INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2011; 1:47-71

FORCED WATER MAIN DESIGN; MIXED ANT COLONY OPTIMIZATION

S. Madadgar[1*, †], A. Afshar[2]
[1]*Department of Civil and Environmental Engineering, Portland State University, Portland, OR 97207, USA*
[2]*Department of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran 16844, Iran*

ABSTRACT

Most real world engineering design problems, such as cross-country water mains, include combinations of continuous, discrete, and binary value decision variables. Very often, the binary decision variables associate with the presence and/or absence of some nominated alternatives or project's components. This study extends an existing continuous Ant Colony Optimization (ACO) algorithm to simultaneously handle mixed-variable problems. The approach provides simultaneous solution to a binary value problem with both discrete and continuous variables to locate and size design components of the proposed system. This paper shows how the existing continuous ACO algorithm may be revised to cope with mixed-variable search spaces with binary variables. Performance of the proposed version of the ACO is tested on a set of mathematical benchmark problems followed by a highly nonlinear forced water main optimization problem. Comparing with few other optimization algorithms, the proposed optimization method demonstrates satisfactory performance in locating good near optimal solutions.

Received: February 2011; Accepted: May 2011

KEY WORDS: ant colony optimization, mixed continuous-discrete problems, forced water main.

[*]Corresponding author: S. Madadgar, Department of Civil and Environmental Engineering, Portland State University, Portland, OR 97207, USA
[†]E-mail address: madadgar@cecs.pdx.edu

1. INTRODUCTION

Many engineering design problems, such as water supply and sewage, water distribution network, and cross-country water mains, include both continuous and discrete decision variables. Optimum design of cross-country water mains and associated pumping stations is a relatively complex problem due to its mixed continuous-discrete decision space. Simultaneous consideration of both discrete variables (i.e. pipe diameter, and pressure classes) and continuous ones (i.e. pumping head) demands an especial algorithm capable of handling such mixed variable problems. Traditionally, either the continuous decision space is discretized which transforms the mixed problem into a discrete one, or the discrete variables are treated as continuous ones and rounded off when the final solution is found and search process is terminated. Both approaches find an approximate solution to the mixed variable problem. Employing the latter approach, it may be formulated as linear and/or nonlinear programming problem ([1] and [2]). The former approach is, however, much more common. Employing the former approach, it may be formulated as a dynamic programming (DP) problem under discretized decision space. As an example, [3] employed DP to find optimal solution to an approximation of the complete pipeline design problem. The solution provided the number and size of pumping stations, diameters, and pressure classes of the pipeline segments at the beginning of each stage interval over the planning period. A DP model is developed in [4] to optimally integrate hydropower plants into a cross country water supply main.

The optimal design problem of water distribution systems using the real-coded genetic algorithm is solved by [5] to find the discrete values for pipe diameters. According to them, this methodology avoided the problem of redundant states often found when using binary (and Gray) coding schemes. Disregarding the discrete nature of some design variables, [2] employed a non-linear mixed integer programming to optimize the design of a water supply pipeline system. [6] conducted the route selection process employing the Geographical Information System (*GIS*) to provide a rational basis for narrowing existing potential alternatives into a final alignment corridor. In a more recent work, [1] established a linear model for the optimal design of a long distance water transmission system to achieve a minimum annual cost. Abbasi et al. [7] extended a simulation-optimization framework to design a water main under transient conditions. They coupled a hydraulic simulation module with ant colony optimization algorithm as a meta-heuristic model to find the optimal specifications of a pipeline system.

During the last decade, evolutionary and meta-heuristic algorithms such as Genetic Algorithms (GAs), Ant Colony Optimization (ACO), Particles Swarm Optimization (PSO), Simulated Annealing (SA), and Honey Bees Mating Optimization (HBMO) have focused on solution of problems with nonlinear, non-convex, continuous and/or discrete search spaces in water resources optimization problems ([8-16]). A very comprehensive review on applications of GAs in water resources planning and management can be found in the study by [8].

Ant colony optimization algorithm was basically presented to solve the problems with discrete search spaces ([17-18]). To apply the ACO algorithms to problems with continuous domain, the search space is traditionally divided into a discrete set of decision values and agents explore the new domain to find the most desirable solution (Jalali et al., 2007). The direct extension of ACO algorithms to continuous domains has been tackled by different researchers ([19-21]). In a quite interesting approach, [22] proposed ACO_R algorithm, the central core of which is well close to original concept of ACO. Recently, [14] suggested two major modifications to improve the performance of ACO_R. Benefiting from adaptation operator and explorer ants, they significantly improved the results of original ACO_R in both benchmark mathematical problems and a real-world reservoir operation optimization problem.

Realizing the existence of many mixed-variable problems with continuous and discrete decision and/or state variables in various fields of engineering and particularly in water resources engineering, this article proposes an ant colony optimization algorithm that directly tackles the optimization problems with mixed variable domains. It is an extension to existing continuous ACO_R algorithm ([14] and [22]) which has been modified to simultaneously deal with mixed-variable problems. In the following sections, the basic concept of the improved ACO_R is presented, and the proposed modifications to enable the algorithm handling both kinds of variables are addressed. The performance of the algorithm is, then, tested on some mathematical functions and compared to those of other algorithms. Finally, to assess the potential of its application to water resources engineering problems, the optimum design of a real-world highly non-linear forced water main is discussed

2. ANT COLONY OPTIMIZATION ALGORITHM FOR CONTINUOUS DOMAIN

Ant colony optimization algorithms borrow the same concept from real ants' foraging behavior. At each decision step in ACO algorithms, the pheromone affect –which resembles the real ants' foraging behavior- is simulated by a probability rule. The probability rule in the original Ant System (AS)([18]) is defined as following:

$$p\left(c_{ij} \mid s^{p}, t\right)=\frac{\left[\tau_{ij}(t)\right]^{\alpha} \cdot\left[\eta\left(c_{ij}\right)\right]^{\beta}}{\sum_{j=1}^{J}\left[\tau_{ij}(t)\right]^{\alpha} \cdot\left[\eta\left(c_{ij}\right)\right]^{\beta}} \quad , \quad \forall c_{ij} \in allowable\ set \tag{1}$$

Where, $p\left(c_{ij} \middle| s^{p}, t\right)$ is the probability of selecting the solution component c_{ij} at step i in iteration t; $\tau_{ij}(t)$ is the pheromone value associated with component c_{ij} at iteration t; $\eta(.)$ assigns the heuristic value to the solution component c_{ij}; α and β are two parameters representing the relative importance of the pheromone trail and heuristic value; and i is the current construction step including j component solutions in the allowable set.

The probability function defined as Eq.1 forms a discrete probability distribution over the allowable set of decision values at each construction step (Figure 1a). The approach is well suited for solution of problems with discrete variables. For continuous search spaces with continuous decision variables, however, ants must sample from continuous probability distribution functions over the search space. Dreo and Siarry [22] applied the probability density function to model the probability distribution over the continuous search space. In this case, ants are allowed to sample continuous values instead of a finite set of solution components in Eq. (1). They employed Gaussian Probability Density Function (PDF) to represent the probability of continuous search domain. In order to overcome the main shortcoming of a single Gaussian function in modeling multimodal areas, a Gaussian kernel PDF replaces the individual PDF which provides a more flexible sampling over search space (Figure 1b). The Gaussian kernel PDF is defined as weighted superposition of several Gaussian functions $g_l^i(x)$ as $G^i(x)$ ([22]):

$$G^i(x) = \sum_{l=1}^{k} \omega_l \; g_l^i(x) = \sum_{l=1}^{k} \omega_l \frac{1}{\sigma_l^i \sqrt{2\pi}} e^{-\frac{(x-\mu_l^i)^2}{2\sigma_l^{i^2}}} \tag{2}$$

Where, k is the number of individual PDFs forming the Gaussian kernel pdf at i^{th} construction step; $\boldsymbol{\omega}$, $\boldsymbol{\mu}^{\mathbf{i}}$ and $\boldsymbol{\sigma}^{\mathbf{i}}$ are the vectors of size k defining the weights, means and standard deviations associated with the individual Gaussian functions at i^{th} construction step, respectively.

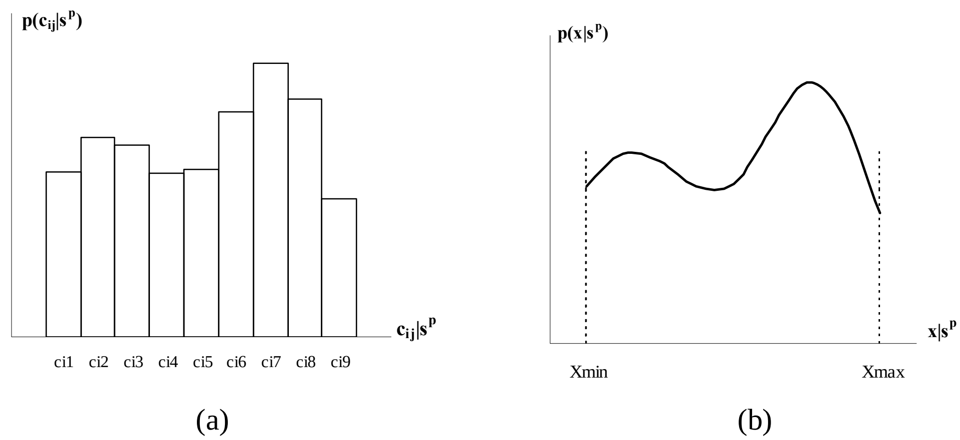


Figure 1. Schematic of a) discrete probability distribution of a set of allowable components $[c_{i1},...,c_{i9}]$ in construction step i, b) continuous probability density function with a possible range of $x \in [x_{\min}, x_{\max}]$ ([14]; adopted from [22]).

To conduct the pheromone updating process, an archive T is defined to store the decision values of a certain number (k) of the superior solutions. To fill the archive, after a complete iteration, all already-archived and newly constructed solutions are evaluated by the fitness function and then ranked according to their fitness values. Then, the superior k

solutions with their decision values, s_l^i; the i^{th} component of solution with rank l are archived in order and rest of the solutions are discarded.

The shape of the Gaussian kernel PDF at each construction step is identified by the vectors $\boldsymbol{\omega}, \boldsymbol{\mu}^{\mathbf{i}}$, and $\boldsymbol{\sigma}^{\mathbf{i}}$ which are determined by the archived solutions. [22] represented the mathematical formulation of these three vectors components at i^{th} construction step as shown in Table 1. A brief description for identification and clarification of different parameters of each Gaussian kernel PDF is presented in Appendix A.

Table 1. Mathematical presentation of the l^{th} component of the vectors at i^{th} construction step (Proposed in [22])

Vector component	**Functional presentation**
μ_l^i	s_l^i
σ_l^i	$\xi \sum_{e=1}^{k} \frac{\left\lvert s_e^i - s_l^i \right\rvert}{k-1}$
ω_l	$\frac{1}{qk\sqrt{2\pi}} e^{-\frac{(l-1)^2}{2q^2k^2}}$

Since sampling the Gaussian kernel PDF is painstaking effort, each ant, before starting the solution construction procedure, chooses a single solution from the archive. Then, at any construction step, it samples the PDFs associated to the chosen solution.. Therefore, the complex task of sampling the Gaussian kernel PDFs is simplified to sampling the individual PDFs. As superior solutions should definitely have more chance to be chosen by the agents, the following probability function is defined to express the chance of selecting the l^{th} solution in the archive ([22]):

$$p_1 = \frac{\omega_1}{\sum_{j=1}^{k} \omega_1}, \; \forall l = 1, \dots, k \tag{3}$$

The Gaussian distribution of ω_l has the standard deviation of qk, where q is a tunable parameter of the algorithm. The value of this parameter has a significant effect on convergence rate of the algorithm. Large values of q cause the algorithm to widely search the decision space in expense of slow convergence to the final solution. In the case of very small values of q, the search process is seriously narrowed around the best found solutions, and rapid pre-mature convergence occurs.

To ensure the reliability of the final solution, agents must widely explore the decision space at initial search stages and gradually narrow around the best solutions. To do so, [14] proposed the *Adaptation Operator* which encourages the agents to concentrate on high qualified areas after a diverse and comprehensive exploration over the space. For this purpose, their model initiates with a relatively large value of q which is adaptively reduced along with algorithm progress. The following expressions describe how the adaptation operator alters the search diversity through sequential iterations ([14]):

$$q_{it} = q_{it-1} * A_{it} \tag{4}$$

$$A_{it} = \begin{cases} 1 & if \ \left(\dfrac{Mean(f_{1\dots m})}{Mean(f_{1\dots n})}\right)_{it} = \left(\dfrac{Mean(f_{1\dots m})}{Mean(f_{1\dots n})}\right)_{it-1} \\ \dfrac{Mean(f_{1\dots m})}{Mean(f_{1\dots n})} & if \ \left(\dfrac{Mean(f_{1\dots m})}{Mean(f_{1\dots n})}\right)_{it} \neq \left(\dfrac{Mean(f_{1\dots m})}{Mean(f_{1\dots n})}\right)_{it-1} \end{cases} \tag{5}$$

In which, A_{it} is the value of adaptation operator in iteration it; $Mean(f_{1\dots m})$ and $Mean(f_{1\dots n})$ are the mean values of fitness function over first m and n $(m < n)$ archived solutions at any iteration, respectively. The terms $(.)_{it}$ and $(.)_{it-1}$ refer the expression between the parentheses to the iterations it and $it-1$, respectively.

The value of A_{it} in Eq. (4) should be less than or equal to one. Then, when a minimization problem is the case (i.e. the solution ranked 1st has the least fitness value), m should be less than n in Eq. (5). This ensures the non-increasing trend of the value of adaptation operator and, consequently, the parameter q. Moreover, Eq. (5) illustrates the necessity of improvement in archived solutions as the required condition to reduce the value of A_{it}. Madadgar and Afshar [14] described how archive updating affects the value of adaptation operator and search diversity.

As implied from the definition of $\mu_l^i = s_l^i$, one may note the severity encountered when the agents trap in local optimums. When, at some solution construction steps, the values of μ_l^i of almost all single PDFs become relatively the same, the values of associated σ_l^i will approach to zero. That is, the mode values of those Gaussian PDFs acquire very large probabilities, and the agents will be naturally encouraged to search through those small areas. Therefore, the agents will seriously trap in sub-optimum points, and even jumping to other PDFs will not further change the result. To help escaping from local optimums, [14] employed *Explorer Ants.* The proposed explorer ants are permitted to probabilistically mutate the trial value sampled from any Gaussian function within a specified *Mutation Range* that may be defined as:

$$MR_{it}^i = \left[x - f\left(\boldsymbol{\sigma}_{it}^i\right),\ x + f\left(\boldsymbol{\sigma}_{it}^i\right)\right] \tag{6}$$

Where, MR_{it}^{i} is the mutation range; x is the initial trial value sampled from the Gaussian PDF; $\boldsymbol{\sigma}_{it}^{i}$ is the vector of standard deviation for Gaussian PDFs; i and it denote the current construction step and current iteration, respectively.

In Eq. (6), the mutation range is expressed as a function of standard deviation, i.e. $f\left(\boldsymbol{\sigma}_{it}^{i}\right)$. Since the values of standard deviations regularly change in consecutive iterations, the mutation range varies through the advancement of the algorithm.

Explorer ants are less imposed and given the chance of more random exploration of the search space. The definition of these ants reduces the impact of PDFs on the search process, especially when the agents trap in local optimums.

Inclusion of the adaptation operator and explorer ants in the original algorithm, ACO_R ([22]), has reasonably improved the performance of the proposed algorithm in some well-known mathematical test functions and operation of the real-world hydropower reservoirs ([14]).

3. PROPOSED ACO ALGORITHM FOR MIXED-VARIABLE PROBLEMS

The convincing performance of the continuous *ACO* algorithm discussed in previous section inspired the authors to extend the algorithm to mixed-variable problems. In following, a simple but efficient procedure is demonstrated which enables the described continuous model to tackle the discrete variables, as well.

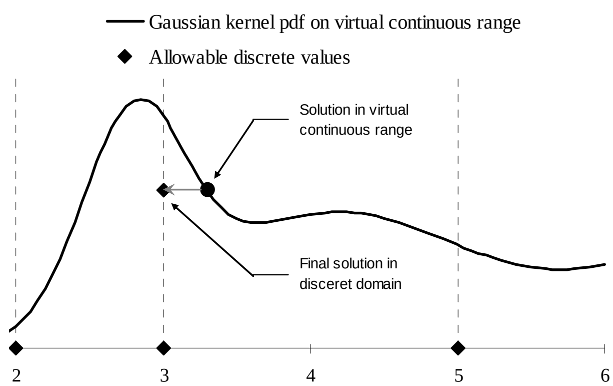


Figure 2. Schematic of handling the discrete variables by proposed approach

To handle both discrete and continuous variables in a search space, two distinct approaches may be regarded. In first approach, one may employ distinct *ACO* methods towards solving each type of decision variables. In one hand, the agents employ an original form of *ACO* as to search through the discrete domains. In the other hand, for continuous domains, an *ACO* developed for continuous search space is applied. Hence, this approach benefits from a combination of *ACO* algorithms instead of a single one. In a second approach, a single *ACO* may be employed for both types of search spaces. This study follows the second approach. To handle the discrete domain, discrete decision space is regarded as sub-domain of a larger virtual continuous domain (Figure 2). The *ACO* algorithm explained in previous section searches for the best continuous values for all decision variables. Then, it converts the values associated to discrete variables to the closest discrete values. As an example, let's assume a discrete variable is allowed to take values from the set $A = \{2, 3, 5\}$. Then, the virtual continuous range $[2, 6)$ is fitted by the Gaussian kernel PDF. Now, the agents construct their solutions through this continuous domain, and the decisions are then converted to the lower discrete values which belong to the actual discrete domain. Afterwards, the solutions are evaluated by the fitness function, and the algorithm steps into the next iteration. Figure 3 clearly shows the steps of the algorithm. As seen, the algorithm makes the agents sample from continuous values for both continuous and discrete decision variables. If the transformations are made towards the lower value of continuous subdivisions, the upper end of entire continuous range is regarded as the next discrete value to the greatest allowable one. Accordingly, any discrete value in allowable set, except the smallest one, belongs to two contiguous subdivisions; one with greater and the other one with smaller continuous values. In this case, the transformation process imposes no bias towards any certain discrete value. Each agent is evaluated by the fitness function just after a complete solution is constructed and transformation to discrete values for according variables is accomplished. The procedure above can be mathematically expressed as:

$$\begin{aligned} &Min\,Z = f(X, Y) \\ &X = g(X') \\ &Y, X' : \textit{vector of the continuous variables} \\ &X : \textit{vector of the discrete variables} \end{aligned} \tag{7}$$

Where Z is the objective function to be optimized; ***X*** and ***Y*** are the vectors of discrete and continuous variables, respectively; and g(.) is the transformation function mapping values of (***X'***) from virtual continuous domain to the discrete domain.

There are two subtle points inherent in the proposed approach to handle the mixed-variable problems. The first one is due to the archiving theme of discrete variables. This study suggests saving the virtual continuous values of discrete variables when archiving procedure is in action. In other words, the archived value of a discrete variable is its continuous value and not its discrete value obtained after transformation process. The advantage of such archiving theme is to avoid immature convergence of the algorithm towards the integer values shared by several solutions in the archive. If the actual discrete values obtained after transformation process were saved in the archive and there were several archived solutions with the same discrete value for a certain integer variable, the search process may rapidly converge to that discrete value and the standard deviation of Gaussian PDFs would quickly approach zero. In a study by [23], the discrete values of integer variables are archived instead of the continuous values, and thus a lower bound for standard deviation of Gaussian PDFs of integer variables is defined to have control on convergence speed. Definition of the lower bound of standard deviation is itself subjective to the number of integer variables in the optimization problem. However, archiving the continuous values of discrete variables as suggested by this study avoids too quick convergence to an integer value.

The second subtle point is the establishment of transformation process before the fitness evaluation. If inversely, the remarkable inaccuracy in solution assessment is highly probable. For clarification purpose, let's assume the mathematical problem as follow:

$$\begin{aligned}
&Max \quad Z = x_1 + 5x_2 \\
&\text{Subject to:} \\
&x_1 + 10\,x_2 \le 20 \\
&x_1 \le 2 \\
&x_1, x_2 \ge 0 \ \text{ and integer}
\end{aligned} \tag{8}$$

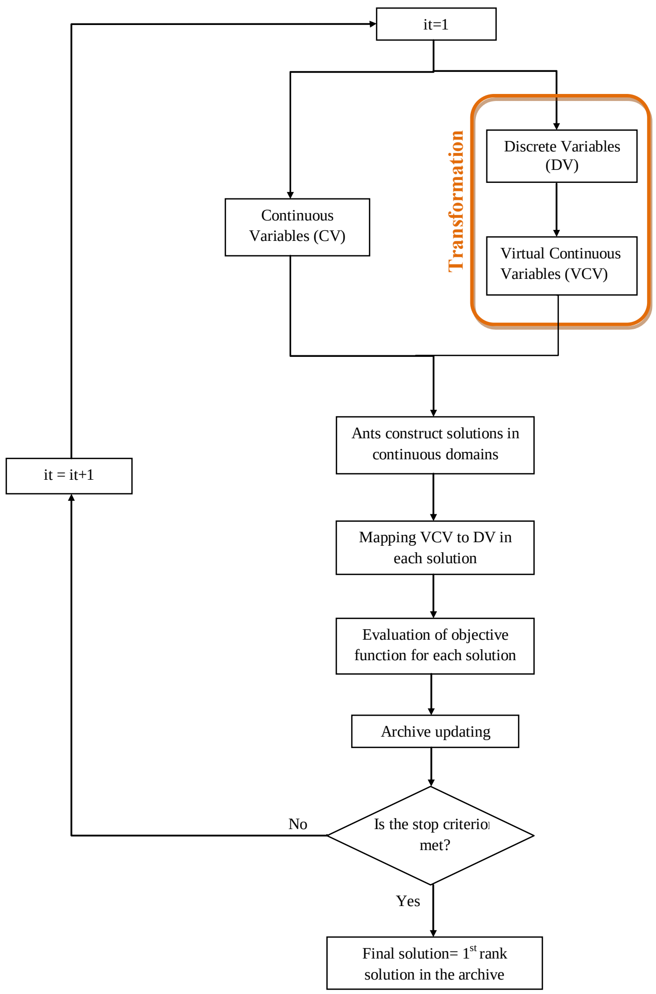


Figure 3. Schematics of proposed ant colony algorithm for mixed-variable problems

The decision and feasible spaces for the mentioned problem with two discrete decision variables are depicted in Figure 4. If a continuous Linear Programming (LP) solver is employed, the optimum solution will be located at point A (Figure 4). Note that if point A is transformed to upper discrete value (i.e. point B; $x_2 = 2$), a non-feasible solution will be resulted. If it is transformed to the lower discrete value (i.e. point C; $x_2 = 1$), the resulted solution is quite far away from the optimal solution which indicates as $(x_1, x_2) = (0,2)$. It is obvious that the algorithm operates and selects the solution in the continuous space, and the transformation process to discrete space is done after fitness evaluation. That is, in advancement of the method, all decisions are made in the continuous space; which may mislead the model to unfavoured spaces. Therefore, the final late transformation process is unable to further modify the result.

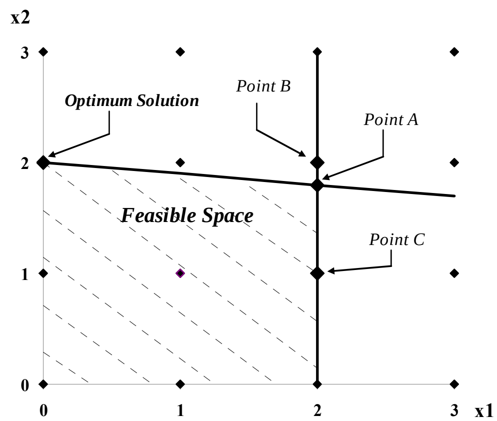


Figure 4. Graphic scheme of the optimization problem defined as Eq. 8

This simple example clarifies the importance of establishing the transformation process before the fitness evaluation in the proposed procedure. As the present model advances, the fitness value is not assigned to any agent unless its decision values are all located in allowable continuous and/or discrete spaces. In the case, the model is not progressed towards misleading areas in the virtual continuous space; and the final solution is highly reliable. In other words, if the fitness of any solution is evaluated after transformation to discrete values, then the optimal solution will be more likely accessible. It should be noted that the problem is solved using the proposed algorithm and the global optimum solution is obtained after 8 function evaluations.

When it comes to real world design problems, before sizing, the designer must initially decide on the presence and/or absence of some nominated alternatives or project's components. This means one has to simultaneously solve a binary value problem and a continuous ACO to locate and size design components of the proposed system. As an example, in application of ACO based algorithms to cross-country pipeline design, before sizing, the designer must decide on existence and/or absence of a pumping station in a given node. Please note that, if pump is not to be assigned to a given node, agents in the proposed algorithm must sample exact value of *zero* form kernel PDF. To resolve this problem one or more "*zero solutions*" are added to the archive in the division of continuous variables. *Zero solutions* are added to the part of discrete-continuous variables (mixed variables) in the archive. Each mixed variable in the *zero solution* taking the value of zero implies the absence of that design component or alternative. The rank of this solution is arbitrary. If inclusion of that design component is slightly promised, the zero solution enters the archive with rank 1st. If an agent chooses a solution other than the *zero* solution, the associated design component or alternative is nominated as a potentially good solution and the generated value is assigned to that variable. Since the archive is able to meet the PDFs and not the single values, inclusion of the *zero solution* is a subtle point of this approach. To prevent any conceptual deviation from the central core of the proposed ACO algorithm, it is suggested to attribute the Gaussian functions to the *zero solution*. This may be achieved by the Gaussian functions with mean and standard deviation of zero. Such a PDF represents the single value of zero. So, any decision value in the *zero solution* is indicated by the described PDF. Definitely, sampling such PDF leads to the value of zero; meaning the absence of the associated design variable or alternative in the final solution.

4. MODEL APPLICATIONS

This paper employs the proposed approach to a set of previously studied test functions and then investigates the performance of the method in a real-world water resources engineering problem.

4.1. Test functions

A set of previously solved benchmark functions are presented in Table 2 to investigate the performance of the proposed mixed-variable method ([24]-[25]). The proposed algorithm is termed ***M-IACO**$_R$* as it modifies the Improved version of ***ACO**$_R$* ([14]) to account for Mixed variable domains. Table 3 summarizes the most effective parameters of *M-IACO*$_R$ method, where ξ in column 5 is a multiplication factor associated with the standard deviation of Gaussian functions ([22]). As used by other researchers, a certain degree of convergence is determined as stop criterion for the algorithm ([16]):

$$\left| f_k - f_{k-\Delta k} \right| < 10^{-5} \quad , \quad \forall\, k > \Delta k \tag{9}$$

In which, f is the objective value of the best-found solution; the subscripts, k and $k-\Delta k$, indicate the iteration numbers where $\Delta k = 50$. To inaugurate any iteration, the stop criterion checks whether the required convergence is already satisfied; and if so, the search process terminates. In other words, the algorithm is assumed to converge to the best solution if the fitness value of the best solution in the k^{th} iteration remains close enough to that obtained in Δk preceding iterations.

Table 2. Summary of test functions

Problem	Taken from	Mathematical formulation	Global optimum
1	Kocis and Grossmann [26]	$Min \;\; f = 2x_1 + x_2 - y$ $Subject\ to:$ $x_1 - 2e^{-x_2} = 0$ $-x_1 + x_2 + y \le 0$ $0.5 \le x_1 \le 1.4$ $y \in \{0,1\}$	$f(x_1, x_2, y) =$ $f(1.375, 0.375, 1) = 2.124$
2	Kocis and Grossmann [27]	$Min \;\; f = 7.5y_1 + 5.5y_2 + 7\upsilon_1 + 6\upsilon_2 + 5x$ $Subject\ to:$ $y_1 + y_2 = 1$ $z_1 = 0.9x_1\left(1 - e^{-0.5\upsilon_1}\right)$ $z_2 = 0.8x_2\left(1 - e^{-0.4\upsilon_2}\right)$ $z_1 + z_2 = 10$ $x_1 + x_2 = x$ $\upsilon_1 \le 10y_1, \; \upsilon_2 \le 10y_2$ $x_1 \le 20y_1, \; x_2 \le 20y_2$ $x_1, x_2, z_1, z_2, \upsilon_1, \upsilon_2 \ge 0 \;\;,\;\; y \in \{0,1\}$	$f(x, y_1, y_2, \upsilon_1, \upsilon_2) =$ $f(13.42799, 1, 0, 3.514237, 0)$ $= 99.23963$
3	Yuan et al. [28]	$Min \;\; f = (y_1 - 1)^2 + (y_2 - 2)^2 + (y_3 - 1)^2 - \ln(y_4 + 1)$ $+ (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2$ $Subject\ to:$ $y_1 + y_2 + y_3 + x_1 + x_2 + x_3 \le 5$ $y_3^2 + x_1^2 + x_2^2 + x_3^2 \le 5.5$ $y_1 + x_1 \le 1.2$ $y_2 + x_2 \le 1.8$ $y_3 + x_3 \le 2.5$ $y_4 + x_1 \le 1.2$ $y_2^2 + x_2^2 \le 1.64$ $y_3^2 + x_3^2 \le 4.25$ $y_2^2 + x_3^2 \le 4.64$ $x_1, x_2, x_3 \ge 0 \;\;,\;\; y_1, y_2, y_3, y_4 \in \{0,1\}$	$f(x_1, x_2, x_3, y_1, y_2, y_3, y_4) =$ $f(0.2, 0.8, 1.907878, 1, 1, 0, 1)$ $= 4.579582$

Table 3. Summary of parameters used in $M - IACO_R$ for test problems

Problem	Population size	Archive size	Initial value of q	ξ	No. of explorer ants
1	3	10	0.1	1.0	1
2	5	20	0.1	1.0	2
3	5	20	0.1	1.0	2

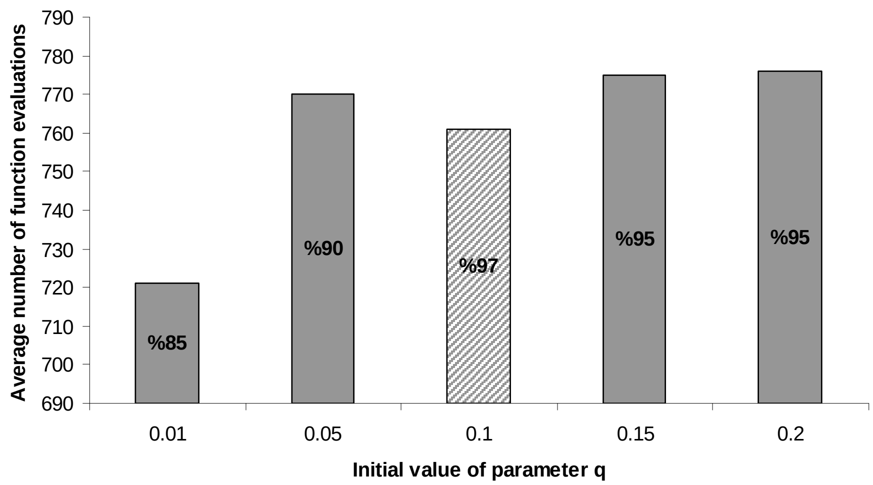
Results are compared based on the mean number of function evaluations along with the percentage of independent runs in which the algorithm has converged to the optimal solution. Table 4 presents the performance of different algorithms on test functions. The reported values include the mean number of function evaluations and the percentage of successful runs over 100 independent performances. For problems number 1 and 2, all tested algorithms locate near optimal solutions satisfying the desired criteria in all 100 independent runs. However, the number of function evaluations of the proposed method $\left(M-IACO_R\right)$ is remarkably less than those of other algorithms. As an example, *R-PSO_c* ([25]) takes an average number of function evaluations of 3500 for problem number 1, whereas the *M-IACO*$_R$ satisfies the same criterion via an average number of function evaluations of 576. In other words, compared to *M-IACO*$_R$, the next best algorithm (i.e. *R-PSO_c*) needs $\frac{3500-576}{576} \cong 5$ times extra function evaluations to obtain the same degree of convergence to near optimal solution. Other employed algorithms require far more function evaluations for the same convergence criterion. This superiority remains more or less valid for other two test problems. For problem number 3, in 97 runs out of 100 runs, the proposed method converges to near optimal solution within an average number of function evaluations of 761. Whereas, among other employed algorithms, *R-PSO_c* has satisfied the stop criterion for all 100 runs with remarkably larger number of function evaluations.

Table 4. Performances of different methods on test functions; stop criterion of any algorithm is assumed as a certain degree of convergence towards analytical solutions

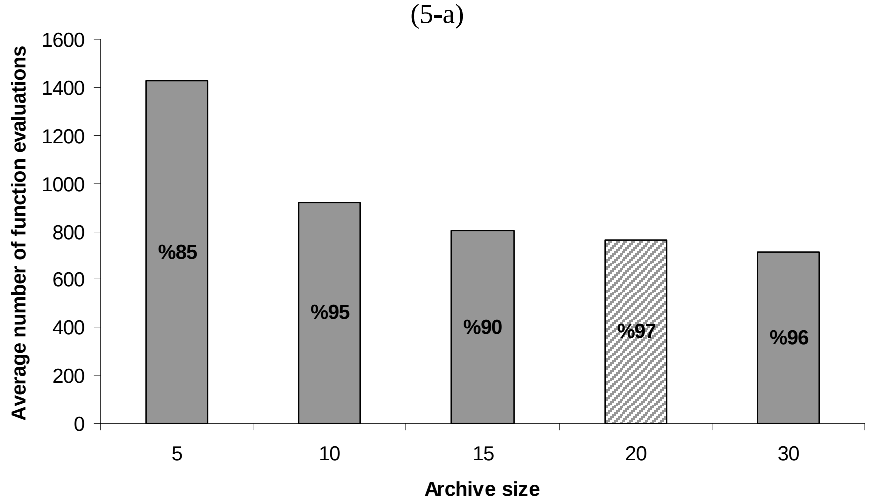
Problem	*GA* [1]	*M-SIMPSA* [1]	Original *PSO* [2]	*R-PSO_c* [2]	*M-LACO*$_R$
1	13939/100	14440/100	-[3]	3500/100	576/100
2	22489/100	42295/100	-[3]	4000/100	763/100
3	102778/60	63751/97	30000/80	30000/100	761/97

[1] [24]
[2] [25]
[3] Converged to non optimal solutions in all executions

To more clarify the impacts of major parameters on the performance of the proposed *M-IACO*$_R$ algorithm, the problem number 3 was solved for a range of archive sizes and initial values for parameter q. Figure 5 depicts the mean number of function evaluations over 100 runs versus archive sizes and initial values of parameter q. The values on the bars indicate the percentage of successful performances over 100 independent runs. As shown, an increase in initial value of q reveals minor impact on the results. This may be interpreted as the influence of adaptation operator which reduces the model sensitivity on initial value of parameter q ([14]). Adaptation operator provides a rather wide search through decision space in initial steps of the model implementation and gradually narrows the search process to the vicinity of more promised areas as the model progresses. Therefore, increasing the initial value of parameter q beside an active adaptation operator does not reduce the model efficiency in terms of convergence as shown in Figure 5(a). In addition, a range of archive sizes tested and the results of 100 independent executions are shown in Figure 5(b). As shown, the changes in results are not significant when the archive size is selected around 20 for problem number 3 (Table 3). Archive size is a parameter controlling convergence speed, and inclusion of explorer ants reduces the sensitivity of convergence rate to the archive size. Without explorer ants, the standard deviation of Gaussian functions declines rapidly if algorithm traps in local optimums; and then the archive size should be finely tuned to avoid too fast convergence before well exploration of the search space. However, introducing the explorer ants to the algorithm relaxes the serious parameter tuning procedure for archive size.



(5-a)



(5-b)

Figure 5. Performance of $M - IACO_R$ on test function number 3 (a) different initial value for parameter q, and (b) different archive size. Hatched bars are due to the values in Table 3

The results show the efficient performance of the proposed approach for mixed variable domains. The efficient performance may partially be due to the fact that the present approach benefits from the same concept towards solving both continuous and discrete variables. Sampling the continuous search space regardless the variable type may cause the model to proceed in both domains with rather equal paces. Moreover, the proposed transformation approach further helps the method to favorably handle the discrete decision variables. It should be also noted that the original and improved versions of ACO_R perform truly efficient in continuous domains ([14], [22]). The proposed adaptation of improved version of ACO_R ([14]) to mixed-variable problems introduces an alternative approach for the extensive range of optimization problems in science and engineering.

4.2. Design of a Forced Water Main

To illustrate the application of the present algorithm in a mixed variable and real world engineering problem, the optimal design of an assumed forced water main as a non-convex and highly non-linear problem is considered.

The system consists of n nodes and $n-1$ reaches. Each node is free to include a pump station with an allowable range of pumping head. The system is assumed to be under steady state conditions and the final design will be coherent with this formulation. For simplicity, the installation of safety instruments to diminish the impacts of possible dynamic pressures is disregarded. The pipe diameters in each section and pumping characteristics at all nodes must be determined, while the layout of the system is known. Therefore, the continuous decision variables include the pumping heads at pump stations; and the discrete variables address the pipes' diameters at each section. The system design should satisfy the pre-defined demand and assumed hydraulic constraints considering static flow regime. One may mathematically define the model as:

$$\begin{aligned}
&Min\ \ Z = \sum_{n=1}^{NN} C_n(hp_n) + \sum_{i=1}^{NR} C_i(D_i) \\
&\text{Subject to}: \\
&V_{\min} \le V_i \le V_{\max} && i = 1,\dots,NR \\
&D_{\min} \le D_i \le D_{\max} && i = 1,\dots,NR \\
&h_{\min} \le h_n \le h_{\max} && n = 1,\dots,NN
\end{aligned} \tag{10}$$

In which, $C_n(hp_n)$ and $C_i(D_i)$ are non-linear cost functions for pumps and pipes used at node n and reach i, respectively; hp_n is pumping head at node n; D_i and V_i are pipe diameter and velocity at reach i, respectively; h_n is piezometric head at node n; NR is number of reaches; NN is number of nodes; and $V_{\min}, V_{\max}, D_{\min}, D_{\max}$ are minimum and maximum allowable velocities and pipe diameters in all reaches, respectively; and $h_{\min}, h_{\max}$ are minimum and maximum allowable piezometric head at all nodes. Detailed definition of the cost functions $C_n(hp_n)$ and $C_i(D_i)$ can be found in appendix B.

For a given water flow (Q) in the system and predefined allowable range for pipe diameters, the velocities will be inevitably between the maximum and minimum values. Therefore, a *feasible diameter set* may be defined such that, for a given Q, resulting velocities fall within the allowable ranges.

The energy equation between two successive nodes in the system may be defined as:

$$h_n + \frac{V_n^2}{2g} + \left(h_p\right)_n = h_{n+1} + \frac{V_{n+1}^2}{2g} + \left(h_{Loss}\right)_i \qquad n = 1, \ldots, NN - 1 \tag{11}$$

Where, h is the piezometric head; h_p is the pumping head; h_{Loss} is the total head loss between two points including both local and friction losses; and the indexes n and $n+1$ refer to the beginning and ending nodes of link i. Since the velocities are rather small in the long pipelines, the according terms may be insignificant.

In a long pipe, the energy loss is considerably attributed to friction losses rather than local ones. Therefore, the latter might be negligible, and head losses only comprise the friction terms. Hazen-Williams equation for friction loss calculation is expressed as:

$$\left(h_f\right)_i = 10.7 \times L_i \times \left(\frac{V_i \times D_i}{C_H}\right)^{1.852} \times \frac{1}{D_i^{4.87}} \qquad i = 1, \ldots, NR \tag{12}$$

Where, h_f is the friction loss; L is the pipe length; and C_H is the Hazen-Williams coefficient.

The system under consideration consists of 18 nodes and 17 reaches with known topographic levels as depicted in Figure 6. Hence, it leads to an optimization problem including 35 decision variables; 17 discrete variables as commercial pipe diameters and 18 continuous variables as potential pumping head in associated nodes. Lengths of the pipes are presented in Table 5. The water flow remains constant in the system as $0.3\left(m^3/s\right)$ and the Hazen-Williams coefficient is assumed as $C_H = 120$. The allowable range for velocity is determined as $[0.4, 2.6]\,m/s$, and according to the system discharge, the pipe diameters ought to fall in the range of $[0.4, 0.8]\,m$.

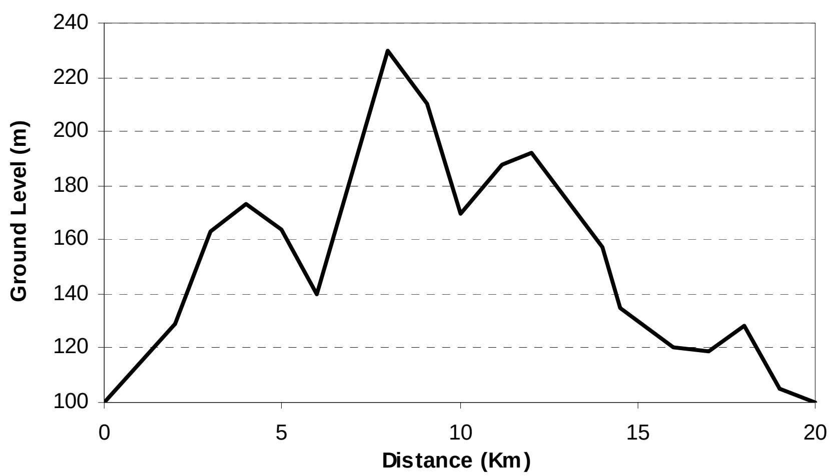


Figure 6. Pipeline topography of the assumed water main

Table 5. The lengths of pipe segments

Link number	1	2	3	4	5	6	7	8	9
Length (m)	2000	1000	1000	1000	1000	2000	1100	900	1200
Link number	10	11	12	13	14	15	16	17	
Length (m)	800	2000	500	1500	1000	1000	1000	1000	

The allowable ranges for pressure and pumping heads are respectively defined as $[3,150]$ and $[3,80]$ meter, respectively. The maximum and minimum permitted values for piezometric head at any point are calculated from the corresponding topographic level adding to the allowable range for pressure head. It should be considered that the pumping head in any node is determined once after the existence of pump station was recognized.

Before making any decision on the pumping head at each node, presence or absence of a pump station at that node of the system must be verified. To do so, one may either use the approach described in section 3 with a so-called "*zero solution*" in the archive or consider a binary decision variable with either 1 or 0 values at each node. Values of 1 and zero for this binary variable refer to presence and/or absence of pump station at the subject node, respectively. If a pump station is assigned to any node, the pumping head is then decided. As is obvious, this method inserts an extra array of discrete variables into the model and increases the computational effort. In this study the former approach is employed. If an agent, at any node, chooses a solution other than the *zero solution*, the node is nominated for a pump station and the generated value shows the pumping head. On the other hand, selection and sampling from the PDF associated with the "*zero solution*" implicitly indicates that a pumping station may not be included in that node.

This approach is able to implicitly handle the presence of pump stations and does not impose any pronounced extra computational effort on the model. It can tackle both the presence of pump stations and their design heads by using original single array of decision variables.

Once an agent selects a solution (selects the pipes' diameters and pumping heads), it should be checked if the solution is feasible. Since the allowable range of pipes' diameter is chosen as to automatically satisfy the acceptable range of velocity, the only probable violation may occur to piezometric heads at nodes. Therefore, velocity constraint in Eq. 10 remains to be satisfied. If the constraints on piezometric head (Eq. 10) are not satisfied, the responsible agent will be penalized by the following expression:

$$Penalty_n = \begin{cases} PF \times (h_{\max} - h_n)^2 & if\ h_n > h_{\max} \\ PF \times (h_{\min} - h_n)^2 & if\ h_n < h_{\min} \end{cases} \quad n = 1,\dots, NN \tag{13}$$

In which, PF (105 in this study) is the penalty factor addressing importance of piezometric head violation. The consequent penalized objective function might be easily approached as:

$$Min\ \ Z = \sum_{n=1}^{NN} C_n(hp_n) + \sum_{i=1}^{NR} C_i(D_i) + \sum_{n=1}^{NN} Penalty_n \tag{14}$$

Imposing the penalized objective function on the violator ants, the model spontaneously inclines towards the feasible areas.

Table 6. Summary of parameters used in present ant model for the forced water main design

Total ants	**20**
Explorer ants	5
No. of iterations	300
Archive size*(k)*	20
ξ	1.1
Initial q	0.3

Table 7. Results obtained by present ant colony optimization algorithm

Run	**M-LACO$_R$**	**Run**	**M-LACO$_R$**
1	**122.71* (1,3)****	11	122.84 (1,3)
2	123.24 (1,2)	12	124.38 (1,2)
3	122.72 (1,2)	13	122.72 (1,2)
4	123.25 (1,2)	14	123.3 (1,3)
5	122.9 (1,2)	15	122.94 (1,3)
6	123.15 (1,2)	16	122.90 (1,2)
7	122.83 (1,2)	17	123.63 (1,3)
8	123.16 (1,3)	18	122.78 (1,2)
9	122.96 (1,3)	19	123.82 (1,2)
10	123.09 (1,2)	20	123.04 (1,2)
	The best	122.71	
	Mean	123.12	
	The worst	124.38	
	S.D.	0.41	

***Annual total cost in thousand dollars**
****Nodes including a pump station**

The presented mixed-ant optimization algorithm was performed for 20 independent runs. For the case under consideration, Table 6 summarizes the values of most effective parameters of the algorithm. The objective value of the best solution found in any execution is presented in Table 7. The results are obtained by 20 ants within 300 iterations. Upon results, the number of pump stations varied from 2 to 3 in different runs; which provides the variety in designs with only little difference in total annual cost. Note that all executions ended up with the final feasible solutions, and the best performance attained the value of 122.71 as total annual cost in thousand dollars. On the other hand, to test the performance of present algorithm, a powerful nonlinear solver, Lingo 9.0, was employed to find the optimal solution to the problem. Presence and/or absence of pump station produces a binary problem, therefore, the mathematical model lends itself to a Mix Integer Non-Linear Programming $(MINLP)$ problem. Lingo 9.0 reported the best local optimal solution with the annual cost of 122.57 thousand dollars. Two alternatives for the locations of pump stations and their design heads were generated with the same annual cost. First solution locates the pump stations at nodes 1 and 3 with 68.4 (*m*) pumping heads at each node; while the alternative solution sets the pump stations at nodes 1 and 4 with pumping heads of 77.92 (*m*) and 58.92 (*m*), respectively. The values of pipes' diameters remain unchanged for both solutions. Hence, the proposed mixed-ant optimization algorithm is capable to locate near optimal solution in such non-linear and complicated case study within rather few numbers of function evaluations.

Table 8. Comparison on system components between the best found solution and reported local optimums

Pipe No.	**Reported local optimums** **Pipe diameter (m)**	**Best found solution** **Pipe diameter (m)**
1	0.8	0.8
2	0.8	0.8
3	0.8	0.8
4	0.8	0.75
5	0.8	0.8
6	0.8	0.8
7	0.45	0.45
8	0.45	0.5
9	0.45	0.45
10	0.45	0.45
11	0.4	0.4
12	0.45	0.45
13	0.4	0.4
14	0.45	0.4
15	0.45	0.5
16	0.4	0.4
17	0.4	0.4
Pumping head (m)	Node 1: 68.42 Node 3: 68.42 Or Node 1: 77.92 Node 4: 58.92	Node 1: 70.05 Node 3: 66.98

Table 8 summarizes the pipe diameters and pumping heads of optimum solutions. Column No. 2 shows the local optimum solutions found by MINLP formulation, while the next column presents the best solution found by $M\text{-}IACO_R$. As seen, the local optimum solutions reported by Lingo 9.0 recommend different pairs of nodes to install the pumping stations. Also, the associated pumping heads are not the same. However, both local optimums result the equal annual cost as 122.57 thousand dollars. On the other hand, the best solution found by $M\text{-}IACO_R$ suggests almost the same pipe diameters as those of local optimums, and the solution generally follows the local optimums in very similar pattern.

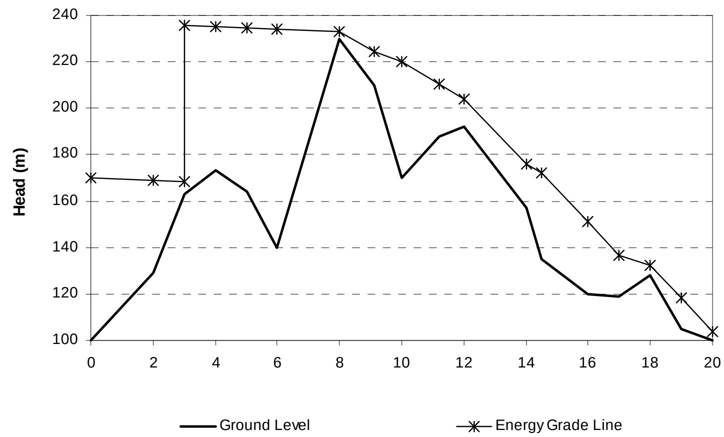


Figure 7. Energy grade line for the best found solution by present ant model over 20 executions

Figure 7 shows the energy grade line for the best found solution by the present algorithm which locates the pump stations at nodes 1 and 3 with pumping heads of 70.05 and 66.98 meters, respectively. Jumps in energy grade line are due to pump stations (nodes 1 and 3), and as it is clear; the energy grade line is reasonably established above the ground level thorough the path.

The average run-time with a personal computer (2.40 GHz CPU and 2 GB RAM) for the water main design problem with 34 decision variables was 12.56 seconds. Reported results are obtained after 6000 function evaluations (20 ants within 300 iterations) which are expected to rapidly increase as the problem grows in size. Moreover, the CPU time becomes larger and larger if the simulation underlying the optimization problem is computationally expensive and addresses extremely non-linear and complex equations. However, similar to other meta-heuristic optimization methods, long run times or even serious deficiencies in finding feasible solutions with limited number of function evaluations may be expected from the proposed ACO in too large-scale problems. . It is highly recommended to test the efficiency of the proposed ACO method in real-life problems like large water distribution networks. Its ability to approach the optimum solution of the large optimization problems within reasonable CPU time and reasonable number of function evaluations may be evaluated in further studies.

The efficient performance of the proposed ACO method in mathematical benchmark problems and tested water main design problem is encouraging to extend its application to multi-objective optimization problems. Various studies have already evaluated the efficiency of different methods in developing the Pareto front of multi-objective problems using the meta-heuristic methods like ACO ([29]). The similar approaches may equally be employed to study the performance of proposed method in problems with contradictory objectives.

5. CONCLUSION

Optimum design of cross-country water mains and associated pumping stations is a relatively complex problem due to its mixed continuous- discrete variables decision space. Realizing the existence of many mixed-variable problems with continuous (i.e., pumping head), discrete (i.e., pipe diameter) as well as binary (i.e., existence or absence of pumping station) decision and/or state variables in various fields of engineering *and* particularly in water resources engineering, this article proposed an ACO based algorithm that directly tackles the optimization problems with mixed variable domains. The extended version of an already existing continuous ACO algorithm was introduced for such mixed-variable problems. It was shown that the proposed transformation from continuous to discrete space, before fitness evaluation, is a subtle point of the algorithm. Inclusion of one or more "*zero solutions*" in the archive in the division of continuous variables effectively resolved the problem of binary decision variables. The method was practiced on a set of mathematical problems and surpassed the results of some other reported algorithms by locating near optimum solutions in remarkably small number of function evaluations. Moreover, its performance in solving a non-convex and highly non-linear forced water main design problem with binary as well as continuous and discrete decision variables was quite satisfactory. To more illustrate the performance of the proposed algorithm, further application into various mixed domain engineering design problems is suggested for prospect studies.

APPENDIX A

Following descriptions are to clarify the parameters of each Gaussian kernel PDF:

- The values of i^{th} variable of all current archived solutions form the vector $\mathbf{\mu^i}$.
- Standard deviation of l^{th} PDF, according to l^{th} solution, at i^{th} construction step, σ_l^i, is proportional to the average distance of s_l^i from other solutions, s_e^i, in the archive. The parameter $\xi(>0)$ resembles the pheromone evaporation coefficient in discrete *ACOs*. The appropriate values of ξ minimizes the chance of further exploration of already scanned areas.
- The l^{th} component of the vector $\boldsymbol{\omega}(\omega_l)$ demonstrates the weight of the l^{th} solution in the archive and reflects the superiority of the solution, i.e. $\omega_1 \geq ... \geq \omega_l \geq ... \geq \omega_k$.
- ω_1 is obtained according to the solution rank (l) Fitness values do not directly enter the equation, which means the weights of archived solutions are not sensitive to the

definition of fitness function. This may be regarded as a strong point of the method.

APPENDIX B

Total cost of the assumed water main includes initial investments and annual operation costs.

The initial investments encompass the costs on:

purchase and installation of the pumps ($Cost_p$)

- pump station house ($Cost_s$)
- accessory equipments ($Cost_{eq}$)
- electrical instruments ($Cost_{el}$)
- purchase and fixing the pipes which is dependent on the pipes' diameters ($Cost_d$)

The annual operation cost is due to the required electricity for pumping the water ($Cost_e$).

The noted costs, at each node or reach, are expressed as follows:

$$\begin{aligned} &Cost_p = Q \times (a_p + b_p \times h_p + c_p \times h_p^2 + d_p \times h_p^3) \\ &a_p = 21225.6 \\ &b_p = 462.368 \\ &c_p = -1.1895 \\ &d_p = 6.2944 \times 10^{-3} \end{aligned} \tag{15}$$

$$\begin{aligned} &Cost_s = a_s + b_s \times Cost_p + c_s \times Cost_p^2 + d_s \times Cost_p^3 \\ &a_s = 4461.41 \\ &b_s = 0.483496 \\ &c_s = 1.28 \times 10^{-7} \\ &d_s = -7.85 \times 10^{-15} \end{aligned} \tag{16}$$

$$\begin{aligned} &Cost_{eq} = a_{eq} + b_{eq} \times Cost_p + c_{eq} \times Cost_p^2 + d_{eq} \times Cost_p^3 \\ &a_{eq} = 4339.24 \\ &b_{eq} = 0.054383 \\ &c_{eq} = -4.18 \times 10^{-10} \\ &d_{eq} = 2.69 \times 10^{-18} \end{aligned} \tag{17}$$

$$\begin{aligned} &Cost_{el} = a_{el} + b_{el} \times (h_p \times Q) + c_{el} \times (h_p \times Q)^2 + d_{el} \times (h_p \times Q)^3 \\ &a_{el} = 28278.1 \\ &b_{el} = 286.957 \\ &c_{el} = -0.04365 \\ &d_{el} = 3.2 \times 10^{-6} \end{aligned} \tag{18}$$

$$
\begin{aligned}
Cost_d &= L \times (a_d + b_d \times D + c_d \times D^2 + d_d \times D^3) \\
a_d &= 7.21125 \\
b_d &= -32.2491 \\
c_d &= 86.3169 \\
d_d &= -48.8413
\end{aligned} \tag{19}
$$

The annual operation cost derived from the required energy for pumping the water may be regarded at each pump station as:

$$Cost_e = C_u \times E_p \tag{20}$$

Where:

$$E_p = \gamma_w \times \frac{Q \times h_p}{1000\eta} \times T \tag{21}$$

In which, C_u is the unit cost of electricity; $E_p\,(KW - hr)$ is the annual electricity consumption; $\gamma_w(N/m^3)$ is the specific weight of water; $Q(m^3/s)$ is the pumped water flow per hour; $h_p\,(m)$ is the pumping head; η is the pumping efficiency; and T is the total hours of pumping in a year.

To calculate the total cost of the assumed system, all explained costs at any node or reach may be incorporated in a unit expression as:

$$Total\,Cost = CRF \times \left(\sum_{n=1}^{NN} Cost_p + Cost_s + Cost_{eq} + Cost_{el} + \sum_{i=1}^{NR} Cost_d \right) + \sum_{n=1}^{NN} Cost_e \tag{22}$$

Where, CRF is Capital Recovery Factor and computed as:

$$CRF = \frac{i(1+i)^n}{(1+i)^n - 1} \tag{23}$$

In which, i is the inflation rate; and n is the estimated length of operation period.

Deep attention to above expressions, Eq. 23 may be paraphrased as follow to derive Eq. 10:

$$
\begin{aligned}
&Total\,Cost = CRF \times \left(\sum_{n=1}^{NN} Cost_p + Cost_s + Cost_{eq} + Cost_{el} + \sum_{i=1}^{NR} Cost_d \right) + \sum_{n=1}^{NN} Cost_e \\
&Total\,Cost = \left\{ CRF \times \left(\sum_{n=1}^{NN} Cost_p + Cost_s + Cost_{eq} + Cost_{el} \right) + \sum_{n=1}^{NN} Cost_e \right\} + \left\{ CRF \times \sum_{i=1}^{NR} Cost_d \right\} \\
&Total\,Cost = C_n(hp_n) + C_i(D_i)
\end{aligned} \tag{24}
$$

REFERENCES

1. Zhu M, Zhang Y, Wang T, Shen B. Mathematical model for the optimal design of long pressurized water supply transmission lines, *In Proceedings of the 2006 International Conference on Hybrid Information Technology*. IEEE Computer Society, Washington DC, USA, 2006;**1**: 573-7.
2. Jabbari E, Afshar A. Optimum layout and design of a water supply line, *Hyd Info Manage*, 2002;**4**(1): 397-406.
3. Martin QW. Linear water supply pipeline capacity expansion model, *J Hydraul Eng-ASCE* 1990;**116**(5): 675-90.
4. Afshar A, Jemma FB, Marino MA. Optimization of hydropower plant integration in water supply system, *J Water Resour Pl-ASCE* 1990;**116**(5): 665-75.
5. Vairavamoorthy K, Ali ME. Optimal design of water distribution systems using genetic algorithms, *Comput Aided Civ Infrastruct Eng* 2000;**15**(2): 374-82.
6. Luettinger J, Clark T. Geographical information system–based pipeline route selection process, *J Water Resour Pl-ASCE* 2005;**131**(3): 193-200.
7. Abbasi H, Afshar A, Jalali MR. Ant-colony-based simulation–optimization modeling for the design of a forced water pipeline system considering the effects of dynamic pressures, *J Hydroinform*, 2010;**12**(2): 212-24.
8. ASCE Task Committee on Evolutionary Computation in Environmental and Water Resources Engineering, accepted 2009, State of the Art for Genetic Algorithms and Beyond in Water Resources Planning and Management *ASCE, J Water Resour Pl-ASCE*, doi:10.1061/(ASCE)WR.1943-5452.0000053.
9. Labadie JW. Optimal operation of multireservoir systems: state of-the-art review, *J Water Resour Pl-ASCE* 2004;**130**(2): 93–111.
10. Cheng CT, Wang WC, Xu DM, Chau KW. Optimizing hydropower reservoir operation using hybrid genetic algorithm and chaos, *Water Resour Manage* 2008; **22**(7): 895-909.
11. Bozorg Haddad O, Afshar A, Mariño MA. Honey-Bees Mating Optimization (HBMO) Algorithm in Deriving Optimal Operation Rules for Reservoirs, *J Hydroinform* 2008;**10**(3): 257-64.
12. Tospornsampan J, Kita I, Ishii M, Kitamura Y. Split-Pipe design of water distribution network using simulated annealing Thailand, *Int J Comput Inf Eng* 2007;**1**(3): 153-63.
13. Vasan A, Raju KS. Comparative analysis of simulated annealing, simulated quenching and genetic algorithms for optimal reservoir operation, *Appl Soft Comput* 2009;**9**(1), 274-81.
14. Madadgar S, Afshar A. An improved continuous ant algorithm for optimization of water resources problems, *Water Resour Manage* 2009;**23**: 2119-39.
15. Sedki A, Ouazar D. Swarm intelligence for groundwater management optimization, *J Hydroinform*, in press, 2011; **13**(3):520-32.
16. Afshar A, Shafii M, Bozorg Haddad O. Optimizing multi-reservoir operation rules: an improved HBMO approach, *J. Hydroinform* 2011; **13**(1): 121-139.
17. Dorigo M. Optimization, Learning and Natural Algorithms, Thesis (PhD). Politecnico di Milano, Milan, Italy, 1992.
18. Dorigo M, Maniezzo V, Colorni A. The ant system: optimization by a colony of

cooperating ants *IEEE Trans Syst Man Cybern* 1996;**26**: 29–42.
19. Bilchev G, Parmee IC. The ant colony metaphor for searching continuous design spaces, *In Fogarty TC (ed) Proceedings of the AISB Workshop on Evolutionary Computation, vol 993 of LNCS*. Springer, Berlin Heidelberg New York, 1995, pp. 25–39.
20. Monmarche N, Venturini G, Slimane M. On how Pachycondyla apicalis ants suggest a new search algorithm, *Future Gener Comput Syst* 2000;**16**: 937–46.
21. Dreo J, Siarry P. A new ant colony algorithm using the hierarchical concept aimed at optimization of multiminima continuous functions, *In Dorigo M, Caro GD, Sampels M (eds) Proceedings of the 3rd International Workshop on Ant Algorithms (ANTS 2002), vol 2463 of LNCS*. Springer, Berlin Heidelberg, New York, 2002, pp. 216–221.
22. Socha K, Dorigo M. Ant colony optimization for continuous domains, *Eur J Oper Res* 2006;**185**(3): 1155-73.
23. Schluter M, Egea JA, Banga JR. Extended ant colony optimization for non-convex mixed integer nonlinear programming, *Comput Oper Res*, 2009; **36**(7), 2217-29.
24. Costa L, Olivera P. Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems, *Comput Chem Eng* 2001;**25**: 257–66.
25. Yiqing L, Xigang Y, Yongjian L. An improved PSO algorithm for solving non-convex NLP/MINLP problems with equality constraints, *Comput Chem Eng* 2007;**31**(3), 153-62.
26. Kocis G.R, Groesmann IE. Relaxation strategy for the structural optimization of process flowsheets. *Ind Eng Chem Res* 1987;**26**(9), 1864-80.
27. Kocis GR, Groasmann IE. A modelling/decompoaition strategy for MINLP optimization of procese flowsheets. *Comput Chem Eng* 1989;**13**,797-819.
28. Yuan X, Zhang S, Pibouleau L, Domenech S. A mixed-integer nonlinear-programming method for process design. *RAIRO, Oper Res* 1989;**22**, 331–46.
29. Afshar A, Sharifi F, Jalali MR. Non-dominated archiving multi-colony ant algorithm for multi-objective optimization: Application to multi-purpose reservoir operation, *Eng Optimiz* 2009;**41**(4), 313-25.