CHAPTER 5

Particle swarm optimisation

J. Izquierdo[1], I. Montalvo[2], R. Pérez-García[1] & M. Herrera[1]
[1]*FluIng-IMM, Universitat Politècnica de València, Valencia, Spain.*
[2]*Ingeniousware, Karlsruhe, Germany.*

Abstract

Due to the various limitations shown by classical techniques of optimisation when dealing with many real-world engineering problems, a number of paradigms have been produced over the last 20 years that claim to be better suited to providing useful engineering solutions for these types of problems. One of the evolutionary algorithms that has shown great potential for solving various optimisation problems is (PSO), which stands for *particle swarm optimisation*. PSO is a multi-agent optimisation system inspired by the social behaviour of a group of migrating birds trying to reach an unknown destination. In this chapter, the basics of PSO are provided together with some modifications that considerably improve the performance of the standard algorithm. This variant features: a mixed continuous-discrete variant of PSO; a mechanism to enrich population diversity; and a self-adapting feature that spares engineers the task of parameter selection and fine-tuning. This variant can find solutions efficiently for various optimisation problems in the water field. In addition, we provide the necessary details for this algorithm to work with multi-objective optimisation problems. Multiobjective optimisation is essential in many decision-making processes. From a practical standpoint, the development of a multi-objective optimisation process enables the combination of economic, engineering, and policy viewpoints when searching for a solution to a problem. Finally, we reference a number of applications of these approaches in various areas of hydrology, hydraulics, and water resource management, and present the details of the complex problem of designing water distribution networks, together with the solutions to two real-world case studies.

Keywords: design, multi-objective optimisation, particle swarm optimisation, reliability, water distribution network.

WIT Transactions on State of the Art in Science and Engineering, Vol 56, © 2012 WIT Press
www.witpress.com, ISSN 1755-8336 (on-line)
doi:10.2495/978-1-84564-664-6/05

1 Introduction

Optimisation in engineering, in general, and in water systems, in particular, is crucial for many reasons. Design is necessary to implement new configurations, improve existing systems, continue satisfying various needs, and expand systems to meet new conditions. Taking into account the uncertainty of data (especially in existing configurations), it is often necessary to solve difficult inverse problems where optimisation techniques are also of key importance. There are many examples of important industrial problems in the water field because there is a great deal of interest in mechanisms for managing sustainable water resources at a reasonable cost.

Optimisation is a typically constrained nonlinear search problem involving both continuous and discrete variables. The problem in hand is frequently a mixed continuous and discrete constrained nonlinear optimisation problem that is often highly dimensional and multimodal. Highly dimensional means that many decision variables influence the solution. Multimodal expresses the idea that there are many local optima in the search space. There is no single search algorithm for solving many real-world optimisation problems without compromising solution accuracy, computational efficiency, and problem completeness.

Classical methods of optimisation involve the use of gradients or higher order derivatives of the fitness function. But these methods are not well suited for many real-world problems since they cannot process inaccurate, noisy, discrete, and complex data. Robust methods of optimisation are often required to generate suitable results.

Over the last 20 years, many researchers, including those in the water field, have embarked on the implementation of a range of *evolutionary algorithms*: genetic algorithms, ant colony optimisation, PSO, simulated annealing, shuffled complex evolution, harmony search, and memetic algorithms, among many others. Some of these techniques are the objective of other chapters in this book.

This chapter presents the principles of PSO, which is one of the evolutionary algorithms that has shown great potential for the solution of various optimisation problems. The PSO algorithm was first developed by Kennedy and Eberhart [1] and is a multi-agent optimisation system inspired by the social behaviour of a group of migrating birds trying to reach an unknown destination. In addition to one of the versions of standard PSO, we describe several modifications (developed by the authors of this chapter) that considerably improve the performance of the standard algorithm when used to find solutions to various optimisation problems in the water field [2–6].

The remainder of this chapter is presented as follows. Firstly, PSO is concisely presented. Secondly, the proposed adaptations, namely, a mixed continuous-discrete variant of PSO, a mechanism to enrich diversity that greatly improves the performance of PSO, and a self-adapting characteristic that avoids the task of parameter selection, are then described. By developing the necessary elements, we then provide an adaptation of this algorithm for multi-objective optimisation problems. Finally, we show the results of specific applications to selected case-studies

regarding a very well-known urban water problem, namely, the design of water distribution systems (WDS). This is a crucial problem of industrial interest in the water field since increasing urban development represents a permanent challenge for the management of many resources – and especially water.

2 Description of particle swarm optimisation

PSO is an evolutionary computation technique that was first developed by Kennedy and Eberhart [1]. The particle swarm idea originated as a simulation of a simplified social system, the graceful but unpredictable choreography of a flock of birds. The word 'swarm' is used after a paper by Millonas [7], who developed several models for artificial life and examined certain principles in swarm intelligence. The selection of the term 'particle' comes from classical mechanics and is justified by the fact that positions and velocities are applied to the population elements, despite the fact that are considered to have zero mass and volume. Kennedy and Eberhart's first idea was to simulate the social behaviour of a flock of birds in their attempt to reach, when flying through the field (search space), their unknown destination (fitness function), e.g. the location of food resources.

2.1 Description of standard PSO

Each problem solution in PSO is a bird of the flock and is referred to as a particle. In this algorithm, birds evolve in terms of their individual and social behaviour and mutually coordinate their movement towards their destination [8].

Each bird keeps track of its coordinates in the problem space, remains aware of its recent trajectory, and remembers a dynamic specific position: the best solution (best local position) it has achieved so far. Birds also communicate among themselves and are able to identify the bird in the best position (best global position). In a coordinated way each bird evolves by changing its velocity so that it accelerates towards both its best position and the best position obtained so far by any bird in the flock without forgetting its recent trajectory. 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. It is worth noting that, according to the description, the process involves not only individual intelligent behaviour (including memory) but also social interaction. In this way, birds somehow follow their recent history, learn both from their own experience (local search), and from group experience (global search).

PSO shows common evolutionary computation features including: (1) initialisation with a population of random solutions; (2) search for optima by updating generations; and (3) particle evolution through the problem space by following specific strategies.

The process initially starts with a group of M particles, which have been randomly generated, representing different solutions of the problem. The ith particle, X_i, is represented by its location in a d-dimensional subset, $S \subset R^d$, where

d corresponds to the number of variables of the problem. Any set of values of the d variables, determining the particle location, represents a candidate solution for the optimisation problem:

Find $\min_{X \in S} F(X)$, subject to appropriate constraints,

where F is the fitness function associated with the problem, a minimisation problem without loss of generality. The optimal solution is then searched for by iteration. The performance of each particle is measured using this fitness function, according to the problem in hand.

During the process, as already explained, each particle i is associated with three vectors:

- current position, $X_i = (x_{i1,} \ldots, x_{id})$;
- best position, $Y_i = (y_{i1,} \ldots, y_{id}) = \text{argmin}(F(X_i(t)), F(X_i(t-1)))$, reached in previous cycles; and
- flight velocity $V_i = (v_{i1,} \ldots, v_{id})$, which makes it evolve.

The bird which is in the best position, $Y^* = \text{argmin}\{F(X_i(t), i = 1, \ldots, M\}$, is identified for every iteration, t.

During each generation, the velocity of each particle is updated in a process based on its recent trajectory, its best encountered position, the best position encountered by any particle, and a number of parameters:

$$V_i \leftarrow \omega V_i + c_1 \, rand\,() \, (Y_i - X_i) + c_2 \, rand\,() \, (Y^* - X_i). \tag{1}$$

The parameters in (1) are as follows: ω is a factor of inertia suggested in [8] that controls the impact of the velocity history on the new velocity; c_1 and c_2 are two positive acceleration constants, called the cognitive and social parameters, respectively; *rand*() represents a function that creates random numbers between 0 and 1 (two independent random numbers enter eqn (1)).

Expression (1) is used to calculate the ith particle's new velocity, a determination that takes into consideration three main terms: the particle's previous velocity, the distance of the particle's current position from its own best position, and the distance of the particle's current position from the swarm's best experience (position of the best particle).

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

$$V_{\min} \leq V_{ij} \leq V_{\max}, \tag{2}$$

in order to control excessive roaming by particles outside the search space. These important parameters are problem dependent. They determine the resolution with which regions between the present position and the target (best so far) positions are searched. If velocities are too great, particles might fly through good solutions. If they are too slow, on the other hand, particles may not explore sufficiently beyond locally good regions – becoming easily trapped in local optima and unable to move far enough to reach a better position in the problem space. Usually, $V_{\min}$ is taken as $-V_{\max}$.

Finally, the position of each particle is updated every generation. This is performed by adding the velocity vector to the position vector,

$$X_i \leftarrow X_i + V_i. \tag{3}$$

Each particle or potential solution moves to a new position according to expression (3).

2.2 Combining continuous and discrete variables

The previously described algorithm can be considered as the standard PSO algorithm, which is applicable to continuous systems and cannot be used for discrete problems. Several approaches have been put forward to tackle discrete problems with PSO [9–12]. The approach we propose for discrete variables involves the use of the integer part of the discrete velocity components. In this way, the new velocity of discrete components will be an integer and, as a consequence, the new updated positions will share this characteristic since the initial population, in its turn, must also have been generated using only integer numbers. According to this simple idea, expression (1) will be replaced by

$$V_i \leftarrow fix(\omega V_i + c_1\, rand\,()\,(Y_i - X_i) + c_2\, rand\,()\,(Y^* - X_i)), \tag{4}$$

for discrete variables, where $fix(\cdot)$ is a function that takes the integer part of its argument. However, it should be taken into account that the new velocity discrete values must be controlled by suitable bounds as in (2). However, there is a singular aspect regarding velocity bounds that must be taken into consideration so that the algorithm can treat both continuous and discrete variables in a balanced way. In [3], it was found that using different velocity limits for discrete and continuous variables produce improved results.

2.3 Enriched diversity

The main drawback of PSO is the difficulty in maintaining acceptable levels of population diversity while balancing local and global searches; and as a result, suboptimal solutions are prematurely obtained [13]. Some evolutionary techniques maintain population diversity by using some more or less sophisticated operators or parameters. Several other mechanisms for forcing diversity in PSO can be found in the literature [14–16]. In general, the random character that is typical of evolutionary algorithms adds a degree of diversity to the manipulated populations. Nevertheless, in PSO these random components are unable to add sufficient diversity.

Frequent collisions of birds in the search space, especially with the leader, can be detected – as shown in [4]. This caused the effective size of the population to fall and the algorithm's effectiveness is consequently impaired. The study in [17] introduces a PSO derivative in which a few of the best birds are selected to check collisions, and colliding birds are randomly re-generated if collision occurs. This random re-generation of the many birds that collide with the best birds has been

shown to avoid premature convergence as it prevents clone populations from dominating the search. The inclusion of this procedure into PSO greatly increases diversity as well as improves convergence characteristics and the quality of the final solutions.

2.4 Self-adapting parameters

The role of the inertia, ω, in (1) and (4) is considered critical for the convergence behaviour of the PSO algorithm. Although inertia was constant in the early stages of the algorithm, it is currently allowed to vary from one cycle to the next. As it facilitates the balancing of global and local searches, it has been suggested that ω could be allowed to adaptively decrease linearly with time – usually in a way that initially emphasises global search and then, with each cycle of the iteration, increasingly prioritises local searches [18]. A significant improvement in the performance of PSO, with decreasing inertia weight across generations, is achieved by using the proposal [19]:

$$\omega = 0.5 + \frac{1}{2\left(\ln\left(t\right)+1\right)}. \tag{5}$$

In the variant we propose, the acceleration coefficients and the clamping velocities are neither set to a constant value, as in standard PSO, nor set as a time-varying function, as in adaptive PSO variants [20]. Instead, they are incorporated into the optimisation problem [5]. Each particle is allowed to self-adaptively set its own parameters by using the same process used by PSO – and given by expressions (1) or (4), and (3). To this end, these three parameters are considered as three new variables that are incorporated into the position vector X_i. In general, if d is the dimension of the problem, and p is the number of self-adapting parameters, the new position vector for particle i will be:

$$X_i = (x_{i1}, \ldots, x_{id}, x_{id+1}, \ldots, x_{id+p}). \tag{6}$$

These new variables do not enter the fitness function, but rather they are manipulated by using the same mixed individual-social learning paradigm used in PSO. Also, V_i and Y_i that give the velocity and thus far best position for particle i increase their dimension, correspondingly.

By using expressions (1) or (4), and (3), each particle is additionally endowed with the ability to self-adjust its parameters by taking into account the parameters it had at its best position in the past, as well as the parameters of the leader, which facilitated this best particle's move to its privileged position. As a result, particles use their cognition of individual thinking and social cooperation to improve their positions, as well as improving the way they improve their position by accommodating themselves to the best-known conditions, namely, their conditions and their leader's conditions when they achieved the thus-far best position.

3 Multi-objective PSO

In multi-objective optimisation, scores of objectives are not scalars but vectors that is to say that the objective space is multidimensional with as many dimensions as objectives considered. A change in the decision space may produce a positive increment in some components (objectives) of these vectors while, at the same time, cause lower values in other objectives. In a departure from the normal behaviour of particles in PSO – derived from the ordered nature of real numbers – particles now use the dominant solution concept when deciding on a *better* position: solution A is said to dominate another solution B when A is better than B in at least one objective, and not worse in the others. Two solutions are called indifferent or incomparable if neither dominates the other. Dominant solutions will always be considered as *better* solutions. Figure 1 illustrates the solution space (decision space), with three decision variables, x_1, x_2, and x_3, and their corresponding points in the objective space, with two objectives, namely, minimisation of z_1 and maximisation of z_2.

The empty particles are incomparable because none is better than the others in both objectives (check in the objective space), but the second-from-top empty particle dominates the solid particles since it is better in both objectives: lower z_1-value and higher z_2-value. Moreover, the particle corresponding to the second-from-bottom empty dot in the objective space seems to dominate the lowest three solid particles (lower z_1-value and equal z_2-value). In contrast, the other three empty particles do not dominate any of the solid particles. Finally, the upper solid particle dominates the three lower solid particles.

The objective of multi-objective optimisation is to obtain at least an approximation of the set of nondominated particles (in the decision space) and its image on the objective space, which is the so-called Pareto front [21] (in red, in Fig. 1).

The leadership in a swarm must be determined in a different manner than in the classical PSO algorithm. The most natural option is to select as leader the closest particle to the so-called *utopia point* in the objective space. The utopia point is

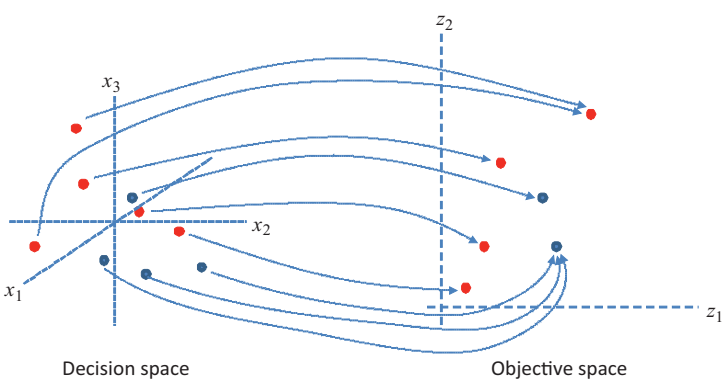

Figure 1: Relation between decision and objective spaces.

defined as the point in the objective space whose components give the best values for every objective. The utopia point is an unknown point since the best value for every objective is not known at the beginning (and perhaps during the whole process). Accordingly, we use a dynamic approximation of this utopia point, termed *singular point*, which is updated with the best values found so far during the evolution of the algorithm [22].

As each objective may be expressed in different units, it is necessary to make a regularisation for evaluating distances in the objective space. Coordinates may be regularised in terms of percentage, considering that for every component, the worst and best values of the corresponding objective are 0% and 100%, respectively. The percentage corresponding to any other value may be calculated using linear interpolation. To establish the distance between any two objective vectors, the components are first regularised in terms of percentage and then the Euclidean distance between them is calculated. The worst and best objective values are not usually known *a priori* and are updated while the solution space is explored.

Figure 2 shows a two-dimensional representation of the concept of a singular point. The most interesting solutions are located near the singular point and not too far from the ends of the Pareto front. For this reason, instead of seeking a complete and detailed Pareto front, we may be more interested in precise details around the singular point. However, situations may occur as shown in Fig. 2 (right) when non-symmetric Pareto fronts with respect to the singular point develop. As a result, poorly detailed sections on the Pareto front may appear. It seems plausible that problem complexity is the cause of this asymmetry in many real-world multi-objective optimisation problems.

It is difficult to find a general heuristic rule for deciding which parts of the Pareto front should be more closely represented and how much detail the representation of the Pareto front should contain. Various methods of inducing the Pareto

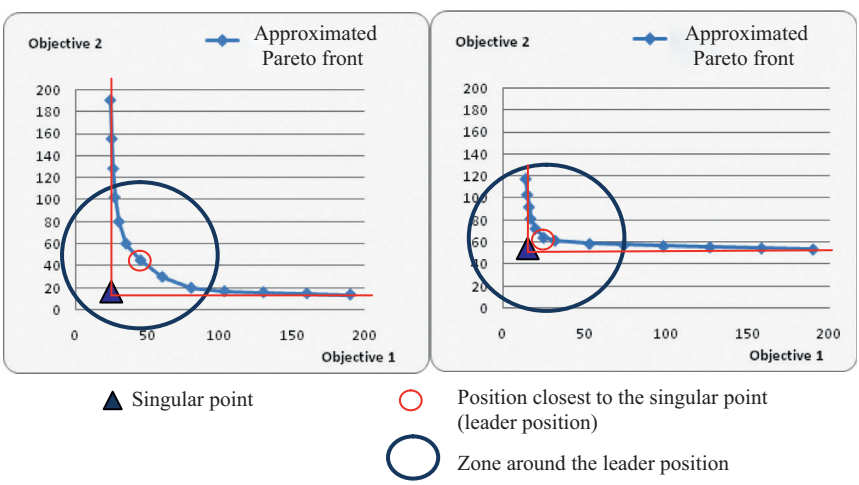

Figure 2: Singular point in an approximated Pareto front.

front completeness may be devised. We describe below a possible approach based on dynamic population increase to enrich the Pareto front density, and another alternative approach based on human computer interaction to complete poorly represented areas of the Pareto front.

3.1 Dynamic population increase

During the search process swarms are able to increase their population when needed in order to better define the Pareto front: a particle whose solution already belongs to the Pareto front may, on its evolution, find another solution belonging to the front. In this situation, a new clone of the particle is placed where the new solution is found, thus increasing the density of particles on the Pareto front. Greater densities on the Pareto front must be restricted to cases where the new clone has at least one of its neighbours located further away than some minimal permissible distance in the objective space. For example, in Fig. 3 (left), a particle J, whose objective vector is located at position P_J, finds a new position $NewP_J$, also belonging to the Pareto front.

The consequence is shown in Fig. 3 (right): a new particle k is added to the swarm by cloning the particle at position $NewP_J$, while particle J will continue to be active and considering the point P_J as its best objective position. This occurs because the new point P_k has at least one neighbour located further away than the minimum permissible distance for at least one of the objectives. In Fig. 3 (right), the point to the left of P_k is located at a distance, regarding Objective 1, that is greater than the minimal distance considered for the increase of density on the Pareto front. It should be noted that two particles are considered as neighbours when no other particle is located between them for at least one of the objectives considered in the problem.

3.2 Human–computer interaction

During evolution there may be areas poorly represented on the Pareto front. Users are allowed to add new swarms for searching in the desired region of the objective space. The concept of singular point is now extended to any desired area in the objective space for particles to search around.

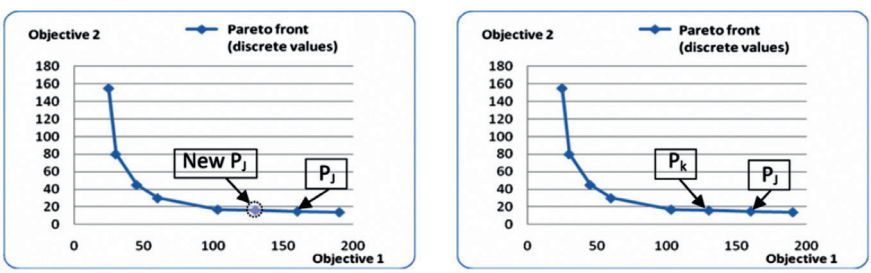

Figure 3: Particle cloning.

Decisions are strongly dependent on the individuals solving the problem and on the problem itself. The user can specify additional points where the algorithm should focus the search, and specify how much detail a region should contain. This must be achieved in real time during the execution of the algorithm. Once a new singular point is added, a new swarm is created with the same characteristics as the swarm created first. Swarms will run in parallel, but they share (and can modify) the information related to the Pareto set. Particles from any swarm can be added to the Pareto set. If the user changes the fixed values for a singular point, then the corresponding swarm selects a new leader considering the new location of the singular point.

Human interaction with the algorithm in real time also enables the incorporation of human behaviour so that the human becomes a member of the swarm by proposing new candidate solutions. Eventually, such a solution can be incorporated into the Pareto front or lead the behaviour of a group of particles. User solutions will always be evaluated in the first swarm created. If a particle is being evaluated then the user request waits until the evaluation of the particle is finished. If a solution proposed by the user is being evaluated then any particle belonging to the first swarm should wait for evaluation. Once any solution is evaluated, the algorithm checks whether it could be incorporated into the Pareto front. Synchronisation is achieved among all the swarms so that they have open access for managing the Pareto front. Proposed solutions could even become leaders of the swarm(s) if they are good enough. At this point, human behaviour begins to have a proactive role during the evolution of the algorithm.

Participation by several human agents with different perspectives on a problem is close to what happens in the practice of engineering decision making, where politicians, economists, engineers, and environment specialists are all involved in final decisions. The idea of incorporating user experience into the search process is a step forward in the development of computer-aided design.

3.3 The algorithm

New particles are used that are based on the behaviour of particles in PSO. Swarms running in parallel may be distributed in different computers, and it must be ensured that the swarms can communicate among themselves; a peer-to-peer scenario could be a good choice for this task. The steps of the algorithm (for a swarm) are summarised below.

1. Set parameters and initialize the number of iterations to zero.
2. Generate random population of M particles: $\{X_i(k)\}_{i=1}^{M}$
3. Evaluate the fitness of particles and set the local best location for each particle equal to its current location.
4. Form the Pareto front and list the particles belonging to the front.
5. Build the singular point.

6. Find the closest particle to the singular point and establish it as swarm leader.
7. While not in termination condition, do the following:
   a. Execute from $i = 1$ to number of particles.
      Start
      i. Change the position of the particle:
         Determine the inertia parameter $\omega(k)$, according to (5).
         Calculate the new velocity, $V_i(k+1)$, for particle i according to (1) or (4).
         Set a new position, $X_i(k+1)$, for particle i according to (3).
      ii. Calculate the new fitness function vector for particle i in its new position.
      iii. If the new fitness function vector for particle i dominates the fitness function vector that the particle had before moving to the new position, then set the new position as the best position currently found by particle i.
      iv. If particle i is in the list of particles belonging to the Pareto front then:
         If the new fitness function vector may also be a point on the Pareto front and this new position has at least one of its neighbours located further than the minimal permissible distance from any of the objectives, then add a new particle j (a clone of i) with P_k and P_{kbest} located at the current position of i;
         else
         try to add (if possible) the particle i (at its new position) to the Pareto front; if the particle is added, remove from the list any dominated solution; dominated clones are eliminated from the swarm.
      v. If particle i is closer to the singular point than any other particle in the swarm, then set particle i as the leader of the swarm with regard to the singular point.
      vi. If particle i is not currently the leader of the swarm, but coincides in position with the leader, then re-generate particle i randomly.
      End
   b. Increase the iteration number.
8. Show the Pareto front and related results.

Some steps in this algorithm (specifically, 3 and 7-a-ii) involve particle fitness calculations. In the next section, we consider various functions characterizing the fitness of a particle –a candidate network. One of them involves the precise knowledge of all the network node pressures, which are obtained by solving the continuity and the energy equations. Various tools to analyze water networks have been developed in the past. Among them, EPANET2 [23], based on gradient-like techniques, is a software program for analysing water networks in the steady-state flow, and is used to evaluate the hydraulic performance of the solutions. Additional analyses in extended period simulation, or any other transient analysis to assess the goodness of solutions, may be performed without any change in the core of the given algorithm. The algorithm and its connection

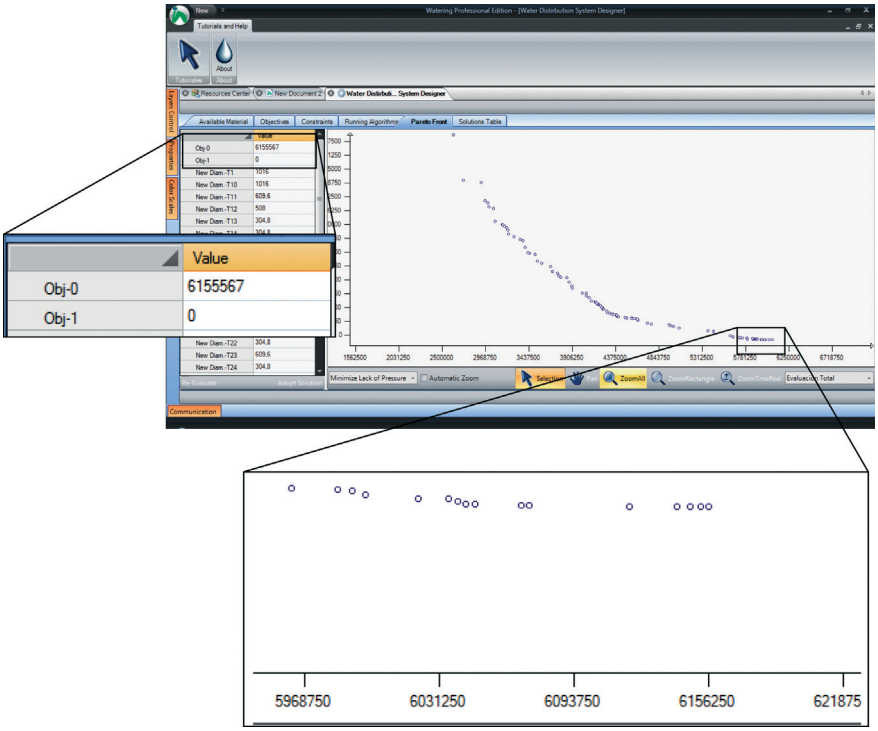

Figure 4: Pareto front for the Hanoi network as seen on the software interface.

with EPANET2 are implemented in a software program called WaterIng[1] [24], developed for water distribution system design and analysis. WaterIng is in constant evolution and may be downloaded from its website – the installation includes a file with network data as an example. A user guide is also available to learn the main concepts of how to design a water distribution system using the software.

A two-dimensional representation of the Pareto front showing the interface of the software implementing the described algorithm is shown in Fig. 4. In this particular case, Objective 0 represents an investment cost, and Objective 1 represents the lack of compliance with certain problem constraints. We have used WaterIng for the application using multi-objective optimisation in the next section.

4 Applications

PSO has shown great potential for the solution of various optimisation problems [3, 11, 13, 19, 25–28]. The authors have used the algorithmic engine of the

[1] www.ingeniousware.net

software mentioned in the previous section to solve problems in various areas [25, 27, 29, 30]. These fields have included: hydrology [31–36], hydraulics [37–46], and WR management [47–51], in addition to the references quoted in this chapter.

We devote the rest of the section to present an important application in water supply management – together with the solutions to two real-world case studies.

4.1 The design of water distribution systems

WDS are undoubtedly alive. They are born, grow, age, and deteriorate, need care (preventive care but also sometimes surgery), are expected to work properly, must meet basic requirements even under adverse circumstances, and so on. Our aim is to ensure a long and quality life for WDS. As a consequence, the design of WDS cannot be thought of as a single, material, and static design. This is one reason why WDS design optimisation is one of the most heavily researched areas in hydraulics. [52–55] and [24] are but a sample of significant references selected with intervals of around five years since 1985 to date. Of special universal relevance is the *annual WDS analysis* (WDSA) conference endorsed by the EWRI-ASCE; and in which WDS design is one of its more relevant recurrent topics.

The optimal design of a WDS aims to determine the values of all involved variables in such a way that all the demands are satisfied, even under certain failure conditions, while the investment and maintenance costs of the system are minimal (see, for instance, [56]). 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; reuse or substitution of existing components; creating a working system configuration that fulfils all water demands (including water quality); adherence to the design constraints; and guaranteeing a certain degree of system reliability [57, 58].

The diameters of the new pipes are the basic variables of the problem. Nevertheless, additional variables that depend on the design characteristics of the system may be required: storage volume, pump head, the type of rehabilitation to be carried out for various parts of the network, etc. The estimation of individual costs will always depend on these variables. The correct approach to assess the costs for each element is important when defining the corresponding objective function, which has to be fully adapted to the problem under consideration: design, enlargement, rehabilitation, operation design, etc. It is also important that this objective function reflects with utmost reliability the total cost of the system during its entire lifetime. Various authors have used, in their optimisation, an objective function that only considers the costs of the pipelines (new and/or additional and duplicated pipelines) while others have taken into account other various costs involved (some examples are [59, 60]).

Satisfying the demands both in quantity and quality represents another objective. Minimum pressure values are frequently specified to guarantee a minimum level of service quality. This condition is sometimes enforced into the problem as constraints for all the consumption nodes. In contrast, in multi-objective approaches, this condition is issued as another objective.

Reliability mainly refers to the ability of the network to provide consumers with adequate and high-quality supply, under normal and *abnormal* conditions. The reliability of water systems can be classified as mechanical reliability and hydraulic reliability. The former usually refers to failures of system components, such as pipe breakage or pumps being out of service. The latter refers to uncertainty coming mainly from nodal demand and pipe roughness. There is no universal agreement about what is the best measure of reliability and what is an acceptable level of reliability (see, for example [61, 62]).

Various approaches exist for assessing the reliability of a water distribution system [63–65]. We consider here the proposal raised in [66]. It ensures that the system offers a certain level of reliability by considering costs incurred by the lack of supply satisfaction. The authors have found that the improvement obtained for various systems by considering these costs in the fitness function implies only moderate increases regarding the initial investment costs.

A more detailed representation of these three objectives follows. The first objective takes into account the pipeline costs (other costs may be easily included); the second objective considers the lack of compliance with minimal values for the pressure at each node of the network; and the third objective evaluates some reliability measure by considering incurred costs for service disruptions.

The mathematical formulation for the first objective, the investment cost of the pipes required for the design, is as follows:

$$C = \sum_{i=1}^{L} c_i l_i, \tag{7}$$

where all (L) individual pipes are summed. $D = (D_1, \ldots, D_L)^t$ is the vector of the pipe diameters. The costs per meter, depending on the diameter of pipe i, D_i, is given by c_i and its corresponding length by l_i. Note that D_i is chosen from a discrete set of available diameters and c_i is a nonlinear function of diameter.

The second objective, P, measures the lack of pressure in the nodes with respect to a prefixed minimal value. This objective is also a function of the selected pipe diameters (through the hydraulic model). For nodes with pressure larger than this minimal value, the associated individual terms vanish, and one uses the usual Heaviside step function H in the explicit expression for P:

$$P = \sum_{j=1}^{N} H(p_{\min} - p_i) \cdot (p_{\min} - p_i). \tag{8}$$

EPANET2 is used to evaluate the actual pressure at consumption nodes for a specific solution. The integration of such software to run various analyses or simulations for potential solutions of the problem is performed during the optimisation process that is developed within the evolutionary algorithms [2, 4, 5] – such as the algorithm presented in this paper.

The presence of loops in water distribution networks adds complexity to the design problem since the optimisation algorithms, due to their nature, attempt to avoid redundancies, in particular, unnecessary loops. This action does not favor

the reliability of the system. Considering explicitly some kind of reliability within the fitness function is one of the most difficult tasks faced by researchers in the area. Many researchers have asked (see [61, 62], for example) "What is the best measure of reliability and what level of reliability is acceptable?"

In this chapter, we consider the proposal in [66] that indirectly assesses reliability from an economic point of view by considering the costs of the water not delivered due to problems in the system. Precisely, the third objective is defined by

$$R = \sum_{k=1}^{L} w_k \cdot L_k \cdot d_k^{-u}, \tag{9}$$

where:

- w_k is a coefficient associated with 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, and length, L, (a and u are known constants)
- t_f is the average number of days required to repair the pipe
- c_f is the daily repair average cost
- c_a is the average cost of the water supplied to affected consumers, in monetary units per unit volume
- $V_f = 86400 \cdot Q_{\text{break}}$ is the daily volume of water that should be supplied to the affected consumer (86400 = number of second per hour) due to the loss of water of Q_{break} cubic meters per second.

All the constant values entering this equation have been taken from [66]. Consideration of other values corresponding to other specific cases is straightforward.

The optimisation problem may be addressed as a single objective problem or as a multi-objective problem. If a single objective approach is implemented, any of the defined objectives can be considered and as many other objectives introduced as desired by way of 'suitable' penalty costs. For example, the following fitness function

$$F(D, V_f) = C + \alpha P + \beta R, \tag{10}$$

considers the minimisation of the piping costs, C, while considering a penalty term αP that considers the lack of compliance with some required minimum pressure level, and another penalty cost, βR, regarding the lack of reliability as the cost incurred for not satisfying the supply due to problems in the system (α and β are suitable penalty factors).

4.2 A single objective problem

The following problem has been considered in [67]. It analyses a sector of the WDS of a Latin American capital (see the layout in Fig. 5) using the PSO variant presented above.

Figure 5: Sector of a WDS in a Latin American capital and the solution when considering reliability.

Table 1: Comparison between costs for both solutions.

Diameter [mm]	Without reliability		With reliability	
	Length [m]	Cost [soles]	Length [m]	Cost [soles]
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 (soles)	2165044.69		2244166.47	

This sector is fed by a tank, and has 294 lines amounting to 18.337 km of pipe and 240 nodes consuming 81.53l/s in total.

Figure 5 also presents the solution obtained by using eqn. 10, which includes the minimum pressure requirement and reliability as penalty terms. The first column of Table 1 specifies the various diameters. This solution is only a mere 3.65% more expensive than the solution obtained by using the fitness function $G = C + \alpha P$ (no reliability consideration). Table 1 presents a comparison between the initial investment costs for both solutions.

Figures 6 and 7 present the solution when not considering reliability. These figures are used to show the performance differences between both solutions following pipe breaks.

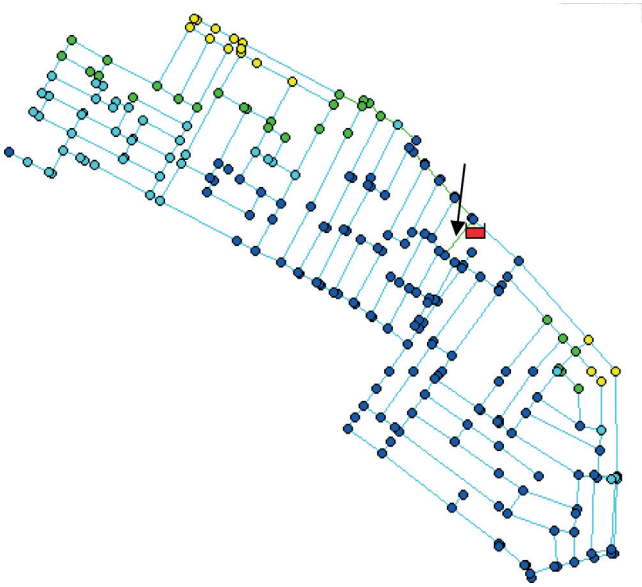

Figure 6: Lower performance after ignoring reliability due to closure of marked pipe.

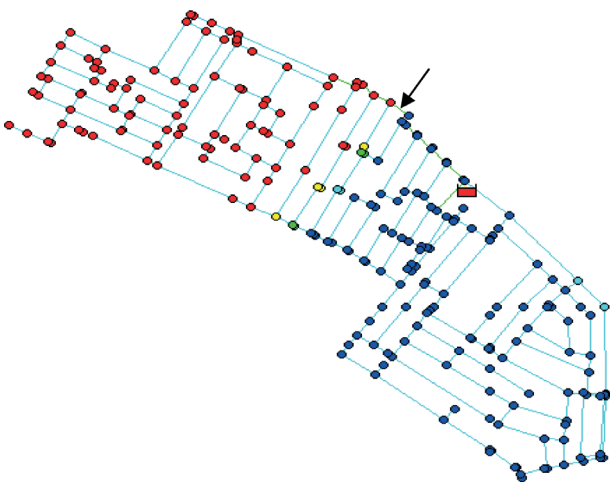

Figure 7: Higher impact on performance after ignoring reliability due to closure of marked pipe.

The effects of closing the pipe indicated by the arrow can be observed in Figure 6. Even though no pressures less than 10 m can be observed, almost half of the nodes (light grey color) do not meet the required minimum pressure of 15 m (dark grey color). Figure 7 shows the higher impact produced by another closed pipe. Now slightly clear points (on the upper left) are consumption nodes with a pressure lower than 10 m. Again, this will not happen for the more reliable design (Figure 5) obtained from F, no matter which pipe is out of service.

The formulation we consider here aims at minimising the cost of a new network with the diameters of the pipes as design variables, while satisfying a minimum pressure in all the nodes and, at the same time, providing a certain amount of enforced reliability by guaranteeing the service under determined failure scenarios. The scenarios considered here 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 when subjected to these circumstances. If the test is negative the design is suitably penalised. In this way, designs will develop increasing reliability. To undergo those tests, the system must be analysed for any of those specific 'breakages'. Only solutions assessed as being feasible by EPANET2 are considered.

4.3 A multi-objective analysis

Multi-objective approaches are clearly preferred for a number of reasons – even though they are more expensive from a computational point of view. It is clear that, sooner or later, a decision must be made to balance the various objectives. Using penalty factors is a decision, somehow arbitrary, taken *a priori*. On the contrary, a multi-objective approach provides a set of solutions that will help *a posteriori* decision making and provide decision makers with a richer range of solutions and alternatives.

We present another case-study corresponding to a different sector of the WDS of the same Latin American capital (Fig. 8). The design involves the three objectives described above in a multi-objective solution: minimising the investment

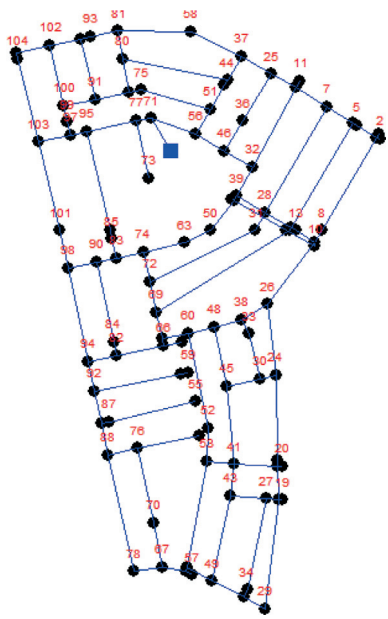

Figure 8: A real-world network.

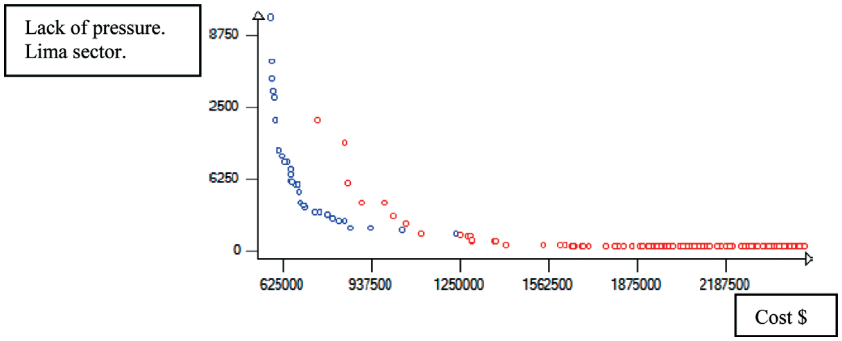

Figure 9: Approximated Pareto front for a real world case.

cost; minimising the lack of pressure at demand nodes; and minimising additional costs because of reliability problems.

This network is fed by a tank and made of 132 lines and 104 consumption nodes, its total length being 9.055 km, and the total consumed flowrate amounting to 47.09l /s.

In Figure 9 a bi-dimensional (cost against lack of pressure) representation of the approximated Pareto front that considers the three mentioned objectives is presented. Two swarms (one covering the left zone of the Pareto front and the other the right more horizontal area) can be identified that build the three-dimensional Pareto front of this problem with three objectives. Observe that the representation is a projection in two dimensions of points in the real Pareto front, which is a surface in a three-dimensional space.

Plenty of enriched information that helps the decision-making process is provided by this type of representation of the Pareto front. For example, it becomes evident, as expected, that after some point, the rate at which the minimum pressure can be increased in the network is much lower than the rate at which initial investment costs must be increased to achieve the desired pressure level.

The development of a multi-objective optimisation process enables the combination of economic, engineering, and policy viewpoints when searching for a solution to a problem. For example, the relationship between the initial investment cost and the minimum pressure in the network may help decide, among other factors, which pressure would be better to use for the final solution. In this case (in which there is a limited budget to implement the design), the decision maker has at his or her disposal a clear guideline to assess how much the quality may be improved if the budget is increased by a certain amount. This is an added value of the multi-objective approach when solving the problem of optimal design of WDS.

5 Conclusions

Classical methods of optimisation are poorly suited for many real world problems since they are unable to process inaccurate, noisy, discrete, and complex data. In this chapter, we have presented the principles of PSO, an evolutionary algorithm that has shown great efficiency for the solution of various optimisation problems.

We have also described three modifications that considerably improve the performance of the standard algorithm for finding solutions to various optimisation problems. Firstly, a proposal for enabling continuous and discrete variables to coexist in one PSO formulation. Secondly, a mechanism for enriching diversity and so improving the performance of PSO. And thirdly, a self-adapting feature that avoids the cumbersome task of parameter selection and fine tuning. The elements to adapt this algorithm to multi-objective optimisation problems have then been provided. Finally, after mentioning various applications in the water field, we have shown the results of specific applications to selected case-studies regarding the design of WDS – using a very well-known urban water problem.

Further improvements should be considered for the multi-objective algorithm presented in this paper. For example, the inclusion of problem-dependent rules – thus taking advantage of expert knowledge – would further facilitate the process of finding solutions by enhancing human–computer interaction while introducing more reality. Also, the consideration of a wider environment where swarms (or even other algorithms – evolutionary or otherwise) with various specialisation tasks could coexist and cooperate in the optimisation process. In addition, all the algorithms used must be further developed to take advantage of emerging technologies in the field of parallel and distributed computing. Some of these research lines have been addressed recently in [67], which is the source for most of the contents of this chapter.

Acknowledgements

This work has been developed with the support of the project IDAWAS, DPI2009-11591, of the Dirección General de Investigación of the Ministerio de Educación y Ciencia, and ACOMP/2010/146 of the Conselleria d'Educació of the Generalitat Valenciana. The first author is also indebted to the Universitat Politècnica de València for the sabbatical leave granted during the first semester of 2011. The use of English in this paper was revised by John Rawlins.

References

[1] Kennedy, J. & Eberhart, R.C., Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks, Piscataway*, NJ, pp. 1942–1948, 1995.

[2] Montalvo, I., Izquierdo, J., Pérez, R. & Tung, M.M., Particle swarm optimization applied to the design of water supply systems. *Computer & Mathematics with Applications*, **56(3)**, pp. 769–776, 2007.

[3] Izquierdo, J., Montalvo, I., Pérez, R. & Fuertes, V.S., Design optimization of wastewater collection networks by PSO. *Computer & Mathematics with Applications*, **56(3)**, pp. 777–784, 2007.

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

[5] Montalvo, I., Izquierdo, J., Pérez, R. & Herrera, M., Improved performance of PSO with self-adaptive parameters for computing the optimal design of water supply systems. *Engineering Applications of Artificial Intelligence*, **23(5)**, pp. 727–735, 2010.
[6] Izquierdo, J., Montalvo, I., Pérez, R. & Tavera, M., Optimization in water systems: A PSO approach. *Proceedings of the Business and Industry Symposium, BIS*, Ottawa, Canada, 2008.
[7] Millonas, M.M., Swarms, phase transition, and collective intelligence. *Artificial Life III*, ed. C.G. Langton, Addison Wesley: Massachusetts, pp. 417–445, 1994.
[8] Shi, Y.F. & Eberhart, R.C., A modified particle swarm optimizer. *IEEE International Conference on Evolutionary Computation*, pp. 69–73, 1998.
[9] Al-Kazemi, B. & Mohan, C.K., Multi-phase discrete particle swarm optimization. *JCIS*, pp. 622–625, 2002.
[10] Rastegar, R., Meybodi, M.R. & Badie, K., A new discrete binary particle swarm optimization based on learning automata. *Proceedings of the 2004 International Conference on Machine Learning and Applications*, Louisville, KY, USA, 2004.
[11] Liao, C.J., Tseng, C.T. & Luarn, P., A discrete version of particle swarm optimization for flowshop scheduling problems. *Comput Oper Res*, **34(10)**, pp. 3099–3111, 2007.
[12] Shi, X.H., Liang, Y.C., Lee, H.P., Lu, C. & Wang, Q.X., Particle swarm optimization-based algorithms for TSP and generalized TSP. *Information Processing Letters*, **103(5)**, pp. 169–176, 2007.
[13] Dong, Y., Tang, B.X.J. & Wang, D., An application of swarm optimization to nonlinear programming. *Computers & Mathematics with Applications*, **49(11–12)**, pp. 1655–1668, 2005.
[14] Angeline, P.J., Using selection to improve particle swarm optimization. *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 84–89, Anchorage, AL, USA, 1998.
[15] Løvbjerg, M., Rasmussen, T.K. & Krink, T., Hybrid particle swarm optimiser with breeding and subpopulations. *Proceedings of the Genetic and Evolutionary Computation Conference*, San Francisco, CA, USA, 2001.
[16] Parsopoulos, K.E., Plagianakos, V.P., Magoulas, G.D. & Vrahatis, M.N., Stretching technique for obtaining global minimizers through particle swarm optimization. *Proceedings of the Workshop on Particle Swarm Optimization*, Indianapolis, IN, USA, 2001.
[17] Herrera, M., Izquierdo, J., Montalvo, I. & Pérez-García, R., Injecting diversity into particle swarm optimization. Application to water distribution system design. *Advances in Computer Science and Engineering*, **6(2)**, pp. 159–179, 2011.
[18] Shi, Y. & Eberhart, R.C., Empirical study of particle swarm optimization. *Proceedings of the IEEE Congress on Evolutionary Computation*. Washington, DC, USA, 1999.
[19] Jin, Y.X., Cheng, H.Z., Yan, J.I. & Zhang, L., New discrete method for particle swarm optimization and its application in transmission network expansion planning. *Electric Power Systems Research*, **77(3–4)**, pp. 227–233, 2007.

[20] Arumugam, M.S. & Rao, M.V.C., On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems. *Appl Soft Comput,* **8(1)**, pp. 324–336, 2008.

[21] V. Pareto, *Cours d'economie politique*, Université de Lausanne, Lausanne, Switzerland, 1896.

[22] Montalvo, I., Izquierdo, J., Schwarze, S. & Pérez-García, R., Multi-objective Particle Swarm Optimization applied to water distribution systems design: An approach with human interaction. *Mathematical and Computer Modelling*, **52**, pp. 1219–1227, 2010.

[23] Rossman, L.A., *EPANET2, User's Manual. Cincinnati* (IN), USA, Environmental Protection Agency, 2000.

[24] Montalvo, I., Martínez-Rodríguez, J.B., Izquierdo, J. & Pérez-García, R., Water distribution system design using agent swarm optimization. *Water Distribution System Analysis 2010 – WDSA2010*. Tucson, AZ, USA, 2010.

[25] Herrera, M., Izquierdo, J., Montalvo, I., García-Armengol, J. & Roig, J.V., Identification of surgical practice patterns using evolutionary cluster analysis. *Mathematical and Computer Modelling*, **50**, pp. 705–712, 2009.

[26] Janson, S., Merkle, D. & Middendorf, M., Molecular docking with multi-objective particle swarm optimization. *Applied Soft Computing*, **8**, pp. 666–675, 2008.

[27] Izquierdo, J., Minciardi, R., Montalvo, I., Robba, M. & Tavera, M., Particle Swarm Optimization for the biomass supply chain strategic planning. *Proceedings of 4th Biennal Meeting, iEMSs 2008: International Congress on Environmental Modelling and Software*, pp. 1272–1280, Barcelona, Spain, 2008.

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

[29] Díaz, J.L., Herrera, M., Izquierdo, J., Montalvo, I. & Pérez-García, R., A Particle Swarm Optimization derivative applied to cluster analysis. *Proceedings of the iEMSs 2008 (International Congress on Environmental Modelling and Software – 4th Biennial Meeting)*, Barcelona, 2008.

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

[31] Gill, M.K., Kaheil, Y.H., Khalil, A., McKee, M. & Bastidas L., Multiobjective particle swarm optimization for parameter estimation in hydrology. *Water Resources Research*, **42**, W07417, p. 14, 2006.

[32] Chau, K.W., Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River. *Journal of Hydrology*, **329(3–4)**, pp. 363–367, 2006.

[33] Jiang, Y., Hu, T., Huang C.C. & Wu, X., An improved particle swarm optimization algorithm. *Applied Mathematics and Computation*, **193(1)**, pp. 231–239, 2007.

[34] Zhang, X., Srinivasan, R., Zhao, K. & van Liew M., Evaluation of global optimization algorithms for parameter calibration of a computationally intensive hydrologic model. *Hydrological Processes*, **23(3)**, pp. 430–441, 2009.

[35] Kuok, K.K., Harun, S. & Shamsuddin, S.M., Particle swarm optimization feedforward neural network for hourly rainfall runoff modeling in bedup basin. *Malaysia. International Journal of Civil & Environmental Engineering IJCEE*, **9(10)**, 2010.

[36] Lü, H., Yu, Z., Horton, R., Zhu, Y., Wang, Z., Hao, Z. & Xiang L., Multiscale assimilation of root zone soil water predictions. Article first published online: 15 March 2011. DOI: 10.1002/hyp.8034.

[37] Jung, B.S. & Karney, B., Fluid transients and pipeline optimization using GA and PSO: the diameter connection. *Urban Water Journal*, **1(2)**, pp. 167–176, 2004.

[38] Muleta, M.K., Boulos, P.F., Orr, C.H. & Ro, J.J., Using Genetic Algorithms and Particle Swarm Optimization for Optimal Design and Calibration of Large and Complex Urban Stormwater Management Models. *ASCE Conference Proceeding* **200**, 113, DOI:10.1061/40856(200)113, 2006.

[39] Kumar, D.N. & Reddy M.J., Multipurpose reservoir operation using particle swarm optimization. *Journal of Water Resources Planning and Management, ASCE*, **133(3)**, pp. 192–201, 2007.

[40] Reddy, M.J. & Kumar, D.N., Optimal reservoir operation for irrigation of multiple crops using elitist-mutated particle swarm optimization. *Hydrological Sciences Journal*, **52(4)**, pp. 686–701, 2007.

[41] Reddy, M.J. & Kumar, D.N., Multi-objective particle swarm optimization for generating optimal trade-offs in reservoir operation. *Hydrological Processes*, **21(21)**, 2007.

[42] Afshar, M.H., Rebirthing particle swarm optimization algorithm: Application to storm water network design. *Canadian Journal of Civil Engineering*, **35(10)**, 2008.

[43] Scheerlinck, K., Pauwels, V.R.N., Vernieuwe, H. & de Baets, B., Calibration of a water and energy balance model: Recursive parameterization versus particle swarm optimization. *Water Resources Research*, **45(10)**, doi:10.1029/2009WR008051, 2009.

[44] Cheng, C., Liao, S., Wu, X. & Chau, K.W., Two stage particle swarm optimisation for long-term operation of a hydroelectric power system. *Transactions Hong Kong Institution of Engineers*, **16(3)**, pp. 8–14, 2009.

[45] Geem, Z.W., Particle-swarm harmony search for water network design. *Engineering Optimization*, **41(4)**, pp. 297–311, 2009.

[46] Zhang, J., Wu, Z., Cheng, C.-T., & Zhang S.-Q., Improved particle swarm optimization algorithm for multi-reservoir system operation, *Water Science and Engineering*, **4(1)**, pp. 61–73, 2011.

[47] Baltar, A.M. & Fontane, D.G., Use of multiobjective particle swarm optimization in water resources management. *Journal of Water Resources Planning and Management,* **134(3)**, pp. 257–268, 2008.

[48] Reddy, M.J. & Kumar, D.N., Performance evaluation of elitist-mutated multi-objective particle swarm optimization for integrated water resources management. *Journal of Hydroinformatics*, **11(1)**, pp. 78–88, 2009.
[49] Yu, L. & Wang, J.-q., Analysis of water resources planning based on particle swarm optimization algorithm. *2nd International Workshop on Intelligent Systems and Applications (ISA)*, Wuhan, China, pp. 1–3, 2010.
[50] Chau, K., River stage forecasting with particle swarm optimization. *Lecture Notes in Artificial Intelligence*, **3029**, pp. 1166–1173, 2004.
[51] Chang, W., A novel particle swarm optimization for optimal scheduling of hydrothermal system. *Energy and Power Engineering*, **2(4)**, 2010.
[52] Walski, T.M., State of the art: Pipe network optimization. *Computer Applications in Water Resources, ASCE*, 1985.
[53] Goulter, I. C., Systems analysis in water-distribution network design: From theory to practice. *Journal of Water Resources Planning and Management, ASCE*, **118(3)**, pp. 238–248, 1992.
[54] Lansey, K.E., Optimal design of water distribution systems. *Water Distribution System Handbook*, ed. L.W. Mays, McGraw-Hill: New York, 2000.
[55] Wu, Z.Y. & Walski, T., Self-adaptive penalty cost for optimal design of water distribution systems. *Journal of Water Resources Planning and Management*, **131(3)**, pp. 181–192, 2005.
[56] Izquierdo, J., Pérez, R. & Iglesias, P.L., Mathematical models and methods in the water industry. *Mathematical and Computer Modelling*, **39(11–12)**, pp. 1353–1374, 2004).
[57] Goulter, I.C. & Coals, A.V., Quantitative approaches to reliability assessment in pipe networks. *Journal of Transportation Engineering*, **112(3)**, pp. 287–301, 1986.
[58] Goulter, I.C. & Bouchart, F., Reliability-constrained pipe network model. *Journal of Hydraulic Engineering*, **116(2)**, pp. 211–229, 1990.
[59] Matías, A.S., *Diseño de redes de distribución de agua contemplando la fiabilidad, mediante algoritmos genéticos*. *PhD thesis*, Universidad Politécnica de Valencia, Valencia, Spain, 2003.
[60] Dandy, G.C. & Engelhardt, M.O., Multi-objective trade-offs between cost and reliability in the replacement of water mains. *Journal of Water Resources Planning and Management*, **132(2)**, pp. 79–88, 2006.
[61] Savic, D.A., (2005) Coping with risk and uncertainty in urban water infrastructure rehabilitation planning. Acqua e città – i convegno nazionale di idraulica urbana, S'Agnello (NA), 28–30.
[62] Martínez-Rodríguez, J.B., Montalvo, I., Izquierdo, J. & Pérez-García, R., Reliability and tolerance comparison in water supply networks. *Water Resources Management*, **25**, pp. 1437–1448, 2011.
[63] Mays, L.W., Review of reliability analysis of water distribution systems. *Stochastic Hydraulic*, Rotterdam, The Netherlands, 1996.
[64] Ostfeld, A. & Shamir, U., Design of optimal reliable multiquality water-supply systems. *Journal of Water Resources Planning and Management*, **122(5)**, pp. 322–333, 1996.

[65] Tanyimboh, T.T. & Setiadi, Y., Joint layout, pipe size and hydraulic reliability optimization of water distribution systems. *Engineering Optimization*, **40(8)**, pp. 729–747, 2008.
[66] Martínez, J.B., Quantifying the economy of water supply looped networks. *Journal of Hydraulic Engineering – ASCE*, **133(1)**, pp. 88–97, 2007.
[67] Montalvo, I., *Diseño óptimo de sistemas de distribución de agua mediante Agent Swarm Optimization*. Doctoral dissertation, Universitat Politècnica de València, Valencia, Spain, 2011.