Mansoura University
Faculty of Engineering
Mansoura Engineering Journal

Modified Jaya Algorithm for Optimal Design of Water Distribution Network

Hossam A.A. Abdel-Gawad

KEYWORDS:
Jaya optimization algorithm, Optimal pipe networks, Free control parameters, Common parameters

Abstract— **For the first time, the recently proposed Jaya metaheuristic optimization algorithm (JA) is used for optimal design of water distribution networks (WDN's). This algorithm has no control parameters, which eliminates the exhausting computational effort required to carry out the essential tune up step for these parameters. A new variant of the algorithm, named free sensitivity analysis Jaya algorithm (FSAJA), is suggested to be free from even the common metaheuristic algorithms parameters, i.e., population size, number of generations and penalty function. Six different variants of the proposed algorithm are investigated to settle the best one using a recently developed two performance criteria. Three famous benchmarks WDN's and a national one, are solved to examine the algorithm. Comparing the performance of FSAJA, with JA and the various evolutionary algorithms available in the literature shows the promising effectiveness, efficiency, and robustness of the proposed algorithm. A new global minimum is achieved for the national WDN.**

I. INTRODUCTION

CAPITAL cost of any water distribution network (WDN) is the major investment part of most water supply systems [1]. Numerous research in the last four decades is interested in determining the optimal cost of the pipe networks under different constraint requirements, i.e. demands and minimum pressure heads at the nodes and limits of velocities within the pipes. In general, two approaches are used to find the optimal solution for any optimization problem: 1) the deterministic approach which requires evaluating of the objective function and their derivatives concerning the different decision variables [2], [3], and 2) the stochastic approach which requires the evaluation of only the objective function, and is more suitable in handling problems based on discrete decision variables [4]–[13].

Despite easiness and robustness of the stochastic algorithms, they need more computational effort with respect to the deterministic ones, i.e., a huge number of objective function evaluations, which is proportional exponentially with the number of the parameters that steer the stochastic algorithm to the optimal solution. Two types of parameters are generally found in nearly all metaheuristic optimization algorithms, the specific control algorithm dependent parameters and the common parameters. Values of the parameters that enhance the algorithm performance are problem dependent and must be readjusted, by a sensitivity analysis, for any changes in characteristics of the studied WDN, e.g. number of the decision variables, number of alternatives of every decision variable, nodes demands and pressure constraints, and any other necessary input data for analyzing the WDN [14].

Unfortunately, performance of nearly all the population algorithms are dependent on a pre-design number of control algorithm parameters (NCAP) [15], [16], e.g. five for Genetic Algorithm (GA), six for Shuffled Complex Evolution (SCE), five for Ant Colony Optimization (ACO), four for Simulated Annealing (SA), four for Cross Entropy (CE), four for Shuffled Frog Leaping Algorithm (SFLA), four for Particle Swarm Optimization (PSO), three for Scatter Search (SS), three for Harmony Search (HS), three for Soccer League Competitions (SLC), three for Improved Mine Blast Algorithm (IMBA), two for Differential Algorithm (DE), and two for Gravitational Search Algorithm (GSA).

Received: (19 May, 2021) - Revised: (10 June, 2021) - Accepted: (16 June, 2021)
* ***Corresponding Author:*** *Hossam A.A. Abdel-Gawad, Associate Professor, Irrigation & Hydraulics Engineering Department, Faculty of Engineering, Mansoura University, Egypt (Email: hossamaaa@mans.edu.eg; hossamgawad@yahoo.com)*

Nomenclature

$\boldsymbol{A}_{NLxNL}$	a square matrix with *NL* rows
AVE^k	average candidates costs in generation k
C_{ave}	average minimum costs in multiple runs
C_{HWi}	Hazen-Williams coefficient of friction for pipe *i*
C_i	cost of pipe *i* per unit length
$C_{\max}$	maximum permissible cost of the network
$C_{\min}$	minimum cost from the best run
C_{opt}	minimum cost ever reached in the literature
C_{std}	standard deviation of minimum costs in 100 runs
D_i	diameter of pipe *i*
$D_{\max}$	maximum available diameter
$D_{\min}$	minimum available diameter
$D_{j,i}^{k/}$	generated diameter
$D_{j,i}^{k}$	diameter for decision variable *i* in candidate *j* at generation *k*
$D_{Ave,i}^{k}$	average diameters for pipe *i* in generation *k*
$D_{Best,i}^{k}$	diameter for pipe *i* at best candidate in generation *k*
$D_{Worst,i}^{k}$	diameter for pipe *i* at worst candidate in generation *k*
E	probability of reaching C_{opt} in one run
$f()$	objective function
ff	feasible objective function
F_j^m	residual pressure head for loop *j* at iteration *m*
$\boldsymbol{F}_{NL}$	vector for residual pressure head at different loops
H_j	pressure head at node *j*
$H_{j\min}$	minimum pressure head at node *j*
Iff	infeasible objective function
K_i	resistance coefficient for pipe *i*
L_i	length of pipe *i*
LP_j	number of pipes in loop *j*
Mm	maximum number of Newton-Raphson iteration
n	exponent of the discharge in the friction equation
ND	number of available commercial pipe diameters
NJ	number of nodes
NL	number of loops
NP	number of pipes
$NPOP$	number of candidates in a population
N_{eval}	minimum number of objective function evaluations to the best cost
N_{OFE}	number of total objective function evaluations in a run
N_{space}	total alternative number of solutions
N_{sim}	number of multiple runs
N_{success}	number of runs reached C_{opt}
$N_{\text{total-opt}}$	actual number of objective function evaluations to catch optimal solution C_{opt}
P^k	penalty at generation *k*
q	uniform random number between (0,1)
Q_i^m	discharge in pipe *i* at iteration *m*
r	uniform random number between (0,1)
$sign_i$	sign for the flow direction in pipe *i*
STD^k	standard deviation of candidates costs in generation k
v	uniform random number between (0,1)

Greek Symbols

α^k	dynamic penalty coefficient at generation *k*
ΔH_j	reduction of the pressure head at node *j*
$\Delta \boldsymbol{Q}_{NL}$	vector of unknown loop's corrections
$\eta_{\text{ave-alg}}$	average global algorithm performance
$\eta_{\text{ave-cost}}$	average effectiveness in multiple runs
$\eta_{\text{ave-eval}}$	average efficiency in multiple runs
$\eta_{\text{best-alg}}$	global algorithm performance for best run
η_{cost}	effectiveness in reaching C_{opt}
η_{eval}	efficiency of the best run
η_{gen}	generation efficiency
$\eta_{\text{gen-success}}$	generation success efficiency

A minimum number of alternatives for any control parameter is two; consequently, the essential minimum effort to carry out any sensitivity analysis is at least 2^{NCAP} times the actual computational effort required for the optimization process itself. To the author's knowledge, unfairly all the literature ignores the exhausted sensitivity computational effort required to tune up the control parameters, in comparing between performances of different algorithms.

Mora-Melia et al. [17] studied the effect of almost all the control parameters for four optimization methods, i.e. GA, PSO, HS, and SFLA, which were applied on four benchmark pipe networks. Each network was solved 200 times for any combination of the control parameters, with at least 25,000 runs for any pipe network handled by a specified algorithm. In their research, the ratio of the computational effort of the sensitivity analysis concerning the actual optimization effort is equal to or more than 124. Sensitivity analysis was carried to tune up three parameters of the SFLA, i.e. accelerator parameter, frog per submemeplex, and evolutionary steps, with 225 different combinations [18].

To avoid the huge computational cost required for the sensitivity analysis of the existed metaheuristic optimization methods, the researchers give attention, in the last decade, for developing a free control parameter algorithm. But still a limited number of these algorithms are applied to optimal design of WDN's as: 1) Self-Adaptive Differential Evolution Algorithm (SADE) [12], 2) Modified dither Differential Evolution algorithm (MdDE) [19], and 3) Fittest Individual Referenced Differential Evolution Algorithms (FDE) [20].

For the first time, the present work adopted a newly developed free control parameter evolutionary algorithm, named by Jaya algorithm (JA), to optimal design of WDN's. JA is designed and presented for the first time by Rao in (2016) [21], [22], and is dependent only on the common population algorithm parameters, i.e.: 1) penalty function, 2) population number, and 3) number of generations. Rao [21] encouraged the researchers to improve JA to be a robust algorithm in different engineering disciplines.

Several papers modified the Jaya algorithm to enhance its performance in handling different optimization problems. The algorithm is modified to solve multi-objective problems, (MO-Jaya), and applied to three machine processes [23]. Modification for Jaya called self-adaptive multipopulation Jaya, (SAMP-Jaya), was suggested and applied to various numbers of benchmark mathematical problems, [24]. SAMP-Jaya results were found better than the corresponding results of the GA, PSO, and Cuckoo Search Algorithm (CSA). Elitism-based SAMP-Jaya algorithm is proposed for design of heat pipes [25]. The standard Jaya algorithm was applied for a water resources management problem to find optimal reservoir operation [26]. Jaya is improved to a new variant, (I-Jaya), by merging it with a clustering strategy, [27], to find optimal structural damage identification. Adaptive multiteam perturbation guiding Jaya, (AMTPG-Jaya), was suggested and applied on different benchmark problems [28], and found superior concerning the standard JA.

In the course of executing the present work, new literature is published, the Lévy distribution is merged with the standard Jaya, (LJA), to supplement JA with a little opportunity of moving large steps to escape from the clustered exploration around a trapped local minimum to new region in the solution domain [29]. Both *k*-means clustering strategy and a new learning technique, based on best and medium candidates, are involved in Jaya to enhance its performance in handling parameter identification of aerofoil [30].

This research presents a modified variant of JA to solve the WDN's optimization problems for the first time [31], and any similar problems that have positive discrete values for the decision variables. The proposed algorithm is designed to be free from the sensitivity analysis of common parameters, i.e. (FSAJA).

The remaining paper consists of four sections. Section two is concerned with the adopted methodology for different steps of the work. It consists of: a) the most efficient mathematical statement for analyzing steady flow in the WDN, b) the optimization problem statement, c) standard JA, d) the proposed FSAJA, a simple hypothetical optimization example was suggested to clarify both of the two algorithm variants, i.e., JA and FSAJA, e) approaches used in handling the common parameters in FSAJA with an illustrative flowchart, and f) criteria used to preference between different possible variants of FSAJA. Description of the handled WDN's is presented in section three. Section four is concerned with choosing the final form of the FSAJA and discussing its performance concerning performances of JA and other algorithms. Finally, conclusions, recommendations and suggested future works are presented in section five.

II. METHODOLOGY

A. Hydraulic Analysis

Determination of the flow rates and the pressure head losses within any suggested pipe diameters of a pipe network is an inevitable step within the optimal design process. The output results are used to check both the velocity limits within the pipes and the minimum required pressure heads at different nodes of the network. Computational effort of the hydraulic analysis is proportional to number of equations/unknowns (NE) used to analyze the WDN. Various mathematical statements can be used to simulate steady flow in the network, and may be considered as one of the following systems [32]: 1) both continuity equations of flow rates at different nodes and equations of head losses summation along different loops, 2) both continuity equations of flow rates at the nodes and pressure head losses equations within the pipes, 3) pressure heads at the nodes which are usually used and adopted by the hydraulic solver EPANET2 [33], and 4) flowrates corrections around different network loops which generate the minimum number of equations, and require minimum computational effort which is proportional to NE^3, in case of solving the generated equations matrix with the Gauss elimination method [34], but needs a prespecified initial balanced guess for the flow rates at different nodes of the network. The last mathematical statement, i.e., the most computational cost efficient, is implemented in the present work and can be represented by nonlinear equations. The linearized form of these equations can be solved iteratively using the Newton-Raphson technique as, [35]:

$$\boldsymbol{A}_{NL\times NL}^{m} * \Delta\boldsymbol{Q}_{NL}^{m} = \boldsymbol{F}_{NL}^{m} \qquad m = 1,2,...,Mm \qquad (1)$$

where:

$$F_j^m = \sum_{i=1}^{LP_j} sign_i \cdot K_i \cdot \left(Q_i^m\right)^n \qquad j = 1,2,...,NL \qquad (2)$$

$$Q_j^m = Q_j^{m-1} + \sum_{i=1}^{ni} \Delta Q_k^{m-1} \qquad i = 1,2,...,LP_j \qquad (3)$$

$$K_i = \frac{10.667\,L_i}{C_{HW_i}^{1.852}\,D_i^{4.871}} \qquad i = 1,2,...,NP \qquad (4)$$

$$A_{j,j}^{m+1} = n \cdot \sum_{i=1}^{LP_j} K_i \cdot \left|Q_i^m\right|^{n-1} \qquad j = 1,2,...,NL \qquad (5)$$

$$A_{j,l}^{m+1} = -n \cdot \sum_{i=1}^{n_{jl}} K_i \cdot \left|Q_i^m\right|^{n-1} \qquad j \neq l \qquad (6)$$

where, $\boldsymbol{A}_{NL\times NL}^{m}$ is a square matrix at iteration m and its elements $A_{j,l=1\to NL}^{m} = \partial F_j^m / \partial \Delta Q_l$, NL is number of loops in the network, $\Delta\boldsymbol{Q}_{NL}^{m}$ is the unknown vector for the flow rates corrections of the NL loops at iteration m and ΔQ_k^m is the flow rate correction for loop k, $\boldsymbol{F}_{NL}^{m}$ is a vector represents the summation of the pressure head losses around the network loops with elements $\Delta\boldsymbol{Q}_{NL}^{m}$, Mm is a prespecified maximum permissible number of iterations to resolve the linearized equations, $sign_i$ is the sign of the flow direction in pipe i, i.e. positive for clockwise direction and negative for anticlockwise direction, K_i is the resistance coefficient of pipe i in S.I. units and depends on the utilized friction head loss equation (Hazen-Williams equation is used in the present work), Q_i is the flow in pipe i, n is a flow rate exponent and equal to 1.852, ni is the number of loops associated with pipe i, LP_j and NP are the number of pipes in loop j and total number of pipes in the network, respectively, L_i, D_i, and C_{HWi} are the length, diameter, and Hazen-Williams coefficient of friction for pipe i, respectively, and n_{jl} is the number of pipes associated with both i and l loops. The flow rate correction vector $\Delta\boldsymbol{Q}^m$ must be added to the initial discharge rates, $\boldsymbol{Q}^m$, in the pipes before processing to the next iteration m + 1.

B. The Optimization Statement

Objective function for minimum capital cost of WDN can be represented as, [36]:

$$Min.f\left(D_1,..,D_{NP}\right)=\sum_{i=1}^{NP}L_i\cdot C_i+P^k\cdot\sum_{j=1}^{NJ}\Delta H_j \qquad (7)$$

with constraints limits:

$$D_{\min}\le D_i\le D_{\max} \qquad i=1,2,...,NP \qquad (8)$$

$$H_j\ge H_{j\min} \qquad j=1,2,...,NJ \qquad (9)$$

where;

$$\Delta H_j=Max.\left(\left[H_{j\min}-H_j\right],0.0\right) \qquad (10)$$

$$P^k=10^8 \qquad (11)$$

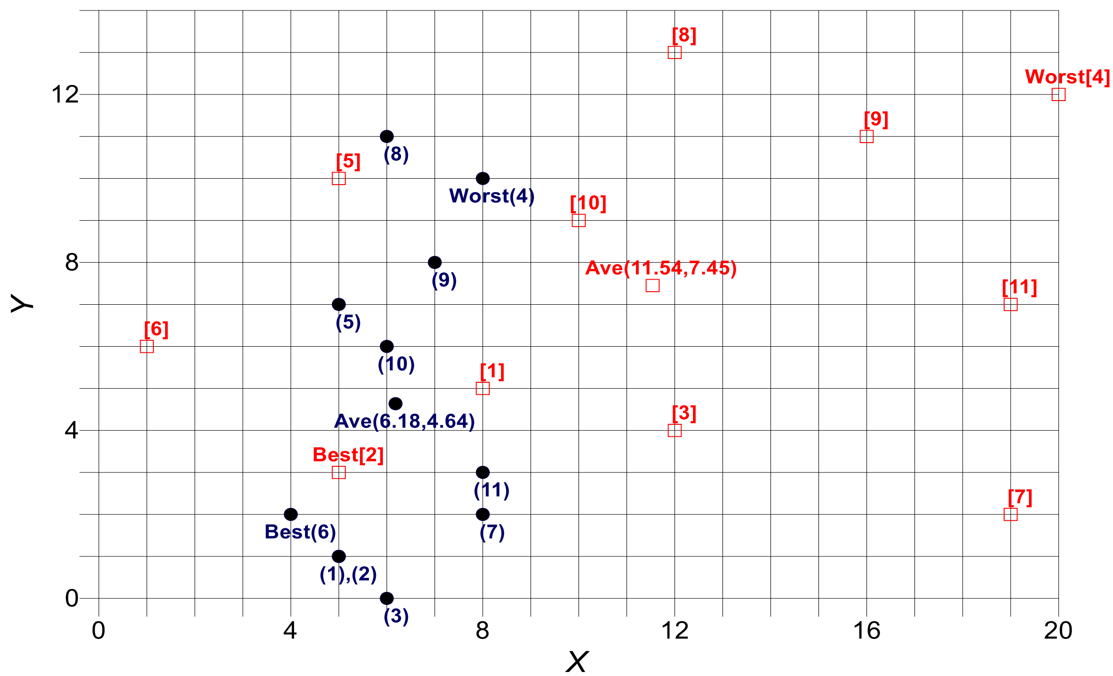

Fig. 1 Description of JA [$NPOP$ = 11, r_X = 0.81, q_X = 0.023, r_Y = 0.273, q_Y = 0.528]

where, *f*( ) is the objective function of the problem, D_i is diameter of pipe *i*, $D_{\min}$, $D_{\max}$ are minimum and maximum discretized range for the available commercial diameters, respectively, *NP* is total number of pipes/decision variables of the network, C_i is cost per unit length of pipe *i* and is diameter dependent, P^k is a penalty constant (static penalty) at the generation number *k*, H_j, $H_{j\min}$ are calculated pressure head and minimum permissible pressure head at node *j*, respectively, *NJ* is total number of nodes, and ΔH_j is the violation in the pressure head constraint at node *j*. The first term on the right-hand side of Eq. (7) represents the actual cost of the pipes, while the second term is a penalization function for the infeasible candidates, which violate the permissible pressure head constraint at any node of the network.

C. Standard Jaya Algorithm

The algorithm is initiated by generating a population that consists of a preselected number of candidates; each represents a vector of uniform random choices of the decision variables (discrete pipe diameters) within their permissible commercial range. Every decision variable in a candidate can be updated as:

$$D_{j,i}^{k/}=D_{j,i}^k+r_i^k\left(D_{Best,i}^k-\left|D_{j,i}^k\right|\right)-q_i^k\left(D_{Worst,i}^k-\left|D_{j,i}^k\right|\right)$$

$$j=1,2,...,NPOP \quad \& \quad i=1,2,...,NP \qquad (12)$$

where, $D_{j,i}^{k/}$ is the new suggested diameter at decision variable *i* in individual *j* within the generation *k*, $D_{j,i}^k$ is the present diameter, $D_{Best,i}^k$ and $D_{Worst,i}^k$ are diameters of pipe *i* at the fittest candidate and the worst one respectively, *NPOP* is population size that represents number of candidate solutions within the generation *k*, and r_i^k and q_i^k are two uniform random values between 0 and 1 for the decision variable *i* at any candidate solution, *j* = 1, 2,...,*NPOP*, in the generation *k*. For every old candidate in the population, a new one is generated and the fittest one survives to the next generation *k*+1.

The JA is based on only one learning phase, which is moving toward the best individual and avoiding the worst one [21], [22]. Using random uniform numbers, for generating a new suggested decision variable, enhances the exploration of the solution domain. Also, considering the absolute value of the decision variable, if its acceptable range is extended in the negative values, improves the exploration process. On the other hand, utilizing the same two random numbers, for the same decision variable in all the candidates within the same generation fixes the scale of movements between different generated candidates and both best and worst candidates which limits to some extent the exploration efficiency.

Illustrative Example

To explain the process of the JA, a simple objective function is considered as: minimize $Z = X^2 + Y^2$, with two decision variables only, X and Y, and their corresponding ranges are the discrete integer numbers from 0 to 14, and from 0 to 20, respectively, see Fig. 1. The logical minimum objective function value is zero at solution (X = 0, Y = 0) and the total number of the available solutions is 15 x 21. A population of 11 candidates is generated randomly within the permissible ranges for the two decision variables as shown in Fig. 1, at the positions of the square symbols. Then two uniform random numbers between 0 and 1 are generated for every decision variable as: r_X = 0.81, q_X = 0.023, r_Y = 0.273, q_Y = 0.528. The best candidate is the second one with (X = 5, Y = 3) and Z_{Best} = 34, while the fourth candidate is the worst one with (X = 20, Y = 12) with Z_{Worst} = 544. For every candidate at the square symbols, a corresponding one is generated at the corresponding circle symbols, for example, in case of candidate 11 at (X = 19, Y = 7) and Z = 410:

$$X^{/} = 19 + 0.81*(5-19) - 0.023*(20-19) = 7.637 \approx 8\,,$$

$$Y^{/} = 7 + 0.273*(3-7) - 0.528*(12-7) = 3.268 \approx 3\,,$$

With final objective function equal to 73 which is < 410, consequently, the generated solution (X = 8, Y =3) replaces the old one in the next generation. The corresponding generated candidate to the third one at (X = 12, Y = 4) and Z = 160, can be estimated as:

$$X^{/} = 12 + 0.81*(5-12) - 0.023*(20-12) = 6.146 \approx 6\,,$$

$$Y^{/} = 4 + 0.273*(3-4) - 0.528*(12-4) = -0.497 < 0\,,$$

then $Y^{/} = 0$, with final Z = 36 < 160, so the generated solution (X = 6, Y = 0) replaces the old one. The generated decision variable between two discrete alternatives is reassigned to the nearest discrete value and the violated $Y^{/}$ at –0.497 is reassigned at the nearest permissible limit of Y at 0. In one generation, the best solution is moved to the sixth candidate at (X = 4, Y = 3) with Z = 25, and the worst objective function enhanced to Z = 164 at (X = 8, Y = 10).

It must be noticed that: 1) all the generated candidates at the circle symbols have better fitness than the corresponding ones in the old generation, 2) the solution domain is reduced from (X = 0 to 20, Y = 0 to 14) to (X = 0 to 8, Y = 0 to 11), and due to the learning philosophy of JA, the dispersion of the population is continually decreasing and moves away from the worst solution to the best one, and 3) the average of the new populations is moved in one generation process from (X = 11.54, Y = 7.45) to (X = 6.18, Y = 4.64) with distance of movement equal to 6 in the direction of the optimal solution at (X = 0, Y = 0). Consequently, JA can be considered as a very efficient algorithm in case of solving relatively simple problems with limited number of local minimums and a generally concave surface for the solution domain.

D. Free Sensitivity Analysis Jaya Algorithm (FSAJA)

The soul of the proposed FSAJA is based on a recently new variant of the algorithm named by comprehensive learning JA (CLJAYA), [٣١]. Instead of using only one learning phase, Eq. (12), three learning phases are adopted as:

$$D_{j,i}^{k\,/} = D_{j,i}^{k} + r_{j,i}^{k}\left(D_{Best,i}^{k} - D_{j,i}^{k}\right) - q_{j,i}^{k}\left(D_{Worst,i}^{k} - D_{j,i}^{k}\right) \qquad if\ \ 0 \le v \le 1/3 \qquad (13)$$

$$D_{j,i}^{k\,/} = D_{j,i}^{k} + r_{j,i}^{k}\left(D_{Best,i}^{k} - D_{j,i}^{k}\right) - q_{j,i}^{k}\left(D_{Ave,i}^{k} - D_{j,i}^{k}\right)$$

$$D_{Ave,i}^{k} = \frac{\sum_{j=1}^{NPOP} D_{j,i}^{k}}{NPOP} \qquad if\ \ 1/3 < v \le 2/3 \qquad (14)$$

$$D_{j,i}^{k\,/} = D_{t,i}^{k} + r_{j,i}^{k}\left(D_{Best,i}^{k} - D_{j,i}^{k}\right) - q_{j,i}^{k}\left(D_{s,i}^{k} - D_{t,i}^{k}\right)$$

$$s \ne t \ne j \qquad if\ \ 2/3 < v \le 1 \qquad (15)$$

where, $r_{j,i}^{k}$ and $q_{j,i}^{k}$ are the two uniform random values between 0 and 1 for decision variable i at candidate j in generation k, s, t are two uniform integer random numbers between 1 and $NPOP$, $D_{Ave,i}^{k}$ is average diameters of the decision variable i for all the candidate solutions in the generation k, $D_{s,i}^{k}$ and $D_{t,i}^{k}$ are the decision variable i at the two randomly selected candidates s, t within the population, and v is a uniform random number which classifies the learning phase for a specified candidate j.

CLJAYA uses the same previous three learning phases, but with the following differences: 1) the random numbers r, q in the three learning phases are candidate independent, and 2) r, q in Eqs. (13, 14) are normal standard random numbers. The above form of the learning process is adopted after a wide investigation for different possible algorithm variants within the literature.

The first learning strategy of the FSAJA inherits the feature of JA, i.e., moving to best solution and away from the worst one. But if they reached best solution is a local minimum, the algorithm is trapped at that minimum. So, in case of handling complex solution domains with a huge number of the local minimum like the WDN problems, another helpful learning strategy must be added to JA to escape from the possibility of trapping at an immature local minimum. Therefore, the second and the third learning strategies, Eq. (14) and (15), are gathered with the standard learning strategy of JA, Eq. (13).

The second strategy is based on guiding the movement of the selected candidate by both best candidate and average position of the population. The average position of the whole population is continuously moving depending on the clustered candidates around the trapped local minimum and some scatter candidates which are still away from that minimum. Consequently, using the information for average position of the population allows escaping from the trapped local minimum and exploring the solution domain more deeply, at the expense of the convergence speed and the corresponding excess in the computational effort to the final minimum. As the generations proceed, the percentage of the clustered candidates around the best solution increases which is weakened the effect of the average population information in giving a candidate big movement to move away from the best.

The third learning strategy has an action similar to the mutation process in different metaheuristic algorithms. While the new suggested candidate is directed to the best solution its movement is distorted with random information based on any two uniformly selected candidates from the population. This strategy gives the newly generated candidate a great chance to explore the solution domain and to move out of the last limits reached by the previous generation. However, with the continuity of the generation process most of the candidates are getting closer to the best reached minimum and that strategy loses its significance, as most of the candidates are nearly became have the same information.

Decision variables, in nearly all the optimization problems, are constrained with upper and lower limits. Standard JA has reassigned the violated decision variable (which is generated out of its permissible range) to the closest limit. That clusters the decision variables at their limits, which weakened exploration of the solution domain. To avoid that disadvantage, a new scheme is suggested to restore the violated decision variable inside its range. The idea is representing the nearest limit with a mirror and replacing the violated decision variable with its image located in the permissible range as:

$$D_{j,i}^{k} = 2D_{\min} - D_{j,i}^{k} \qquad \text{if } D_{j,i}^{k} < D_{\min} \tag{16}$$

$$D_{j,i}^{k} = 2D_{\max} - D_{j,i}^{k} \qquad \text{if } D_{j,i}^{k} > D_{\max} \tag{17}$$

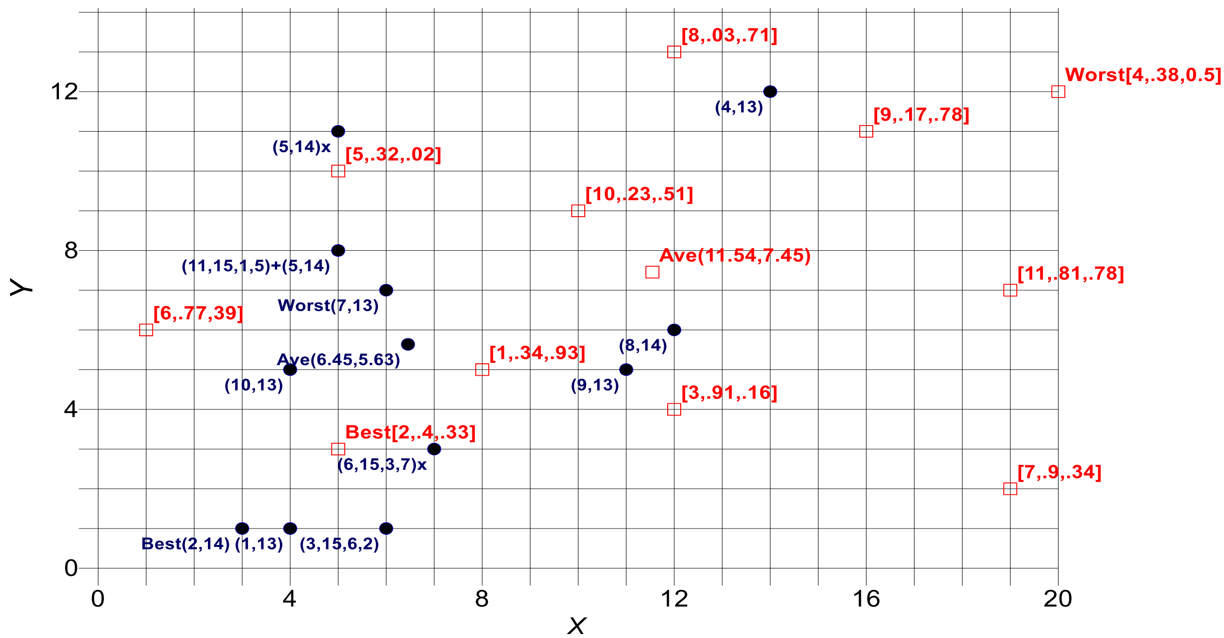

Fig. 2 Description of FSAJA [*NPOP* = 11]

Candidates in generation k are represented by squares with their corresponding notations in square brackets, [j, r, q], which included the candidate number j, and the corresponding two uniform random numbers r, q used to generate the suggested new candidate, respectively. Suggested candidates are represented by circles and their notations in brackets, (j, Eq.), with number of the candidate, j, and the used equation to generate it, respectively. In case of adopting Eq. (15), the two subsequent numbers represent s and t, respectively. Ave means the average of the two decision variables for all the 11 candidates in generation k.

For the previous illustrative example, the same generated 11 candidates for the purpose of explanation JA as shown Fig. 1 are reused to demonstrate the process of the FSAJA in Fig. 2. In case of FSAJA, for every decision variable in any candidate, two uniform random numbers are generated between 0 and 1; consequently, for every candidate with 2 decision variables, four uniform random numbers must be generated. For the sake of simplicity only, it is assumed in the present example that $r_X = q_Y$ and $q_X = r_Y$; consequently, the initial candidates are supplemented with square brackets containing the candidate number followed by r_X and q_X, respectively. The average solution for all the candidates is found at (X = 11.54, Y = 7.45). For every candidate at a square symbol, a corresponding one is generated at a circle symbol which supplemented by a bracket containing the candidate number followed by the equation used in the generation process, and in case of using Eq. (15), the following two numbers represent the randomly selected candidates used, i.e., *s* and *t*, respectively. Every candidate has an equal chance to create a new one using any of the three Eqs. (13, 14, and 15). Consider the candidate number 1 with (X = 8, Y = 5) and Z = 89, the four generated random numbers are $r_X = q_Y = 0.34$, and $q_X = r_Y = 0.93$, respectively, see Fig. 2. Then the corresponding generated candidate using Eq. (13) is:

$$X' = 8 + 0.34*(5-8) - 0.93*(20-8) = -4.18 = 4.18 \approx 4,$$

$$Y' = 5 + 0.93*(3-5) - 0.34*(12-5) = 0.76 \approx 1,$$

with Z = 17, thus it replaces the old one. Consider the fifth candidate at (X = 5, Y = 10) with Z = 125, its corresponding generated one is produced from Eq. (14) which depends on the average positions of the whole candidates in the population (X = 11.54, Y = 7.45) as:

$$X' = 5 + 0.32*(5-5) - 0.02*(11.54-5) = 4.87 \approx 5,$$

$$Y' = 10 + 0.02*(3-10) - 0.32*(7.45-10) = 10.68 \approx 11,$$

with Z = 146 which is bigger than 125; consequently, the old candidate survives to the next generation. Consider the sixth candidate at (X = 1, Y = 6) and Z = 37, Eq. (15) is used to generate its corresponding candidate with two uniformly selected random candidates (s = 3, t = 7), located at (X = 12, Y = 4) and (X = 19, Y = 2), respectively. The generated candidate is:

$$X^{/} = 1 + 0.77*(5-1) - 0.39*(12-19) = 6.81 \approx 7,$$

$$Y^{/} = 6 + 0.39*(3-6) - 0.77*(4-2) = 3.29 \approx 3,$$

with Z = 58 > 37; consequently, the old candidate, at (X = 1, Y = 6), survives to the next generation. From Fig. 2, the following notes can be observed:

The fifth (X = 5, Y = 11) and sixth (X = 1, Y = 6) candidates in the old generation at the squares were processed through the second and third learning strategy, respectively, which lead to two worse candidates, that means the surviving of these two old candidates to the next generation. The generated fifth candidate moves away from the direction that connects the best and the worst solutions, which gives the candidate the opportunity of exploring the new region around it more precisely. This behavior is preferable in case of handling a complex solution domain with a lot of dips.

The average position of the new generation is (X = 6.45, Y = 5.63), considering the surviving candidates only, has a smaller movement from the average position in the previous generation (X = 11.54, Y = 7.45) in comparison with the corresponding movement in JA, see Fig. 1.

The decrease in the dispersion of the candidates has a slower rate than the corresponding rate of the JA, which gives a bigger chance of exploring different local minimums within the solution domain. In addition, the third learning provides the candidates with random movement depend on two uniformly random selected candidates.

E. Common Parameters

Despite the great advantage of the free specific parameter JA, it still needs some preliminary computational effort to adjust its performance, against the remaining common parameters. The proposed FSAJA is designed, as described in the following paragraphs, to be free even from those common parameters, i.e., 1) population size, 2) number of generations, and 3) the form of the penalty function, which is used only in optimization problems that restrict their evaluation function response with some constraints to be acceptable.

1) *Population size*: population may be varied from small size, with limited diversity within the solution domain and a consequent fast termination to a local minimum, to a large size with the much enhanced diversity and exploration for the solution domain but with slower movement to a more likely global minimum. Optimal population size is the one that has maximum likelihood to reach the best solution in minimum computational effort.

In the present work, both static and dynamic population sizes are studied. Wide range for initial number of the candidate solutions is considered to explore the best one. In case of using static population size, the number of initial candidates remains constant for all the generations. On the other hand, in the dynamic approach, the population size is randomly changed between subsequent generations as [37]:

$$NPOP^{k} = round\left(NPOP^{k-1} \times (1+r^{k})\right)$$
$$3 \le 0.1(NP) \le NPOP^{k} \le NP \qquad (18)$$

where, $NPOP^k$ is the population size at generation k with a lower limit equal to the biggest value for $0.1NP$ and 3, r^k is a uniform random number between –0.5 and 0.5, and *round*() means return the integer number. After every generation, the objective function's values, Eq. (7), are ranked in ascending order, i.e., the best will be the first and the worst will move to the last. If $NPOP^k < NPOP^{k-1}$, only the first best $NPOP^k$ candidates move to the next generation k. But if $NPOP^k > NPOP^{k-1}$, the whole candidates in generation k–1 are survived and fittest ($NPOP^k - NPOP^{k-1}$) candidates are duplicated to fill the extra size of generation k.

2) Generations number: a suggested criterion is used to terminate the optimization process instead of using a constant prespecified number of generations. The criterion is based on diversification of the population, within the solution domain. As there is diversification of the different candidates, there is considerable opportunity of exploring the solution domain and avoiding local minima. As that diversification approaches zero, the population is clustered at local/global minima, and the process must be terminated. Any of the following two criteria are used to terminate the optimization process:

$$\frac{STD^{k}}{AVE^{k}} < 0.0001 \qquad \text{or} \qquad Iter. > 30 \qquad (19)$$

where, STD^k, AVE^k are the standard deviation and average of the population objective functions, Eq. (7), in generation k, and *Iter*. is the number of generations processed with no improvement in the best feasible solution. Practically, the first criterion is almost controlling the termination process.

3) Penalty function: the form of the penalty function is a common parameter that influences performance of any population algorithm. To eliminate the computational effort required to tune up that parameter, the self-adaptive penalty, Eq. (20), as suggested by Afshar and Mariño, [3], is used here:

$$P^{k} = \alpha^{k-1} \times P^{k-1} = \frac{Min.\left(ff_{1\to NPOP}\right)^{k-1}}{Min.\left(Iff_{1\to NPOP}\right)^{k-1}} \times P^{k-1} \qquad (20)$$

where, $ff_{1\to NPOP}$ are costs of feasible candidate solutions, when $\Delta H_{1\to NJ}$ = 0, $Iff_{1\to NPOP}$ are total penalized costs of infeasible candidate solutions in a generation, $Min.(ff_{1\to NPOP})^{k-1}$, and $Min.(Iff_{1\to NPOP})^{k-1}$ are minimum/best feasible and infeasible costs at generation number k–1, respectively, and α is a dynamic penalty coefficient. The coefficient of the penalty, α, is adjusted after each generation to keep the search around the boundary of the feasible domain, by giving minor violations in the pressure head a relatively small penalization. This process enables exploiting and surviving different information's within the candidate solutions just outside the feasible solution domain. The initial penalty coefficient, P^0, can be considered as very large or very small value and it is self-adjusted exponentially with the generation process to approach a stable value.

An illustrative flowchart for different alternative forms of the proposed FSAJA is shown in Fig. 3.

F. FSAJA Performance

It is inevitable to have criteria to judge the performance of any algorithm and compare it with the various other ones. Any algorithm performance depends on the following points: 1) Effectiveness: the proximity of the optimal solution to the global known solution, 2) Efficiency: the number of objective function evaluations necessary to reach the optimum solution and the total permissible objective function evaluations in any run, and 3) Reliability: the ratio of the number of runs reached the global solution to the total number of runs. Recently, Djebedjian et al. [16] suggested two criteria to investigate the performance of different metaheuristic algorithms in case of using one or multiple runs. Then, they apply their criteria to compare between different algorithms using all the available WDN's literature data. These criteria are used here and can be represented as in Eqs. (21) and (25):

$$\eta_{\text{best-alg}} = \eta_{\text{cost}} \cdot \left[0.005\left(1+\eta_{\text{eval}}\right) + 0.99\ \eta_{\text{gen}} \right] \tag{21}$$

where,

$$\eta_{\text{cost}} = \frac{C_{\text{opt}}}{C_{\min}} \cdot \frac{C_{\max} - C_{\min}}{C_{\max} - C_{\text{opt}}} \tag{22}$$

$$\eta_{\text{eval}} = 1 - \frac{\log_{10} N_{\text{eval}}}{\log_{10} N_{\text{space}}} \tag{23}$$

$$\eta_{\text{gen}} = 1 - \frac{\log_{10} N_{\text{OFE}}}{\log_{10} N_{\text{space}}}, \qquad N_{\text{space}} = ND^{NP} \tag{24}$$

$$\eta_{\text{ave-alg}} = \eta_{\text{cost}} \cdot \left[0.005\left(\eta_{\text{ave-cost}} + \eta_{\text{ave-eval}}\right) + 0.99\ \eta_{\text{gen-success}} \right] \tag{25}$$

where,

$$\eta_{\text{ave-cost}} = \frac{C_{\text{opt}}}{C_{\text{ave}}} \cdot \frac{C_{\max} - C_{\text{ave}}}{C_{\max} - C_{\text{opt}}} \tag{26}$$

$$\eta_{\text{ave-eval}} = 1 - \frac{\log_{10} N_{\text{ave-eval}}}{\log_{10} N_{\text{space}}} \tag{27}$$

$$\eta_{\text{gen-success}} = 1 - \frac{\log_{10}\left[N_{\text{OFE}} \cdot \left(N_{\text{sim}} / N_{\text{success}} \right) \right]}{\log_{10} N_{\text{space}}} \tag{28}$$

where, $\eta_{\text{best-alg}}$, $\eta_{\text{ave-alg}}$ are the performance of the algorithm using the best run results and average of multiple runs results, respectively; η_{cost} is the effectiveness for the best run and equal to one if the final minimum cost, $C_{\min}$, reached is equal to the global optimum cost ever known, C_{opt}; η_{eval} is efficiency of the best run; N_{eval} is minimum number of function evaluations to reach, $C_{\min}$, in the best run; η_{gen} is efficiency which measured by the ratio of maximum/actual permissible objective function evaluations, N_{OFE}, processed to terminate the run to the available total number of solutions, N_{space}; ND is number of available commercial discrete pipe diameters; $\eta_{\text{ave-cost}}$ is average effectiveness for multiple runs; $\eta_{\text{ave-eval}}$ is average efficiency of multiple runs; C_{ave} is average final optimum costs in multiple runs; $N_{\text{ave-eval}}$ is average number of function evaluations to the first reach to the final optimum in multiple runs; and N_{sim} and N_{success} are number of total runs and number of runs succeeded in reaching the global optimum, respectively.

III. CASE STUDIES

Three benchmarks WDN's in addition to one national WDN are solved in the present work to investigate the performance of the proposed FSAJA. The three benchmark WDN's, used by Djebedjian et al., [16], are adopted here for the sake of comparing the performance of FSAJA to performances of other algorithms presented in [16]. The studied WDN's are: 1) the two-loop WDN, 2) the New York WDN, and 3) the Hanoi WDN. The fourth WDN is a national one named by El-Mostakbal city WDN.

For parsimony, only main data and some selected references include the full hydraulic data of the three benchmark WDN's are presented in Table (1) and Fig. 4. The full data for the El-Mostakbal WDN is presented for the sake of completeness.

Figure 5 shows the layout of the El-Mostakbal WDN, and Tables (2)-(4) present the available commercial pipes, cost per unit meter in Egyptian pounds (LE), pipe lengths, node levels and different demands (Q_{out}). The minimum required pressure head at different nodes of the El-Mostakbal WDN is considered equal to 22 m, and the hydraulic gradient level at the supplying tank is equal to 58.89 m. Each of the studied networks is supplied with one tank.

TABLE (1)
MAIN DATA OF THE STUDIED WDN'S

WDN	NP	NJ	NL	ND	C_{HW}	Ref.
Two-Loop	8	7	2	14	130	[16], [38]
New York	21	20	2	16	100	[16], [39]
Hanoi	34	32	3	6	130	[16], [40], [41]
El-Mostakbal	44	33	12	10	100	[42]–[44]

TABLE (2)
AVAILABLE DIAMETERS AND COST/M
(EL-MOSTAKBAL WDN)

Diameter (m)	Cost (LE)	Diameter (m)	Cost (LE)	Diameter (m)	Cost (LE)
0.15	188	0.40	570	0.8	1485
0.20	255	0.50	735	1.0	2505
0.25	333	0.60	1110	1.2	3220
0.30	419				

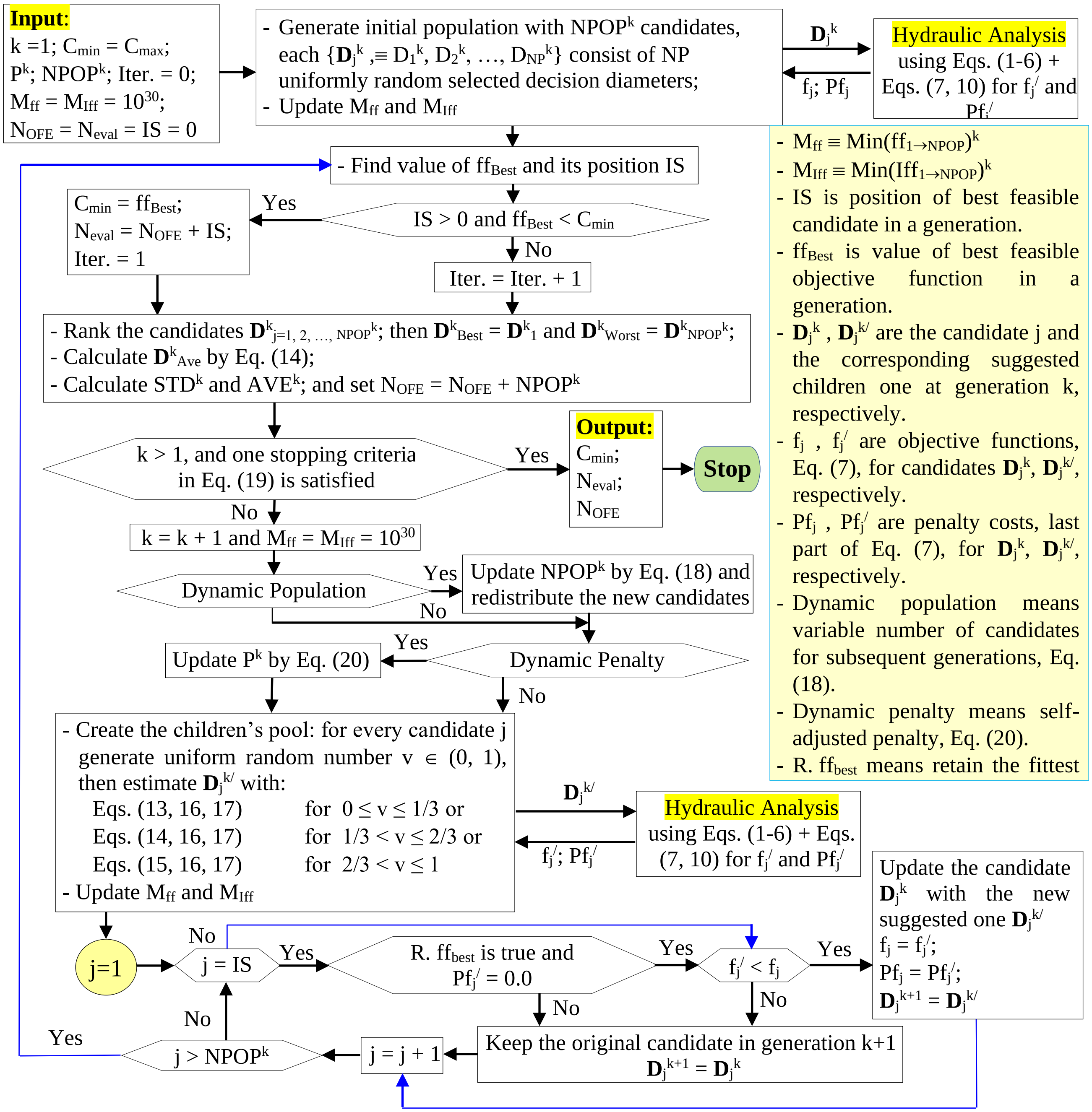

Fig. 3 Flowchart illustrates one run for different alternative forms of the proposed FSAJA.

IV. RESULTS AND DISCUSSION

A. Selection for Best Performed FSAJA Variant

This work is intended to design a completely free sensitivity analysis variant of JA. Thus, some alternatives within the algorithm process must be discovered to find the best recommended one. These alternatives are: 1) the population size is constant or dynamic Eq. (18), 2) the penalty function is constant with a very big value, Eq. (11), or self-adjusted dynamic penalty during the generation process, Eq. (20), 3) the fittest feasible solution can be replaced with a fittest infeasible solution or not (i.e., lose the fittest feasible solution or retain the fittest feasible solution), and 4) the initial population size. To investigate the effect of these variants, six alternative algorithm variants are adopted to select the best one.

For every combination of the alternatives, the constructed algorithm variant is processed for a group of 100 runs (always with an initial seed number equal to the run number). Each group is reprocessed with 13 different population sizes to cover a wide range as (1, 2, 3, 4, 5, 10, 15, 20, 25, 40, 50, 75, 100) * *NP*. Results for every group of 100 runs are analyzed to determine: 1) the best ever optimum cost reached within all the runs and the number of its recurrence within the 100 runs, 2) the average and standard deviation for total executed number of function evaluations and minimum costs reached for the 100 runs, and 3) the average number of function evaluations, to the first reach to the minimum cost in any run, C_{ave}.

The optimum cost and diameters of the three benchmark WDN's are early settled and known in advance from the literature. These optimum costs are 419,000, 3, 8637,600, and 6,081,087 monetary units for the two-loop, New York, and Hanoi WDN's, respectively.

On the other hand, the optimum cost for the El-Mostakbal WDN is still under investigation and its magnitude is improving with the recently published researches, [43], [44]. Despite all suggested variants of FSAJA reach several better minima than the last recently achieved in the literature, the solution process of FSAJA is usually terminated at local minimum, see local minimum 1 in Tables (3) and (4) in columns 5 and 6, respectively. Only a few runs reach to a better minimum, i.e., local minimum 2 and global minimum as presented in Tables (3) and (4), columns 6-7 and 7-8, respectively. Relative discrepancies in the cost between local minimum 1, and the global minimum is less than 0.05%. Therefore, the FSAJA run is assumed successful if it reaches the local minimum 1 or any better minimum (i.e., local minimum 2 or the global minimum).

Figures 6 and 7 show the calculated best and average global performances, respectively, for the six selected algorithm variants shown in the graph's legends. Any data point in Fig. 7 represents the average global performance of 100 runs for a specified variant of FSAJA at certain initial population size, while it represents only the best global performance (for the best run) in Fig. 6. The best run is the one that reached the minimum cost, within all the 100 runs, using minimum number of function evaluations, in case of its recurrence.

From Fig. 6, it can be observed that the best global performance is generally decreasing with increasing the population size, with better performances when using a dynamic population size. Despite some literature, [10], used one run to compare between performances of different algorithms, it seems to be unfair for the following points: 1) due to ignoring the computational effort executed for the other worst performed different runs to reach this best run from the comparison process, and 2) any algorithm can reach the global minimum more likely as the population size increases, which increases the algorithm reliability, this behavior cannot be measured using the best run results only.

All the data points in Fig. 7 that represented the two-loop WDN and the New York WDN, reached the global minimum cost of at least one run or more.

However, some data points that represented the initial population size (1, 2, 3) **NP* for Hanoi and El-Mostakbal WDN's fail to reach the global minimum cost for the whole 100 runs. In general, the average global performance of the studied six variants decreases as the population size increases; consequently, the best performed range of the population size can be considered the smallest size that ensures reaching the global cost, i.e., 4**NP*. Irregular responses for the six algorithm variants are noticed for the four WDN's; with often better performance in case of using the algorithm variant that is based on: 1) constant population size, 2) dynamic self-adjusted penalty, and 3) retaining the fittest feasible candidate between generations. Thus, that variant is recommended to be the final form for the FSAJA.

The global reached cost of the El-Mostakbal WDN, see Tables (3) and (4), has a lower relative percentage than the lowest one presented in the literature, [44], by 0.12% with magnitude equal to 4,926,560.7 monetary units.

B. Comparison between FSAJA and Other Algorithms for the Three Benchmark WDN's

Table (5) shows a comparison between the performances of JA and FSAJA with 100 runs for the three benchmark WDN's. The performance of JA is weakened rapidly as N_{space} increases. For all the comparison points, seen in columns of Table (5), the performance of the FSAJA is always superior concerning the corresponding one of JA. In the case of the Hanoi WDN, JA cannot reach global cost even after $4*10^7$ objective function evaluations.

Performance of the FSAJA is compared to performances of different metaheuristic algorithms previously applied to the three benchmark WDN's as shown in Tables (6)-(8).

TABLE (3)
PIPE LENGTHS AND OPTIMAL DIAMETERS FOR DIFFERENT RESEARCH (EL-MOSTAKBAL WDN)

Pipe no.	Pipe length (m)	Optimal Diameters					Pipe no.	Pipe length (m)	Optimal Diameters				
		[43] *	[44] **	FSAJA					[43] *	[44] **	FSAJA		
				Local Min. 1	Local Min. 2	Global Min.					Local Min. 1	Local Min. 2	Global Min.
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
1	2463	0.60	0.60	0.60	0.60	0.60	24	134.2	0.20	0.15	0.15	0.15	0.15
2	100	0.50	0.50	0.50	0.50	0.50	25	163.4	0.20	0.15	0.15	0.15	0.15
3	328	0.15	0.50	0.50	0.50	0.50	26	309.15	0.15	0.40	0.40	0.40	0.40
4	80	0.15	0.50	0.50	0.50	0.50	27	92.7	0.20	0.40	0.40	0.40	0.40
5	149.3	0.15	0.15	0.15	0.15	0.15	28	84.9	0.15	0.20	0.15	0.20	0.20
6	67	0.15	0.15	0.15	0.15	0.15	29	226.3	0.15	0.15	0.15	0.15	0.15
7	184.3	0.15	0.15	0.15	0.15	0.15	30	100	0.40	0.15	0.15	0.15	0.15
8	288	0.50	0.15	0.15	0.15	0.15	31	217	0.30	0.15	0.15	0.15	0.15
9	100	0.50	0.15	0.15	0.15	0.15	32	101	0.15	0.15	0.15	0.15	0.15
10	341.65	0.15	0.15	0.15	0.15	0.15	33	156.5	0.15	0.15	0.15	0.15	0.20
11	152.5	0.15	0.50	0.50	0.50	0.50	34	185	0.15	0.15	0.15	0.15	0.15
12	70.7	0.15	0.50	0.50	0.50	0.50	35	145.8	0.15	0.25	0.30	0.30	0.25
13	172	0.15	0.15	0.15	0.15	0.15	36	230.5	0.20	0.40	0.40	0.30	0.30
14	109	0.15	0.15	0.15	0.15	0.15	37	262.6	0.30	0.15	0.15	0.15	0.15
15	104.7	0.15	0.15	0.15	0.15	0.15	38	109.9	0.15	0.25	0.30	0.30	0.30
16	155	0.50	0.15	0.15	0.15	0.15	39	114.9	0.15	0.20	0.20	0.25	0.25
17	123.5	0.50	0.15	0.15	0.15	0.15	40	181.9	0.25	0.20	0.15	0.20	0.20
18	98.4	0.15	0.15	0.15	0.15	0.15	41	120	0.15	0.15	0.15	0.15	0.15
19	164.6	0.15	0.15	0.15	0.15	0.15	42	257.4	0.15	0.15	0.20	0.15	0.15
20	127.6	0.15	0.50	0.50	0.50	0.50	43	184	0.15	0.15	0.15	0.15	0.15
21	225.5	0.50	0.15	0.15	0.15	0.15	44	370.6	0.15	0.20	0.15	0.20	0.20
22	198	0.40	0.15	0.15	0.15	0.15	Optimal Cost		4,968,881.5	4,932,467.1	4,928,997.3	4,928,614.0	**4,926,560.7**
23	357.9	0.15	0.15	0.15	0.15	0.15							

* Particle Swarm Optimization results presented by El-Ghandour and Elbeltagi [43] (personal communication)
** Whale Optimization Algorithm results presented by Ezzeldin and Djebedjian [44] (revised results - personal communication)

TABLE (4)
NODE LEVELS AND EXCESS PRESSURE HEADS AT OPTIMAL SOLUTIONS (EL-MOSTAKBAL WDN)

Node no.	Level (m)	Q_{out} (L/s)	Excess pressure head at different nodes (m)					Node no.	Level (m)	Q_{out} (L/s)	Excess pressure head at different nodes (m)				
			[43] *	[44] **	FSAJA						[43] *	[44] **	FSAJA		
					Local Min. 1	Local Min. 2	Global Min.						Local Min. 1	Local Min. 2	Global Min.
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
1	15	-352.49	21.89	21.89	21.89	21.89	21.89	18	15	24	2.77	4.94	4.95	4.94	4.94
2	15	0	12.83	12.83	12.83	12.83	12.83	19	15	19.2	4.08	1.89	1.79	1.83	1.82
3	14	24	12.93	12.93	12.93	12.93	12.93	20	15	34.09	5.82	0.70	0.52	0.59	0.56
4	14	0	9.01	10.66	10.66	10.66	10.66	21	15	0	7.46	1.99	1.87	1.91	1.90
5	14	19.2	8.06	10.11	10.11	10.11	10.11	22	15.5	20.8	4.99	0.41	0.19	0.26	0.24
6	14	0	9.11	9.26	9.25	9.25	9.25	23	15.5	0	2.90	3.31	2.44	3.18	2.85
7	14	0	9.59	8.87	8.87	8.87	8.87	24	15	16	2.53	4.50	4.52	4.51	4.52
8	14	17.6	10.89	7.82	7.81	7.81	7.81	25	15.5	16	1.42	3.50	3.44	2.11	2.37
9	14	20.8	10.30	6.85	6.83	6.83	6.83	26	15.5	0	1.91	2.49	2.68	1.69	1.62
10	14	19.2	7.24	9.24	9.24	9.24	9.24	27	15.5	0	2.91	2.34	2.15	1.73	2.09
11	14	0	7.09	8.92	8.92	8.92	8.92	28	15.5	0	3.53	1.27	1.30	1.02	1.15
12	14	0	8.21	7.81	7.79	7.80	7.80	29	15.5	24	2.24	0.72	1.15	0.80	0.79
13	14	0	8.92	7.10	7.08	7.09	7.09	30	15.5	0	1.87	1.85	2.22	1.35	1.30
14	14	0	9.60	6.43	6.40	6.41	6.40	31	15.5	19.2	0.51	0.31	0.37	0.02	0.06
15	14	19.2	9.09	4.46	4.40	4.42	4.41	32	15.5	19.2	1.60	0.25	0.08	0.04	0.07
16	14	0	6.72	5.00	4.95	4.97	4.96	33	15.5	16	0.11	1.15	0.34	0.36	0.50
17	14	24	5.22	8.40	8.40	8.40	8.40	**Optimal Cost**			**4,968,881.5**	**4,932,467.1**	**4,928,997.3**	**4,928,614.0**	**4,926,560.7**

* Particle Swarm Optimization results presented by El-Ghandour and Elbeltagi [43] (personal communication)
** Whale Optimization Algorithm results presented by Ezzeldin and Djebedjian [44] (revised results - personal communication)

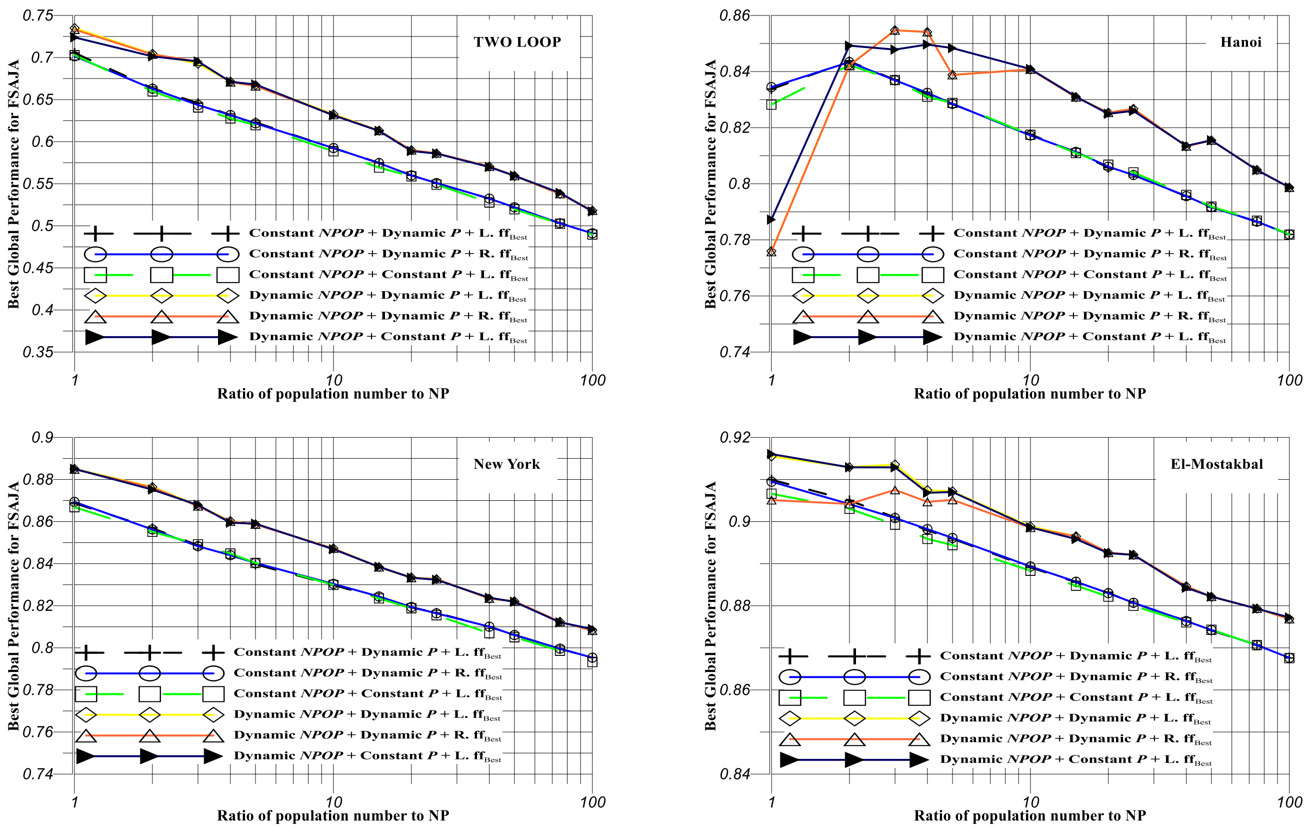

Fig. 6 Best Global FSAJA performance vs. the ratio of population number to NP
[L. ff_{Best} = FSAJA can lose fittest feasible solution between generations, R. ff_{Best} = FSAJA retain the fittest feasible solution]

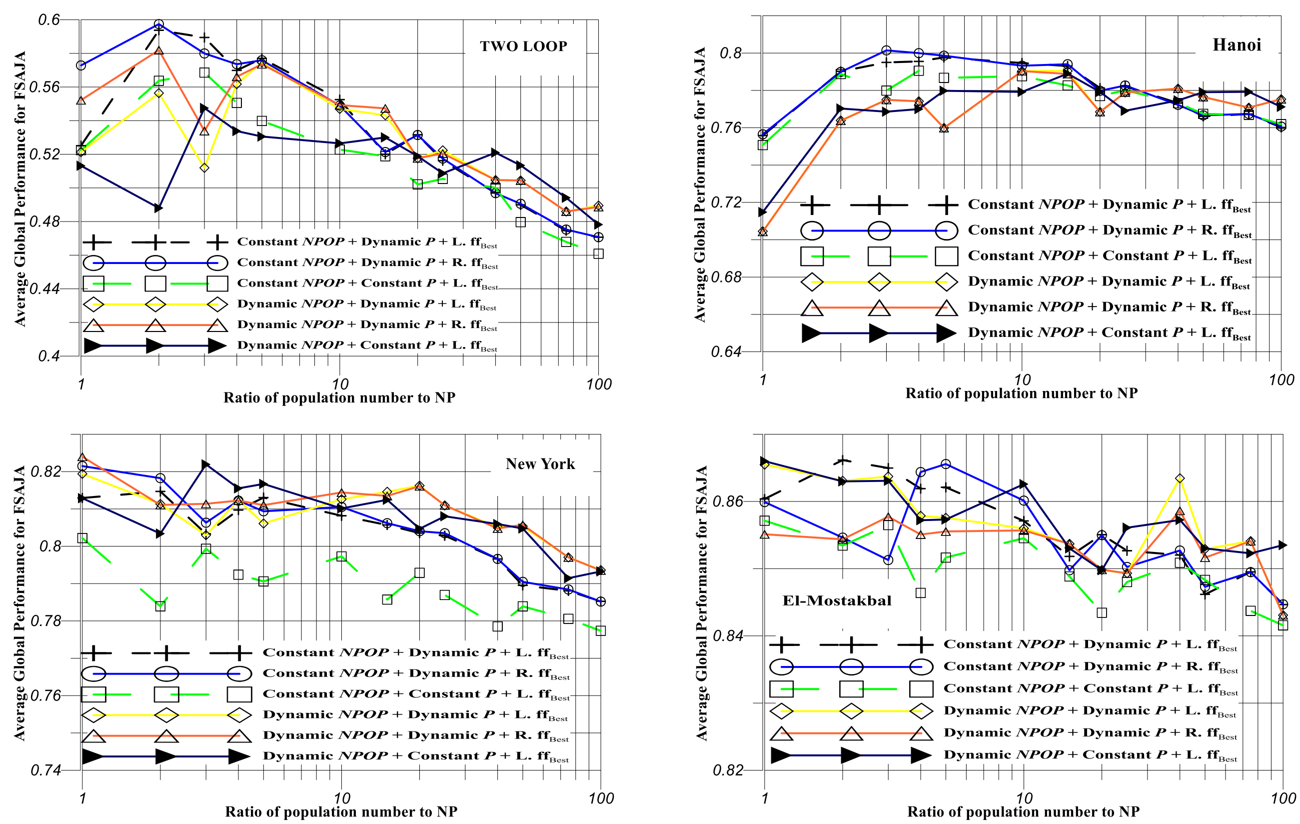

Fig. 7 Average Global FSAJA performance vs. the ratio of population number to NP
[L. ff_{Best} = FSAJA can lose fittest feasible solution between generations, R. ff_{Best} = FSAJA retain the fittest feasible solution]

TABLE (5)
COMPARISON BETWEEN PERFORMANCES OF JA AND FSAJA FOR THE THREE BENCHMARK WDN'S

WDN	Algorithm	C_{min}	*NPOP*	*Ave* N_{OFE}	$N_{success}/N_{sim}$	N_{eval}	$N_{ave\text{-}eval}$	C_{ave}/C_{opt}	C_{std}/C_{opt}
Two-Loop	JA	419,000	50	2,500	4%	1932	1894	1.07	0.128
	FSAJA	419,000	4*8	2,514	27%	578	1188	1.02	0.029
New York	JA	38,637,600	300	120,000	1%	47,604	38,319	1.19	0.117
	FSAJA	38,637,600	4*21	9,229	20%	4,505	6,650	1.03	0.036
Hanoi	JA	6,257,115	400	400,000	0%	283,201	135,384	1.30	1.534
	FSAJA	6,081,087	4*34	28,646	18%	20,343	24,457	1.04	0.032

Ave N_{OFE} is the average number of the total objective function evaluations within the 100 runs.
C_{std} is the standard deviation of minimum reached costs in 100 runs.

The algorithm's abbreviations are explained after their first appearance at the foot of any table. Two criteria are used in the comparison, i.e.: 1) total executed number of objective function evaluations to reach a global minimum ($N_{total\text{-}opt} = N_{OFE} * N_{sim}/N_{success}$) which represents both the actual efficiency (computational effort) and reliability/robustness of any algorithm, and 2) Inverse of the probability of reaching the global minimum in one run E [45], i.e., ($1/E = N_{ave\text{-}eval} * N_{sim}/N_{success}$) which represents both of the algorithm reliability and convergence rate to the optimum.

Table (6) shows the different results for the two-loop WDN. Total and expected minimum numbers of objective function evaluations for FSAJA to reach the global cost are equal to 7,979 and 4,402, respectively. With exception of four algorithms (i.e., FDE with N_{OFE} = 1000, 5000; PSO; and PSO-DE), the FSAJA is equivalent or performs better than the other algorithms. All the presented algorithms in the table need an exhausted sensitivity computational effort, for their control parameters, except the FDE.

Table (7) presents the compared results for the New York WDN. FSAJA actual and expected minimum numbers of objective function evaluations to reach the global cost are equal to 46,145 and 33,249, respectively. The FDE with N_{OFE} = 1000, 5000, 10,000 performed better than FSAJA; also both ACO and BLD-DE prove a better performance. The remaining algorithms have a worst or equivalent performance as FSAJA.

Hanoi WDN needs high number of objective function evaluations than the previous two WDN's to reach a global cost, this is due to its feasible domain that includes large number of the local minimum [45]. Consequently, that WDN can be considered as more complex than the previous two WDN's. From thirty-three comparisons between different algorithms and FSAJA, see Table (8), only 13 ones have better performances concerning FSAJA, the remaining have equivalent or worst behaviors.

For the three benchmarks of the WDN's, the best performed is the FDE algorithm which is a free control parameter. However, that algorithm is based on adjusted common parameters when handled the three benchmarks WDN's, e.g., *NPOP* is considered equal to 100 and an adjusted form for the penalty function is used, which depends on the summation of shortage in the pressure heads at the nodes multiplied by $C_{max}/(0.1*H_{min})$. Considering the required computational effort for adjusting different control parameters for the remaining algorithms, the FSAJA may be superior to the whole algorithms. Future papers are encouraged to focus on presenting the actual total objective function evaluations used to tune up both control and common parameters of any algorithm in the research data to ensure a fair comparison between different algorithms.

TABLE (6)
COMPARISON BETWEEN DIFFERENT ALGORITHMS AND FSAJA (TWO-LOOP WDN)

Algorithm	$N_{total\text{-}opt}$	1/E	Algorithm	$N_{total\text{-}opt}$	1/E
ACO [43]	178,571	>18,929	GA1 [46]	500,000	56,250
B-GA [47]	66,667	--	GA2 [46]	666,667	110,000
DE [48]	25,000	11,875	GA3 [46]	2,000,000	415,000
DE [49]	25,000	14,968	GENOME [50]	300,000	--
DE [46]	40,000	13,500	GHEST [51]	83,333	12,083
EGA1 [46]	181,818	22,273	MA [52]	15,960	14,111
EGA2 [46]	400,000	65,000	MA [43]	62,500	> 35,631
EGA3 [46]	1,000,000	600,000	PSHS [53]	38,462	**1,792**
FDE1 [20]	**2,222**	**1,333**	PSO [9]	**7,650**	5,138
FDE2 [20]	**5,814**	**1,298**	PSO [43]	10,417	> 6,875
FDE3 [20]	11,364	**1,477**	PSO-DE [54]	**7,000**	**3,080**
FDE4 [20]	101,010	**4,214**	SFLA [43]	63,636	> 31,459
GA [49]	333,333	191,300	SLC [55]	12,500	5,128
GA [43]	500,000	>151,500	SS [11]	11,490	**3,215**
JA	62,500	47,325	FSAJA	7,979	4,402

- Bold values represent better results than that obtained by FSAJA
- ACO = Ant Colony Optimization; B-GA = Bounded Genetic Algorithm; DE = Differential Evolution; EGA1 = elitist GA with truncation; EGA2 = elitist GA with tournament; EGA3 = elitist GA with roulette-wheel method of selection; FDE = Fittest individual referenced Differential Evolution; FDE1 = FDE with N_{OFE} equal to 1000; FDE2 = FDE with N_{OFE} equal to 5000; FDE3 = FDE with N_{OFE} equal to 10,000; FDE4 = FDE with N_{OFE} equal to 100,000; GA = Genetic Algorithm; GA1 = GA with truncation; GA2 = GA with tournament; GA3 = GA with roulette-wheel; GENOME = Genetic Algorithm Pipe Network Optimization Model; GHEST = Genetic Heritage Evolution by Stochastic Transmission; MA = Memetic Algorithm; PSHS = Combined Particle-Swarm Harmony Search; PSO = Particle Swarm Optimization; PSO-DE = Combined Particle Swarm Optimization and Differential Evolution; SFLA = Shuffled Frog Leaping Algorithm; SS = Scatter Search; and SLC = Soccer League Competition.

TABLE (7)
COMPARISON BETWEEN DIFFERENT ALGORITHMS AND FSAJA
(NEW YORK WDN)

Algorithm	$N_{total-opt}$	1/E	Algorithm	$N_{total-opt}$	1/E
ACO [43]	937,500	>349,688	GA [43]	375,000	>288,375
ACO^{1} [56]	**33,333**	**24,083**	GA^{1} [36]	66,667	**28,100**
ACO^{2} [56]	**33,333**	**26,950**	GA^{2} [36]	57,143	**19,171**
BLP-DE [57]	**7,500**	**3,486**	GA [58]	789,889	--
CGA [59]	400,000	88,648	GA_{mod} [58]	555,556	--
cGA [60]	**18,007**	--	GHEST [61]	50,000	**7,795**
dDE [59]	215,054	**14,209**	HD-DDS [62]	58,140	53,488
DE [58]	136,426	--	MA [43]	50,000	>144,940
DE [48]	14,144	**7,771**	MMAS [8]	83,333	51,185
DE [49]	50,505	**18,456**	NLP-DE1 [63]	20,202	**8,361**
DE [58]	555,556	--	NLP-DE2 [63]	20,202	**10,738**
DE1 [64]	348,837	47,356	PSO [58]	2,145,923	--
DE2 [64]	2,500,000	**31,050**	PSO [58]	300,300	--
DE3 [64]	535,714	43,682	PSO [43]	357,143	>21,786
DE4 [64]	348,837	51,340	PSO variant [62]	266,667	--
DE5 [64]	312,500	192,673	SADE [12]	10,029	7,172
FDE1 [20]	**2,128**	**1,004**	SDE [59]	206,186	**13,253**
FDE2 [20]	**5,952**	**1,731**	SFLA [43]	857,143	>56,879
FDE3 [20]	**11,236**	**1,893**	SFLA [65]	--	**8,872**
FDE4 [20]	100,000	**4,193**	SGA [59]	444,444	121,753
GA [58]	214,133	--	SLC1 [55]	125,000	**9,776**
GA [66]	220,264	108,921	SLC2 [55]	100,000	15,764
GA [49]	156,250	42,100	SS [11]	92,985	88,589
JA	$12*10^6$	3,831,800	FSAJA	46,145	33,249

- Bold values represent better results than that obtained by FSAJA.
- ACO^{1} = ACO with internal self-adaptive penalty method; ACO^{2} = ACO with external self-adaptive penalty method; BLP-DE = Combined Binary Linear Programming and Differential Evolution; CGA = convergent Genetic Algorithm; cGA = Creeping mutation Genetic Algorithm; dDE = Dither Differential Evolution; DE1 = DE random mutation; DE2 = DE best mutation 1; DE3 = DE best mutation 2; DE4 = DE CurrentToBest2; DE5 = DE rand2; GA^{1} = GA with internal self-adaptive penalty method; GA^{2} = GA with external self-adaptive penalty method; HD-DDS = Hybrid Discrete Dynamically Dimensioned Search; MMAS = Max-Min Ant System; NLP-DE1 = DE seeded with two tailored pipe diameters; NLP-DE2 = DE seeded with four tailored pipe diameters; SADE = Self-Adaptive Differential Evolution; SDE = Standard Differential Evolution; SGA = Standard Genetic Algorithm; SLC1 = SLC without relegation and promotion; and SLC2 = SLC with relegation and promotion.

It is worth mentioning that the considered population size with 4**NP* candidates is not the optimal one for all the studied four distribution networks, but it is a compromise size that gives good results for different WDN problems under different levels of complexity. Optimal population size for both the two-loop and the New York WDN's is 2**NP*, with ($N_{total-opt}$, $1/E$) equal to (2,878; 4,798) and (23,004; 32,220), respectively. While the optimal population size for the Hanoi WDN is 3**NP* with $N_{total-opt}$ and $1/E$ equal to 124,044 and 144,942, respectively. The most complicated problem is El-Mostakbal WDN with optimal population size 4**NP* and with $N_{total-opt}$ and $1/E$ equal to 494,288, and 602,413, respectively.

TABLE (8)
COMPARISON BETWEEN DIFFERENT ALGORITHMS AND FSAJA
(HANOI WDN)

Algorithm	$N_{total-opt}$	1/E	Algorithm	$N_{total-opt}$	1/E
BLP-DE [57]	**40,816**	**33,824**	GA_{trad} [58]	15,151,515	--
DE [58]	517,063	--	GHEST [51]	125,000	**91,838**
DE [67]	163,043	**72,283**	GSA [15]	**9,375**	**8,635**
DE [49]	**102,041**	**32,263**	HD-DDS [62]	1,250,000	1,250,000
DE^{a} [68]	**48,309**	**30,164**	HS [69]	4,166,667	--
DE^{b} [68]	**121,951**	**59,420**	ISEDPSO [70]	401,929	**18,864**
DE1 [64]	348,837	**86,726**	NLP-DE1^{d} [63]	**82,474**	**35,679**
DE2 [64]	7,500,000	166,500	NLP-DE2^{e} [63]	**81,633**	**43,655**
DE3 [64]	357,143	233,181	PEDPSO [70]	277,778	**26,000**
DE4 [64]	357,143	225,514	PSHS [53]	4,166,667	--
dDE [59]	625,000	**79,625**	PSO [58]	5,000,000	--
FDE1 [20]	**1,587**	**838**	SADE [12]	**89,138**	**72,062**
FDE2 [20]	**5,556**	**913**	SDE [59]	543,478	**83,935**
FDE3 [20]	**10,526**	**1,206**	SLC1 [55]	**125,000**	**36,385**
FDE4 [20]	**103,093**	**2,129**	SLC2 [55]	**100,000**	**71,789**
GA_{mod} [58]	7,462,687	--	SS [11]	**94,438**	**67,420**
JA	$>4*10^7$	--	FSAJA	159,143	135,871

- Bold values for only better results than that obtained by FSAJA.
- DE^{a} = Population size equal to 20; DE^{b} = Population size equal to 100; GSA = Gravitational Search Algorithm; HS = Harmony Search; ISEDPSO = Improved Sequential combination of PSO and Estimation of Distribution Algorithm (EDA); PEDPSO = Parallel hybridization of PSO and Estimation of Distribution Algorithm (EDA).

V. CONCLUSIONS AND RECOMMENDATIONS

A novel variant of the Jaya algorithm, FSAJA, is proposed to be completely independent of any control algorithm or common parameter. The algorithm consists of the following structure: 1) three learning phases, Eqs. (13)-(15), with equivalent probability of considering any of them to any candidate, the corresponding generated candidate using any selected learning phase replaces the original candidate only if it has a fittest objective function, Eq. (7); 2) the image strategy is adopted to replace the generated violated decision variable with an interior one within its permissible limits, Eqs. (16) or (17); 3) population size is fixed and equal to four times number of the decision variables; 4) the penalty is assumed initially very big value and it is self-adjusted with the generation process, Eq. (20), to a lower magnitude that gives infeasible candidates around the solution domain, the opportunity of sharing their information's in the next generations; 5) fittest feasible candidate in a generation cannot be replaced, with a generated more fittest infeasible one, and 6) satisfying any of the following two stopping criteria to terminate the run: a) if the ratio between the standard deviation of candidates objective functions and their average, goes to be less than 0.0001 at any generation, or b) the processed number of generations without improvement in the fittest candidate is more than 30, Eq. (19).

Using three benchmarks of WDN's, the FSAJA performance is found to be superior concerning the standard JA. Also, the comparison between FSAJA and different evolutionary algorithms, published in the literature, proves that

FSAJA has an equivalent or better performance for most of the comparisons. If the computational efforts executed to tune up the different control parameters of the compared algorithms are considered, FSAJA may prove more superior performance. The FDE algorithm still has the precedence performance with respect to all evolutionary algorithms.

El-Mostakbal WDN is presented here in full detail; it has a complex solution domain with great difficulty in reaching its global minimum. However, FSAJA reaches several local minima's lower than the recently reported in the literature with a new global cost equal to 4,926,560.7 Egyptian pounds. But most of the best runs in the different groups that consist of 100 runs are terminated at a higher local minimum with a cost equal to 4,928,997.3 Egyptian pounds which is still superior to any minimum reached in the literature. El-Mostakbal WDN is a challenging one and it is recommended to be used for investigating the effectiveness of different algorithms.

A more research effort is required to investigate a relationship between the pipe network characteristics and optimal population size.

REFERENCES

[1] G. C. Dandy and M. O. Engelhardt, "Multi-objective trade-offs between cost and reliability in the replacement of water mains," J. Water Resour. Plan. Manag, vol. 132, no. 2, pp. 79–88, Mar. 2006, doi: 10.1061/ (ASCE) 0733-9496(2006)132:2(79).

[2] K. E. Lansey and L. W. Mays, "Optimization model for water distribution system design," J. Hydraul. Eng., vol. 115, no. 10, pp. 1401–1418, Oct. 1989, doi: 10.1061/(ASCE)0733-9429(1989)115:10(1401).

[3] N. Duan, L. W. Mays, and K. E. Lansey, "Optimal reliability-based design of pumping and distribution systems," J. Hydraul. Eng., vol. 116, no. 2, pp. 249–268, Feb. 1990, doi: 10.1061/(ASCE)0733-9429(1990)116:2(249).

[4] C. M. da Conceição and S. Joaquim, "Water distribution network design optimization: simulated annealing approach," J. Water Resour. Plan. Manag., vol. 125, no. 4, pp. 215–221, Jul. 1999, doi: 10.1061/(ASCE)0733-9496(1999)125:4(215).

[5] Z. W. Geem, "Optimal cost design of water distribution networks using harmony search," Eng. Optim., vol. 38, no. 3, pp. 259–277, Apr. 2006, doi: 10.1080/03052150500467430.

[6] M. M. Eusuff, and K. E. Lansey, "Optimization of water distribution network design using the shuffled frog leaping algorithm," J. Water Resour. Plan. Manag., vol. 129, no. 3, pp. 210–225, May 2003, doi: 10.1061/(ASCE)0733-9496(2003)129:3(210).

[7] H. R. Maier, A. R. Simpson, A. C. Zecchin, W. K. Foong, K. Y. Phang, H. Y. Seah, and C. L. Tan, "Ant colony optimization for design of water distribution systems," J. Water Resour. Plan. Manag., vol. 129, no. 3, pp. 200–209, May 2003, doi: 10.1061/(ASCE)0733-9496(2003)129:3(200).

[8] [8] A. C. Zecchin, A. R. Simpson, H. R. Maier, M. Leonard, A. J. Roberts, and M. J. Berrisford, "Application of two ant colony optimisation algorithms to water distribution system optimisation," Math. Comput. Model, vol. 44, no. 5, pp. 451–468, 2006, doi: 10.1016/j.mcm.2006.01.005.

[9] C. R. Suribabu and T. R. Neelakantan, "Design of water distribution networks using particle swarm optimization," Urban Water J., vol. 3, no. 2, pp. 111–120, Jun. 2006, doi: 10.1080/15730620600855928.

[10] L. Perelman and A. Ostfeld, "An adaptive heuristic cross-entropy algorithm for optimal design of water distribution systems," Eng. Optim., vol. 39, no. 4, pp. 413–428, Jun. 2007, doi: 10.1080/03052150601154671.

[11] M.-D. Lin, Y.-H. Liu, G.-F. Liu, and C.-W. Chu, "Scatter search heuristic for least-cost design of water distribution networks," Eng. Optim., vol. 39, no. 7, pp. 857–876, Oct. 2007, doi: 10.1080/03052150701503611.

[12] F. Zheng, A. C. Zecchin, and A. R. Simpson, "Self-adaptive differential evolution algorithm applied to water distribution system optimization," J. Comput. Civ. Eng., vol. 27, no. 2, pp. 148–158, Mar. 2013, doi: 10.1061/ (ASCE) CP.1943-5487.0000208.

[13] W. Bi, G. C. Dandy, and H. R. Maier, "Improved genetic algorithm optimization of water distribution system design by incorporating domain knowledge," Environ. Model. Softw, vol. 69, pp. 370–381, 2015, doi: 10.1016/j.envsoft.2014.09.010.

[14] H. M. Lee, D. Jung, A. Sadollah, E. H. Lee, and J. H. Kim, "Performance comparison of metaheuristic optimization algorithms using water distribution system design benchmarks," In: Harmony Search and Nature Inspired Optimization Algorithms. Springer, Singapore, pp. 97–104, 2019.

[15] H. Fallah, S. Ghazanfari, C. R. Suribabu, and E. Rashedi, "Optimal pipe dimensioning in water distribution networks using Gravitational Search Algorithm," ISH J. Hydraul. Eng., pp. 1–14, Jun. 2019, doi: 10.1080/09715010.2019.1624630.

[16] B. Djebedjian, H. A. A. Abdel-Gawad, and R. M. Ezzeldin, "Global performance of metaheuristic optimization tools for water distribution networks," Ain Shams Eng. J., vol. 12, no. 1, pp. 223–239, 2021, doi: 10.1016/j.asej.2020.07.012.

[17] D. Mora-Melia, P. L. Iglesias-Rey, F. J. Martinez-Solano, and P. Ballesteros-Pérez, "Efficiency of evolutionary algorithms in water network pipe sizing," Water Resour. Manag., vol. 29, no. 13, pp. 4817–4831, 2015, doi: 10.1007/s11269-015-1092-x.

[18] D. Mora-Melia, P. L. Iglesias-Rey, F. J. Martínez-Solano, and P. Muñoz-Velasco, "The efficiency of setting parameters in a modified shuffled frog leaping algorithm applied to optimizing water distribution networks," Water , vol. 8, no. 5, 2016, doi: 10.3390/w8050182.

[19] F. Zheng, "Comparing the real-time searching behavior of four differential-evolution variants applied to water-distribution-network design optimization," J. Water Resour. Plan. Manag, vol. 141, no. 10, p. 4015016, Oct. 2015, doi: 10.1061/(ASCE)WR.1943-5452.0000534.

[20] N. Moosavian and B. J. Lence, "Fittest individual referenced differential evolution algorithms for optimization of water distribution networks," J. Comput. Civ. Eng., vol. 33, no. 6, p. 4019036, Nov. 2019, doi: 10.1061/(ASCE)CP.1943-5487.0000849.

[21] R. V. Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems," Int. J. Ind. Eng. Comput., vol. 7, pp. 19–34, 2016, doi: 10.5267/j.ijiec.2015.8.004.

[22] R. V. Rao and G. G. Waghmare, "A new optimization algorithm for solving complex constrained design optimization problems," Eng. Optim., vol. 49, no. 1, pp. 60–83, Jan. 2017, doi: 10.1080/0305215X.2016.1164855.

[23] R. V. Rao, D. P. Rai, J. Ramkumar, and J. Balic, "A new multi-objective Jaya algorithm for optimization of modern machining processes," Adv. Prod. Eng. Manag., vol. 11, no. 4, pp. 271–286, 2016, doi: 10.14743/apem2016.4.226.

[24] R. V. Rao and A. Saroj, "A self-adaptive multi-population based Jaya algorithm for engineering optimization," Swarm Evol. Comput., vol. 37, pp. 1–26, 2017, doi: 10.1016/j.swevo.2017.04.008.

[25] R. V. Rao, A. Saroj, and S. Bhattacharyya, "Design optimization of heat pipes using elitism-based self-adaptive multipopulation Jaya algorithm," J. Thermophys. Heat Transf., vol. 32, no. 3, pp. 702–712, Mar. 2018, doi: 10.2514/1.T5348.

[26] V. Kumar and S. M. Yadav, "Optimization of reservoir operation with a new approach in evolutionary computation using TLBO algorithm and Jaya algorithm," Water Resour. Manag, vol. 32, no. 13, pp. 4375–4391, 2018, doi: 10.1007/s11269-018-2067-5.

[27] Z. Ding, J. Li, and H. Hao, "Structural damage identification using improved Jaya algorithm based on sparse regularization and Bayesian inference," Mech. Syst. Signal Process., vol. 132, pp. 211–231, 2019, doi: 10.1016/j.ymssp.2019.06.029.

[28] R. V. Rao, H. S. Keesari, P. Oclon, and J. Taler, "An adaptive multi-team perturbation-guiding Jaya algorithm for optimization and its applications," Eng. Comput., vol. 36, no. 1, pp. 391–419, 2020, doi: 10.1007/s00366-019-00706-3.

[29] G. Iacca, V. C. dos Santos Junior, and V. Veloso de Melo, "An improved Jaya optimization algorithm with Lévy flight," Expert Syst. Appl., vol. 165, p. 113902, 2021, doi: 10.1016/j.eswa.2020.113902.

[30] Z. H. Ding, Z. R. Lu, and F. X. Chen, "Parameter identification for a three-dimensional aerofoil system considering uncertainty by an enhanced Jaya algorithm," Eng. Optim., pp. 1–21, Feb. 2021, doi: 10.1080/0305215X.2021.1872558.

[31] E. H. Houssein, A. G. Gad, and Y. M. Wazery, "Jaya algorithm and applications: A comprehensive review", Metaheuristics and Optimization in Computer and Electrical Engineering, N. Razmjooy, M. Ashourian, and

Z. Foroozandeh, Eds. Cham: Springer International Publishing, 2021, pp. 3–24.
[32] P. R. Bhave and R. Gupta, Analysis of Water Distribution Networks. Oxford, U.K: Alpha Science International, 2006.
[33] L. A. Rossman, "EPANET users manual," 1994.
[34] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, Numerical Recipes: The Art of Scientific Computing, 3rd ed. USA: Cambridge University Press, 2007.
[35] G. Z. Watters, Analysis and Control of Unsteady Flow in Pipelines. United States: Butterworths, Boston, MA, 1984.
[36] M. H. Afshar and M. A. Mariño, "A parameter-free self-adapting boundary genetic search for pipe network optimization," Comput. Optim. Appl., vol. 37, no. 1, pp. 83–102, 2007, doi: 10.1007/s10589-007-9016-1.
[37] R. V. Rao, Jaya: An Advanced Optimization Algorithm and its Engineering Applications, 1st ed. Springer Publishing Company, Incorporated, 2018.
[38] E. Alperovits and U. Shamir, "Design of optimal water distribution systems," Water Resour. Res., vol. 13, no. 6, pp. 885–900, Dec. 1977, doi: 10.1029/WR013i006p00885.
[39] H. A. A. Abdel-Gawad, "Optimal design of pipe networks by an improved genetic algorithm," In Proceedings of the Sixth International Water Technology Conference, IWTC 2001, Alexandria, Egypt (pp. 155–166).
[40] O. Fujiwara and D. B. Khang, "A two-phase decomposition method for optimal design of looped water distribution networks," Water Resour. Res., vol. 26, no. 4, pp. 539–549, Apr. 1990, doi: 10.1029/WR026i004p00539.
[41] D. A. Savic and G. A. Walters, "Genetic algorithms for least-cost design of water distribution networks," J. Water Resour. Plan. Manag, vol. 123, no. 2, pp. 67–77, Mar. 1997, doi: 10.1061/(ASCE)0733-9496(1997)123:2(67).
[42] M. Abou Rayan, B. Djebedjian, N. G. El-Hak, and A. Herrick, "Optimization of potable water network (case study)," in 7th Int. Water Technology Conf., International Water Technology Association, Alexandria, Egypt, 2003, pp. 507–522.
[43] H. A. El-Ghandour and E. Elbeltagi, "Comparison of five evolutionary algorithms for optimization of water distribution networks," J. Comput. Civ. Eng., vol. 32, no. 1, p. 4017066, Jan. 2018, doi: 10.1061/(ASCE)CP.1943-5487.0000717.
[44] R. M. Ezzeldin and B. Djebedjian, "Optimal design of water distribution networks using whale optimization algorithm," Urban Water J., vol. 17, no. 1, pp. 14–22, Jan. 2020, doi: 10.1080/1573062X.2020.1734635.
[45] D. Mora-Melià, F. J. Martínez-Solano, P. L. Iglesias-Rey, and J. H. Gutiérrez-Bahamondes, "Population size influence on the efficiency of evolutionary algorithms to design water networks," Procedia Eng., vol. 186, pp. 341–348, 2017, doi: 10.1016/j.proeng.2017.03.209.
[46] S. N. Poojitha, G. Singh, and V. Jothiprakash, "Improving the optimal solution of GoYang network – using genetic algorithm and differential evolution," Water Supply, vol. 20, no. 1, pp. 95–102, Sep. 2019, doi: 10.2166/ws.2019.139.
[47] J. Reca, J. Martínez, and R. López, "A hybrid water distribution networks design optimization method based on a search space reduction approach and a genetic algorithm," Water, vol. 9, no. 11, 2017, doi: 10.3390/w9110845.
[48] C. R. Suribabu, "Differential evolution algorithm for optimal design of water distribution networks," J. Hydroinformatics, vol. 12, no. 1, pp. 66–82, Sep. 2009, doi: 10.2166/hydro.2010.014.
[49] X. Dong, S. Liu, T. Tao, S. Li, and K. Xin, "A comparative study of differential evolution and genetic algorithms for optimizing the design of water distribution systems," J. Zhejiang Univ. Sci. A, vol. 13, no. 9, pp. 674–686, 2012, doi: 10.1631/jzus.A1200072.
[50] J. Reca and J. Martínez, "Genetic algorithms for the design of looped irrigation water distribution networks," Water Resour. Res., vol. 42, no. 5, May 2006, doi: 10.1029/2005WR004383.
[51] A. Bolognesi, C. Bragalli, A. Marchi, and S. Artina, "Genetic heritage evolution by stochastic transmission in the optimal design of water distribution networks," Adv. Eng. Softw., vol. 41, no. 5, pp. 792–801, 2010, doi: 10.1016/j.advengsoft.2009.12.020.
[52] R. Baños, C. Gil, J. Reca, and F. G. Montoya, "A memetic algorithm applied to the design of water distribution networks," Appl. Soft Comput., vol. 10, no. 1, pp. 261–266, 2010, doi: 10.1016/j.asoc.2009.07.010.
[53] Z. W. Geem, "Particle-swarm harmony search for water network design," Eng. Optim., vol. 41, no. 4, pp. 297–311, Apr. 2009, doi: 10.1080/03052150802449227.
[54] A. Sedki and D. Ouazar, "Hybrid particle swarm optimization and differential evolution for optimal design of water distribution systems," Adv. Eng. Informatics, vol. 26, no. 3, pp. 582–591, 2012, doi: 10.1016/j.aei.2012.03.007.
[55] N. Moosavian and B. K. Roodsari, "Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks," Swarm Evol. Comput, vol. 17, pp. 14–24, 2014, doi: 10.1016/j.swevo.2014.02.002.
[56] M. H. Afshar, "Penalty adapting ant algorithm: Application to pipe network optimization," Eng. Optim., vol. 40, no. 10, pp. 969–987, Oct. 2008, doi: 10.1080/03052150802236079.
[57] F. Zheng, A. R. Simpson, and A. C. Zecchin, "Coupled binary linear programming–differential evolution algorithm approach for water distribution system optimization," J. Water Resour. Plan. Manag., vol. 140, no. 5, pp. 585–597, May 2014, doi: 10.1061/(ASCE)WR.1943-5452.0000367.
[58] D. Graeme, W. Andrew, and R. Hayley, "A methodology for comparing evolutionary algorithms for optimising water distribution systems," Water Distribution Systems Analysis 2010. pp. 786–798, doi: 10.1061/41203(425)73.
[59] F. Zheng, A. R. Simpson, and A. Zecchin, "A performance comparison of differential evolution and genetic algorithm variants applied to water distribution system optimization," World Environmental and Water Resources Congress 2012. pp. 2954–2963, doi: 10.1061/9780784412312.296.
[60] M. H. Afshar, "Application of a compact genetic algorithm to pipe network optimization problems," Sci. Iran., vol. 16, no. 3, 2009, [Online]. Available: http://scientiairanica.sharif.edu/article_3103.html.
[61] C. R. Suribabu and T. R. Neelakantan, "Particle swarm optimization compared to other heuristic search techniques for pipe sizing," J. Environ. Informatics, vol. 8, no. 1, 2006.
[62] B. A. Tolson, M. Asadzadeh, H. R. Maier, and A. Zecchin, "Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for water distribution system design optimization," Water Resour. Res., vol. 45, no. 12, Dec. 2009, doi: 10.1029/2008WR007673.
[63] F. Zheng, A. R. Simpson, and A. C. Zecchin, "A combined NLP-differential evolution algorithm approach for the optimization of looped water distribution systems," Water Resour. Res., vol. 47, no. 8, Aug. 2011, doi: 10.1029/2011WR010394.
[64] F. Zheng, A. R. Simpson, and A. Zecchin, "Performance study of differential evolution with various mutation strategies applied to water distribution system optimization," World Environmental and Water Resources Congress 2011. pp. 166–176, doi: 10.1061/41173(414)18.
[65] K. M. Aghdam, I. Mirzaee, N. Pourmahmood, and M. P. Aghababa, "Adaptive mutated momentum shuffled frog leaping algorithm for design of water distribution networks," Arab. J. Sci. Eng., vol. 39, no. 11, pp. 7717–7727, 2014, doi: 10.1007/s13369-014-1367-1.
[66] F. Zheng, A. R. Simpson, and A. C. Zecchin, "A method for assessing the performance of genetic algorithm optimization for water distribution design," Water Distribution Systems Analysis 2010. pp. 771–785, doi: 10.1061/41203(425)72.
[67] W. Bi, Y. Xu, and H. Wang, "Comparison of searching behaviour of three evolutionary algorithms applied to water distribution system design optimization," Water, vol. 12, no. 3, p. 695, 2020.
[68] R. Sheikholeslami, A. Kaveh, A. Tahershamsi, and S. Talatahari, "Application of charged system search algorithm to water distrubition networks optimization," Int. J. Optim. Civ. Eng., vol. 4, no. 1, pp. 41–58, 2014.
[69] Z. W. Geem, "Improved harmony search from ensemble of music players," In International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (pp. 86–93). Springer, Berlin, Heidelberg, 2006.
[70] X. Qi, K. Li, and W. D. Potter, "Estimation of distribution algorithm enhanced particle swarm optimization for water distribution network optimization," Front. Environ. Sci. Eng., vol. 10, no. 2, pp. 341–351, 2016, doi: 10.1007/s11783-015-0776-z.

Title Arabic:

خوارزمية جايا المعدلة للتصميم الأمثل لشبكة توزيع المياه

Abstract Arabic:

لأول مرة، يتم استخدام خوارزمية جايا (JA) المقترحة مؤخرًا من أجل التصميم الأمثل لشبكات توزيع المياه. لا تحتوي هذه الخوارزمية على متغيرات تحكم، مما يلغي الجهد الحسابي المستنفذ والمطلوب لتنفيذ خطوة الضبط الأساسية لهذه المتغيرات. تم اقتراح شكل جديد للخوارزمية، وسميت بخوارزمية جايا الخالية من تحليل الحساسية (FSAJA) وذلك الشكل لا يحتاج حتى معايرة المتغيرات الشائعة المتعارف على وجودها بين طرق التحسين المختلفة. تم فحص ستة بدائل مختلفة للخوارزمية المقترحة لتحديد أفضلها. ثم تم اختبار البديل الأفضل بحل ثلاثة مشاكل قياسية لشبكات توزيع المياه بالإضافة لشبكة وطنية. تظهر مقارنة أداء كل من الخوارزمية المقترحة FSAJA مع كل من الخوارزمية الأصلية JA والخوارزميات التحسينية المختلفة المتوفرة في الأبحاث، الفعالية الواعدة والكفاءة والمتانة للخوارزمية المقترحة.