Advances in Computer Science and Engineering
Volume 6, Number 2, 2011, Pages 159-179
Published Online: April 8, 2011
This paper is available online at http://pphmj.com/journals/acse.htm

INJECTING DIVERSITY INTO PARTICLE SWARM OPTIMIZATION. AN APPLICATION TO WATER DISTRIBUTION SYSTEM DESIGN

MANUEL HERRERA*, JOAQUÍN IZQUIERDO, IDEL MONTALVO and RAFAEL PÉREZ-GARCÍA

Fluing - Instituto de Matemática Multidisciplinar
Universitat Politècnica de València
Camino de Vera, s/n, 46022, Valencia, Spain
e-mail: mahefe@gmmf.upv.es
jizquier@gmmf.upv.es
imontalvo@gmmf.upv.es
rperez@gmmf.upv.es

Abstract

Particle Swarm Optimization is a well established optimization technique. Nevertheless, one of its main drawbacks comes from the fact that it is difficult to maintain acceptable levels of population diversity and to balance local and global searches. In this paper, we describe a discrete variant of PSO with increased diversity whose performance is initially investigated by applying it to a discrete, real-world problem: the design of Water Distribution Systems. Two traditional benchmark problems in the Hydraulic Engineering literature are considered: the Hanoi new water distribution network and the New York Tunnel water supply system. The

Keywords and phrases: water distribution systems, optimal design, evolutionary algorithms, particle swarm optimization, population diversity.

This work is performed with the support of the project IDAWAS, DPI2009-11591, of the Dirección General de Investigación del Ministerio de Educación y Ciencia, and ACOMP/2010/146 of the Conselleria de Educación of the Generalitat Valenciana.

*Corresponding author

Received November 21, 2010

obtained results exhibit considerable improvements regarding both convergence characteristics and the quality of the final solutions. A really important conclusion is that a small representative sample of the algorithm's runs can be used to consistently achieve near optimal results at a much reduced computational cost. This is of paramount importance from an engineering perspective. Finally, to show the scalability of the model, we have applied the algorithm to a real-world water distribution network.

1. Introduction

Particle Swarm Optimization (PSO) is an Evolutionary Algorithm that has shown great potential and good perspective for the solution of various optimization problems [1-10]. The PSO algorithm is a multi-agent optimization system inspired by the social behaviour of a group of migrating birds trying to reach an unknown destination. Like other evolutionary algorithms, PSO does not guarantee the global optimum and has premature convergence to local optima, especially in complex multi-modal search problems. Nevertheless, PSO can be easily implemented, and it is computationally inexpensive, since memory and CPU speed requirements are low. Another advantage is that PSO does not require specific operators, as particles update themselves with internal velocity. PSO algorithms also have memory and receive information only from the best particle in the history, which is a simpler mechanism of information transmission than those used in Genetic Algorithms (GA) or Ant Colony Optimization (ACO), for example. The evolution only looks for the best solution, and particles tend to converge to the best solution quickly. Nevertheless, PSO's main drawback is that it is difficult to maintain acceptable levels of population diversity and to balance local and global searches and hence suboptimal solutions are prematurely obtained [1].

Some evolutionary techniques maintain population diversity by using certain more or less sophisticated operators or parameters, such as the mutation parameter in the case of GAs. Several other mechanisms for forcing diversity can be found in the literature. There are many examples coming from different areas, such as artificial immune systems [11-13] or genetic programming [14, 15]. Regarding PSO, the literature is abundant. In [16], PSO is endowed with an explicit selection mechanism similar to that used in more traditional evolutionary computations. In [17], a hybrid PSO that combines the traditional velocity and position update rules with the ideas of breeding and subpopulations is proposed. Function ‘‘stretching’’ to PSO for the alleviation of the local minima problem is introduced in [18, 19]. Proposed in [20], IPSO is a method in which the population is partitioned into several sub-swarms that are made to evolve based on PSO; the algorithm shuffles the population at periodic stages in the evolution, each time reassigning new sub-swarms to ensure information sharing. The DPSO version [21] introduces random mutations on the particles with small probability, which is hard to determine along with the evolution. The same authors introduce DEPSO in [22], which provides bell-shaped mutations with consensus on the population diversity, while keeping the particle swarm dynamics. Finally, several hybrid techniques using rationales not specifically belonging to PSO have been proposed. The study in [23] employs the technique of initializing the particle swarm optimizer using the nonlinear simplex method to explore the search space more efficiently and to detect better solutions. PSO with a Gaussian mutation is presented in [24]. Inspired by the GAs theory, [25] presents a hybrid evolutionary algorithm by crossing PSO and GAs, which possesses better ability to find the global optimum than that of the standard PSO algorithm. In [26], PSO and simulated annealing are integrated.

The random character typical of evolutionary algorithm’s features adds, in general, a degree of diversity to their genotypes, phenotypes, or individuals integrating the manipulated populations. Nevertheless, in PSO, those random components are generally unable to add a sufficient amount of diversity. Regarding the discrete PSO performance, the authors have detected frequent collisions of particles in the search space, which can also be theoretically justified. This, in fact, caused the effective population size to be lower and the algorithm effectiveness to be consequently impaired. The proposed PSO derivative we propose, which is a theoretically and experimentally founded extension of [27], greatly increases the ability of simultaneous exploration and exploitation and is a kind of affordable action to effectively limit particles’ collisions.

To show its performance, it is applied to two discrete problems in hydraulic engineering. The results show that it remarkably improves the calculation efficiency and is an effective global optimization tool for the design of water supply systems. The scalability of the problem has been demonstrated after applying the model with success to various district metered areas of a Latin-American capital within a joint project with a multi-national water company.

After the introduction, first, the new feature is presented within the framework of PSO. Next, the application to two discrete standard benchmarking problems in Hydraulic Engineering is presented. It includes a comparison with the results obtained by other authors and discussions about the representativeness of samples of algorithm executions and the consistency with which near optimal results are achieved at a much reduced computational cost. Finally, the application to a real-world problem is presented and a conclusion section closes the paper.

2. Injecting Diversity into Standard PSO

The original idea of Kennedy and Eberhart [28] was to simulate the social behaviour of a flock of birds (agents) in their endeavour to reach their unknown destination (fitness function) when flying through the field (search space), for example, in search of the location of food resources. In PSO, each bird of the flock is a potential solution and is referred to as a particle. Initially, a number of particles are randomly generated. Then, particles evolve in terms of their individual and social behaviour, and mutually coordinate their movement towards their destination [29].

The *i*th particle represents a solution of the optimization problem and is characterized by its location in a *D*-dimensional space, where *D* corresponds to the number of variables of the problem. During the process, each particle *i* is associated with three vectors, $X_i = (x_{ij})_{j=1}^{D}$, $P_i = (p_{ij})_{j=1}^{D}$ and $V_i = (v_{ij})_{j=1}^{D}$, representing its current location, the best location it has reached so far, which is updated in each iteration, and its velocity, which enables it to evolve to a new location. Also, in each cycle (iteration), the particle that best fits the objective function is obtained; its location, $G_i = (g_{ij})_{j=1}^{D}$, plays an important role in the calculation of the movement evolution of every other bird.

In a coordinated way, each bird evolves by changing its location

$$X_i = X_i + V_i, \tag{1}$$

with updated new velocity

$$V_i = \omega_i V_i + c_1 r_1 (P_i - X_i) + c_2 r_2 (G - X_i), \tag{2}$$

so that it accelerates towards both its best position, P_i, and the best position obtained so far by any bird in the flock (best global position), *G*. This enables each bird to explore the search space from its new location. The process is repeated until the best bird reaches a certain desired location. The elements in equation (2) are as follows: c_i, $i = 1, 2,$ are the acceleration constants and represent the weighting of the stochastic acceleration terms that pull each particle simultaneously towards its best position and the best global position; these constants are also sometimes referred to as learning rates or factors; r_i, $i = 1, 2,$ are uniform independent pseudo-random numbers between 0 and 1; and ω is an inertia term proposed in [29] that controls the impact of the velocity history into the new velocity and can be suitably adapted during the calculation process. This operator allows a balance between local and global searches and typically decreases with time so that though a global search is initially favoured, the trend shifts towards a local search as the solution process evolves, resulting in less iteration on average to find an optimal solution.

It is worth noting here that, according to the description, the process involves not only intelligent behaviour but also social interaction. This way, birds learn both from their own experience (local search) and from the group experience (global search).

In addition, on each dimension, particle velocities are clamped to minimum and maximum velocities, which are user defined parameters

$$V_{\min} \leq V_j \leq V_{\max}, \tag{3}$$

to control excessive roaming of particles outside the search space. These are very important parameters that are problem dependent. They determine the resolution with which regions between the present position and the target (best so far) positions are searched. If $V_{\max}$ is too big, then particles might fly through good solutions. If $V_{\max}$ is too small, on the other hand, particles may not explore sufficiently beyond locally good regions and could easily be trapped in local optima, unable to move far enough to reach a better position in the problem space.

Increasing diversity

Several researchers have analyzed PSO empirically [21, 30, 31] and theoretically [32-35], and have shown that particles evolve in different oscillating waves and converge quickly, sometimes prematurely, especially for small values of

ω [21]. This is particularly evident for any particle X whose best value P is too close to G. According to (2), those particles become inactive at certain stages of evolution [21, 24, 36].

The analysis can be reduced, without loss of generality, to the one-dimensional case:

$$v = \omega v + c_1 r_1 (p - x) + c_2 r_2 (g - x). \tag{4}$$

Equation (4) shows that if v becomes small, it will not be able to take on large values again if $|p - x|$ and $|g - x|$ are both small, too. This fact is especially critical in the discrete case, where $|p - x|$ and $|g - x|$ can be effectively zero. This will represent a loss of exploration capability for the particle in some generations, since v will be damped quickly with the ratio ω. Such a circumstance can occur even at early stages for the best particle, for which $|p - x|$ and $|g - x|$ are zero. In the long run, however, it is expected that $p = g$, for a number of particles as all the particles in the population "agree" upon a single best point which becomes the unique attractor; this represents a clear lost of diversity.

Effectively, by writing $c = c_1 r_1 + c_2 r_2$, the equivalent attraction point h can be given by

$$h = \frac{c_1 r_1}{c} p + \frac{c_2 r_2}{c} g. \tag{5}$$

As a result, equation (4) can be re-written as

$$v = \omega v + c(h - x). \tag{6}$$

If p and g are different, h changes from iteration to iteration even if no better solutions are discovered, that is to say, even if p and g remain constants; as a consequence, v will change suitably. However, if p equals g, then equation (5) implies that, irrespective of the generated random numbers, $h = p = g$. With $|h - x|$ small, which typically occurs in the latter stage of the evolution process, then v, according to (6), is damped with small ω values and as a result, the particle is clearly inactive and has no chance to improve.

In effect, after conducting a specific study (not included in this work) on the discrete PSO performance, the authors detected frequent collisions of birds in the search space, especially with the leader [27]. This, in fact, caused the effective population size to be (much) lower; consequently, the algorithm effectiveness was impaired. This led us to try to devise some kind of affordable action to effectively limit bird collisions. To check all of the birds for all possible collisions was deemed extravagant and unnecessary. Therefore, only a few of the best birds were selected to check collision, and a new bird was randomly regenerated if a collision occurred. The random regeneration of the many birds that tended to collide with the best birds was shown to avoid premature convergence, as it prevented clone populations from dominating the search. The inclusion of this procedure into the discrete PSO produces greatly increased diversity and, according to the results shown in the next paragraph, improved convergence characteristics and the quality of the final solutions.

The modified algorithm can be given by the following pseudo-code, with k as iteration number.

Let $k = 1$.

Generate a random population of M particles: $\{X_i(k)\}_{i=1}^{M}$.

Evaluate the fitness of the particles.

Record the local best locations $\{P_i(k)\}_{i=1}^{M}$.

Record the global best location, $G(k)$, and the list of the m best particles to check collisions.

While (not termination-condition) do

- Determine the inertia parameter $\omega(k)$

- Begin cycle from 1 to number of particles M

  Start

    Calculate new velocity, $V_i(k+1)$, for particle i according to (2), and take its integer part (for discrete optimization)

    Update position, $X_i(k+1)$, of particle i according to (1)

    Calculate fitness function for particle i

If particle i has better fitness value than the fitness value of the best particle in history, then set particle i as the new best particle in history and update the list of the m best particles

If particle i is not currently one of the m best particles but coincides with one of the selected m best particles, then re-generate particle i randomly

End

- Let $k = k + 1$.

Show the solution given by the best particle

The parameters used by this algorithm have been selected after preliminary tuning experiments following a number of suggestions [4, 5, 29, 37]: $c_1 = 3$, $c_2 = 2$, $\omega = 0.5 + \frac{1}{2(\ln k + 1)}$, $V_{\max} = 50\%$ of variable range, $V_{\min} = -V_{\max}$; Number of particles $(\text{population size}) = 100$.

Different termination conditions may be stated [37]. In this paper, the process is stopped if no improvement in the solution had been obtained after 800 iterations.

The performance of the approach introduced herein to avoid collisions with the best particles (different values of m have been tried), can be observed from the results obtained for the two benchmark problems studied in the next paragraph.

3. WDS Benchmarking Problems and a Real-world Case Study

WDS design is a wide problem in hydraulic engineering that consists in determining the values of all involved variables in such a way that the investment and maintenance costs of the system are minimal, subject to a number of constraints [38]. A general strategy for solving the optimal design problem of a WDS involves the balancing of several factors: finding the lowest costs for layout and sizing using new components, reusing or substituting existing components, creating a working system configuration that fulfils all water demands, adhering to the design constraints, and guaranteeing a certain degree of reliability for the system [39, 40]. The benchmark cases we address here have been used traditionally in the literature and are standard examples used to demonstrate the application of a wide range of tests and analyses. The fitness function that has traditionally been used only takes pipeline costs into account. Nevertheless, a generalization to broader classes of fitness functions is straightforward. Hence, in order to facilitate comparisons with results obtained by other authors, we use the following objective function to estimate the costs:

$$F(D) = \sum_{i=1}^{P} C(D_i) \cdot L_i, \tag{7}$$

where P is the number of pipes in the network, $D = (D_i)$ is the vector of pipe diameters (which is P-dimensional and its components belong to a discrete set of commercially available diameters), $C(D_i)$ is the unit cost of diameter D_i, and L_i is the length of the ith pipe. It has to be noted that C is a non-linear function of diameter. Also, in order to restrict ourselves to the same rules used in the literature to deal with the benchmark problems, only three kinds of constraints are considered here: continuity equations, energy equations (strongly nonlinear), and lack of satisfaction of minimum pressures at demand nodes. As a consequence, the total cost of the network is considered as the sum of the network cost (7) and a penalty cost, defined as

$$F = \sum_{i=1}^{P} C(D_i) \cdot L_i + \sum_{j=1}^{K} p_j \cdot v_j^2, \tag{8}$$

where K is the number of constraints, v_j is the jth constraint violation, and p_j represents the penalty parameter corresponding to constraint j with a large value to ensure that infeasible solutions will have a cost greater than any feasible solution.

The problems faced in the optimal design of WDSs are great. Furthermore, this simple variant for the design of a water supply system is NP-hard. For instance, one of the networks considered in this paper, with 21 pipes and 15 potential commercial pipe diameters, has 16^{21} possible pipe diameter combinations (including a null option) that constitute the search space of the problem. This modest network would require a considerable amount of time for an exhaustive search algorithm to navigate the entire search space of almost $2 \cdot 10^{25}$ potential solutions.

Hanoi water supply system

The Hanoi pipe network has been considered several times in the literature [41-45]. This network consists of a single fixed head source at an elevation of 100m, 34 pipes and 31 demand nodes organized in three loops, and two ramified branches. The problem is to find the diameters (from a set of six commercially available diameters) for the 34 pipes such that the total cost of the network is minimal and the pressure at each node of consumption is at least 30m. The complete setting can be found in [46].

New York tunnel supply system

The second case is the New York Tunnel water supply network, which, similar to the Hanoi water distribution problem, has been studied extensively by various researchers [42, 43, 47]. A complete detailed description of the case can be seen in [48]. The system has a fixed head reservoir, 21 tunnels, and 19 nodes. The objective of the New York Tunnel (NYT) problem is to determine the most economically effective design for adding to the existing system of tunnels that constituted the primary water distribution system of the city of New York. Because of age and increased demands, the existing gravity flow tunnels were found to be inadequate to meet the pressure requirements for the projected consumption level. The construction of additional gravity flow tunnels parallel to the existing ones is considered. All 21 tunnels are considered for duplication. There are 15 available discrete diameters and one extra possible decision, which is the “do nothing” option.

Sample representativeness

For the WDSs under consideration, designs were optimized 100 times initially, then 1000 times, and finally, 2000 times. This way, three independent samples of different size were obtained. The final minimal fitness values were analyzed using the Kruskal-Wallis test (K-W test) [49] for nonparametric analysis. The test statistic T is defined as:

$$T = \frac{1}{S^2}\left(\sum_{i=1}^{k}\frac{R_i^2}{n_i} - N\,\frac{(N+1)^2}{4}\right), \tag{9}$$

being

$$S^2 = \frac{1}{N-1}\left(\sum_{\text{all ranks}} R(X_{ij})^2 - N\,\frac{(N+1)^2}{4}\right), \tag{10}$$

where k is the number of groups (three, in this case), N is the total number of runs (here 3100), n_i is the sample size for group i (100, 1000, and 2000, respectively), R_i is the sum of the ranks for group i, and $R(X_{ij})$ is the rank of all samples.

The null hypothesis (no difference between statistical measures for the samples) was not rejected after obtaining p-values of 0.703 and 0.6448 for the Hanoi and New York problems, respectively. Box plots of the distribution of the fitness values for both cases are given in Figure 1.

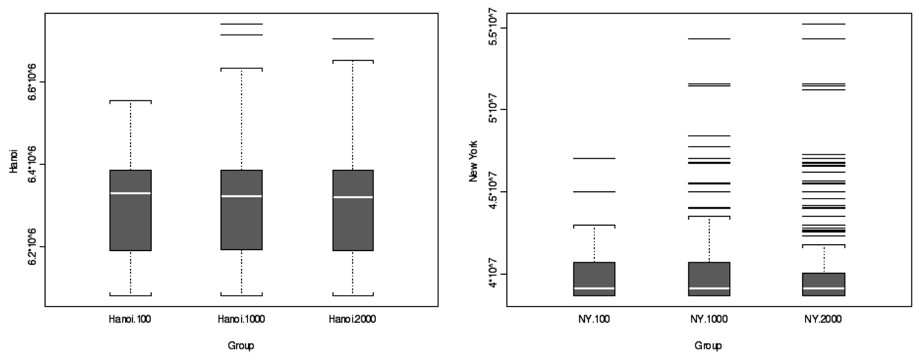

Figure 1. Box plots of the distribution of the fitness values for each sample on the two WDSs considered.

As a consequence, the samples of the first 100 runs will be considered hereafter.

Obtained results and different comparisons

The best solution found after the inclusion of the re-generation option with $m = 1$ is shown in Table 1 for both networks together with other best solutions found in the literature.

Table 1. Optimal design cost (million dollars) for the Hanoi and the New York networks

Hanoi network			**New York tunnel system**		
Reference	Method	Cost	Reference	Method	Cost
[42]	GA	6.093	[42]	GA	38.64
[46]	GA	6.182	[48]	GA	38.8
[43]	GA	6.195	[47]	ACO	38.64
[45]	ACO	6.367	[43]	GA	40.42
[9]	PSO	6.133	[9]	PSO	38.64
This work	PSO	6.081	This work	PSO	38.64

The final minimal fitness values for the 100 runs were compared with the ones obtained in [9] by standard discrete PSO (that is, without performing the re-generation-on-collision feature herein described) by using the Mann-Whitney U-test [49] for nonparametric analysis. The statistic U is essentially the same as the T of Kruskal-Wallis, but is used for only two groups. The null hypothesis (no difference between central statistical measures for the PSO versions) was rejected after obtaining a p-value of 0. Box plots of the distribution of fitness values for both versions of PSO are given in Figure 2 for the two WDSs considered here. It can be observed that the inclusion of the re-generation option (group 2), clearly outperforms the standard PSO (group 1).

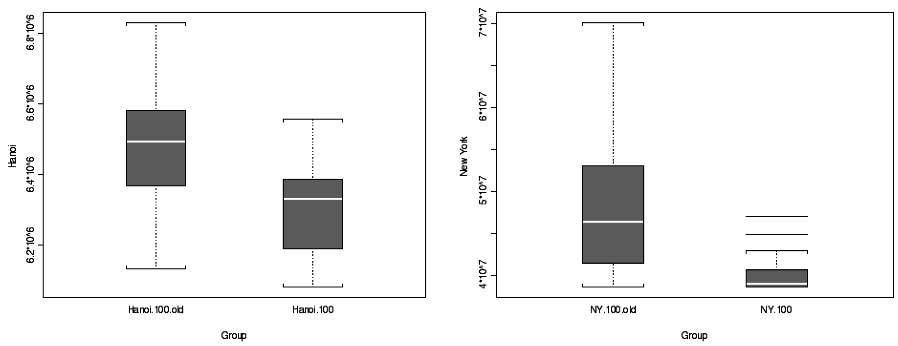

Figure 2. Box plots of the distribution of the fitness values for the conventional PSO (1) and the derivative presented here (2) for the Hanoi and New York systems.

It is also worthwhile to observe that the average cost was 6.297 million dollars, only 3.56% higher than the best known solution, for the Hanoi system. In the case of the New York system, the result is 39.738 million dollars, or 2.91% over the best known solution. These figures do not need further explanation regarding the quality of the algorithm described in this paper.

Cost and probability of suboptimal designs

Many 'best' solutions found in the literature regarding these two problems have been obtained after never-ending computer dedication and, as a consequence, with a huge computational effort. This is a significant drawback for the application of evolutionary algorithms to the solution of 'real-world' problems where cost and time constraints prohibit repeated runs of the algorithm and evaluations of the network.

The average number of generations needed to obtain the best solution for the Hanoi system is 700, with 105 being the minimum number of generations to obtain the best solution. Regarding the New York system, these figures are 230 generations for the best solutions and 16 for the minimum number of generations to obtain the best solution. These figures make it clear that the algorithm is quite inexpensive. For example, the solution for the New York system of 45.73 million dollars obtained in [50] was found after 80,000 evaluation trials. In [51], two sets of solutions based on different hydraulic coefficients are reported. The number of generations allowed for their Genetic Algorithm was 10,000. Genetic Algorithms, where the population size of 20 and the maximum number of generations was set to 2,000, were used in [52]. This improvement in the efficiency is mainly due to the self-adaptive fitness formulation for evolutionary constraint optimization they propose.

Using the obtained results, the probability for a single run of obtaining a solution differing by less than a certain percent from the best known solution was obtained. Different conclusion can be obtained. For example, by running the re-generation PSO algorithm (described in this paper) only once, the probability of obtaining the best known solution is almost 30% for the NYT system and 5% for the Hanoi system. However, from a practical point of view in which 'early' almost-optimal solutions are much better than 'too late' best solutions, other pieces of important information are also outstanding. For example, one single run of our algorithm guarantees a solution that is less expensive than 5.5% of the best known solution with a probability of 86%, for both studied problems. Additionally, there is an almost complete guarantee that in only one single run of the algorithm, we will obtain a solution with a cost under 1.1 times the best known solution cost.

These probabilities can be shown to be strongly problem dependent. As a consequence, these results cannot be directly extrapolated to other problems. However, it is seen that the algorithm presented in this paper was able to find the optimum or near-optimum solution with considerably low computational effort.

Application to a real-world water distribution system

In the real-world case study [53], we consider here the minimum pressure allowed is 15m and the available commercial diameters are given in Table 2. This table also includes the Hazen-Williams coefficient, *C*, used in the hydraulic model, and the unit cost of the pipes of available diameters.

Table 2. Commercially available diameters

Diameter (mm)	CH-W	Cost ($units)
100	140	117.14
150	140	145.16
200	140	191.42
250	140	241.09
300	140	333.16

The problem is solved by using two fitness functions, namely, *F*, defined in (8), and *FR*, defined as

$$FR = \sum_{i=1}^{P} C(D_i) \cdot L_i + \sum_{j=1}^{K} p_j \cdot v_j^2 + \sum_{i=1}^{P} w_i \cdot L_i \cdot D_i^{-u}, \tag{11}$$

which, following [54], adds reliability to the design from an economic point of view, by considering the costs of the water not delivered due to problems in the system. Here, w_i is a coefficient associated to each pipe, of the form $a \cdot t_f \cdot (c_f + c_a \cdot V_f)$; $a \cdot L \cdot D^{-u}$ gives the number of expected failures per year of one pipe, as a function of diameter, D_i, and length, l_i, (*a* and *u* are known constants); t_f is the average number of days required to repair the pipe; c_f is the daily repairing average cost; c_a is the average cost of the water supplied to affected consumers, in monetary units per unit volume; and $V_f = 86400 \cdot Q_{\text{break}}$ is the daily volume of water that should be supplied to the affected consumer due to the loss of water of Q_{break} in cubic meters per second.

The scenarios considered in (11) follow the approach of ‘breaking’ by turn all the pipes of a specific design to check if all the constraints are fulfilled by the design subjected to this circumstance. If the test is negative, then the design is suitably penalized. This way, designs will develop increasing reliability. To undergo those tests, the system must be analyzed for any of those specific ‘breakages’. Taking into account, the expensiveness of the algorithm presented in this paper, all these runs have been performed in a much reduced time, and performing a very low number of function evaluations.

The layout of the network can be seen in Figure 3. For the understanding of the results, a code for colors has been used. Regarding pipes, blue, green, yellow and red colors represent 100, 150, 200 and 250mm pipes, respectively. Regarding nodes, dark blue represents pressure above 15m; light blue, between 14 and 15m; green, between 12 and 14m; yellow, between 10 and 12m; and, finally, nodes having a pressure under 10m are represented in red.

This network, which is fed by a tank, has 294 lines amounting to 18.337km of pipes and 240 nodes consuming 81.53l/s in total. Figure 3 (left) presents the solution obtained by using *FR* (including reliability). This solution is only a mere 3.65% more expensive than the one obtained by using *F* (no reliability consideration). The diameters for this case can be observed in Figure 3 (right). Table 3 presents a comparison between the initial investment costs for both solutions.

The effect of closing the pipe pointed by the arrow can be observed in Figure 3 (right) for the solution without reliability. It shows the great impact produced by a closed pipe. It can be shown that this does not happen for the more reliable design obtained from *FR* (left), no matter which pipe is out of service.

Table 3. Comparison between costs for both solutions

Diameter (mm)	Without reliability		With reliability	
	Length (m)	Cost ($units)	Length (m)	Cost ($units)
100	17731.10	2077021.41	15822.31	1853425.63
150	606.39	88023.28	2077.69	301597.04
200	0.00	0.00	328.79	62937.56
250	0.00	0.00	108.70	26206.24
300	0.00	0.00	0.00	0.00
Total cost ($units)	2165044.69		2244166.47	

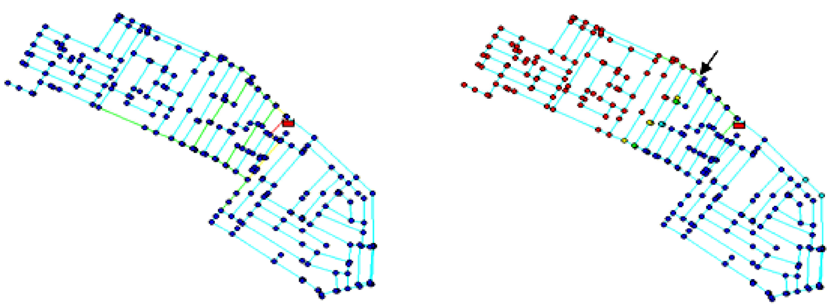

Figure 3. Designs with and without reliability consideration.

Table 4 shows the pressure values at the most critical nodes when pipe indicated in Figure 3 is closed.

Table 4. Pressure at most critical nodes

Node	Without reliability	With reliability
ID	Pressure (m)	Pressure (m)
1111345	18.81	3.21
1102108	18.77	3.24
1112395	18.85	3.24
1106799	18.90	3.33
1098891	18.75	3.35
1103578	19.19	3.59
1113234	19.21	3.59
1107987	19.26	3.64
1100151	19.04	3.65
1099662	19.14	3.75
1094132	19.33	4.02
1062222	19.23	4.52
1049416	19.30	4.89
1047213	19.32	4.97

4. Conclusions

In this paper, discrete PSO is endowed with a re-generation-on-collision formulation, which greatly improves the performance of standard discrete PSO for water systems design. The performance of the formulation introduced in this paper has been illustrated by application to two benchmark networks, and the results have been compared with those obtained using other evolutionary algorithms. Comparison of the results shows that this formulation is able to find optimum or near-optimum solutions much more efficiently, with considerably less computational effort. The improved performance of the algorithm described here is due to the richer population diversity it introduces. The main advantages of the method are that it does not require sophisticated operators or parameters and is thus simpler than other evolutionary techniques; it does not need initial feasible particles, nor do the re-generated particles need to be feasible; and finally, it is robust in handling diverse fitness functions and different constraints. Furthermore, having a low number of generations is a major advantage in real water distribution systems where cost and time constraints prohibit repeated runs of the algorithm and hydraulic evaluations. From the studied benchmark problems, it can be inferred that obtaining ‘good’ solutions with the proposed algorithm is quite inexpensive. Therefore, the algorithm is desirable when one aims at quickly obtaining good solutions that are not necessarily very close to the optimum.

Investigating new abilities, as the one introduced in this paper, of these particles to decide, as a group, how to move inside the search space, and change their behaviour during the search processes, as well as finding very good solutions in a relatively short period of time, constitutes an open-door environment that could be perfectly exploited to address multi-objective formulations regarding optimization problems in different fields.

References

[1] Y. Dong, J. Tang, B. Xu and D. Wang, An application of swarm optimization to nonlinear programming, Computers Mathematics with Applications 49(11-12) (2005), 1655-1668.

[2] M. Herrera, J. Izquierdo, I. Montalvo, J. García-Armengol and J. V. Roig, Identification of surgical practice patterns using evolutionary cluster analysis. Mathematical and Computer Modelling Doi: 10.1016/j.mcm.2008.12.026. 2009.

[3] S. Janson, D. Merkle and M. Middendorf, Molecular docking with multi-objective particle swarm optimization, Appl. Soft Comput. 8(1) (2008), 666-675.

[4] Y. X. Jin, H. Z. Cheng, J. Y. Yan and L. Zhang, New discrete method for particle swarm optimization and its application in transmission network expansion planning, Electric Power Systems Research 77(3-4) (2007), 227-233.

[5] C. J. Liao, C. T. Tseng and P. Luarn, A discrete version of particle swarm optimization for flowshop scheduling problems, Computers and Operations Research 34(10) (2007), 3099-3111.

[6] J. Izquierdo, I. Montalvo, R. Pérez and V. S. Fuertes, Design optimization of wastewater collection networks by PSO, Computers and Mathematics with Applications 56(3) (2008a), 777-784.

[7] J. Izquierdo, R. Minciardi, I. Montalvo, M. Robba and M. Tavera, Particle swarm optimization for the biomass supply chain strategic planning, Proceedings of iEMSs 2008: International Congress on Environmental Modelling and Software, Barcelona, Spain, 2008b, pp. 1272-1280.

[8] J. Izquierdo, I. Montalvo, R. Pérez and V. S. Fuertes, Forecasting pedestrian evacuation times by using swarm intelligence, Physica A: Statistical Mechanics and its Applications 388(7) (2009), 1213-1220.

[9] I. Montalvo, J. Izquierdo, R. Pérez and M. M. Tung, Particle swarm optimization applied to the design of water supply systems, Computers & Mathematics with Applications 56(3) (2008a), 769-776.

[10] Q. K. Pan, F. Tasgetiren and Y. C. Liang, A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem, Computers and Operations Research 35(9) (2008), 2807-2839.

[11] R. E. Smith, S. Forrest and A. S. Perelson, Searching for diverse, cooperative populations with genetic algorithms, Evolution Computation 1(2) (1993), 127-149.

[12] P. Hajela and J. Lee, Constrained genetic search via schema adaptation, An immune network solution, Struct. and Mult. Optim. 12(1) (1996), 11-15.

[13] J. Yoo and P. Hajela, Immune network simulations in multicriterion design, Structural and Multidisciplinary Optimization 18(2-3) (1999), 85-94.

[14] M. Tomassini, L. Vanneschi, F. Fernandez and G. Galeano, Diversity in Multipopulation Genetic Programming, E. Cantu-Paz, J. A. Foster, K. Deb, D. Davis, R. Roy, U. M. O'Reilly, H. G Beyer Beyer, R. Standish, G. Kendall, S. Wilson, M. Harman, J. Wegener, D. Dasgupta, M. A. Potter, A. C. Schultz, K. Dowsland, N. Jonoska and J. Miller, editors, Genetic and Evolutionary Computation -- GECCO-2003, Springer-Verlag, pp. 1812-1813, Chicago, 2003.

[15] G. Folino, C. Pizzuti, G. Spezzano, L. Vanneschi and M. Tomassini, Diversity analysis in cellular and multipopulation genetic programming, R. Sarker, R. Reynolds, H. Abbass, K. Chen Tan, B. McKay, D. Essam and T. Gedeon, editors, Proceedings of the 2003 Congress on Evolutionary Computation CEC2003, IEEE Press, pp. 305-311, Canberra, 2003.

[16] P. J. Angeline, Using selection to improve particle swarm optimization, Proc. of the IEEE Congress on Evolutionary Computation 1998, Anchorage, Alaska, USA, 1998b, pp. 84-89.

[17] M. Løvbjerg, T. K. Rasmussen and T. Krink, Hybrid particle swarm optimiser with breeding and subpopulations, Proceedings of the Third Genetic and Evolutionary Computation Conference, 2001.

[18] K. E. Parsopoulos, V. P. Plagianakos, G. D. Magoulas and M. N. Vrahatis, Stretching technique for obtaining global minimizers through particle swarm optimization, Proceedings on Particle Swarm Optimization Workshop, Indianapolis (IN), USA, 2001a.

[19] K. E. Parsopoulos, V. P. Plagianakos, G. D. Magoulas and M. N. Vrahatis, Improving particle swarm optimizer by function “stretching”, Advances in Convex Analysis and Global Optimization Nonconvex Optimization and Applications series, The Netherlands, Kluwer Academic Publishers, The Netherlands, 2001b, pp. 445-457.

[20] Y. Jiang, T. Hu, C. Huang and X. Wu, An improved particle swarm optimization algorithm, Applied Mathematics and Computation 193(1) (2007b), 231-239.

[21] X. F. Xie, W. J. Zhang and Z. L. Yang, A dissipative particle swarm optimization, Congress on Evolutionary Computation, Honolulu, Hawaii, USA, 2002a.

[22] W. J. Zhang and X. F. Xie, DEPSO: hybrid particle swarm with differential evolution operator, IEEE International Conf. on Systems, Man and Cybernetics 4 (2003), 3816-3821.

[23] K. E. Parsopoulos and M. N. Vrahatis, Initializing the particle swarm optimizer using the nonlinear simplex method, Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation, WSEAS Press, 2002, pp. 216-221.

[24] N. Higashi and H. Iba, Particle swarm optimization with gaussian mutation, Proc. of the IEEE Swarm Intelligence Symp. 2003, Indianapolis, Indiana, USA, 2003, pp. 72-79.

[25] X. H. Shi, Y. H. Lu, C. G. Zhou, H. P. Lee, W. Z. Lin and Y. C. Liang, Hybrid evolutionary algorithms based on PSO and GA, Proc. of IEEE Congress on Evolutionary Computation 2003, Canbella, Australia, 2003, pp. 2393-2399.

[26] X. H. Wang and J. J. Li, Hybrid particle swarm optimization with simulated annealing, Proc. of the Third International Conf. on Machine Learning and Cybernetics, Shanghai, 2004, pp. 2402-2405.

[27] I. Montalvo, J. Izquierdo, R. Pérez and P. L. Iglesias, A diversity-enriched variant of discrete PSO applied to the design of water distribution networks, Engineering Optimization 40(7) (2008b), 655-668.

[28] J. Kennedy and R. C. Eberhart, Particle swarm optimization, Proc. of the IEEE International Conf. on Neural Networks, Piscataway, NJ, 1995, pp. 1942-1948.

[29] Y. Shi and R. C. Eberhart, A modified particle swarm optimizer IEEE international conference on evolutionary computation, Piscataway, NJ, 1998, pp. 69-73.

[30] P. J. Angeline, Evolutionary optimization versus particle swarm optimization: philosophy and performance difference, Proc. of the Seventh Annual Conf. on Evolutionary Programming, 1998a, pp. 601-610.

[31] J. Kennedy, Bare bones particle swarms, Proc. of the IEEE Swarm Intelligence Symp., Indianapolis, Indiana, USA, 2003, pp. 80-87.

[32] M. Clerc and J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, IEEE Transactions on Evolutionary Computation 6(1) (2002), 58-73.

[33] M. Jiang, Y. P. Luo and S. Y. Yang, Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm, Inf. Process. Lett. 102(1) (2007a), 8-16.

[34] I. C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, Inf. Process. Lett. 85(6) (2003), 317-325.

[35] K. Yasuda, A. Ide and N. Iwasaki, Adaptive particle swarm optimization, Proceedings of IEEE International Conference on Systems, Man and Cybernetics, 2003, pp. 1554-1559.

[36] X. F. Xie, W. J. Zhang and Z. L. Yang, Adaptive particle swarm optimization on individual level, Proc. of the 6th International Conf. on Signal Processing, Beijing, China, 2002b, pp. 1215-1218.

[37] X. H. Shi, Y. C. Liang, H. P. Lee, C. Lu and Q. X. Wang, Particle swarm optimization-based algorithms for TSP and generalized TSP, Information Processing Letters 103(5) (2007), 169-176.

[38] J. Izquierdo, R. Pérez and P. L. Iglesias, Mathematical models and methods in the water industry, Mathematical and Computer Modelling 39 (2004), 1353-1374.

[39] I. C. Goulter and F. Bouchart, Reliability-constrained pipe network model, Journal of Hydraulic Engineering-Asce 116(2) (1990), 211-229.

[40] I. C. Goulter and A. V. Coals, Quantitative approaches to reliability assessment in pipe networks, Journal of Transportation Engineering 112(3) (1986), 287-301.

[41] M. C Cunha and J. Sousa, Water distribution network design optimization: simulated annealing approach, Journal of Water Resources Planning and Management 125(4) (1999), 215-221.

[42] A. S. Matías, Diseño de redes de distribución de agua contemplando la fiabilidad, mediante Algoritmos Genéticos, Departamento de Ingeniería Hidráulica y Medio Ambiente, Universidad Politécnica de Valencia, Doctoral dissertation, 2003.

[43] D. A. Savic and G. A. Walters, Genetic operators and constraint handling for pipe network optimization, Evolutionary Computing, AISB Workshop, 1995, pp. 154-165.

[44] A. C. Zecchin, Max-min ant system applied to water distribution system optimization, MODSIM 2003, International Congress on Modelling and Simulation, 2003.

[45] A. C. Zecchin, A. R. Simpson, H. R. Maier and J. B. Nixon, Parametric study for an ant algorithm applied to water distribution system optimization, IEEE Transactions on Evolutionary Computation 9(2) (2005), 175-191.

[46] Z. Y. Wu and A. R. Simpson, Competent genetic-evolutionary optimization of water distribution systems, Journal of Computing in Civil Engineering 15(2) (2001), 89-101.

[47] 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, Journal of Water Resources Planning and Management 129(3) (2003), 200-209.

[48] G. C. Dandy, A. R. Simpson and L. J. Murphy, An improved genetic algorithm for pipe network optimization, Water Resources Research 32(2) (1996), 449-458.

[49] M. Hollander and D. A. Wolfe, Nonparametric Statistical Methods, 2nd ed., John Wiley & Sons, 1999.

[50] I. Lippai, J. P. Heaney and M. Laguna, Robust water system design with commercial intelligent search optimizers, J. Comput. Civ. Eng. 13(3) (1999), 135-143.

[51] D. A. Savic and G. A. Walters, Genetic algorithms for least-cost design of water distribution networks, Journal of Water Resources Planning and Management 123(2) (1997), 67-77.

[52] R. Farmani, J. A. Wright, D. A. Savic and G. A. Walters, Self-adaptive fitness formulation for evolutionary constrained optimization of water systems, Journal of Computing in Civil Engineering 19(2) (2005), 212-216.

[53] I. Montalvo, Optimal design of water distribution systems by agent swarm optimization, Departamento de Ingeniería Hidráulica y Medio Ambiente, Universidad Politécnica de Valencia, Doctoral dissertation 2010.

[54] J. B. Martínez, Quantifying the economy of water supply looped networks, Journal of Hydraulic Engineering, ASCE 133(1) (2007), 88-97.