)\,Li \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(5)}$$

Where, N is the total number of pipes in the individual loops, c_i (D_i) the cost of pipe i with diameter D_i, per unit length of a pipe i and Li is the length of the pipe in the loop. The equation is developed from the relation of **costs of pipe diameter (m), per meter length** relation. Costs of different uPVC pipe diameters with 10bar are collected from the design report. The developed curve and its equation are shown below.



Pipe Diameter (m)	Cost (birr/m)
0.05	156.62
0.065	218.94
0.08	289.52
0.1	450
0.15	675.77
0.2	855

Figure 5-Cost-diameter Curve

$ci = 11295\ (Di)^{1.429}$

Where, ci and Di are variables for Cost in birr of a pipe with diameter Di in meter, respectively.

Thus, equation 5 is replaced with the following equation and written in Matlab M-file.

C = $(11295*D_1^{1.429})*L_1 + (11295*D_2^{1.429})*L_2 + 11295*D_3^{1.429})*L_3 + 11295*D_4^{1.4616})*L_4$………… (6)

Where '**C**' is the total cost of the pipes in the loop, D_1….are pipe diameters (decision variables), and L_1…..are length of pipes with diameter D_1……

As explained before, the first constraint is the sum of head loss around the loop must be equal to zero, taking clockwise direction as positive and anti-clockwise as negative,

$\sum HL = 0$…………………. (7)

In the original design, **Hazen – William formula** was used to compute head loss. The same is done in this analysis too.

$HL = 10.67*L_i\ Q_i^{1.85} / (C^{1.85} * D_i^{4.87})$…Hazen – William equation for head loss, HL… (8)

Where Q_i is discharge in pipe (m3/sec), Di is pipe diameter (m), Li is length of pipe (m) and C is Hazen-William's coefficient.

The **second** constraint is formulated for pressure values at nodes must not be less than 15m, using Bernoulli's equation of energy between successive nodes in the loop in the direction of flow from inlet to outlet. The **third** constraint is velocity of flow in pipes of the loop should be greater than 0.6m/sec. The pressure constraints are written for each node in the loop following the direction of flow from entry to outlet of the loop. Since the velocity at entering and leaving point of the pipes are equal, the velocity head between two successive nodes cancelled each other.

Therefore, Bernoulli's equation between entering point (node1) and node 2 is written as follows.

z1 + p1= z2 + p2 + hL1……… (9) Bernoulli's equation between two nodes.

Where, Z1 and Z2 are elevations at two consecutive nodes, following the direction of flow

p1 and p2 are pressure at two consecutive nodes, following the direction of flow

hL, head loss in pipe from node1 to 2.

The value of pressure, p1, at entering point of flow into each loop is taken from the design output of Epanet, which passed through calibration test.

But from (9), p2 = (z1– z2) - hL1+ p1……………. (10)

Since p2 shall be greater than the minimum pressure 15m (design criteria), equation (10) becomes

(z1– z2) - hL1+ p1 > 15…….constraint for pressure at node2

For the third node (at the end of second pipe in the loop), the pressure constraint is written as follows.

z2 + p2= z3 + p3 + hL2 ----- (11)

Where p3, z3 are pressure and elevation at node 3 respectively and hL2 is head loss in the second pipe of the loop under consideration.

In this equation, p2 is substituted by (z1 - z2) +p1- hL1 (equation 10) to use the known variable p1, then solving for p3 equation11 becomes

P3 = (z2-z3) + (z1 - z2) +p1- hL1-hL2, since p3 should be greater than 15m,

(z2-z3) + (z1- z2)-hL1-hL2+p1 > 15…….constraint for pressure at node3

Pressure constraint for the next node goes like this pattern until the final node (the outlet of the flow from the loop).

For the final node, for the loop contains four pipes (4nodes), from flow entry to leaving the loop, the equation is written as;

(z3-z4)+ (z2 – z3) + (z1-z2) - hL1-hL2 –hL3+p1 > 15…….constraint for pressure at node4

Where z4 is elevation at node 4 and hL3 is head loss in the third pipe of the loop under consideration. The other variables are as explained before.

In Matlab, the ‘greater than’ inequality is written changing into ‘less than or < as an example pressure constraint for node 3 is written as follows by putting semicolon at the end.

- (z2-z3) - (z1- z2) +hL1+hL2-p1 +15;……………….. (12)

The same is procedure is followed to other nodes.

In all the above constraints, the decision variable ‘D’ is found in head loss function, hL, for all respective pipes in the loops

hL1 = $10.67 * L_1 Q_1^{1.85} / (C^{1.85} * D_1^{4.87})$

hL2 = $10.67 * L_2 Q_2^{1.85} / (C^{1.85} * D_2^{4.87})$

hL3 = $10.67 * L_3 Q_3^{1.85} / (C^{1.85} * D_3^{4.87})$

hL4 = $10.67 * L_4 Q_4^{1.85} / (C^{1.85} * D_4^{4.87})$……..

The lower and upper boundary of the market available pipe diameter are specified on the command window of Matlab and entered in ga (Genetic Algorithm) function, which will be discussed here after.

Constrained Minimization Using Genetic Algorithm

For constrained minimization of an objective function, Matlab provides the command ga, which stands for Genetic Algorithm. The Objective function and constraint function are coded in a Matlab M-file and saved as ‘objfun’.m and ‘constraint function’.m respectively in the working directory. Details how to call the objective and constraint functions involving linear equality, nonlinear equality, nonlinear equality or nonlinear inequality on Matlab command window is explained in Annex section.

Coding the objective Function

To illustrate how the aforementioned objective function was created in Matlab, one loop is taken and coded in Matlab. The file is named as Loop1.m and the following function is written in it.

```
function C =Loop1(D)
 C =
(11295*(D(1))^1.429)*181+(11295*(D(2))^1.429)*203+(11295*(D(3))^1.429)*
153....
     +(11295*(D(4))^1.429)*386+(11295*(D(5))^1.429)*150;
```

Where, `(11295*D(1)^1.429)` is Cost of the pipe per meter in birr(c), and `194` is length of the pipe in meter for Pipe1

Coding the Constraint Function

Constraint for pressure at node one (z1 – z2) - hL1+ p1 > 15, is written in Matlab as - (z1 – z2) + hL1- p1 + 15

Constraint for pressure at node two (z2-z3) + (z1- z2)-hL1-hL2+p1 > 15, is also written in Matlab as - (z2-z3) - (z1- z2) +hL1+hL2-p1 +15

It continues like this for the rest of nodes up to the node at which the flow leaves the loop.

A MATLAB file named as Loop1constraint.m is created and all the constraints in the above form are written/coded as follows;

```
function [c,ceq] = Loop1constraint(D)
%headloss constraint
c1 = -1.03*10^-5*D(1)^-4.87-5.49*10^-6*D(2)^-4.87+8.94*10^-7*D(3)^-
4.87+1.48*10^-5*D(4)^-4.87....
    +6.82*10^-9*D(5)^-4.87;
%pressure constraints
c2 = -(920-919)-31.72+1.03*10^-5*D(1)^-4.87+15;
p2 = (920-919)+31.72-1.03*10^-5*D(1)^-4.87;
c3 = -(919-918)-p2+5.49*10^-6*D(2)^-4.87+15;
p3 = (919-918)+p2-5.49*10^-6*D(2)^-4.87;
c4 = -(920-916)-p3+6.82*10^-9*D(3)^-4.87+15;
p4 = (920-916)+p3-6.82*10^-9*D(3)^-4.87;
c5 = -(916-915.3)-31.72+1.48*10^-5*D(4)^-4.87+15;
p5 = (916-915.3)+31.72-1.48*10^-5*D(4)^-4.87;
c6 = -(915.3-918)-p5+8.94*10^-7*D(5)^-4.87+15;

%velocity constraints
c7 = D(1)-(4*4.72*10^-3/0.6*3.14)^0.5;
c8 = D(2)-(4*3.16*10^-3/0.6*3.14)^0.5;
c9 = D(3)-(4*0.1*10^-3/0.6*3.14)^0.5;
```

```
c10 = D(4)-(4*3.81*10^-3/0.6*3.14)^0.5;
c11 = D(5)-(4*1.38*10^-3/0.6*3.14)^0.5;

c = [c2;c3;c4;c5;c6;c7;c8;c9;c10;c11];
ceq = [c1];
```

c1 is nonlinear equality constraint stands for the sum of head loss around a loop is equal to zero.

c2 to c6arenonlinearinequalities stands for pressure constraints, based on Bernoulli's equation written for each node within the loop.

c7 to c11 are nonlinear inequalities stands for velocity constraints written for each pipes within the loop.

P2 to p5 are pressure values obtained by optimization.

c is to indicate that c2……c11 are nonlinear inequality constraints.

ceq is to show that c1 is no nonlinear equality constraint

Minimizing Using Genetic Algorithm (ga)

To minimize the objective function using the Genetic Algorithm program in Matlab, there is a function handle to access the objective function, the number of decision variables, Lower and Upper bounds of the pipe diameters called ga function handle. It also helps to access the nonlinear constraint function written in Matlab M-file.

For instance, to access and minimize the objective function with the nonlinear constraints for loop1in Matlab M-file, the following 'function handle' is written on command page of Matlab.

```
ObjectiveFunction = @Loop1;
nvars = 5;    % Number of variables
LB = [0.025 0.025 0.025 0.025  0.025 ];   % Lower bound
UB = [0.1 0.1 0.1 0.1 0.1 ];  % Upper bound
ConstraintFunction = @Loop1constraint;
```

```
rng(1,'twister') % for reproducibility
[D,fval] = ga(ObjectiveFunction,nvars,[],[],[],[],LB,UB,ConstraintFunction)
```

The number of pipes diameters to be optimized indicated in ‘nvars’. The market available pipe diameters also specified in LB as lower boundary and UB as upper boundary.

The new pipe diameters (the optimized pipe size), then, assigned for the pipes in the loop under consideration in the Epanet Model.

5 RESULT AND DISCUSSION

5.1 RESULT

After running the functions, on command page of Matlab, the new optimized pipe diameters are shown at the bottom of the function handle. The optimization **result** of Loop1 submitted as an example shown below.

```
>> ObjectiveFunction = @Loop1;

nvars = 5;    % Number of variables

LB = [0.025 0.025 0.025 0.025  0.025 ];   % Lower bound

UB = [0.1 0.1 0.1 0.1 0.1 ];  % Upper bound

ConstraintFunction = @Loop1constraint;

rng(1,'twister') % for reproducibility

[D,fval] = ga(ObjectiveFunction,nvars,[],[],[],[],LB,UB,ConstraintFunction)

Optimization terminated: average change in the fitness value less than
options.FunctionTolerance

 and constraint violation is less than options.ConstraintTolerance.

D =     0.0635    0.0544    0.0411    0.0638    0.0415
fval =    1.9706e+05
```

The optimized Matlab output of the pipe diameter then approximated to the nominal pipe diameter, as DN80, DN65, DN50, DN80, and DN25.

In the same wayloop2, loop3, Loop4, loop5, loop6, loop7and Loop8 (figure-2) are optimized and their new pipes diameter shown in the Annex section. Substituting the new optimized pipe diameters in the Epanet software, the following Epanet output is obtained and tabulated below for comparison.

Matlab results of optimized pipe diameters for the selected loops are attached in the Appendices section.

These optimized pipe results can be checked by copy and paste the objective and Constraint functions from appendices and saving in the working directory of Matlab (creating any folder). To get the optimized pipes in each loop using the objective and constraint function created in Matlab directory (folder), the following statements has to be copied from the appendices and paste on Matlab command window for that particular loop.

Copy from

"ObjectiveFunction=@Loopx"

to "

D,fval]=ga(ObjectiveFunction,nvars,[],[],[],[],LB,UB,ConstraintFunction)"

and paste on Command window of Matlab. The variable 'x' is to represent the Loop number; 1, 2, 3…etc.

Then run the Matlab command to get the optimized Pipe diameters.

Table 2 Demand and pressure before and after optimization

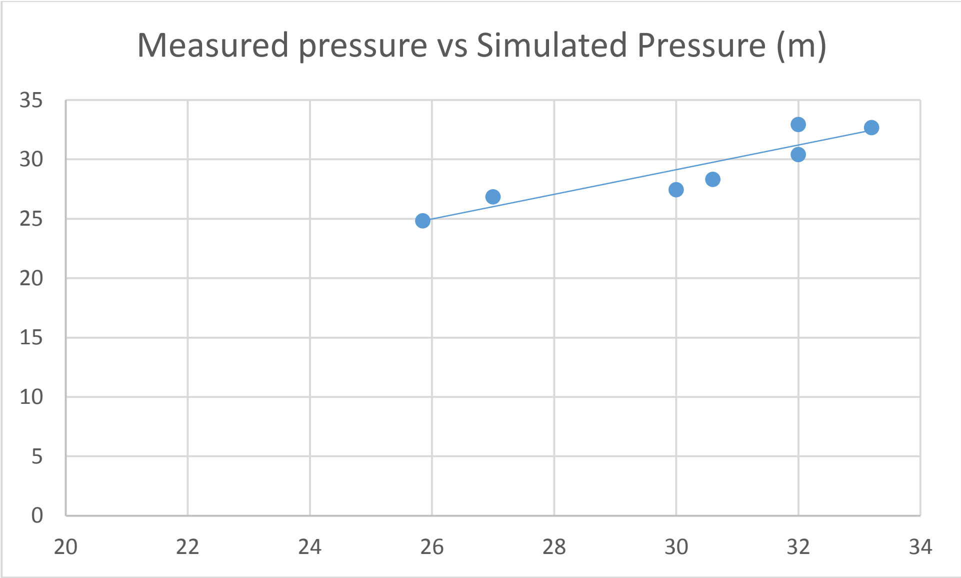
Before optimization				**After optimization**			
Network Table - Nodes				**Network Table - Nodes**			
	Demand	Head	Pressure		Demand	Head	Pressure
Node ID	LPS	m	m	Node ID	LPS	m	m
Junc 67	1.2	947.65	21.15	Junc 67	1.2	945.34	18.84
Junc 71	1.74	946.97	24.97	Junc 71	1.74	937.11	15.11
Junc 73	1.2	948.03	27.03	Junc 73	1.2	942.73	21.73
Junc 75	1.01	948.61	31.61	Junc 75	1.01	945.41	28.41
Junc 76	1.2	949.94	28.94	Junc 76	1.2	949.43	28.43
Junc 77	2.08	948.89	33.89	Junc 77	2.08	945.2	30.2
Junc 78	0.04	948.79	31.79	Junc 78	0.04	945.02	28.02
Junc 90	0.76	951.29	35.99	Junc 90	0.76	948.14	32.84
Junc 91	0.9	951.67	35.67	Junc 91	0.9	951.33	35.33
Junc 92	0.76	951.67	31.67	Junc 92	0.76	951.67	31.67
Junc 93	3.74	951.15	25.15	Junc 93	3.74	950.87	24.87
Junc 95	2.9	951.64	22.64	Junc 95	2.9	951.76	22.76
Junc 101	1.2	951.85	26.85	Junc 101	1.2	951.85	26.85
Junc 103	0.34	951.71	26.21	Junc 103	0.34	948.24	22.74
Junc 105	0.95	951.67	33.17	Junc 105	0.95	951.46	32.96
Junc 107	2.42	951.68	31.68	Junc 107	2.42	951.72	31.72
Junc 108	0.72	951.61	32.61	Junc 108	0.72	949.5	30.5
Junc 109	0.67	951.59	31.59	Junc 109	0.67	949.48	29.48
Junc 110	0.74	951.66	29.66	Junc 110	0.74	944.5	22.5
Junc 111	0.08	951.66	30.66	Junc 111	0.08	951.7	30.7
Junc 112	0.57	951.58	30.58	Junc 112	0.57	943.39	22.39
Junc 113	0.39	951.65	28.65	Junc 113	0.39	943.47	20.47
Junc 117	0.42	951.38	33.38	Junc 117	0.42	946.24	28.24
Junc 118	1.11	951.21	35.71	Junc 118	1.11	938.63	23.13
Junc 119	0.44	951.2	37.2	Junc 119	0.44	946.69	32.69

Finally, calibration test is made for the pressure values obtained by Matlab against the actual measured values on ground as shown below

Simulation Calibration and Validation

Table 3- Time series of pressure calibration (Measured vs. Matlab)

Time	Junction to Water points at	Measured Pressure on the site(x)	Model (Matlab) Pressure(y)
8:00Hrs	**Mariam church (J-101)**	27.00	26.85
9:00Hrs	**EEPCO(J-76)**	30.6	28.43
10:00Hrs	**Genet hotel (J-93)**	25.85	24.87
11:00Hrs	**Tele. Office (J-108)**	32	30.5
12:00Hrs	**Back of Total (J-90)**	33.2	32.84
13:00Hrs	**Near to finance office(J-78)**	30	28.02
14:00Hrs	**Near to water supply service (J-105)**	32	32.96



$$R^2 = \frac{\sum(X-\overline{X})(Y-\overline{Y})}{\sqrt{\sum(X-\overline{X})^2\,\Sigma(Y-\overline{Y})^2}}$$

Where,

- R is Person correlation
- X: Measured pressure on the field
- Y: Computed pressure by the model
- $\bar{X}$: Mean value of X
- $\bar{Y}$: Mean value of Y

Hence,

$$R^2 = 0.930702$$

The resulting correlation factor shows that the simulated pressure value (by Matlab) has strong correlation with the measured value obtained from the ground.

Table 4 Pipe size and velocity before and after optimization

Before Optimization			After optimization		
Network Table - Links			Network Table - Links		
	Diameter	Velocity		Diameter	Velocity
Link ID	mm	m/s	Link ID	mm	m/s
Pipe 104	80	0.32	Pipe 104	25	0.58
Pipe 107	50	0.35	Pipe 107	40	0.75
Pipe 109	80	0.42	Pipe 109	50	0.71
Pipe 110	80	0.53	Pipe 110	65	0.84
Pipe 111	50	0.54	Pipe 111	50	0.74
Pipe 117	32	0.29	Pipe 117	25	0.25
Pipe 132	50	0.18	Pipe 132	80	0.76
Pipe 133	80	0.79	Pipe 133	65	1.09
Pipe 135	80	0.74	Pipe 135	80	0.81
Pipe 138	80	0.51	Pipe 138	65	0.58
Pipe 139	100	0.36	Pipe 139	80	0.64
Pipe 150	80	0.28	Pipe 150	32	0.9
Pipe 155	150	0.01	Pipe 155	25	0.13
Pipe 156	80	0.64	Pipe 156	65	0.98
Pipe 158	150	0.07	Pipe 158	25	0.2
Pipe 160	200	0.26	Pipe 160	80	0.94
Pipe 163	100	0.05	Pipe 163	25	1.06
Pipe 164	50	0.12	Pipe 164	25	0.47
Pipe 165	50	0.13	Pipe 165	25	0.34
Pipe 166	80	0.06	Pipe 166	25	0.82
Pipe 167	80	0.05	Pipe 167	25	0.33
Pipe 172	50	0.07	Pipe 172	25	0.46
Pipe 173	80	0.17	Pipe 173	32	0.68
Pipe 175	100	0.2	Pipe 175	50	0.7
Pipe 176	150	0.38	Pipe 176	65	0.95
Pipe 177	150	0.27	Pipe 177	65	1.24
Pipe 178	80	0.02	Pipe 178	25	1.01
Pipe 179	100	0.15	Pipe 179	65	0.5
Pipe 180	100	0.11	Pipe 180	32	0.92
Pipe 181	80	0.14	Pipe 181	32	0.74

Table 5 Comparison of Pipe cost before and after optimization

	Before Optimization				After Optimization			
Loop	Pipe diameter (mm)	Pipe Length (m)	Price per meter (Birr)	Total price (Birr)	pipe diameter (mm)	Pipe Length (m)	Price per meter (Birr)	Total price (Birr)
loop - 1	200	181	855.00	154,755.00	80	181	289.52	52,403.12
	150	203	675.77	137,181.31	65	203	218.94	44,444.82
	100	153	450.00	68,850.00	50	153	156.62	23,962.86
	100	150	450.00	67,500.00	80	150	289.52	43,428.00
	50	386	156.62	60,455.32	25	386	58.02	22,393.90
Loop -2	100	298	450.00	134,100.00	80	298	289.52	86,276.96
	80	153	289.52	44,296.56	80	153	289.52	44,296.56
	50	299	156.62	46,829.38	50	299	156.62	46,829.38
	80	1000	289.52	289,520.00	65	1000	218.94	218,940.00
Loop - 3	80	147	289.52	42,559.44	25	147	58.02	8,528.94
	100	308	450.00	138,600.00	65	308	218.94	67,433.52
	80	146	289.52	42,269.92	32	146	82.56	12,053.09
	150	303	675.77	204,758.31	50	303	156.62	47,455.86
Loop -4	50	168	156.62	26,312.16	25	168	58.02	9,747.36
	80	164.5	289.52	47,626.04	25	164.5	58.02	9,543.52
	100	120	450.00	54,000.00	25	120	58.02	6,961.84
	50	151.5	156.62	23,727.93	25	151.5	58.02	8,789.32
Loop - 5	80	373	289.52	107,990.96	32	373	82.56	30,793.17
	80	149	289.52	43,138.48	25	149	58.02	8,644.28
	80	110	289.52	31,847.20	25	110	58.02	6,381.68
	50	369	156.62	57,792.78	32	369	82.56	30,462.95
Loop - 6	150	291	675.77	196,649.07	50	291	156.62	45,576.42
	80	147	289.52	42,559.44	25	147	58.02	8,528.94
	100	153	450.00	68,850.00	50	153	156.62	23,962.86
	100	292	450.00	131,400.00	25	292	58.02	16,941.84
Loop - 7	50	299	156.62	46,829.38	25	299	58.02	17,346.57
	80	416	289.52	120,440.32	40	416	116.47	48,451.52
	50	312.5	156.62	48,943.75	50	312.5	156.62	48,943.75
	80	313	289.52	90,619.76	65	313	218.94	68,528.22
	80	209.5	289.52	60,654.44	50	209.5	156.62	32,811.89
Loop - 8	150	167	675.77	112,853.59	65	167	218.94	36,562.98
	80	323.5	289.52	93,659.72	25	323.5	58.02	18,769.47
	50	197	156.62	30,854.14	65	197	218.94	43,131.18
	80	315	289.52	91,198.80	25	315	58.02	18,276.30
Total Cost (Birr)				2,801,384.38	Total Cost(Birr)			1,178,281.88
Cost difference Before and After Optimization(Birr)								1,623,102.50

Following the optimized pipes, cost reduction due to the decrease of the pipe size is summarized in the above table. The cost reduction as a result of pipe optimization is birr 1,623,102.50 that is 15.87% of the total cost of pipes and fittings.

5.2 DISCUSSION

5.2.1 Pressure

As shown in the result (table2), the pressure values at all nodes are greater than 15m. The maximum and minimum pressure **before optimization** was 37.2m and 21.15m at nodes 120 and 67 respectively as shown in table-2. **After optimization,** the maximum and minimum pressure became 35.33m and 15.11 at Node 91 and 71 respectively. However, in both cases, the optimized pipes results in decrease in pressure, they are above to the minimum pressure design criteria (constraint) that is 15m.

5.2.2 Velocity

The velocity in the optimized loops got increased from what they were before (As shown in table3).In some loops Matlab has given the optimized pipe size below DN25mm that give velocity greater than 0.6m/sec. However, the minimum pipe size that has to be laid in main lines has to be DN25mm (Federal Democratic Republic of Ethiopia, Ministry of Water Resources, 2006). Due to this reason, the optimized pipe sizes that were found below 25mm made to be increased to the minimum uPVC-PN10 pipe size, DN25mm. As a result, the constraint that the velocity not has to be less than 0.6m/sec is violated.

5.2.3 Flow rate

It can also be observed from the table-2 that after optimization, the flow in the pipe keep satisfying the required demand at each node that were before optimization was carried out. The demands at each node are designed for the peak hour demand (WAD consulting Share Company, 2012). The new optimized pipes have the capacity to convoy the peak hour demand as noted in table-2.

5.2.4 Head loss

Head loss is related to velocity and pipe roughness. The maximum head loss with therefore be governed by the maximum velocity criterion (Ministry of Water Resources, 2006). As it is seen from table-3, head loss increased in optimized pipes than those were before. But as none of the pipe section has velocity greater than the maximum velocity, the head loss values in the optimized pipes are acceptable.

5.2.5 Reliability of the network

Reliability is defined as the probability that a system does not fail. In other word it is the probability of system failure subtracted from one (Garg S. Kumar, 1996). There are qualitative and quantitative evaluations of reliability. Awash Sebat killo water supply system is reviewed based on the following constraints.

Pressure

A good distribution system should supply water at consumer's tap with adequate pressure, Awash WDN designed to fulfill the minimum pressure requirement (15m head). And the model values (Epanet and Matlab) are strongly correlated with the measured value collected from the existing network. The whole water supply area is in one pressure zone. All Water points, Junctions and consumer connections are getting water from the 250m3 Reservoir, elevated at 24 meter height directly pumped from 500m3 ground service Reservoir.

Water quality

There is a Water treatment plant to treat the raw water coming from Awash River. The Water is treated to the standard of Ethiopia as well as World health organization. The Water quality is regularly checked at the point of end user.

Continuous system

Awash water supply distribution system is a loop system. The advantage of the loop system is water is supplied to the beneficiaries in either direction. In case of pipe line brake down or for maintenance purpose, a loop network continuous supplying water.

As discussed before the optimized pipe can supply the peak hour demand at the desired pressure.

In general, the system is reliable both in the efficiency of water supply system and the water supply service has a capacity to manage the system. (Ministry of water, Irrigation and Electricity, 2017)

6 CONCLUSION AND RECOMMENDATION

As discussed in literature review, up to this point of time many researches are conducting for better and new approach for optimal water supply distribution networks.

In this thesis, a methodology is outlined to search for an optimal pipe network by using Epanet/Water cad and Matlab (to use the built in genetic algorithm function) at the same time. This methodology is applied to analyze the existing water supply distribution system on selected pipe loops which have larger diameters in order to have an economical pipe size.

In this thesis, **the formulated methodology was applied to analyze the existing distribution system** however it can also be employed to WDN at the design phase to make the selection of economical pipe sizes easier.

7 REFERENCES

1. C. Nalluriand R. E. Featherstone, 2009,Civil Engineering Hydraulics,
2. WAD Consulting Share Company, Feasibility and Design report for Awash Sebat killo town(Final), 2012
3. Garg S. kumar, Water Supply Engineering, 1996
4. International Council for Research and Innovation in building and construction, 2002
5. International Journal of Mechanical Engineering and Technology, Optimization Technique for water supply Network, a critical review, 2014.
6. M.CISTY, Z.BAJTEK, 2008, Optimal Design of Water Distribution Systems by A combination of Stochastic Algorithms and Mathematical Programming, Slovak Journal of Civil Engineering.
7. Matlab R2016a,Numerical computing, 2016
8. Abiy Girma,2013, Water, Sanitation & Hygiene (WaSH) in Ethiopia
9. Ministry of Water Resources, Urban Water Supply Design Criteria, 2006
10. Ministry of water, Irrigation and Electricity, Towns' Water Supply Service Indicators,2017
11. Ministry of Water, irrigation and Electricity, 2015, GTP-II water sector
12. Paula Dawe, 2000,Workshop on Hydraulic Network Modeling with Water cad
13. Svedula pp Mujumda, 2006,Water Resource systems Modeling technique & analysis
14. Tenth International Water Technology Conference, IWTC 10, 2006, Alexandria, Egypt
15. https://www.researchgate.net/post/What_is_the_advantage_of_using_Genetic_Algorithm_in_the_optimization_problems [accessed Jul 24, 2017].
16. http://www.edc.ncl.ac.uk/highlight/rhjanuary2007g01.php(accessed Jul 29, 2017).
17. http://www.math.colostate.edu/~gerhard/MATH331/lab/fmincon.html#Example%201.

8 APPENDICES

8.1 Objective and Constraint functions of each loops with their result

```
function C =Loop1(D)
 C =
(11295*(D(1))^1.429)*181+(11295*(D(2))^1.429)*203+(11295*(D(3))^1.429)*
153....
     +(11295*(D(4))^1.429)*386+(11295*(D(5))^1.429)*150;
function [c,ceq] = Loop1constraint(D)
%headloss constraint
c1 = -1.03*10^-5*D(1)^-4.87-5.49*10^-6*D(2)^-4.87+8.94*10^-7*D(3)^-
4.87+1.48*10^-5*D(4)^-4.87....
    +6.82*10^-9*D(5)^-4.87;
%pressure constraints
c2 = -(920-919)-31.72+1.03*10^-5*D(1)^-4.87+15;
p2 = (920-919)+31.72-1.03*10^-5*D(1)^-4.87;
c3 = -(919-918)-p2+5.49*10^-6*D(2)^-4.87+15;
p3 = (919-918)+p2-5.49*10^-6*D(2)^-4.87;
c4 = -(920-916)-p3+6.82*10^-9*D(3)^-4.87+15;
p4 = (920-916)+p3-6.82*10^-9*D(3)^-4.87;
c5 = -(916-915.3)-31.72+1.48*10^-5*D(4)^-4.87+15;
p5 = (916-915.3)+31.72-1.48*10^-5*D(4)^-4.87;
c6 = -(915.3-918)-p5+8.94*10^-7*D(5)^-4.87+15;

%velocity constraints
c7 = D(1)-(4*4.72*10^-3/0.6*3.14)^0.5;
c8 = D(2)-(4*3.16*10^-3/0.6*3.14)^0.5;
c9 = D(3)-(4*0.1*10^-3/0.6*3.14)^0.5;
c10 = D(4)-(4*3.81*10^-3/0.6*3.14)^0.5;
c11 = D(5)-(4*1.38*10^-3/0.6*3.14)^0.5;

c = [c2;c3;c4;c5;c6;c7;c8;c9;c10;c11];
ceq = [c1];
```

ObjectiveFunction = @Loop1;

nvars = 5;   % Number of variables

LB = [0.025 0.025 0.025 0.025 0.025];   % Lower bound

UB = [0.1 0.1 0.1 0.1 0.1]; % Upper bound

ConstraintFunction = @Loop1constraint;

rng(1,'twister') % for reproducibility

[D,fval] = ga(ObjectiveFunction,nvars,[],[],[],[],LB,UB,ConstraintFunction)

Optimization terminated: average change in the fitness value less than options.FunctionTolerance

and constraint violation is less than options.ConstraintTolerance.

D =   0.0635   0.0544   0.0411   0.0638   0.0415
fval =   1.9706e+05

```
function C = Loop2(D)
 C =
(11295*D(1)^1.429)*298+(11295*D(2)^1.429)*153.5+(11295*D(3)^1.429)*299+
(11295*D(4)^1.429)*1000;

function [c,ceq] = Loop2constraint(D)
%headloss constraint
c1 = -4.13*10^-6*D(1)^-4.87-6.66*10^-6*D(2)^-4.87-1.9*10^-6*D(3)^-
4.87+1.09*10^-5*D(4)^-4.87;
%pressure constraints
c2 = -(929-926)-22.76+4.13*10^-6*D(1)^-4.87+15;
p2 = (929-926)+22.76-4.13*10^-6*D(1)^-4.87;
c3 = -(926-921)-p2+6.66*10^-6*D(2)^-4.87+15;
p3 = (926-921)+p2+6.66*10^-6*D(2)^-4.87;
c4 = -(921-926.5)-p3+1.9*10^-6*D(3)^-4.87+15;
p4 = (921-926.5)+p3-1.9*10^-6*D(3)^-4.87;
c5 = -(929-926.5)-22.76+1.09*10^-5*D(4)^-4.87+15;

%velocity constraints
c6 = D(1)-(4*2.2*10^-3/0.6*3.14)^0.5;
c7 = D(2)-(4*4.08*10^-3/0.6*3.14)^0.5;
c8 = D(3)-(4*1.45*10^-3/0.6*3.14)^0.5;
c9 = D(4)-(4*1.93*10^-3/0.6*3.14)^0.5;

c = [c2;c3;c4;c5;c6;c7;c8;c9];
ceq = [c1];
```

```
>> ObjectiveFunction = @Loop2;

nvars = 4;    % Number of variables

LB = [0.025  0.025  0.025  0.025];   % Lower bound

UB = [0.15  0.15  0.15  0.15];  % Upper bound

ConstraintFunction = @Loop2constraint;

rng(1,'twister') % for reproducibility

[D,fval] = ga(ObjectiveFunction,nvars,[],[],[],[],LB,UB,ConstraintFunction)
```

Optimization terminated: average change in the fitness value less than options.FunctionTolerance

and constraint violation is less than options.ConstraintTolerance.

D =   0.0587   0.0708   0.0519   0.0595

fval =   3.4753e+05

```
function C = Loop3(D)
 C =
(11295*D(1)^1.429)*147+(11295*D(2)^1.429)*308+(11295*D(3)^1.429)*146+(1
1295*D(4)^1.429)*303;

function [c,ceq] = Loop3constraint(D)
%headloss constraint
c1 = -1.31*10^-7*D(1)^-4.87-9.9*10^-6*D(2)^-4.87+5.68*10^-7*D(3)^-
4.87....
    1.83*10^-7*D(4)^-4.87;
%pressure constraints
c2 = -(914-915.5)-32.69+1.31*10^-7*D(1)^-4.87+15;
p2 = (914-915.5)+32.69-1.31*10^-7*D(1)^-4.87;
c3 = -(915.5-912)-p2+9.9*10^-6*D(2)^-4.87+15;
p3 = (915.5-912)+p2-9.9*10^-6*D(2)^-4.87;
c4 = -(914-911)-p3+5.68*10^-7*D(3)^-4.87+15;
p4 = (914-911)+p3-5.68*10^-7*D(3)^-4.87;
c5 = -(911-912)-32.69+1.83*10^-7*D(4)^-4.87+15;
%velocity constraints
c6 = D(1)-(4*0.5*10^-3/0.6*3.14)^0.5;
c7 = D(2)-(4*3.5*10^-3/0.6*3.14)^0.5;
c8 = D(3)-(4*0.74*10^-3/0.6*3.14)^0.5;
c9 = D(4)-(4*0.6*10^-3/0.6*3.14)^0.5;
c = [c2;c3;c4;c5;c6;c7;c8;c9];
ceq = [c1];
```

ObjectiveFunction = @Loop3;

nvars = 4;   % Number of variables

LB = [0.025 0.025 0.025 0.025];   % Lower bound

UB = [0.1 0.1 0.1 0.1]; % Upper bound

ConstraintFunction = @Loop3constraint;

rng(1,'twister') % for reproducibility

[D,fval] = ga(ObjectiveFunction,nvars,[],[],[],[],LB,UB,ConstraintFunction)

Optimization terminated: average change in the fitness value less than options.FunctionTolerance

and constraint violation is less than options.ConstraintTolerance.

D =   0.0296   0.0550   0.0367   0.0250

fval =   9.8246e+04

```
function C = Loop4(D)
 C =
(11295*D(1)^1.429)*168+(11295*D(2)^1.429)*164.5+(11295*D(3)^1.429)*120+
(11295*D(4)^1.429)*151.5;

function [c,ceq] = Loop4constraint(D)
%headloss constraint
c1 = 3.57*10^-8*D(1)^-4.87+9.72*10^-8*D(2)^-4.87-1.15*10^-7*D(3)^-
4.87....
    -1.84*10^-8*D(4)^-4.87;
%pressure constraints
c2 = -(921-920)-30.7+3.57*10^-8*D(1)^-4.87+15;
p2 = (921-920)+30.7-3.57*10^-8*D(1)^-4.87;
c3 = -(920-921)-p2+9.72*10^-8*D(2)^-4.87+15;
p3 = (920-921)+p2-9.72*10^-8*D(2)^-4.87;
c4 = -(921-922)-p3+1.15*10^-7*D(3)^-4.87+15;
p4 = (921-922)+p3-1.15*10^-7*D(3)^-4.87;
c5 = -(922-921)-p4+1.84*10^-8*D(4)^-4.87+15;
%velocity constraints
c6 = D(1)-(4*0.23*10^-3/0.6*3.14)^0.5;
c7 = D(2)-(4*0.40*10^-3/0.6*3.14)^0.5;
c8 = D(3)-(4*0.52*10^-3/0.6*3.14)^0.5;
c9 = D(4)-(4*0.17*10^-3/0.6*3.14)^0.5;
c = [c2;c3;c4;c5;c6;c7;c8;c9];
ceq = [c1];
```

ObjectiveFunction = @Loop4;

nvars = 4;   % Number of variables

LB = [0.025 0.025 0.025 0.025];   % Lower bound

UB = [0.1 0.1 0.1 0.1]; % Upper bound

ConstraintFunction = @Loop4constraint;

rng(1,'twister') % for reproducibility

[D,fval] = ga(ObjectiveFunction,nvars,[],[],[],[],LB,UB,ConstraintFunction)

Optimization terminated: average change in the fitness value less than options.FunctionTolerance

and constraint violation is less than options.ConstraintTolerance.

D =   0.0251   0.0255   0.0255   0.0250

fval =   3.5603e+04

```
function C = Loop5(D)
 C =
(11295*D(1)^1.429)*373+(11295*D(2)^1.429)*149+(11295*D(3)^1.429)*110+(1
1295*D(4)^1.429)*369;

function [c,ceq] = Loop5constraint(D)
%headloss constraint
c1 = 3.97*10^-7*D(1)^-4.87+1.62*10^-8*D(2)^-4.87-1.93*10^-7*D(3)^-4.87-
....
    7.84*10^-8*D(4)^-4.87;
%pressure constraints
c2 = -(925-922)-26.85+3.97*10^-7*D(1)^-4.87+15;
p2 = (925-922)+26.85-3.97*10^-7*D(1)^-4.87;
c3 = -(922-923)-p2+1.62*10^-8*D(2)^-4.87+15;
p3 = (922-923)+p2-1.62*10^-8*D(2)^-4.87;
c4 = -(925-926.5)-26.85+1.93*10^-7*D(3)^-4.87+15;
p4 = (925-925.5)+26.85-1.93*10^-7*D(3)^-4.87;
c5 = -(925.5-923)-26.85+7.84*10^-8*D(4)^-4.87+15;
%velocity constraints
c6 = D(1)-(4*0.55*10^-3/0.6*3.14)^0.5;
c7 = D(2)-(4*0.16*10^-3/0.6*3.14)^0.5;
c8 = D(3)-(4*0.72*10^-3/0.6*3.14)^0.5;
c9 = D(4)-(4*0.23*10^-3/0.6*3.14)^0.5;
c = [;c2;c3;c4;c5;c6;c7;c8;c9];
ceq = [c1];
```

```
>> ObjectiveFunction = @Loop5;

nvars = 4;    % Number of variables

LB = [0.025 0.025 0.025 0.025];   % Lower bound

UB = [0.1  0.1  0.1  0.1];  % Upper bound

ConstraintFunction = @Loop5constraint;

rng(1,'twister') % for reproducibility

[D,fval] = ga(ObjectiveFunction,nvars,[],[],[],[],LB,UB,ConstraintFunction)
```

Optimization terminated: average change in the fitness value less than options.FunctionTolerance

and constraint violation is less than options.ConstraintTolerance.

D =   0.0288   0.0251   0.0268   0.0250

fval =   6.3628e+04

```
function C = Loop6(D)
 C =
(11295*D(1)^1.429)*291+(11295*D(2)^1.429)*147+(11295*D(3)^1.429)*292+(1
1295*D(4)^1.429)*153;

function [c,ceq] = Loop6constraint(D)
%headloss constraint
c1 = -1.26*10^-8*D(1)^-4.87-1.29*10^-5*D(2)^-4.87+2.45*10^-6*D(3)^-
4.87+....
    1.31*10^-7*D(4)^-4.87;
%pressure constraints
c2 = -(915.3-918)-32.84+1.26*10^-8*D(1)^-4.87+15;
p2 = (915.3-918)+32.84-1.26*10^-8*D(1)^-4.87;
c3 = -(918-915.5)-p2+1.29*10^-5*D(2)^-4.87+15;
p3 = (918-915.5)+p2-1.29*10^-5*D(2)^-4.87;
c4 = -(915.3-914)-32.84+2.45*10^-6*D(3)^-4.87+15;
p4 =  (915.3-914)+p3-2.45*10^-6*D(3)^-4.87;
c5 = -(914-915.5)-32.84+1.31*10^-6*D(4)^-4.87+15;
%velocity constraints
c6 = D(1)-(4*0.14*10^-3/0.6*3.14)^0.5;
c7 = D(2)-(4*4.12*10^-3/0.6*3.14)^0.5;
c8 = D(3)-(4*1.68*10^-3/0.6*3.14)^0.5;
c9 = D(4)-(4*0.5*10^-3/0.6*3.14)^0.5;
c = [c2;c3;c4;c5;c6;c7;c8;c9];
ceq = [c1];
```

>> ObjectiveFunction = @Loop6;

nvars = 4;   % Number of variables

LB = [0.025   0.025   0.025   0.025];   % Lower bound

UB = [0.1 0.1   0.1 0.1]; % Upper bound

ConstraintFunction = @Loop6constraint;

rng(1,'twister') % for reproducibility

[D,fval] = ga(ObjectiveFunction,nvars,[],[],[],[],LB,UB,ConstraintFunction)

Optimization terminated: average change in the fitness value less than options.FunctionTolerance

and constraint violation is less than options.ConstraintTolerance.

D =   0.0251   0.0555   0.0399   0.0349

fval =   1.0849e+05

```
function C = Loop7(D)
 C =
(11295*D(1)^1.429)*299+(11295*D(2)^1.429)*416+(11295*D(3)^1.429)*312.5+
(11295*D(4)^1.429)*313....
 +(11295*D(5)^1.429)*209.5;


function [c,ceq] = Loop7constraint(D)
%headloss constraint
c1 = 1.9*10^-6*D(1)^-4.87+1.36*10^-7*D(2)^-4.87-6.77*10^-6*D(3)^-4.87-
....
    1.26*10^-6*D(4)^-4.87....
    -8.97*10^-7*D(5)^-4.87;
%pressure constraints
c2 = -(921-926.5)-28.43+1.9*10^-6*D(1)^-4.87+15;
p2 = (921-926.5)+28.43-1.9*10^-6*D(1)^-4.87;
c3 = -(926.5-922)-p2+1.36*10^-7*D(2)^-4.87+15;
p3 = (926.5-922)+p2-1.36*10^-7*D(2)^-4.87;
c4 = -(921-917)-p3+6.77*10^-6*D(3)^-4.87+15;
p4 = (921-917)+p3-6.77*10^-6*D(3)^-4.87;
c5 = -(917-921)-28.43+1.26*10^-6*D(4)^-4.87+15;
p5 = (917-921)+28.43-1.26*10^-6*D(4)^-4.87;
c6 = -(921-922)-p5+8.97*10^-7*D(5)^-4.87+15;
%velocity constraints
c7 = D(1)-(4*1.45*10^-3/0.6*3.14)^0.5;
c8 = D(2)-(4*0.29*10^-3/0.6*3.14)^0.5;
c9 = D(3)-(4*2.8*10^-3/0.6*3.14)^0.5;
c10 = D(4)-(4*1.4*10^-3/0.6*3.14)^0.5;
c11 = D(5)-(4*0.94*10^-3/0.6*3.14)^0.5;
c = [c2;c3;c4;c5;c6;c7;c8];
ceq = [c1];
```

```
ObjectiveFunction = @Loop7;

nvars = 5;    % Number of variables

LB = [0.025 0.025 0.025 0.025 0.025];   % Lower bound

UB = [0.1 0.1 0.1 0.1 0.1];  % Upper bound

ConstraintFunction = @Loop7constraint;

rng(1,'twister') % for reproducibility

[D,fval] = ga(ObjectiveFunction,nvars,[],[],[],[],LB,UB,ConstraintFunction)
```

Optimization terminated: average change in the fitness value less than options.FunctionTolerance

and constraint violation is less than options.ConstraintTolerance.

D =   0.0448   0.0276   0.0650   0.0442   0.0450

fval =   2.0800e+05

```
function C = Loop8(D)

C =
(11295*D(1)^1.429)*167+(11295*D(2)^1.429)*323.5+(11295*D(3)^1.429)*197+
(11295*D(4)^1.429)*315;


function [c,ceq] = Loop8constraint(D)
%headloss constraint
c1 = -2.95*10^-9*D(1)^-4.87-9.22*10^-6*D(2)^-4.87-1.26*10^-8*D(3)^-
4.87+....
    1.09*10^-5*D(4)^-4.87;
%pressure constraints
c2 = -(918.5-920)-31.67+2.95*10^-9*D(1)^-4.87+15;
p2 = (918.5-920)+31.67-2.95*10^-9*D(1)^-4.87;
c3 = -(913-915)-p2+9.22*10^-6*D(2)^-4.87+15;
p3 = (913-915)+p2-9.22*10^-6*D(2)^-4.87;
c4 = -(913-915)-p3+1.26*10^-8*D(3)^-4.87+15;
p4 =  (913-915)+p3-1.26*10^-8*D(3)^-4.87;
c5 = -(920-915)-31.67+1.09*10^-5*D(4)^-4.87+15;
%velocity constraints
c6 = D(1)-(4*0.06*10^-3/0.6*3.14)^0.5;
c7 = D(2)-(4*3.25*10^-3/0.6*3.14)^0.5;
c8 = D(3)-(4*0.12*10^-3/0.6*3.14)^0.5;
c9 = D(4)-(4*3.61*10^-3/0.6*3.14)^0.5;
c = [c2;c3;c4;c5;c6;c7;c8;c9];
ceq = [c1];
```

```
>> ObjectiveFunction = @Loop8;

nvars = 4;    % Number of variables

LB = [0.025 0.025  0.025 0.025];   % Lower bound

UB = [0.1 0.1 0.1 0.1];  % Upper bound

ConstraintFunction = @Loop8constraint;

rng(1,'twister') % for reproducibility

[D,fval] = ga(ObjectiveFunction,nvars,[],[],[],[],LB,UB,ConstraintFunction)

Optimization terminated: norm of the step is less than  2.2204e-16

 and constraint violation is less than options.ConstraintTolerance.


D =     0.0251    0.0590    0.0251    0.0598

fval =    1.4875e+05
```

8.2 Matlab symbols & Constrained Optimization

8.2.1 Mathematical Symbols in Matlab

% = It is used to write explanation or comment on Command window of Matlab. Anything written following this symbol will not be executed by Matlab

^ = This symbol helps to write exponents to variables or numbers.

…. = A symbol to write these 4 consecutive points on Matlab command window help to continue a statement to the new line

; = This semi colon is used on Command window of Matlab, if one wants to prevent the mathematical function not to be executed. (Matlab R2016a, 2016)

8.2.2 Constrained Optimization using Matlab's ga (genetic algorithm)

For constrained minimization of an objective function f(x) (for maximization use -f), Matlab provides the command ga. The objective function must be coded in a function file (M-file). This file will be called objfun and saved as objfun.m in the working directory.

Basic calls

Without any extra options, ga is called as follows:

- with linear inequality constraints Ax<b only :[x,fval]= ga('objfun',A,b)

- with linear inequality constraints and linear equality constraints Aeq·x=beq only:

[x,fval]=ga('objfun',A,b,Aeq,beq)

- with linear inequality and equality constraints, and in addition a lower bound of the form x>lb only:

[x,fval]= ga('objfun',A,b,Aeq,beq,lb)

If only a subset of the variables has a lower bound, the components of lb corresponding to variables without lower bound are -Inf. For example, if the variables are (x,y), and x>1 but y has no lower bound, then lb=[1;-Inf].

- with linear inequality and equality constraints and lower as well as an upper bound of the form x<ub only:

[x,fval]= ga('objfun',A,b,Aeq,beq,lb,ub)

If only a subset of the variables has an upper bound, the components of ub corresponding to variables without upper bound are Inf. For example, if the variables are (x,y) and x>1 but y has no upper bound, then lb=[1;Inf].

- with linear inequality and equality constraints, lower and upper bounds, and nonlinear inequality and equality constraints:

[x,fval]=ga('objfun',A,b,Aeq,beq,lb,ub,'constraint')

The last input argument in this call is the name of a function file (denoted constraint in these notes and saved as constraint.m in the working directory), in which the nonlinear constraints are coded.

Constraint function file:

constraint.m is a function file (any name can be chosen) in which both the inequality functions c(x) and the equality constraints ceq(x) are coded and provided in the form of column vectors. The function call

[c,ceq]=constraint(x)

must retrieve c(x) and ceq(x) for given input vector x. If only inequality constraints are given, define ceq=[]. Likewise, if only equality constraints are given, define c=[].

Interpretation:

The retrieved ceq(x) is interpreted by ga as equality constraint ceq(x) = 0. The inequalities associated with c(x) are interpreted as -c(x)= 0. Thus, if a constraint of the form $c(x)>0$ is given, rewrite this as $-c(x)<0$ and code $-c(x)$ in the constraint function file.

Placeholders:

As shown above, the constraints have to passed to ga in the following order:

1. Linear inequality constraints
2. Linear equality constraints
3. Lower bounds
4. Upper bounds
5. Nonlinear constraints

If a certain constraint is required, all other constraints appearing before it have to be inputted as well, even if they are not required in the problem. If this is the case, their input argument is replaced by the placeholder [] (empty input).

Examples:

- If lb and (A,b) are given, but there are no other constraints, the syntax is:

```
[x,fval]=ga('objfun',A,b,[],[],lb)
```

- If ub and (Aeq,beq) are the only constraints:

```
[x,fval]=ga('objfun',[],[],Aeq,beq,[],ub)
```

- If only nonlinear constraints are given:

```
[x,fval]= ga('objfun',x0,[],[],[],[],[],[],'constraint')
```

and function file constraint.m must be provided.