Desalination and Water Treatment
www.deswater.com

doi: 10.5004/dwt.2019.24146

161 (2019) 1–13
September

Multi-swarm optimizer applied in water distribution networks

Douglas F. Surco[a,b,*], Diogo H. Macowski[a,b], João G. L. Coral[c], Flávia A.R. Cardoso[a], Thelma P.B. Vecchi[a], Mauro A.S.S. Ravagnani[b]

[a]*Department of Civil Engineering, Federal University of Technology, Via Rosalina Maria dos Santos, 1233 CEP 87301-899 Campo Mourão, Paraná, Brazil, emails: fukunaga@utfpr.edu.br (D.F. Surco), dmacowski@utfpr.edu.br (D.H. Macowski), flaviareitz@gmail.com (F.A.R. Cardoso), thelmapbv@gmail.com (T.P.B. Vecchi)*
[b]*Department of Chemical Engineering, State University of Maringá, Av. Colombo, 5790 CEP 87020-900 Maringá, Paraná, Brazil, email: massravagnani@uem.br*
[c]*Operational Development, Umuarama Regional Unit, Companhia de Saneamento do Paraná, Sanepar, Av. Presidente Castelo Branco, 3525 CEP 87501-050 Umuarama, Paraná, Brazil, email: eng.joaocoral@gmail.com*

Received 16 September 2018; Accepted 20 March 2019

ABSTRACT

In the present work, a new approach is presented for the optimization of multi-modal nonlinear programming problems with constraints or a nondifferentiable objective function. The model is applied in the optimization of water distribution networks (WDN). An algorithm is proposed to solve the problem, based on multi-swarm optimization (MSO) with multiple swarms that work corporately – a master swarm and several slave swarms – named multi-swarm corporative particle-swarm optimizer (MSC-PSO). There are discrete and continuous decision variables and the problem can be treated as a mixed discrete nonlinear programming (MDNLP) one. The combinations of the algorithm search parameters are obtained in a simple manner, allowing viable and promising solutions. A benchmark problem from the literature is studied, in which the installation costs of a WDN are to be minimized with a computational time of 50 s. The implementation of the algorithm is proven to be efficient, with reduction in pipe installation cost up to 1.08% when compared with results from the literature. The algorithm is also implemented in a primary network, installed in the town of Esperança Nova, Paraná, Brazil, with reduction of 4.28% in the total cost when compared to the current WDN in operation. The computational time for this case was 69 s.

Keywords: Multi-swarm optimization algorithms; Particle-swarm optimization; Water distribution network; Optimization; MDNLP

1. Introduction

The optimization of a water distribution network (WDN) is a multimodal nonlinear programming problem with a nondifferentiable objective function and with real and discrete decision variables. The analytical solution to the optimization problem is highly complex, requiring simultaneous analysis of the mass conservation equation in each node and of the energy conservation equation in each network loop. Furthermore, the minimum pressures in the demand nodes, as well as the minimum and maximum velocities in the pipes, must be considered.

In the 1970s, several researchers [1–3], among others, studied WDN optimization using deterministic methods such for solving linear programming (LP), nonlinear programming (NLP), and dynamic programming (DP) problems formulations, with adaptations used in order to work around constraints or differentiability and continuity problems of the objective functions. In the case of the deterministic methods, the solution is strongly dependent on the initial proposition.

* Corresponding author.

In the last four decades, however, heuristic methods have been used due to their efficiency in dealing with the aforementioned problems. Nevertheless, the use of hydraulic simulators allows complex mathematical treatments to be avoided.

One of the most successful heuristic methods in the SI (swarm intelligence) group is PSO (particle swarm optimization). First introduced by Kennedy and Eberhart [4], the method is based on patterns found in the individual and social behaviors of certain natural species that require communication between individuals for the survival of the group. This behavior is observed in several animal species, some examples being schools of fish, flights of birds, and swarms of bees. Each particle (bird, bee, fish) represents a potential solution to the system, mathematically represented by a vector, x_i, whose N components are the decision variables of the problem. At a certain instant, when the particle reaches its best position (best evaluation of the objective function), its coordinates are named Personal Best (Pbest) and are represented by vector $\boldsymbol{p}_i$. The best position reached by any particle in the group is named Global Best (Gbest) and is represented by vector $\boldsymbol{g}$. The future location of particle i in the search space will be directed by these vectors $\boldsymbol{p}_i$ and $\boldsymbol{g}$. The particles of the PSO algorithm move towards a global or local optimum. When the leader particle (Gbest) stops at a local optimum, the group converges prematurely. This causes the search for the global optimum solution to be challenging, especially in highly nonlinear multimodal problems.

According to Xu et al. [5], PSO convergence is quicker than other evolutionary algorithms [6], such as genetic algorithms (GA), developed by Holland [7] inspired by the theory of evolution; ant colony optimizations (ACO), developed by Dorigo [8], based on ant foraging; simulated annealing (SA), developed by Kirkpatrick et al. [9], based on metal annealing; shuffled complex evolution (SCE), developed by Duan et al. [10]; harmony search (HS), the algorithm that mimics the musical harmony phenomenon, created by Geem et al. [11] inspired by the improvisation process of musicians; memetic algorithms (MA), developed by Dawkins, [12], inspired in GA with na individual learning procedure able to refine local search and the shuffled frog leaping algorithm (SFLA), developed by Eusuff and Lansey [13] was inspired in MA and PSO in global optima search.

Being one of the most used optimization heuristic methods, particle swarm optimization (PSO) was used Trigueros et al. [14] and Ravagnani et al. [15] for the optimization of reuse water networks. Several studies presented solutions to the WDN optimization problem using PSO, such as [16] and [17]. Surco et al. [18] presented a modified PSO algorithm, potentializing the particles for better exploration of the search space and described the influence of the parameters in the optimization process.

Considering other evolutive algorithms applied on WDN optimization, Zheng et al. [19] used ACO with parameters adaptive strategies in searching better results. Reca et al. [20] developed a GA jointly with search space reduction methods by limiting pipe diameters. El-Ghandour and Elbeltagi [21], used five evolutionary algorithms (GA, PSO, ACO, MA and SFLA) to solve the WDN optimization and concluded that PSO has the best performance in achieving the better results.

Multi-Swarm Optimization is an extension of the PSO algorithm. Interacting with several swarms, it is adequate for multimodal problem optimization. Chen et al. [22] presented a model inspired by mutualism, which is the occurrence of information exchange between the particles of a certain swarm as well as between particles from a swarm and the best ones from other swarms. The authors used 17 mathematical benchmark functions to prove the efficiency of the model.

In the present work, a new approach is presented for the optimization of nonlinear programming problems of a multi-modal nature, with constraints or a nondifferentiable objective function, where variables are mixed (discrete and continuous numbers). The model is then used to optimize WDN at small and large scales. An algorithm based on MSO is proposed in order to solve the problem. A new model is proposed, named Multi-Swarm Corporative Particle Swarm Optimizer (MSC-PSO). An initialization strategy for the particles is also employed for better distribution throughout the search space, increasing the particles search of exploration. Results obtained for a case study from the literature shown the applicability of the proposed approach in finding better optima when compared to previous evolutive algorithms.

2. PSO algorithm

The coordinates of particle i are represented by the components of vector $\boldsymbol{x}_i = (x_{i,1},\ldots,x_{i,j},\ldots,x_{i,N})$ in N-dimensional space. When component $x_{i,j}$ is a real number (continuous), it is limited by x_j^L (lower limit) and x_j^U (upper limit), that is, $x_{i,j} \in [x_j^L, x_j^U]$. When $x_{i,j}$ is a discrete number, it belongs to the set $\boldsymbol{x}_j^{\mathrm{SET}} = \{x_{j1}, x_{j2},\ldots,x_{j\mathrm{ND}}\}$, which has ND elements.

According to Kennedy and Eberhart [4], the new position of particle i at $(t + 1)$ is represented by Eq. (1):

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1) \tag{1}$$

where $v_{i,j}(t + 1)$ is the velocity of component j of particle i, given by Eq. (2):

$$v_{i,j}(t+1) = v_{i,j}(t) + c_1 r_1 \left(p_{i,j} - x_{i,j}(t)\right) + c_2 r_2 \left(g_j - x_{i,j}(t)\right) \tag{2}$$

where c_1 and c_2 are, respectively, the cognitive and social acceleration coefficients and r_1 and r_2 are uniformly distributed random numbers in the interval (0,1). The use of Eq. (2) requires prior knowledge of the values of variables $p_{i,j}$ (component j of vector Pbest) and g_j (component j of vector Gbest). The velocity of the particles is limited, that is, $v_{i,j} \in [v_L, v_U]$, where v_L and v_U are the predetermined lower and upper limits.

Kennedy and Eberhart [23] implemented the PSO algorithm for discrete binary variables by introducing Eq. (3):

$$x_{i,j} = \begin{cases} 1 & if\ rand(\ ) < S(v_{i,j}) \\ 0 & if\ rand(\ ) \geq S(v_{i,j}) \end{cases} \tag{3}$$

where $S(v_{i,j})$ is a sigmoid limiting transformation represented by Eq. (4) and *rand*( ) is a uniformly distributed random number in the interval (0,1).

$$S\left(v_{i,j}\right)=\frac{1}{1+e^{v_{i,j}}} \tag{4}$$

Shi and Eberhart [24] presented a modified PSO algorithm with the introduction of a new parameter named inertial weight (w), recommending that it decrease along the iteration process instead of being static. This modification is shown in Eq. (5), which replaces Eq. (2).

$$v_{i,j}\left(t+1\right)=wv_{i,j}\left(t\right)+c_1r_1\left(p_{i,j}-x_{i,j}\right)+c_2r_2\left(g_j-x_{i,j}\right) \tag{5}$$

Due to the importance of this parameter, several formulations have been proposed for w as a function of the iteration number. According to Suribabu and Neelakantan [16], the inertia weight influences the exploration of the search space. A larger inertia weight makes global exploration easier, while a smaller one improves local exploration.

3. Multi-swarm optimization

Considering multi-swarm optimization, Niu et al. [25] presented the multi-swarm cooperative particle swarm optimizer (MCPSO) model, inspired by the symbiosis existing in natural ecosystems. It is based on the master-slave model, where the total population comprises a master swarm and several slave swarms. Each slave swarm moves as a single PSO would, while the master swarm moves according to its own knowledge added to those of the slave swarms. Six 30-dimensional mathematical benchmark functions were used to evaluate the performance of the model.

Kaveh and Laknejadi [26] presented the multi-objective optimization multi-swarm charged system search (MO-MSCSS) algorithm for problems that use several PSO swarms, in an attempt to solve optimization problems for metallic structures. Four unconstrained multi-objective problems were used to evaluate the performance of the model. The results were compared with those of six known multi-objective models from the literature (MOEA/D, NSGA-II, SPEA2, MOPSO, sMOPSO, and cMOPSO) and of two constrained multi-objective problems of dimensioning metal structures. Niu et al. [25] proposed two versions of MCPSO:

- Competitive version (COM-MCPSO), whose particle movement is guided by Eqs. (6) and (7):

$$v_{i,j}^{M}\left(t+1\right)=wv_{i,j}^{M}\left(t\right)+c_1r_1\left(p_{i,j}^{M}-x_{i,j}^{M}\left(t\right)\right)+\Phi c_2r_2\left(g_i^{M}-x_{i,j}^{M}\left(t\right)\right)+\left(1-\Phi\right)c_3r_3\left(g_j^{S}-x_{i,j}^{M}\left(t\right)\right) \tag{6}$$

$$x_{i,j}^{M}\left(t+1\right)=x_{i,j}^{M}\left(t\right)+v_{i,j}^{M}\left(t+1\right) \tag{7}$$

where M corresponds to master swarm, S corresponds to slave swarm, r_3 is a uniformly distributed random number in the interval (0,1), c_3 is the acceleration coefficient, $p_{i,j}^{M}$ is component j of vector Pbest^{M} corresponding to particle i, g_j^{S} is component j of vector Gbest^{S} (the best Gbest of all the slave swarms), g_j^{M} is component j of vector Gbest^{M}, Φ is the migration factor given by Eq. (8) (for minimization problems):

$$\Phi=\begin{cases}0 & \text{Gbest}^{S}<\text{Gbest}^{M}\\ 0.5 & \text{Gbest}^{S}=\text{Gbest}^{M}\\ 1 & \text{Gbest}^{S}>\text{Gbest}^{M}\end{cases} \tag{8}$$

- Collaborative version (COL-MCPSO), whose trajectories are adjusted according to Eqs. (9) and (10):

$$v_{i,j}^{M}\left(t+1\right)=wv_{i,j}^{M}\left(t\right)+c_1r_1\left(p_{i,j}^{M}-x_{i,j}^{M}\left(t\right)\right)+c_2r_2\left(g_{i,j}^{M}-x_{i,j}^{M}\left(t\right)\right)+c_3r_3\left(g_j^{S}-x_{i,j}^{M}\left(t\right)\right) \tag{9}$$

$$x_{i,j}^{M}\left(t+1\right)=x_{i,j}^{M}\left(t\right)+v_{i,j}^{M}\left(t+1\right) \tag{10}$$

In COM-MCPSO and COL-MCPSO, the particles move drawing information from the master swarm and from the leader of each slave swarm. Each slave swarm carries out a competition, without information exchange between swarms. Each slave swarm sends its best local particle to the master swarm. The master swarm selects the best leader among the slave swarms to compose the new orientation of the particles of its swarm.

4. MCPSO improvement proposal: multi-swarm corporative particle swarm optimizer (MSC-PSO)

In the collaborative version described by Eqs. (9) and (10), the master swarm commands every swarm, using its knowledge together with the collective knowledge of the slave swarms. When variables are discrete, a particle belonging to a slave swarm may take the best position. However, even if the said particle reaches the best theoretical position, its contribution cannot lead other swarms, given that the orientation of the speed is determined by four vectors. As a countermeasure to this deficiency, a corporative algorithm is proposed in the present work, one where a particle from a slave swarm has the opportunity of redirecting the groups to obtain a better result. The mathematical formulation that guides the trajectories of the particles is given by Eqs. (11)–(14).

Equations for the speed and for the new position of a slave swarm particle:

$$v_{i,j}^{S}\left(t+1\right)=wv_{i,j}^{S}\left(t\right)+c_1r_1\left(p_{i,j}^{S}-x_{i,j}^{S}\left(t\right)\right)+c_2r_2\left(g_j^{S}-x_{i,j}^{S}\left(t\right)\right)+c_3r_3\left(g_j^{M}-x_{i,j}^{S}\left(t\right)\right) \tag{11}$$

$$x_{i,j}^{S}\left(t+1\right)=x_{i,j}^{S}\left(t\right)+v_{i,j}^{S}\left(t+1\right) \tag{12}$$

Equations for the speed and for the new position of a master swarm particle:

$$v_{i,j}^{M}\left(t+1\right)=wv_{i,j}^{M}\left(t\right)+c_1r_1\left(p_{i,j}^{M}-x_{i,j}^{M}\left(t\right)\right)+c_3r_3\left(g_j^{M}-x_{i,j}^{S}\left(t\right)\right) \tag{13}$$

$$x_{i,j}^{M}\left(t+1\right)=x_{i,j}^{M}\left(t\right)+v_{i,j}^{M}\left(t+1\right) \tag{14}$$

where g_j^{M} is component j of particle Gbest^{M}.

According to Eqs. (11)–(14), particles belonging to the slave swarms do not move independently, but rather using their knowledge of the best position reached by particles of their own swarm as well as the position of $Gbest^M$ by instant *t*.

Furthermore, strategies that improve the distribution of particles in the search space can be employed in the initialization of the particles belonging to the slave swarms, as shown in Fig. 1.

The *Two polar points* strategy distributes the particles of a swarm around *Point1* and the particles of another swarm around *Point2*, within a radius R^S ($R^S = \psi$. *MaxDist*), where ψ is a constant in the interval (0.5,0.6) and *MaxDist* is the Euclidean distance between *Point1* and *Point2*, calculated by Eq. (15). The components of *Point1* and *Point2* are, respectively, the lower and upper limits of the decision variables.

$$MaxDist = \sqrt{\sum_{j=1}^{N} \left(x_j^U - x_j^L \right)^2} \quad (15)$$

The search space of Fig. 1 is two-dimensional and variables x_1 and x_2 are limited ($x_1 \in [x_1^L, x_1^U]$ and $x_2 \in [x_2^L, x_2^U]$). *Point1* has coordinates (x_1^L, x_2^L). *Point2*, (x_1^U, x_2^U).

An initial particle x_j belonging to slave swarm 1 must obey the distance criterium:

$$\sqrt{\sum_{j=1}^{N} \left(x_j^L - x_j^{S_1} \right)^2} \leq R^S \quad (16)$$

If this inequality is not satisfied, particle x_j is discarded and another initialization vector is created, according to Inequality (16). The same criterium is applied to all the initialization particles belonging to slave swarm 2.

5. Optimization of WDN

WDNs are important systems in urban centers and its optimal design has been fundamental since the beginning of the human society development. In the present paper, the WDN problem is studied considering networks with one or more reservoirs, demand nodes and the existence of pipes and accessories (valves and pumps), in such a way that the water must be available for the consumer at adequate pressures. The single pipe approach is used and the optimization problem consists in minimizing the WDN piping cost subject to the mass balance in the nodes and the energy balance in the loops, considering pressure and velocity limits. For the WDN optimization, a set of commercial diameters (D_{SET}) is made available to be used in the network. The optimization of the network comprises the search for the diameters (belonging to D_{SET}) of the pipes so the network is designed with the lowest possible cost, assuring minimum pressures in the consumption nodes and velocities among minimum and maximum limits.

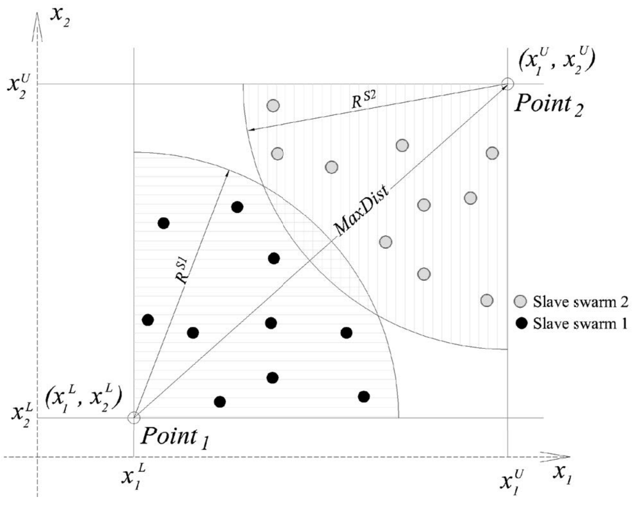

Fig. 1. *Two polar points* slave swarm particle initialization strategy.

5.1. Development of the WDN optimization model

If a network has *N* pipes and *K* nodes, the model can be formulated as:

$$\text{Min } C_P = \sum_{j=1}^{N} L_j \times Cost\left(D_j\right) \quad (17)$$

subjected to the following constraints:

$$\sum q\left(k\right) = 0, \quad \forall\, k = 1,...,K \quad (18)$$

$$\sum h_f\left(j\right) = 0, \quad \forall\, j \in \text{set of pipes in a loop} \quad (19)$$

$$pr\left(k\right) \geq pr_{min}\left(k\right) \quad (20)$$

$$v_{w_{\min}} \leq \left| v_{w_j} \right| \leq v_{w_{\max}} \quad (21)$$

$$D_j \in D_{\text{SET}} = \left\{ D_1, D_2, ..., D_{\text{ND}} \right\} \quad (22)$$

where C_P is the objective function for the minimization of the total pipe installation cost of the network, L_j is the length of pipe *j*, and $Cost(D_j)$ is the installation cost per unit length of the pipe with diameter D_j. Eqs. (18) and (19) are constraints corresponding to the laws of conservation of mass and energy, which assure the steady state of the network. $q(k)$ is the flow rate entering and leaving node *k* and $h_f(j)$ is the pressure drop in pipe *j* belonging to a certain loop. Inequality (20) assures that all pressures are adequate for the system and for the consumers. Inequality Eq. (21) keeps flow velocity in the pipes within the allowed limits. Eq. (22) shows that the selected diameters must belong to the set of available diameters, D_{SET}.

In the present work, the pressure drop, h_f, is calculated in the international system using the Hazen–Williams equation:

$$h_f\left(j\right) = \frac{10.674\, q_j^{1.852}\, L_j}{C_j^{1.852}\, D_j^{4.871}} \quad (23)$$

where C_j is the Hazen–Williams roughness coefficient (dimensionless), q_j is the flow rate (m^3/s), D_j is the diameter (m), and L_j is the length (m) of pipe *j*.

For the hydraulic simulation in the iterative process, an external file is used, named Epanet2.dll. Developed by the North-American Environmental Protection Agency (US EPA), it is freely distributed as a programmer's toolkit [27]. The file is incorporated in the programming code, in a compatible language (Delphi, Pascal, C/C++, among others), used for optimization and for simulation.

5.2. Development of the MSC-PSO algorithm

The set of available diameters for the network, D_{SET}, is made up of *ND* diameters. Organizing in crescent order, $D_1 < D_2 < \ldots < D_{ND}$, where D_1 is named D_{min} and D_{ND} is named D_{max}. Each diameter has a corresponding cost ($/m), $cost_1$, $cost_2$, …, $cost_{ND}$ and a corresponding Hazen–Williams roughness coefficient, C_1, C_2, …, C_{ND}, as shown in Table 1.

Particles are initialized randomly according to Eq. (24), where *r* is a uniformly distributed random number in the interval (0,1). The result is a diameter with a continuous value (D_C) that does not belong to D_{SET}. One way of solving this problem is to carry out a discretization process on the diameters. To that end, a procedure proposed in a study by Surco et al. [18] is used, according to Eq. (25) and Fig. 2.

Diameter D_C is thus converted into a diameter D_U or D_L belonging to D_{SET}.

$$x_{i,j} = D_{min} + r \times \left(D_{max} - D_{min}\right) \quad (24)$$

Table 1
Available diameters (D_{SET}) and their respective properties

Index	Diameter	Cost	Roughness coefficient
1	D_1	$cost_1$	C_1
2	D_2	$cost_2$	C_2
ND	D_{ND}	$cost_{ND}$	C_{ND}

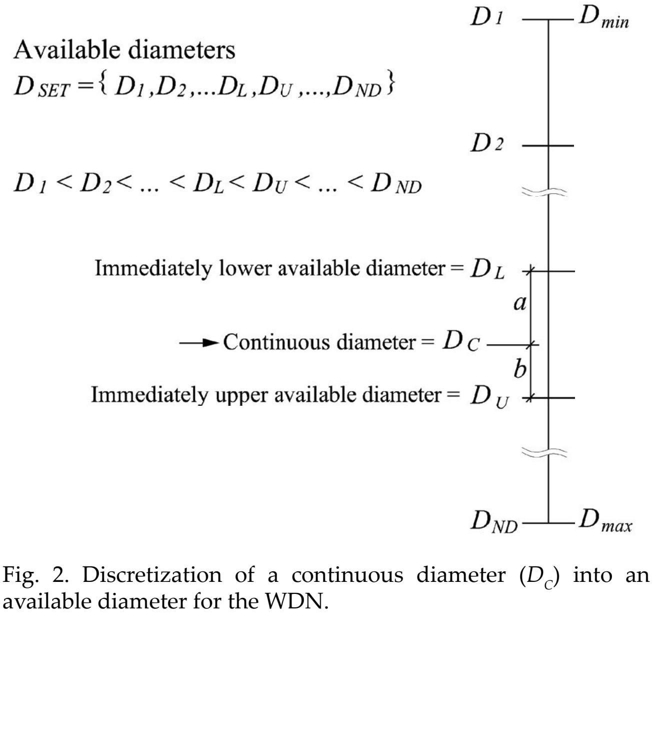

Fig. 2. Discretization of a continuous diameter (D_C) into an available diameter for the WDN.

$$x_{i,j} = \begin{cases} D_L & \text{if } a \le b \\ D_U & \text{if } a > b \end{cases} \quad (25)$$

$x_{i,j}$ is a discrete value, belonging to D_{SET}, that corresponds to the diameter of pipe *j* of particle *i*.

In Eq. (5), $v_{i,j}$ is limited by Inequality (26) and $x_{i,j}(t + 1)$ is limited by Eq. (27):

$$\left|v_{i,j}\right| \le v_U \quad (26)$$

$$x_{i,j}\left(t+1\right) = \begin{cases} D_{min} & \text{if } x_{i,j}\left(t+1\right) < D_{min} \\ D_{max} & \text{if } x_{i,j}\left(t+1\right) > D_{max} \end{cases} \quad (27)$$

Fig. 3 shows that the particle in iteration *t* has the position $x_{i,j}$ and the velocity $v_{i,j}$ and corresponds to the diameter D_n. The velocity $v_{i,j}(t + 1)$ needs to be limited by v_U in such a way that $x_{i,j}(t + 1)$ can assume other diameter value belonging to the set of available diameters. The manner in what the parameter v_U is determined is better explained in Section 5.3.

Particle *i* (x_i) is an *N*-dimensional vector (*N* pipes). Each vector x_i has a corresponding Pbest vector (p_i) and velocity vector (v_i). Vector Gbest is the position with the best performance achieved by some particle of the swarm. The vectors of the particles belonging to the master swarm are formatted as follows:

- $\boldsymbol{x}_i^M = (x_{i,1}^M, x_{i,2}^M, \ldots, x_{i,N}^M)$, position;
- $\boldsymbol{v}_i^M = (v_{i,1}^M, v_{i,2}^M, \ldots, v_{i,N}^M)$, velocity;
- $\boldsymbol{p}_i^M = (p_{i,1}^M, p_{i,2}^M, \ldots, p_{i,N}^M)$, Pbest^M;
- $\boldsymbol{g}^M = (g_1^M, g_2^M, \ldots, g_N^M)$, Gbest^M with respective performance $\boldsymbol{C}_G^M$, which is the best solution found by the MSC-PSO algorithm so far.

The vectors of particle *i* belonging to the N_S slave swarms have analogous formatting:

- $\boldsymbol{x}_i^{S_1}, \boldsymbol{x}_i^{S_2}, \ldots, \boldsymbol{x}_i^{S_{N_S}}$, position;
- $\boldsymbol{v}_i^{S_1}, \boldsymbol{v}_i^{S_2}, \ldots, \boldsymbol{v}_i^{S_{N_S}}$, velocity;
- $\boldsymbol{p}_i^{S_1}, \boldsymbol{p}_i^{S_2}, \ldots, \boldsymbol{p}_i^{S_{N_S}}$, Pbest^S;
- $\boldsymbol{g}^{S_1}, \boldsymbol{g}^{S_2}, \ldots, \boldsymbol{g}^{S_{N_S}}$, Gbest^S.

For each iteration, the position vectors of each particle are updated and evaluated according to Eq. (17). The pressures in each node and the velocities in each pipe are obtained using hydraulic simulator Epanet2. If particle *i* presents *NV* hydraulic violations (nodes with pressures lower than the specified minimum or velocities in the pipes outside the limits), it is penalized with a stipulated *Penalty* value for each violation. The total is then added to the value of the objective function according to Eq. (28):

$$C_{PPi} = \sum_{j=1}^{N} L_j \times Cost\left(x_{i,j}\right) + NV \times Penalty \quad (28)$$

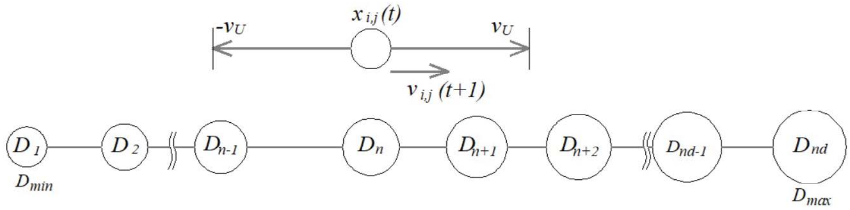

Fig. 3. Limits of $v_{i,j}$ and $x_{i,j}$.

where L_j is the length of pipe j and $Cost(x_{i,j})$ is the cost of pipe j with diameter $x_{i,j}$.

If particle i achieves a better performance (C_{PPi}), this value is stored in column vector $\boldsymbol{f}$; hence the best performances of the particles are stored in this N-dimensional column vector.

For the master swarm, $\boldsymbol{f}^M = (f_1^M, f_2^M, \ldots, f_N^M)^T$, where f_1^M represents the best performance of particle $\boldsymbol{x}_1^M$ achieved so far. f_G^M is the performance of the leader of the master swarm.

For the N_S slave swarms, $\boldsymbol{f}^{S_1} = \left(f_1^{S_1}, f_2^{S_1}, \ldots, f_N^{S_1}\right)^T, \ldots, \boldsymbol{f}^{S_{N_S}} = \left(f_1^{S_{N_S}}, f_2^{S_{N_S}}, \ldots, f_N^{S_{N_S}}\right)^T$ and for the leaders of each slave swarm, $f_G^{S_1}, f_G^{S_1}, \ldots, f_G^{S_{N_S}}$, where $f_G^{S_1}$ represents the performance of $\boldsymbol{g}^{S_1}$ so far.

The vectors whose components have indexes D_{max} and D_{min} are, respectively, named $\boldsymbol{g}_{max} = (D_{max}, D_{max}, \ldots, D_{max})$ and $\boldsymbol{g}_{min} = (D_{min}, D_{min}, \ldots, D_{min})$, where the costs of the WDN are C_{Pmax} and C_{Pmin}, respectively. If particle $\boldsymbol{g}_{max}$ is viable, that is, it has no hydraulic violations, the WDN can be fed by gravity.

After a given maximum number of iterations (t_{max}), the output consists of the diameters of each pipe, stored in vector $\boldsymbol{g}^M = (g_1^M, g_2^M, \ldots, g_N^M)$, the optimized cost f_G^M, and the results of hydraulic variables such as the pressure vector $\boldsymbol{p}_R$ and the velocity vector $\boldsymbol{v}_R$ for vector $\boldsymbol{g}^M$.

Using one master swarm, two slave swarms ($N_S = 2$), and N_p particles for every swarm, the MSC-PSO algorithm applied in WDN optimization is described as follows:

1. Calculate *MaxDist* (distance between *Point1* and *Point2*) using Eq. (15). Calculate cover radius R^S using ψ in interval (0.5, 0.6) to position the particles of the slave swarm.
2. Initialize the particles of the master swarm according to Eqs. (24) and (25). The particles of the slave swarm follow the *Two polar points* strategy, as shown in Fig. 1. Velocity vectors start at rest. The Pbest vectors of all particles initialize with the same $\boldsymbol{g}_{max}$ components and the Gbest vectors of all swarms initialize with the same $\boldsymbol{g}_{min}$ components. The performance of every Pbest and Gbest is set to C_{Pmax}.
3. For each particle (i = 1 to N_P) of a swarm, for the slave swarms and master swarm in parallel:
   - 3.1 Calculate the value of the penalized objective function, $C_{\mathrm{PPi}}{}^S$ for particles of the slave swarm and C_{PPi}^M for particles of the master swarm, according to Eq. (28).
   - 3.2 Update Pbest$_i$: compare performance C_{PPi}^S with performance f_i^S from vector $\boldsymbol{p}_i^S$. If C_{PPi}^S is better, update the performance of vector $\boldsymbol{p}_i^S$ as well as its components: ($f_i^S \leftarrow C_{\mathrm{PPi}}^S$) and ($\boldsymbol{p}_i^S \leftarrow \boldsymbol{x}_i^S$). Likewise, for the master swarm, if $C_{\mathrm{PPi}}^M < f_i^M$, update the performance and the components of vector $\boldsymbol{p}_i^M$: ($f_i^M \leftarrow C_{\mathrm{PPi}}^M$) and ($\boldsymbol{p}_i^M \leftarrow \boldsymbol{x}_i^M$).
   - 3.3 Update Gbest^S: if C_{PPi}^S is lower than f_G^S, update the new components of Gbest^S ($\boldsymbol{g}^S \leftarrow \boldsymbol{x}_i^S$) and its performance ($f_G^S \leftarrow C_{\mathrm{PPi}}^S$). Particle $\boldsymbol{x}_i^S$ can become a Gbest^M. If ($C_{\mathrm{PPi}}^S < f_G^M$), update the performance ($f_G^M \leftarrow C_{\mathrm{PPi}}^S$) and the components of vector $\boldsymbol{g}^M$($\boldsymbol{g}^M \leftarrow \boldsymbol{x}_i^S$). Store pressures and velocities, obtained for particle $\boldsymbol{x}_i^S$ through Epanet2, in vectors $\boldsymbol{p}_R$ and $\boldsymbol{v}_R$, respectively.
   - 3.4 Update Gbest^M: for each particle in the master swarm, compare its performance, C_{PPi}^M, with that of the Gbest of the master swarm, f_G^M. If the particle has a better performance, update the performance and the components of vector $\boldsymbol{g}^M$, ($f_G^M \leftarrow C_{\mathrm{PPi}}^M$) and ($\boldsymbol{g}^M \leftarrow \boldsymbol{x}_i^M$), and store the pressures and velocities, obtained for particle $\boldsymbol{x}_i^M$ through Epanet2, in vectors $\boldsymbol{p}_R$ and $\boldsymbol{v}_R$, respectively.
4. Update the new position of particle $\boldsymbol{x}_i^S(t+1)$ or $\boldsymbol{x}_i^M(t+1)$ according to Eqs. (12) and (14), respectively. If the number of iterations (t) is less than or equal to t_{max}, *go* back to step 3.
5. End: Return the results ($\boldsymbol{g}$, f_G^M, $\boldsymbol{p}_R$, $\boldsymbol{v}_R$).

The block diagram of the MSC-PSO algorithm is shown in Fig. 4.

5.3. Determining the parameters of the MSC-PSO algorithm

According to Surco et al. [28], in the optimization process using the PSO algorithm, there are different sets of PSO parameters that lead to the same optimal solution. These parameters are w, c_1, c_2, c_3, N_P, v_U and *Penalty*.

To obtain parameter w, Eq. (29) is used, with $\lambda = 0.95$, as suggested by Shrivatava et al. [29]. Other values for λ were tested in the present study, but $\lambda = 0.95$ was the one which presented the best performance for the algorithm.

$$w = w_{min} + \left(w_{max} - w_{min}\right) \times \lambda^{(t-1)} \tag{29}$$

where w is a dynamic parameter. In iteration $t = 1$, $w = w_{max}$, and in the end of the iterations, $w = w_{min}$. In the present paper, the considered values were $w_{max} = 0.9$ and $w_{min} = 0.5$. It is done in such a way that in the beginning of the iterations the particles movement has more intensity, allowing a more embracing search for possible solutions. During the iterations, velocities became smaller, inducing a more meticulous search. Other values for w_{max} and w_{min} were tested but these values, 0.9 and 0.5 were the ones that presented the best performance in the search of promising solutions.

According to Kennedy and Eberhart [4], the acceleration coefficients can be initialized with a generic value equal to 2.

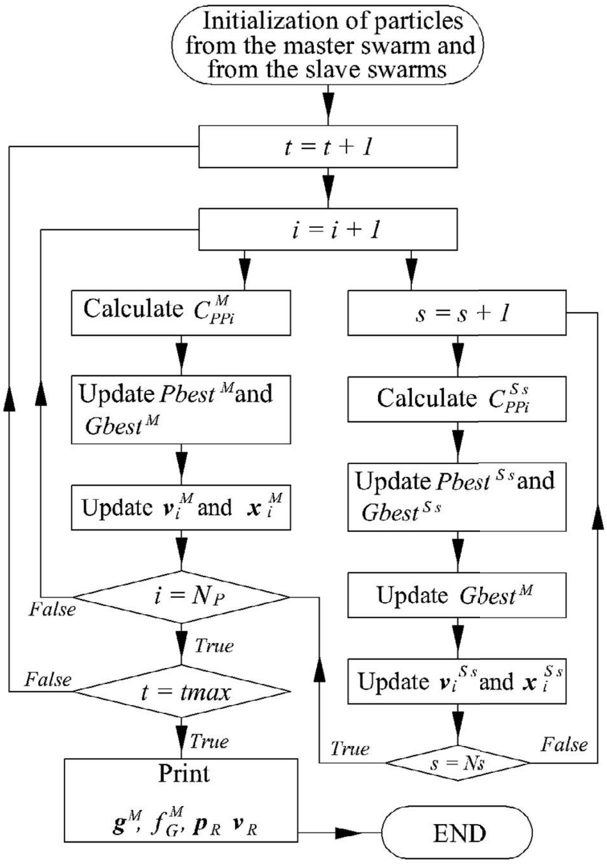

Fig. 4. Block diagram of the MSC-PSO algorithm.

The estimated number of possible solutions (search space) of a WDN with N pipes and ND available commercial diameters is given by $(ND)^N$. The number of particles, N_p, depends on the number of variables to be optimized and is predetermined, in the present work, by Inequality (30):

$$N_P \leq \frac{(N \times ND)}{3} \tag{30}$$

Parameter v_U determines the maximum velocity of the particle. For small values, the search is not very fast and can present nonviable solutions because of its discrete nature. For larger values, the search space is fully spanned in a few interactions. A good strategy is to limit the maximum velocity, v_U, as in Inequality (31):

$$\frac{\Delta D_{max}}{2} \leq v_U \leq \Delta D_{max} \tag{31}$$

ΔD_{max} is the maximum difference between two consecutive commercial diameters available for the WDN, given by:

$$\Delta D_{max} = \max\{\Delta D_n\} \tag{32}$$

where $\Delta D_n = D_n - D_{n-1}$ is the difference between two consecutive diameters in mm.

The main purpose of this strategy is to ensure that the component $x_{i,j}$ $(t + 1)$ can assume other diameter value near to the achieved in iteration t, in order that the optimization process allow the evaluation of all available diameters. Parameter *Penalty* can be limited by Inequality (33):

$$Penalty \leq \frac{C_{Pmax}}{1.5K} \tag{33}$$

Parameters v_U and *Penalty* are the most sensitive in the WDN optimization problem.

6. Case studies

In order to verify the performance of the MSC-PSO algorithm in solving the WDN optimization problem, two case studies were carried out. The first one is a benchmark problem known as two-source network. The second one is a real network in the town of Esperança Nova, located in the state of Paraná, Brazil.

6.1. Two-source network

The two-source network was previously studied by Kadu et al. [30]. Authors developed a modified genetic algorithm and the software GA-NET, using the hydraulic simulator GRA_NET. Suribabu [31] used a heuristic approach based on the information of the flow velocities in the pipes and Epanet was used as hydraulic simulator. Ezzeldin et al. [17] used a modified PSO algorithm, named IDPSO (integer discrete particle swarm optimization) to obtain the solution to the problem. The method of Newton–Raphson was used to solve the hydraulic problem. The two-source network comprises 26 nodes, 34 pipes, 9 loops, and 2 reservoirs.

There is a set of 14 commercially available diameters for the optimization of this network, all of them with a Hazen–Williams roughness coefficient of 130. Their costs per unit length are presented in Table 2.

Table 2
Set of available diameters with corresponding roughness coefficients and costs for the two-source network

D (mm)	*C* (H-W)	Cost ($/m)
150	130	1,115
200	130	1,600
250	130	2,154
300	130	2,780
350	130	3,475
400	130	4,255
450	130	5,172
500	130	6,092
600	130	8,189
700	130	10,670
750	130	11,874
800	130	13,261
900	130	16,151
1000	130	19,395

Source: Kadu et al. [30].

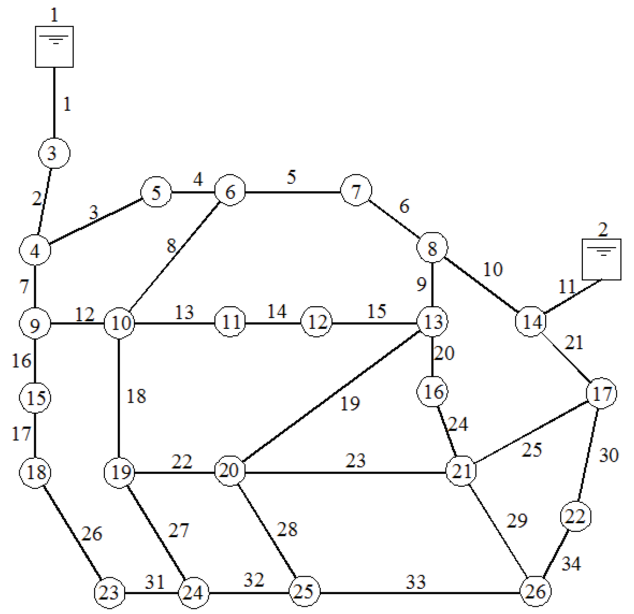

Fig. 5. Layout of the two-source network (adapted from Kadu et al. [30]).

The layout of the two-source network is presented in Fig. 5.

The lengths of the pipes, as well as the minimum values of the hydraulic gradient level and the demands of each node, are shown in Table 3.

For the optimization of the two-source network problem, a master swarm and two slave swarms were used. *Point1*, named g_{min} = (150, 150, ..., 150) with dimension 34, has an Euclidean distance of 4,956.31 mm from *Point2*, named g_{max} = (1,000, 1,000, ..., 1,000). The particles from slave swarm 1 (S_1) were initialized within a radius of 2,973.79 mm from *Point1* (0.6 *MaxDist*). The particles from slave swarm 2 (S_2) were initialized within a radius similar to that of S_1 from *Point2*.

The maximum cost of the network is $651,575,025.00, corresponding to vector g_{max}. The minimum cost is $37,458,425.00, corresponding to vector g_{min}.

The PSO parameters were set at the same values for every swarm as follows: inertia weights w_{max} = 0.9 and w_{min} = 0.5, acceleration coefficients $c_1 = c_2 = c_3 = 2$, maximum particle velocity v_U = 75 mm, *Penalty* = $15,000,000, population of each swarm N_p = 150, and maximum iteration number = 250.

The total installation cost for the network, using the MSC-PSO algorithm to optimize the diameters, is $125,019,790.00, with 50 s computation time in a computer using a 1.6 GHz Intel Core I5 CPU. This minimum value is found in Table 4 together with those from other algorithms.

The solution found using the MSC-PSO algorithm with an initialization strategy for the slave swarms saves $1,349,075.00 (1.08% drop) relative to Kadu et al. [30] and $824,205.00 (0.66% drop) relative to Ezzeldin et al. [17].

For comparison effects, two other multi-swarm algorithms were developed in the present paper. The first one is named MS-COM and uses the algorithm COM-MCPSO and the second one is named MS-COL and uses the algorithm COL-MCPSO. For the three cases, Epanet 2.0 is the hydraulic simulator used to calculate the nodes pressure and the pipes velocities in the WDN.

Table 3
Pipe length, node demand, and minimum node HGL

Pipe	Length (m)	Node	Demand (m^3/min)	Minimum HGL (m)
1	300	1	–	100
2	820	2	–	95
3	940	3	18.4	85
4	730	4	4.5	85
5	1,620	5	6.5	85
6	600	6	4.2	85
7	800	7	3.1	82
8	1,400	8	6.2	82
9	1,175	9	8.5	85
10	750	10	11.5	85
11	210	11	8.2	85
12	700	12	13.6	85
13	310	13	14.8	82
14	500	14	10.6	82
15	1,960	15	10.5	85
16	900	16	9.0	82
17	850	17	6.8	82
18	650	18	3.4	85
19	760	19	4.6	82
20	1,100	20	10.6	82
21	660	21	12.6	82
22	1,170	22	5.4	80
23	980	23	2.0	82
24	670	24	4.5	80
25	1,080	25	3.5	80
26	750	26	2.2	80
27	900	–	–	–
28	650	–	–	–
29	1,540	–	–	–
30	730	–	–	–
31	1,170	–	–	–
32	1,650	–	–	–
33	1,320	–	–	–
34	3,250	–	–	–

Source: Kadu et al. [30].

Table 5 presents the optimized results achieved by the models COM-MCPSO, COL-MCPSO, and MSC-PSO. As it can be noted, the MSC-PSO, proposed in the present work has the best result. In the COL-MCPSO model, the performances of the slave swarms are better than that of the master swarm.

Fig. 6 shows the progress of the optimization using COM-MCPSO, where the master swarm reached a value of $126,275,495 at iteration 148, slave swarm 2 reached the same value at iteration 152, and slave swarm 1 reached a value of $127,113,005 at iteration 248. Each slave swarm evolves independently at each iteration.

Table 4
Optimized diameters and costs for the two-source network

Pipe	Kadu et al. [30]	Suribabu [31]	Ezzeldin et al [17]	Present study
	D (mm)	*D* (mm)	*D* (mm)	*D* (mm)
1	1,000	1,000	900	900
2	900	1,000	900	900
3	350	400	400	350
4	250	200	250	300
5	150	150	150	150
6	250	250	200	250
7	800	1,000	800	800
8	150	150	150	150
9	600	450	400	450
10	700	600	500	500
11	900	1,000	900	750
12	700	800	700	700
13	500	500	600	500
14	450	350	450	500
15	150	150	150	150
16	450	500	500	500
17	350	300	350	350
18	400	450	350	400
19	450	150	200	150
20	150	150	150	150
21	600	900	700	700
22	150	150	150	150
23	150	450	500	450
24	400	300	350	350
25	500	750	700	700
26	200	150	250	250
27	350	300	300	250
28	250	250	300	300
29	150	150	200	200
30	300	300	250	300
31	150	150	150	150
32	150	150	150	150
33	150	150	150	150
34	200	150	150	150
Cost($)	126,368,865.00	140,177,210.00	125,843,995.00	125,019,790.00

Source: Adapted from Ezzeldin et al. [17].

Table 5
Optimized model costs

Model	Master swarm	Slave swarm 1	Slave swarm 2
COM-MCPSO	126,275,495	127,113,005	126,275,495
COL-MCPSO	130,479,465	127,113,005	129,052,290
MSC-PSO	125,019,790	125,019,790	127,473,335

Fig. 7 shows the progress using the COL-MCPSO model, where the slave swarms had a better performance than the master swarm. Comparing COM-MCPSO and COL-MCPSO, the former presents better results and faster progress.

Fig. 8 shows the progress of cost optimization for the proposed model, MSC-PSO, which obtained the best value for pipe installation costs among compared models. In this model, the master swarm uses all information from the slave swarms.

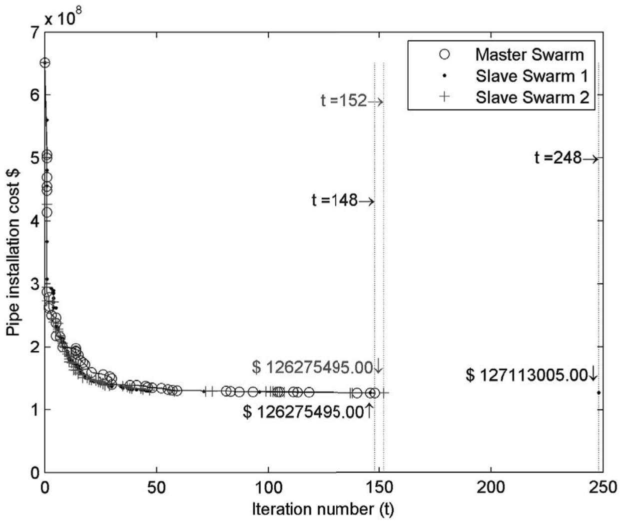

Fig. 6. Progress of cost optimization for the COM-MCPSO model.

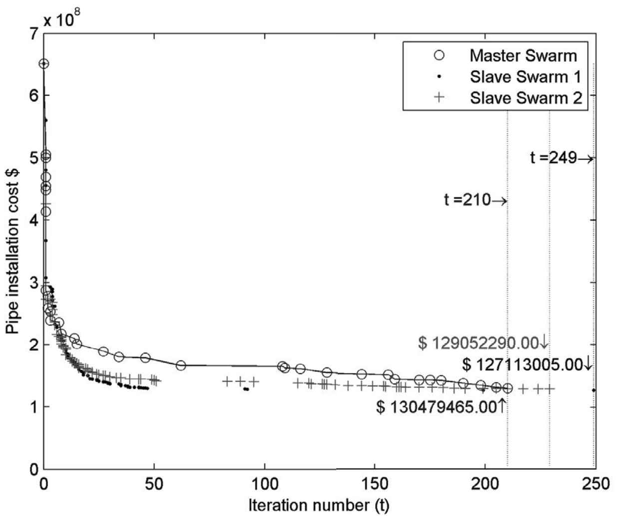

Fig. 7. Progress of cost optimization for the COL-MCPSO model.

The models were also tested without the initialization strategy for slave-swarm particle initialization, as show in Table 6, where the MSC-PSO model also stands out, although the optimized cost is still greater than when using the initialization strategy.

6.2. *Esperança Nova network*

Esperança Nova is a Brazilian town, located in the state of Paraná, with an estimated population of 1,875 people and an area of 138.56 km² (in 2016). The existing network has been in service for over 20 years, with some extensions in the last few years [28].

The network, shown in Fig. 9, is fed by a tank with an elevation of 14.0 m. It comprises 131 nodes and 166 pipes. The available diameters for the network are presented, with their respective costs and roughness coefficients, in Table 7.

Assuming a diameter of 100 mm for every pipe in the network, its maximum cost is found to be US $292,395.13. This value corresponds to vector g_{max}. The minimum cost, corresponding to vector g_{min}, is US $174,932.58. One of the constraints to be considered is the maximum flow velocity ($v_{w_{max}}$), which is 3 m/s for every pipe.

The maximum distance (*MaxDist*) between vectors *Point1* = (32, 32, …, 32) and *Point2* = (100, 100, …, 100) is 876.12 mm. The particles from slave swarm S_1 were randomly distributed around *Point1* within a radius of 525.67 mm (60% of *MaxDist*), while those from slave swarm S_2 were randomly distributed around *Point2* within a radius similar to that from S_1.

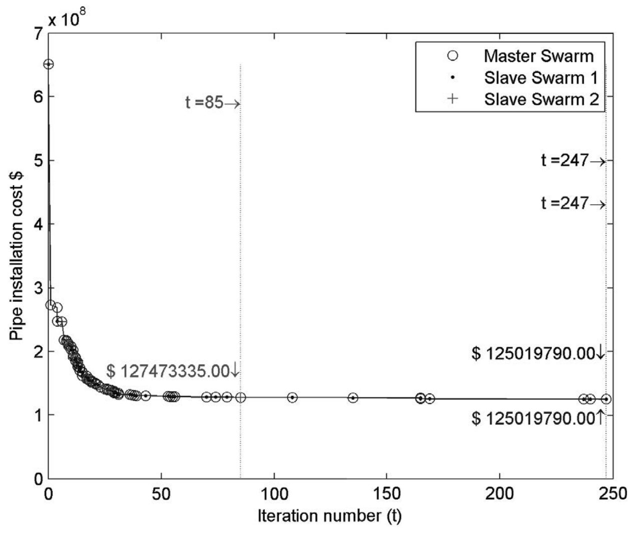

Fig. 8. Progress of cost optimization for the MSC-PSO model.

Table 6
Results without the *Two polar points* strategy

Model	Master swarm	Slave swarm 1	Slave swarm 2
COM-MCPSO	127,351,100	127,351,100	135,654,620
COL-MCPSO	130,479,465	127,113,005	129,052,290
MSC-PSO	126,676,235	126,676,235	129,221,480

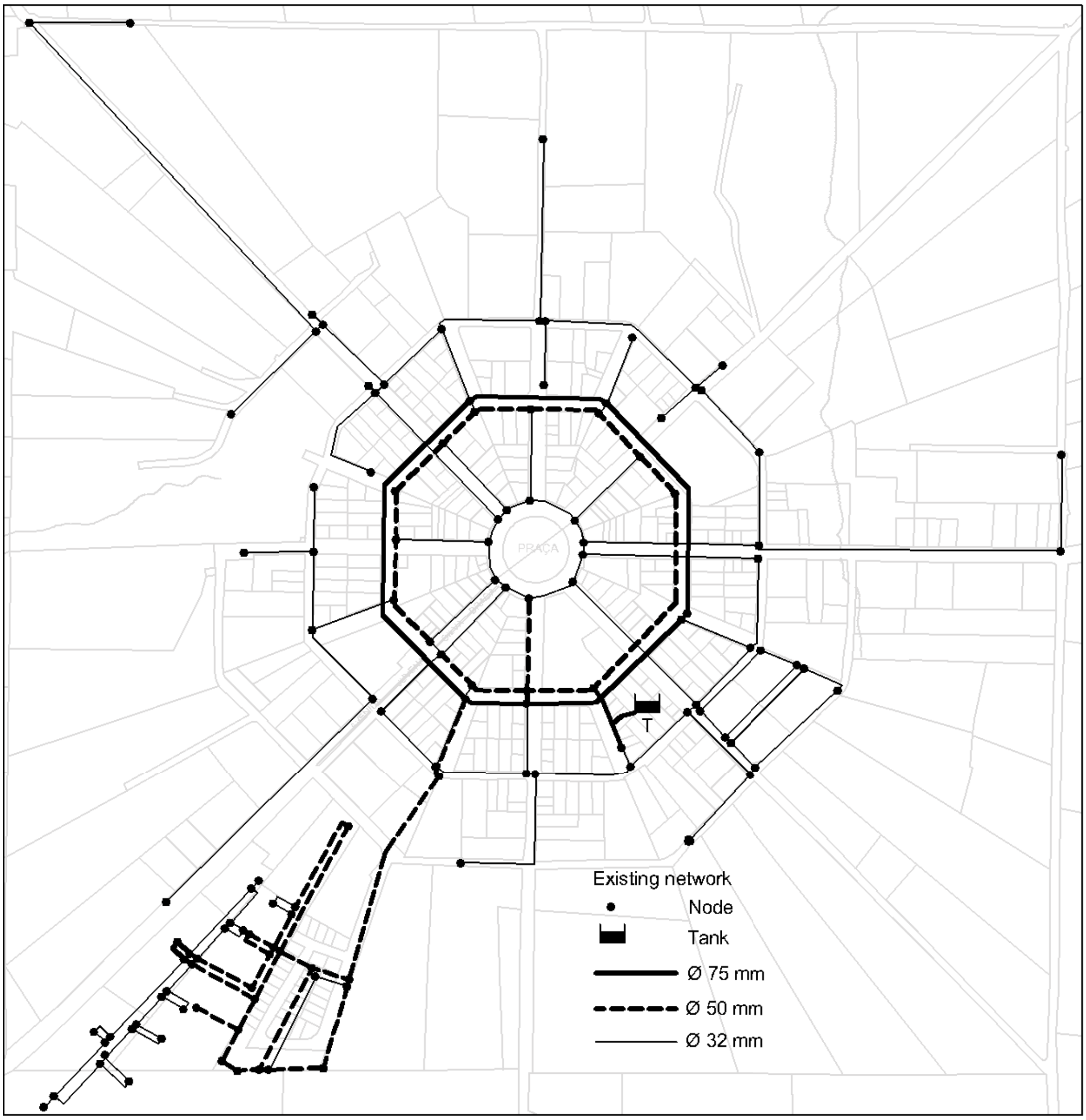

Fig. 9. WDN currently installed in Esperança Nova.

Table 7
Available diameters for the Esperança Nova network

Diameter (mm)	Cost (US $/m)	Roughness coefficient *C* (H-W)
32	13.31	130
50	14.30	130
75	17.82	130
100	22.24	130

The PSO parameters were, for all swarms, inertia weights w_{max} = 0.9 and w_{min} = 0.5, acceleration coefficients $c_1 = c_2 = c_3 = 2$, maximum particle velocity (v_U) = 25 mm, *Penalty* = US $1,000, population of each swarm (N_p) = 200, and maximum number of iterations = 170.

Considering the prices from Table 7, the cost of the existing WDN is US $183,780.79, while the total cost using MSC-PSO optimization is US $175,921.10, with 69 s computation time in a computer using a 1.6 GHz Intel Core I5 CPU, presenting a cost reduction of 4.28%.

Table 8 shows a comparison between the results of optimizations carried out using the models studied in the present work, with a particle initialization strategy. The proposed improved model, MSC-PSO, presented the best result.

The MSC-PSO algorithm presents a result of US $176,001.66 when no strategy is used for the initial distribution of particles from the slave swarms.

As previously commented, there are seven parameters (w, c_1, c_2, c_3, N_p, v_U, and *Penalty*) to be adjusted to achieve the optimal solution. Initially the parameters (w, c_1, c_2, c_3) must be defined. Next, parameters N_p, v_U, and *Penalty* are found

Table 8
Comparison of results optimized by each model

Model	Master swarm	Slave swarm 1	Slave swarm 2
COM-MCPSO	176,001.66	177,176.65	176,001.66
COL-MCPSO	176,139.86	177,176.65	176,140.00
MSC-PSO	175,921.11	175,921.10	175,921.00

Table 9
Definition of the parameters

				Maximum values			Used values		
Network	C_{Pmax} $	N	K	ΔD_{max} mm	N_P	*Penalty* $	v_U mm	N_P	*Penalty* $
Two-source	651,575,025.00	34	26	100	159	16,707,051.92	75	150	15,000,000.00
Esperança Nova	292,395.13	166	131	25	221	1,488.02	25	200	1,000.00

considering N_P limited by Inequality (30), v_U limited by Inequality (31) and *Penalty* limited by Inequality (33).

A value of 2 is attributed for parameters c_1, c_2 and c_3. The dynamic parameter, w, starts at 0.9 and ends at 0.5. Table 9 presents the parameters used in the optimization process and a comparison with maximum values defined by Inequalities (30), (31), and (33) and Eq. (32).

7. Conclusions

In the present work, an optimization model for WDN was presented, with multiple swarms classified as master or slave, with a new algorithm, MSC-PSO, formulated as an MDNLP problem based on the PSO algorithm. Initialization strategies were employed for the particles, assuring a better random distribution throughout the search space. The behavior of the swarms was analyzed during the optimization process, with the proposed model, MSC-PSO, using the experience from the swarms to redirect particles in search of better results. The Free software Epanet 2.0 was used, mainly for calculations of nodal pressures and water velocities in the pipes. The model was applied in two WDN case studies, the two-source network, with better results than those found in other studies, and the Esperança Nova network, with a reduction of 4.28% in the cost of the network relative to what is currently installed. In both studied cases, the initialization strategy for slave swarm particles was proven efficient in the search for better results.

Considering the results achieved in each one of the cases studied, one can conclude that the proposed methodology, using multi-swarm optimization is very effective in solving this type of WDN optimization problem. Besides solving literature problems, it is also able to solve real problems, as the presented in the city of Esperança Nova. It seems to be very promising in solving large-scale problems.

Acknowledgements

The authors gratefully acknowledge the financial support from the National Council for Scientific and Technological Development - CNPq (Brazil) and the Coordination for the Improvement of Higher Education Personnel - Process 88881.171419/2018-01- CAPES (Brazil).

Symbols

AttMax — Maximum number of attempts
c_1, c_2, c_3 — Cognitive and social acceleration coefficients
C_j — Hazen–Williams roughness coefficient for pipe j
cMOPSO — Cluster multi-objective particle swarm optimization
C_P — Objective function of pipe installation costs
C_{PPi} — Value of the penalized objective function for particle i
f_G^M — Value of the objective function for particle Gbest
C_{Pmax} — Maximum total cost of the WDN
C_{Pmin} — Minimum total cost of the WDN
D_j — Diameter of pipe j
D_{max} — Maximum available diameter for the pipes
D_{min} — Minimum available diameter for the pipes
D_{SET} — Set of available diameters for the network
f_i — Best value of the penalized objective function for vector x_i
g — Global best solution vector
g_{max} — Vector where all the pipes have diameter D_{max}
h_f — Head loss
i — Particle i
j — Pipe j
k — Node k
K — Number of demand nodes
L_j — Length of pipe j
MOEA/D — Multi-objective evolutionary algorithm based on decomposition
MOPSO — Multi-objective particle swarm optimization
n — Index of the diameter
N — Total number of pipes
ND — Total number of available diameters
NV — Number of hydraulic violations
N_P — Total number of particles in the swarm
NSGA-II — Non-dominated Sorting Genetic Algorithm II

SPEA2 — Strength Pareto-Evolutionary Algorithm
$\boldsymbol{p}_i$ — Vector of best position of particle i
$\boldsymbol{p}_{i,j}$ — Component j of vector p_i
$\boldsymbol{p}_R$ — Vector of nodal pressures for solution g
$pr(k)$ — Pressure head on node k
$pr_{min}(k)$ — Minimum pressure head on node k
q — Flow rate
r, r_1, r_2, r_3 — Uniformly distributed random numbers in (0,1)
sMOPSO — Sigma-method multi-objective particle swarm optimization
t — Iteration number t
t_{max} — Maximum number of iterations
$\boldsymbol{v}_i$ — Velocity vector of particle i
$v_{i,j}$ — Velocity component j of particle i
$v_w(j)$ — Flow velocity in pipe j
v_U — Maximum velocity of the particle
$\boldsymbol{v}_R$ — Solution vector for water velocities in solution G
WDN — Water distribution network
Penalty — Penalty value
w — Inertia weight
w_{max} — Maximum inertia weight
w_{min} — Minimum inertia weight
$\boldsymbol{x}_i$ — Current position vector of particle i
$x_{i,j}$ — Diameter of pipe j belonging to particle i

References

[1] E. Alperovits, U. Shamir, Design of optimal water distribution systems, Water Resour. Res., 13 (1977) 885–900.

[2] K.E Lansey., L.W. Mays, Optimization model for water distribution system design, J. Hydraul. Eng, 115 (1989) 1401–1418.

[3] O. Fujiwara, D.B. Khang, A two-phase decomposition method for optimal design of looped water distribution networks, Water Resour. Res., 26 (1990) 539–549.

[4] J. Kennedy, R.C. Eberhart, Particle Swarm Optimization, Proc. IEEE International Conference on Neural Networks, Perth, WA, IEEE Xplore, 1995, pp. 1942–1948.

[5] X. Xu, Y.Tang, J. Li, C. Hua, X. Guan, Dynamic multi-swarm particle swarm optimizer with cooperative learning strategy, Appl. Soft Comput., 29 (2015) 169–183.

[6] T. Blackwell, A study of collapse in bare bones particle swarm optimization, IEEE Trans. Evol. Comput., 16 (2012) 354–372.

[7] J.H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, University of Michigan Press, Oxford, England, 1975.

[8] M. Dorigo, Optimization, Learning and Natural Algorithms, Ph.D Thesis, Politecnico di Milano, Italy, 1992.

[9] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science, 220 (1983) 671–680.

[10] Q. Duan, S. Sorooshian, V. Gupta, Effective and efficient global optimization for conceptual rainfall-runoff models, Water Resour. Res., 28 (1992) 1015–1031.

[11] Z. W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search. Simulation, 76 (2001) 60–68.

[12] R. Dawkins, The Selfish Gene, Oxford University Press, Oxford, 1976.

[13] M.M. Eusuff, K.E. Lansey, Optimization of water distribution network design using the shuffled frog leaping algorithm, J. Water Resour. Plann. Manage., 129 (2003) 210–225.

[14] D.E.G. Trigueros, A.N. Módenes, M.A.S.S. Ravagnani, F.R. Espinoza-Quiñones, Reuse water network synthesis by modified PSO approach, Chem. Eng., J. 183 (2012) 198–211.

[15] M.A.S.S. Ravagnani, D.E.G. Trigueros, A.N. Módenos, F.R. Espinoza-Quiñones, Synthesis of reuse water networks by PSO approach, LNCS, 8790 (2014) 282–294.

[16] C.R. Suribabu, T.R. Neelakantan, Design of water distribution networks using particle swarm optimization, Urban Water J. 3 (2006) 111–120.

[17] R. Ezzeldin, B. Djebedjian, T. Saafan, Integer discrete particle swarm optimization of water distribution networks, J. Pipeline Syst. Eng. Pract., 5 (2014) 04013013–1 04013013–11.

[18] D.F. Surco, T.P.B. Vecchi, M.A.S.S. Ravagnani, Optimization of water distribution networks using a modified particle swarm optimization algorithm, Water Sci. Technol. Water Supply, 18 (2017) 660–678.

[19] F. Zheng, A.C. Zecchin, J.P. Newman, H.R. Maier, G.C. Dandy, An adaptive convergence-trajectory controlled ant colony optimization algorithm with application to water distribution system design problems, IEEE Trans. Evol. Comput., 21 (2017) 773–791.

[20] J. Reca, J. Martínez, R. López, A hybrid water distribution networks design optimization method based on a search space reduction approach and a genetic algorithm, Water, 9 (2017) 845.

[21] H.A. El-Ghandour, E. Elbeltagi, Comparison of five evolutionary algorithms for optimization of water distribution networks, J. Comput. Civ. Eng., 32 (2018) 04017066-1 04017066-10.

[22] H. Chen, Y. Zhu, K. Hu, Discrete and continuous optimization based on multi-swarm coevolution, Nat. Comput., 9 (2010) 659–682.

[23] J. Kennedy, R.C. Eberhart, A discrete binary version of the particle swarm algorithm, In 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Orlando, USA, 1997, pp. 4104–4108.

[24] Y. Shi, R. Eberhart, A modified particle swarm optimizer, In 1998 IEEE International Conference on Evolutionary Computation Proceedings, Anchorage, 1998, 69–73.

[25] B. Niu, Y. Zhu, X. He, H. Wu, MCPSO: a multi-swarm cooperative particle swarm optimizer, Appl. Math. Comput., 185 (2007) 1050–1062.

[26] A. Kaveh, K. Laknejadi, A new multi-swarm multi-objective optimization method for structural design, Adv. Eng. Software, 58 (2013) 54–69.

[27] L.A. Rossman, EPANET 2 Users Manual, EPA Enviromental Protection Agency, Cincinnati, 2000.

[28] D.F. Surco, T.P.B. Vecchi, M.A.S.S. Ravagnani, Rehabilitation of water distribution networks using particle swarm optimization, Desal. Water Treat., 106 (2018) 312–329.

[29] M. Shrivatava, V. Prasad, R. Khare, Multi-objective optimization of water distribution system using particle swarm optimization, IOSR-J. Mech. Civ. Eng., 12 (2015) 21–28.

[30] M.S. Kadu, R. Gupta, P.R. Bhave, Optimal design of water networks using a modified genetic algorithm with reduction in search space, J. Water Resour. Plann. Manage, 134 (2008) 147–160.

[31] C.R. Suribabu, Heuristic-based pipe dimensioning model for water distribution networks, J. Pipeline Syst. Eng. Pract., 3 (2012) 115–124.