Additional Information

Water Distribution System Computer-aided Design by Agent Swarm Optimization

I. Montalvo

3SConsult GmbH, Karlsruhe, Germany

J. Izquierdo*, R. Pérez-García

FluIng-IMM, Universitat Politècnica de València, Valencia, Spain

&

M. Herrera

BATir Département, Université Libre de Bruxelles, Brussels, Belgium

Abstract: *Optimal design of water distribution systems (WDS), including the sizing of components, quality control, reliability, renewal and rehabilitation strategies, etc., is a complex problem in water engineering that requires robust methods of optimization. Classical methods of optimization are not well suited for analyzing highly-dimensional, multimodal, non-linear problems, especially given inaccurate, noisy, discrete and complex data. Agent Swarm Optimization (ASO) is a novel paradigm that exploits swarm intelligence and borrows some ideas from multi-agent based systems. It is aimed at supporting decision-making processes by solving multi-objective optimization problems. ASO offers robustness through a framework where various population-based algorithms co-exist. The ASO framework is described and used to solve the optimal design of WDS. The approach allows engineers to work in parallel with the computational algorithms to force the recruitment of new searching elements, thus contributing to the solution process with expert-based proposals.*

1 INTRODUCTION

In many fields of science and engineering, the optimization techniques employed have conditioned the way in which those problems have been approached over the years. For example, the use of linear programming implied the linearization of the objective function and constraints. Techniques based on the gradient required the derivability of the function to be guaranteed – or the division of the problem into parts so that only differentiable terms could be used. For years, consciously or unconsciously, the problem was adapted to the optimization techniques in use. But these techniques have been shown to be poorly suited for many real-world highly-dimensional, multimodal, strongly non-linear problems; while, at the same time, they must process inaccurate, noisy, discrete and complex data. Robust methods of optimization are often required to generate suitable results.

Many researchers have embarked on the implementation of various evolutionary algorithms: genetic algorithms (GA) (Goldberg, 1989); ant colony optimization (ACO) (Dorigo et al., 1996); particle swarm optimization (PSO) (Kennedy and Eberhart, 1995); simulated annealing (Kirkpatrick et al., 1983; Černý, 1985); shuffled complex evolution (Duan et al., 1993); harmony search (Geem et al., 2001); and memetic algorithms (Moscato, 1989), among many others. These derivative-free global search algorithms have been shown to obtain good and engineering-relevant solutions for large-scale real-world problems of a varied nature (Adeli and Kumar, 1995; Sarma and Adeli, 2000;

Kim and Adeli, 2001; Vrugt and Robinson, 2007; Arumugan and Rao, 2008; Dridi et al., 2008; Izquierdo et al., 2009; Herrera et al., 2009; Vitins and Axhausen, 2009; Zeferino et al., 2009; Plevris and Papadrakakis, 2011; Jafarkhani and Masri, 2011; Putha et al., 2012; Sgambi et al., 2012; Hsiao et al., 2012; Tao et al., 2012; Shafahi et al., 2013; Hejazi et al., 2013; Fuggine et al., 2013; Amini et al., 2013; Duchesne et al., 2013, to cite just a few). More recently, these algorithms have started to be adapted to multi-objective problems in various areas (Deb, 2001; Savic, 2002; Vamvakeridou-Lyroudia, 2005; Dandy and Engelhardt, 2006; Payá et al., 2008; Janson et al., 2008; Montalvo et al., 2010; Xie and Waller, (2012); Raich and Liszkai, (2012), among others).

Some of the advantages of the growing use of evolutionary algorithms in optimization include (Montalvo, 2011):

- Evolutionary algorithms can deal with problems containing continuous variables as well as naturally discrete variables and binary variables in the yes/no decisions that are so frequent in many real-world problems.
- Evolutionary algorithms only work with the information of the objective function and this prevents complications associated with the determination of the derivatives and other auxiliary information.
- Evolutionary algorithms are generic optimization procedures and can directly adapt to any objective function, even if it is not described by closed expressions, and is described by complex procedures.
- Because evolutionary algorithms work with a population of solutions, many solutions can be obtained that can be of great interest from an engineering point of view.
- Because evolutionary algorithms are versatile and flexible, analyzing systems under various loading conditions, or forcing terms, can be performed within the optimal design process.

Thus, unlike most of the classical optimization algorithms, evolutionary algorithms enable the use of any form of quantitative assessment of the desired objectives without conditioning the approach to the problem.

Nevertheless, despite its benefits, each algorithm has its own drawbacks and is better adapted to certain problems than to others. The heuristics behind a certain evolutionary algorithm endow its elements (agents) with specific capabilities for efficiently solving some kinds of problems, while being inefficient with problems of a different nature. For example, the first author has found that PSO experienced major difficulties in a problem of container manipulation in a harbor and even in the design of WDS – the application-object of this paper – when the candidate diameters for the pipes were listed in an unordered way (unpublished results). In these cases, the main reason is

clearly the lack of a logical order in the options, which makes the agents roam too abruptly.

In fact, the individual knowledge (specific current information and variables states) of an agent is very limited and probably subjective. An effective search of optimal solutions is only possible as a result of interaction among many agents. Agent-based applications are becoming more frequent day by day (Gutiérrez-García and Sim, 2012; Rodríguez-Seda et al., 2012 ; Nejat and Damnjanovic 2012; Badawy et al., 2013; Fougères and Ostrosi, 2013; Pinto et al., 2013, to cite only a few). Agents possess individual behavior, and associations of interacting agents result in collective structures, called swarms, that represent the emergent behavior of groups of agents. These structures are also considered agents at a higher abstraction level. Swarms, in turn, can interact with other existing swarms.

A particle from a PSO swarm, an ant from an ACO system, and a chromosome from a GA structure do exhibit different behavior. Yet, they all share a common feature: each represents a potential solution for the problem in hand. In a combined environment, a PSO particle could help reinforce the pheromone on ant paths, an ant could be reproduced with a chromosome; a chromosome could be the leader of a particle swarm, and so on. This conceptual framework is not a fixed meta-heuristic but a dynamic environment where a new algorithm (agent or swarm) can be added in runtime to contribute to the solution of the given problem. In this framework, algorithms share a common pool of solutions. This means that all the algorithms share the solutions embodied by their populations. This sharing is very simple from a computational point of view. Each algorithm evolves in its own way, but is not restricted from sharing the solution information of the other algorithms. By sharing the information, one algorithm can benefit from what any other is doing. This idea makes success possible in a wider scope of problems. In addition, agents are endowed with sets of problem-dependent specific rules with two clear objectives, namely fine-tuning the agent behavior to specialize in the optimization problem being tackled, while, at the same time, reducing the search space, thus enabling decisions to be produced with increased reliability and within a reasonable time frame.

The framework broadly described above has been termed ASO, for Agent Swarm Optimization. The main contribution of ASO derives from the combination of: a multi-agent point of view; the coexistence of different agent breeds (including humans); the interaction among them; and the fact that agents can be endowed with problem-specific rules. ASO dynamically combines the strengths of multiple metaheuristics and demonstrates good performance, especially in the field of WDS design. ASO can thus be considered as an innovative computing application that can be efficiently used in civil and infrastructure engineering.

In the rest of the paper, the necessary details to completely develop ASO are provided and a real-world application, namely the design of such crucial infrastructures as water distribution systems, is addressed. The design optimization of a large WDS from a multi-objective standpoint within a reasonable time frame remains a challenging and burdensome problem, mainly due to the very high level of computational complexity involved, since numerous, expensive hydraulic simulations are needed. To exemplify the algorithm, a real-world network is considered and various solutions provided.

The rest of the paper, which is substantive extension of Izquierdo et al. (2012), is organized as follows. Firstly, the multi-objective framework is defined and the ASO features presented. The problem of the WDS design is then outlined and qualitative results for a case-study are presented. The conclusions close the paper.

2 MULTI-OBJECTIVE OPTIMIZATION USING ASO

Economic criteria usually condition the difference between a solution and a better solution. Nevertheless, decision-making also needs to fulfill many other technical and non-technical targets and constraints that are involved in the problem in hand. A typical scenario includes the consideration of multiple conflicting objectives and involves finding an acceptable trade-off between them. We spare the readers the basic definitions of multi-objective optimization, including the concept of dominance (Pareto, 1896). ASO fully incorporates the concept of dominance to determine which solutions will be considered as dominant.

2.1 ASO in multi-objective optimization

The idea behind ASO is a PSO-based environment developed by the authors to mimic the judgment of an engineer (Montalvo et al., 2010). It was built by using various prior features and improvements regarding swarm intelligence. Multi-agent systems, and the necessary adaptation to multi-objective performance, including human interaction, are also integrated in ASO.

The first feature derives from the philosophy behind PSO. It consists of a variant of the standard PSO that can deal with various types of variables, and includes a mechanism for increased diversity (Montalvo et al., 2008; Herrera et al., 2011). This enables self-management of the parameters involved so that engineers are spared the task of parameter selection and fine-tuning (Montalvo et al., 2009). Although the authors have applied this algorithm mainly to WDS design, it has proven very efficient in solving optimization problems in other fields (Izquierdo et al., 2008a; Herrera et al., 2009).

The emergent behavior of a PSO swarm is strongly reminiscent of the philosophy behind the multi-agent (MA)

paradigm (Sycara, 1998; Wooldridge, 2002). In an MA system each agent has a limited capacity and/or incomplete information to resolve a problem – and therefore has a limited view of the solution. There is no overall control of the system; values are decentralized and the computation is asynchronous (Sycara, 1998). Each agent acting alone cannot solve the problem in its entirety, but a group of agents, with the coexistence of differing views, is better able to find a solution by interacting together. This idea can be clearly extrapolated to the case of multi-objective optimization, since the result of the many interactions occurring within an MA system is improved performance. Associations of agents interacting among themselves result in a collective structure, called a swarm, that represents the collective behavior of a group of agents. This structure can also be considered as an agent on a higher abstraction level. Each swarm has its own behavior and, in its turn, is able to interact with other swarms.

Taking into account the desirability of solving real-world, multi-objective optimization problems, and the benefits offered by MA systems, a departure from the standard behavior of particles in PSO must be performed. In addition to using the concept of dominance, various other aspects must also be re-stated. We then re-define the concept of leadership, adopt a normalization procedure, propose two mechanisms to enrich the Pareto front, incorporate human interaction within the framework, and propose endowing the agents with specific, problem-dependent behavioral rules. As stated in the Introduction, these are the main contributions of this work.

2.1.1 Leadership

Firstly, the concept of leadership in a swarm must be re-defined. The most natural option is to select as leader the closest particle to the so-called the ideal or *utopia point* in the objective space (Wierzbicki, 1998). The *utopia point* is 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 something unknown at the start (and perhaps during the whole process). Accordingly, we use a dynamic approximation of this *utopia point*, termed a *singular point* (see Fig. 1), which is updated with the best values found so far during the evolution of the algorithm (Montalvo et al., 2010). To enrich the solution and eliminate the natural tendency of new solutions to approach the *singular point* (Vrugt et al., 2003, 2006), searches in certain desired regions of the Pareto front must be encouraged. To this purpose, in ASO new sub-swarms, following instructions given by the user/engineer are added to specialize around displaced *singular points*. This fits the methodology of *reference point* approaches (Wierzbicki, 1998), in which aspiration levels and, possibly, reservation levels, may be provided by the user, and which will mainly reflect his/her intuition or understanding of the problem.

Objectives will be differently weighted to determine the location of the *singular point* corresponding to each sub-swarm.

2.1.2 Normalization

Secondly, because each objective may be expressed in different units, it is necessary to make some regularization for evaluating distances in the objective space. Once a regularization mechanism has been enforced – to establish the distance between any two objective vectors the Euclidean distance between them is calculated. Note that the worst and best objective values are not usually known a priori; they are updated while the solution space is being explored.

2.1.3 Pareto front enrichment

Two-dimensional representations of the concept of a *singular point* can be seen in Fig. 1. In some problems, the most interesting solutions are located near the *singular point* (Vrugt et al., 2006) and not too far from the peripheral areas of the Pareto front. In these cases, instead of seeking a complete and detailed Pareto front, only precise details around the *singular point* are obtained. Nevertheless, situations can occur, as shown in Fig. 1 (right), when unbalanced Pareto fronts develop with respect to the *singular point*. Consequently, poorly detailed sections on the Pareto front may appear that may be worth exploring. It seems plausible that problem complexity is the cause of this asymmetry in many real-world, multi-objective optimization problems.
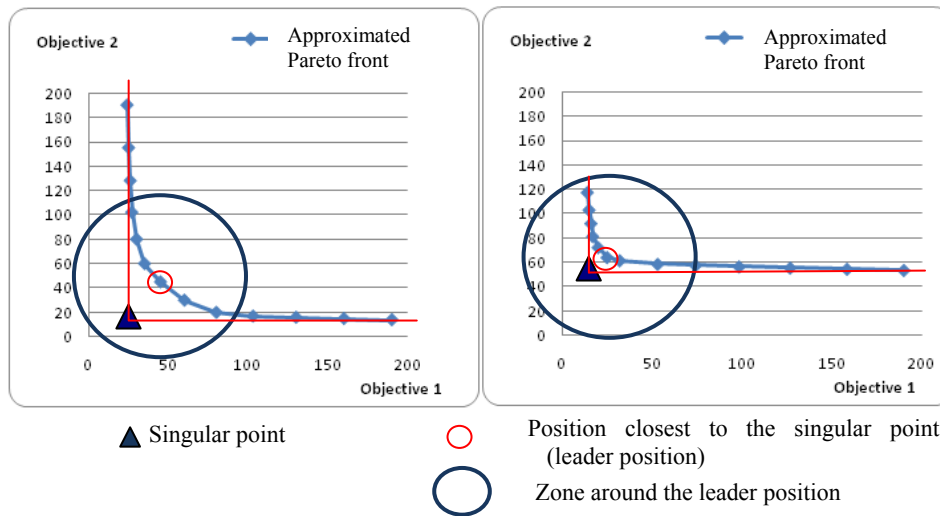
Figure 1 Two examples of *singular point* in an approximated Pareto front

It is not easy 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. Those decisions are strongly dependent on the people making the decision and on the problem itself.

In fact, for making final decisions in real-world problems where all objectives do not go in the same direction, additional information is always needed (Coello Coello et al., 2007). This additional information can be established a priori, for example when objectives are represented in only one expression by giving a specific weight to each of them. Additional information can also be used at the end of the search process for deciding, for example, which solution from a Pareto front should be selected. As a third possibility, the one used in this paper, additional information can be used during the search process itself.

Users relying on their intuition or understanding of the problem may provide suitable *reference points* (Wierzbicki, 1998) to express their aspiration levels. In our case this idea is implemented not only for deciding which regions of the Pareto front are more interesting, but also for proposing solutions that may lead or enrich the way other agents behave. The addition of new sub-swarms by the user during the solution search process can help solve this problem but may be insufficient. Therefore, we describe one possible approach based on dynamic population increases to raise the Pareto front density (Reyes-Sierra and Coello Coello, 2006; Dupont et al., 2008), and another approach based on human-computer interaction to complete poorly represented areas of the Pareto front. This is achieved in runtime during the execution of the algorithm.

2.1.4 Agent cloning

In the first approach, during the search process, swarms are able to increase their populations autonomously when needed in order to better define the Pareto front: an agent 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. The cloned agent inherits the experience (information) of the parent-agent. As a consequence, the solution represented by the cloned agent will no longer be exploited by the original agent, since the latter will continue exploiting the original information that entitled it to be a point in the Pareto front. In this way, the population is increased dynamically and so represents with more detail the Pareto front of the problem being solved. Nevertheless, greater densities on the Pareto front must be restricted to the case where the new clone has at least one of its neighbors located further away than some minimal permissible distance in the objective space. For example, in

Figure 2 (left), agent J, whose objective vector is located at position P_J, finds a new position $NewP_J$. The consequence is represented in Figure 2 (right): a new agent k is added to the swarm by cloning the particle with objective vector at position $NewP_J$, while particle J will continue to be active and considers the point P_J as its best position in the objective space. This happens because the new objective vector P_k has at least one neighbor located further away than the minimal permissible distance in at least one of the objectives. In Figure 2 (right), the particle with objective vector at the left of P_k is located at a distance, with respect to objective 2, that is greater than the minimal distance considered for the increase of density in the Pareto front. It has to be noted that two objective vectors are considered to be neighbors when no other objective vector is located between them in at least one of the objectives considered in the problem.
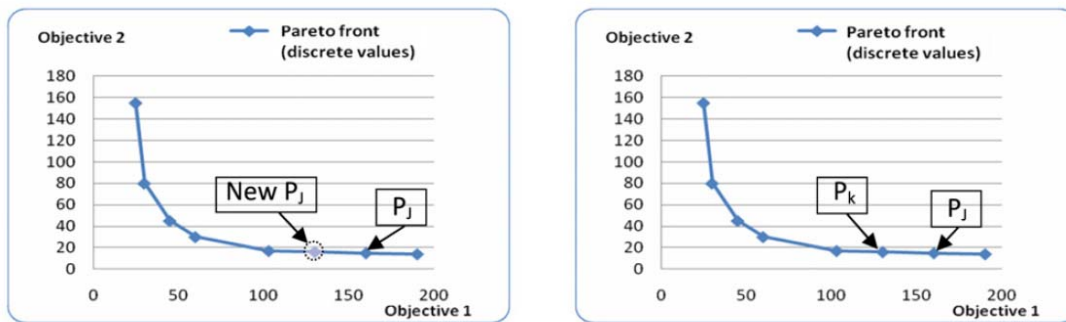

Figure 2 An example of agent cloning

The incorporation of new agents in runtime makes it possible for the algorithm to evolve both in its structure and in its capacity to find good solutions. Nevertheless, when more solutions are incorporated in the Pareto front, it takes longer to check if a new potential solution can also be part of the Pareto front. The approach we describe in 2.1.6 helps in this issue.

2.1.5 Human computer interaction

The second way to enrich the Pareto front is by human interaction. As mentioned, users are allowed to add new swarms for searching in the desired region of the objective space. It is achieved in runtime during the execution of the algorithm. Specialists interacting with the algorithms are able to decide relevant regions of the Pareto front for adding new swarms, and use their expertise for proposing new solutions to existent swarms. This interaction enables the incorporation of human behavior, so the humans turn out to be other members of the process by proposing new candidate solutions. This means that a new solution may be

proposed to the algorithm at any time and the algorithm should be able to fit it on the Pareto front, if appropriate. If accepted, a new *singular point* is added and a new swarm is created with the same characteristics as the first created swarm. Proposed solutions can even become leaders of the swarm if they are good enough.

At this point, human behavior begins to have a proactive role during the evolution of the algorithm; and it can be said that the times when experts just sat in front of their computers waiting for results are over. This represents the main difference with "classical" multi-agent systems, where agents are normally considered as part of a software code; in our case, humans are also considered as agents actively involved in the solution search process.

The participation of several human agents with different perspectives on a problem is very close to what happens in the practice of engineering decision-making, where politicians, economists, engineers, and others are 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.

2.1.6 Swarm hierarchies

When the number of solutions on the Pareto front increases, many comparisons are needed before concluding that a new solution also deserves belonging to the Pareto front. A time consuming task for any population-based algorithm used to solve multi-objective optimization problem is determining which solutions belong to the Pareto front when there are already many solutions belonging to the front (Deb, 2001). In ASO, various (probably many) swarms may coexist. Thus the various (probably many) new solutions they generate will pose a hard bottleneck for evaluation. With just one single instance of the Pareto Front it is impossible to aspire to any kind of parallelization. A hierarchy of swarms is used for fulfilling this task by profiting from parallel and distributed computing. Different swarms specialize in different subsets of the approximated Pareto front. To discover if a solution belongs to the approximated Pareto front, swarms first check if the solution is dominated by any of those solutions belonging to its own Pareto subset. If the solution is not dominated then the swarm asynchronously asks its superior to check if the solution is dominated or not. The process is repeated at every hierarchical level if the solution is found to be non-dominated; in that case all swarms involved in the checking process will have the information about the new non-dominated solution. While swarms are waiting for asynchronous responses from their superior, solutions are assumed to belong to their Pareto subsets. A swarm, when receiving a request to check if a solution belongs to its own subset of the Pareto front, uses only those solutions belonging to its own Pareto subset that did not come from the requesting swarms. Figure 3 represents a hierarchy of swarms.
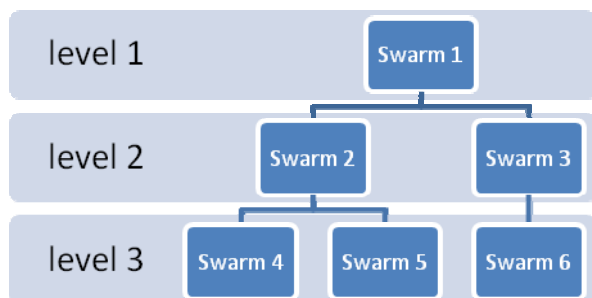
Figure 3 Hierarchy of swarms

When new swarms are added to solve a problem they must be placed in the hierarchy of swarms already solving the problem. Each swarm has a maximum number of connected swarms in the lower level. New swarms will be connected to any of those existent swarms that still have connection capacity.

The combination of various swarms within the same algorithm is efficient because it conducts a neighborhood search in which each of the swarms specializes, and the best improvement step in terms of Pareto optimality is followed to yield a new solution. The practice of incorporating different search mechanisms also reduces the probability of the search becoming trapped in local optima.

The implementation was made using the capabilities of Microsoft.Net Framework 4.0 to run different swarm instances in parallel and synchronize their work. The distributed computing was based on the capabilities of Windows Communication Foundation (included in Microsoft.Net Framework) to communicate and synchronize swarm instances running on different processes/machines.

2.1.7 Rule-driven agents

Within the ASO framework, GA chromosomes, ACO ants, and PSO particles, in particular, and evolutionary algorithm agents in general, are genuine agents in a multi-agent system. They can be endowed with specific, problem-dependent behavioral rules purposely designed to heuristically approach the solution process. These rules try to mimic the judgment of a human expert when approaching a solution to a problem.

Evolutionary algorithms generally have not previously taken advantage of this feature and, as a result, have been bound to analyze a larger solution space than necessary. Including rules may reduce the search space by several orders of magnitude.

As a consequence, the solution is both efficient and closer to reality. Efficiency derives from the fact that just checking a number of usually simple rules avoids many expensive calculations or simulations (hydraulic simulations in the application we present in this paper). Finally, the fact that the rules have strong problem-dependent meanings definitely brings the solution nearer to reality. For example, in the application dealt with in this paper, one rule just states that downstream pipe diameters should be normally equal to or smaller than upstream pipe diameters. In the section devoted to the application of ASO we present other examples of problem-specific rules.

3 SPECIAL CHARACTERISTICS AND ADVANTAGES OF ASO

Agent swarm optimization enables the resolution of problems using a multi-objective approach and integrates various algorithms in runtime on a single platform. The mixture of different algorithms and the incorporation of new agents in runtime within ASO are possible because ASO makes use of parallel and distributed computing to enable the incorporation of new agents, as well as the asynchronous behavior of agents, and the inclusion of

different users working on the same problem. The inclusion of different users is particularly interesting because additional information or criteria is, in general, used at some stage of the solution of multi-objective optimization problems. ASO gives the users the opportunity to participate as active agents, enabling them to propose potential solutions to the problem in runtime. Proposals made by users can be used by other actors to reorient or enhance their searches. Users benefit from the speed and search ability of artificial agents, while agents take advantage of the creativity, ideas, and experience of users.

Depending on the needs of the project, or any requirements to be imposed in the near future, it may be necessary to take into account the different aspects and various ways in which the objectives can be evaluated. The way to evaluate and decide the goals is an evolutionary process, and in this sense the optimization algorithm must be able to adapt to new situations and conditions. The inclusion of additional objectives or changes in the method of evaluating existing objectives can be made immediately with the presented algorithm.

Designing a general algorithm able to solve any present and future optimization problem may be an endless endeavor. However, such an algorithm must be flexible and extensible enough to attempt this task. ASO may be able to meet this philosophy.

Firstly, new agents can be included at any time, and these may comprise agents with different behaviors that may be better suited to solving the optimization problem in hand. Thus, from the optimization point of view, ASO can be seen as just a method. However, secondly, and maybe more importantly, from the software engineering perspective, ASO is certainly a framework (an extensible framework) that can meet the requirements we encounter today, and may be able to meet new requirements that will appear in the future.

In addition, ASO can also be considered a computer-aided framework from the software engineering point of view because the core of ASO is an abstraction that provides general functionalities for including different evolutionary optimization algorithms in the solution search process that is executed for solving a specific optimization problem. These functionalities are provided by the implementation of the various software interfaces defined in ASO. The same applies to including new evolutionary algorithms that are intended to work within the framework of ASO.

3.1 The algorithm

An idea of how ASO works may be summarized in the following pseudo-code – presented for the case of PSO swarms for the sake of simplicity. Specific details about how the relevant operators work for WDS optimization have been added in italics.

For each swarm in parallel:
1. Connect to the hierarchical structure.
2. Run the consideration of external requests in parallel.
3. Set up parameters and initialize the number of iterations to zero.
4. Generate a random population of M agents.
   Generate M instances of networks with randomly chosen variables (diameters, etc.).
5. Evaluate the fitness of the agents and set the best location for every agent equal to its current location.
   For each network (agent) the various objectives (cost, reliability, etc.) are evaluated; this implies the analysis of the networks using a hydraulic simulator, and various calculations specified in Section 4.
6. Form the Pareto front; make a list of agents belonging to it.
7. Update the *singular point*.
8. Find the closest agent to the *singular point* and establish it as the swarm leader.
9. While not in termination-condition, do the following:
   a. Execute asynchronously from $i = 1$ to number of agents.
      START
      i. Ask agent i to change its position in the solution space.
         Modify the candidate network according to the EA requirements (mutation, position updating, etc.).
      ii. Calculate the new fitness function vector for agent i in its new position.
         For this network (agent) calculate the various objectives.
      iii. If the new fitness function vector for agent i dominates the fitness function vector that the agent had before moving to the new position, set the new position as the best position found up to now by agent i.
      iv. If agent 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 neighbors located further than the minimal permissible distance from any of the objectives, then add a new agent j (a clone of i) located at the current position of i;
         else
         try to add (if possible) the agent i (at its new position) to the Pareto front; if the agent is added, remove from the list any dominated solution; eliminate dominated clones from the swarm. This step involves the interchange of swarms located at different hierarchical levels.
      v. If agent i is closer to the *singular point* than any other particle in the swarm then set agent i as the leader of the swarm with regard to the *singular point*.

vi. If agent i is not currently the leader of the swarm, but coincides in position with the leader, then re-generate agent i randomly.
   END
b. Increase the iteration number.
10. Show the Pareto front and related results.

These steps may be understood also for the general case in which agents are different from particles in PSO. In that case, the main changes would happen at step (i.) where the agent changes its own position. Step (vi.) should only be used for agents that behave similarly to the particles in PSO for enriching the population diversity. More information about PSO as applied to water distribution system design and the advantages of enriching the search can be found in Montalvo et al. (2008, 2009, 2010) and Montalvo (2011).

The combination of all these calculations and possibilities is clearly inaccessible from a practical point of view without a computer application that offers an appropriate user interface. In the next paragraph we present the problem of WDS design and provide the solution for a real-world case-study using a specific computer tool implementing ASO.

4 APPLICATION TO WATER DISTRIBUTION SYSTEM DESIGN

In this section we first concisely describe the problem of water distribution system design. We then provide details about a real-world water distribution system and the solution obtained using the ideas and the framework described above.

4.1 The problem: water distribution system design

Water distribution system design is a wide and open problem in hydraulic engineering that may involve the addition of new elements in a system; the rehabilitation or replacement of existing elements; decision-making on operation; reliability and protection of the system; among other actions. Designs are necessary in order to carry out new configurations, or to enlarge or improve existing systems to meet new conditions (Goulter and Coals, 1986; Goulter and Bouchart, 1990; Walski, 2003).

Various objectives may be considered in the WDS optimal design problem. In this section, we describe these objectives, namely: cost of components; satisfaction of water demand quality; adherence to hydraulic constraints; and resilience of the system during stressed conditions.

Although the basic variables of the problem are the diameters of the new pipes, additional variables that depend on the design characteristics of the system may also be required: storage volume, pump head, type of rehabilitation to be carried out for various parts of the network, etc. The various objectives will always depend on these variables.

The correct approach to assess any objective fitness is important when defining the objective function, which has to be fully adapted to the problem under consideration in terms of design, enlargement, rehabilitation, operational design, etc. In general, for a specific network, some of the areas may be already built, while others must be newly designed. For the existing pipes several actions may be taken: rehabilitation (with several available alternatives with associated costs), replacement, simply duplication, or no action. In addition, it is important that the objective functions reflect with utmost reliability the system during its entire lifetime (Kleiner et al., 2001). Complete definitions of the objectives, which objectives to use, and the best way to evaluate them could be assembled in a separate topic that is outside the scope of this paper. Nevertheless, we present here some classical and basic ideas to consider and evaluate objectives in water distribution system design.

4.1.1 *Cost of components*

A general objective cost function includes several terms, several scenarios or working conditions, and a time horizon for the whole infrastructure. The function

$$C_{\text{WDS}} = \sum_k \left[P_{\text{wc}}^k \left(a_{\text{pipe}} \cdot C_{\text{pipe}} + a_{\text{pump}} \cdot C_{\text{pump}} + \right. \right. \\ \left. \left. + a_{\text{valv}} \cdot C_{\text{valv}} + a_{\text{tank}} \cdot C_{\text{tank}} + C_{\text{Oper}} \right) \right] \quad (1)$$

includes various individual working conditions (WC), which depend on the values adopted by two types of variables: namely, demand models and roughness coefficient values (accounting for pipe ageing); P_{wc}^k represents the probability for the k-th working condition. Typically, independent random variables are used to model both types of variables. Under the assumption that design is made to work for N_{dm} demand models and N_{rc} sets of roughness coefficient values, the design is performed for $N_{\text{wc}} = N_{\text{dm}} \cdot N_{\text{rc}}$ working conditions. These conditions have individual probabilities, P_{wc}^k, $k = 1, \ldots, N_{\text{wc}}$, given by the product of the corresponding probabilities regarding demand models and roughness values. This function also considers the operational costs of the network, C_{Oper}, along a certain temporal horizon and this obliges the use of the amortization rates, a_{xxx}, to multiply any of the investment costs, namely, C_{pipe}, C_{pump}, C_{valve}, and C_{tank}, representing costs for pipes, pumping systems, valves, and storage tanks, respectively.

In general, C_{WDS} is a non-linear, partially stochastic function dependent on continuous, discrete, and binary variables.

4.1.2 *Hydraulic constraints*

Among the several formulations available to define these constraints, we use the one that considers the continuity equations, which are linear, plus the energy equations,

typically non-linear. The complete set of equations may be written, by using block matrix notation as

$$\begin{pmatrix} A_{11}(q) & A_{12} \\ A_{12}^t & 0 \end{pmatrix} \begin{pmatrix} q \\ H \end{pmatrix} = \begin{pmatrix} -A_{10}H_f \\ Q \end{pmatrix}, \qquad (2)$$

where A_{12} is the connectivity matrix describing the way demand nodes are connected through the lines; q is the vector of the flowrates through the lines; H the vector of unknown heads at demand nodes; A_{10} describes the way fixed head nodes, with components in vector H_f, are connected through the lines, and Q is the vector of demands. Finally, $A_{11}(q)$ is a diagonal matrix accounting for the line resistances and characterizing potential pumps.

System (2) is a non-linear problem, whose solution is the state vector $x = (q^t, H^t)^t$ (flowrates through the lines and heads at the demand nodes) of the system.

Since most water systems involve a huge number of equations and unknowns, system (2) is usually solved using some gradient-like technique. Various tools to analyze water networks using gradient-like techniques have been developed. Among them, EPANET2 (Rossman, 2000), is used in a generalized way.

To be integrated in the algorithm later described, we have modified the EPANET2 Toolkit to support pressure-driven demands as described in Xu and Goulter (1997); the idea of pressure-driven demands has also been considered in other works (Wu et al., 2006; Giustolisi et al., 2008, among many others).

The integration of the extended version of EPANET2 to run different analyses or simulations for potential solutions of the problem is performed during the optimization process that is developed within the evolutionary algorithms (Montalvo et al., 2007, 2008, 2009) integrated in the ASO platform.

4.1.3 *Demand satisfaction*

WDS design is typically performed subject to several performance constraints in order to achieve an adequate service level. The most used constraint requires a certain minimum pressure level at each node of the system. Other constraints may include maximum pipe flow velocities, and minimum concentrations of chlorine, for example. For many years nodal pressure constraints were considered as strong constraints in the sense that they should be strictly satisfied. Nevertheless, the possibility of violating by a small degree some of these constraints opens the door to various strategies for adopting sub-optimal designs or soft solutions that may be more acceptable from other (global or political) perspectives. This is openly favored by multi-objective approaches – such as the one we present in this paper.

In many studies, these constraints have been included as penalty terms in the cost function, making up a priori decisions on the weights assigned to the various constraints.

However, in this paper we consider the satisfaction of demand as a new objective that must be fulfilled.

There are various ways of expressing a lack of compliance with conditions of pressure, velocity, disinfectant, etc. For example, an objective function considering nodal pressure given by minimum values of node pressures may be expressed by

$$f = \sum_{j=1}^{N} H(p_{min} - p_j) \cdot (p_{min} - p_j). \qquad (3)$$

Here N is the number of demand nodes in the network. For nodes with pressures, p_j, greater than this minimal value, p_{min}, the associated individual terms vanish, and the Heaviside step function H is used in this explicit expression for this purpose. Extensions of (3) may be provided to consider maximum and minimum bounds for other variables, such as limit velocities, or limiting the level of chlorine in each pipe in the case of water quality optimization. This expression is also a function of the selected pipe diameters through the hydraulic model presented in the previous subsection.

These constraints may involve values that could be a topic of discussion as, for example, the most convenient minimum pressure for supplying water in a town. Considering the lack of pressure as a second objective not only helps the algorithm to better explore the boundaries defined by the constraints (where good solutions can be found) but also enables users to evaluate the impact of a small relaxation in minimum pressure constraints. In this sense, the multi-objective approach considered in ASO plays an essential role making it possible to perform an analysis of the consequences of different pressure values. The same procedure could be easily applied to other constraints.

4.1.4 *WDS resilience*

The resilience of a WDS is assessed in terms of various types of properties, such as reliability and tolerance.

In this paper we consider a simple reliability formulation as in Xu and Goulter (1997), Tanyimboh et al. (2001), and Kalungi and Tanyimboh (2003), which takes into account only individual pipe failures, a hypothesis widely justified in the scientific literature (Park and Leibman, 1993; Gupta and Bhave, 1994; Khomsi et al., 1996; Xu and Goulter, 1997; Xu and Goulter, 1999; Martínez-Rodríguez et al., 2011).

Considering an average time for the duration of a pipe failure, reliability R is defined as:

$$R = \left(q^{nf} \, pf_0 + \sum_{k=1}^{L} q^k \, pf_k \right) \Big/ q^{req}, \qquad (4)$$

where L is the number of pipes, q^{nf} is the total flow delivered to the network when there are no failures; q^k the total flow delivered to the network when pipe k fails; pf_0 is the probability of the whole system working without

failure, pf_k the failure probability of pipe k, and q^{req} is the total required demand by the network (the sum of all nodal demands).

As WDSs should behave satisfactorily under normal conditions when there are no failures, it is worthwhile making a separate and specific analysis of their behavior under only failure states. Accordingly, the concept of tolerance to failure T has been introduced (Tanyimboh et al., 2001; Kalungi and Tanyimboh, 2003) using the expression:

$$T = \sum_{k=1}^{L}(q^k / q^{req}) pf_k \left/ \sum_{k=1}^{L} pf_k \right. . \qquad (5)$$

Any of these functions, R and T, may be used as objectives to fulfill in the optimization process. Various forms of economically quantifying reliability or tolerance problems can be found in the literature. We have used the one given in Martínez-Rodríguez et al. (2011).

4.1.5 *Specific rules*

A number of rules may be specified to approach the solution process to the problem.

For the sizing of pipes in a WDS, it is a rule of thumb to reduce the diameter of pipes as the system progresses from upstream to downstream. This rule was implemented in the agents, thus enabling designs to be produced with significant reliability and within a reasonable time frame. Not including this rule causes additional difficulties when trying to size the pipes of a large network using general evolutionary algorithms. Applying this rule in looped networks is possible thanks to the way it was implemented in the software solution. Agents do not follow a static pipe order to decide which diameter to use. Agents order the pipes to decide which will be designed first. Depending on previous decisions (current diameters already assigned to pipes), agents select the next pipe to be designed and also the candidate diameter to use. In case of 'doubt' (this is the case at the beginning of the process) agents use a random hypothesis of what upstream and downstream should be. Note that the upstream and downstream nodes assumed by the algorithm may not be the real upstream and downstream points under the various work conditions of the network. In any case, the rule guarantees that a pipe with demand nodes at both ends (and thus connected to other pipes at both ends) will never have a larger diameter than all the pipes connected to it.

Other new rules may be considered to further facilitate the process of finding solutions. Some of them have already been individually used in various works so far – for example: adjusting diameters if one or more consumption nodes experience too little or too much pressure as in Keedwell and Khu (2006); increasing the diameter of pipes joining at a certain node where the pressure is too high (Afshar et al., 2005); increasing or reducing diameters depending on the energy dissipation in certain pipes (Todini, 2000); increasing the diameters of pipes

experiencing higher unitary loses or reducing the diameters of pipes with lower loses (Saldarriaga et al., 2008), etc. Different new rules may be devised, such as reducing the diameter of a pipe if it undergoes a large reduction of chlorine concentration. These are examples of rules that, obviously, reflect the nature of the problem in hand.

4.1.6 *Implementation of ASO*

ASO and its connection with EPANET2, modified with the pressure-driven demand feature (Wu et al., 2006), has been implemented in a software package called WaterIng[1] (Montalvo, 2011), which was developed for water distribution system design and analysis. WaterIng is in constant development and may be downloaded from its website – the installation includes a file with network data as an example. An introductory guide is also available to learn the main concepts of how to design a water distribution system using the software.

The application of ASO to the most popular benchmarking problems in the WDS-design literature has produced the best solutions ever found for these problems (Montalvo, 2011). The authors have also used a preliminary version of this package that considers only PSO swarms for addressing other real-world complex problems such as the design of waste water systems (Izquierdo et al., 2007), the calibration of a WDS (Izquierdo et al., 2008b), the optimal design of a biomass supply chain at regional level (Izquierdo et al., 2008a), and the clustering of a water company database to classify pipes with the aim of rehabilitation (Díaz et al., 2008), among others.

The case-study presented in the next subsection has been completely developed with WaterIng. The multi-objective model implemented by this software has shown robustness and good explanatory outcomes. Decision makers are provided with a set of informed solutions to select the best design with regard, for example, to available resources and/or other criteria.

The platform currently integrates three of the best known evolutionary algorithms, namely, GA, ACO and PSO. The initial population size was set to 20 for the three evolutionary algorithms. Other parameters are the following. Regarding GA: mutation probability = 0.0234; crossover probability = 0.8; reproduction method = fixed intervals; crossover method = all single, mutation method = bit to bit. Regarding ACO: coefficient   = 1; coefficient = 0.5; coefficient of phero-evaporation = 0.98, factor of reward = 1.1e7, initial pheromone = 26. Finally, regarding PSO: inertia = adaptive varying with iteration number; individual and collective intelligence parameters = self-adaptive fine tuning – as described in Montalvo et al., (2009).
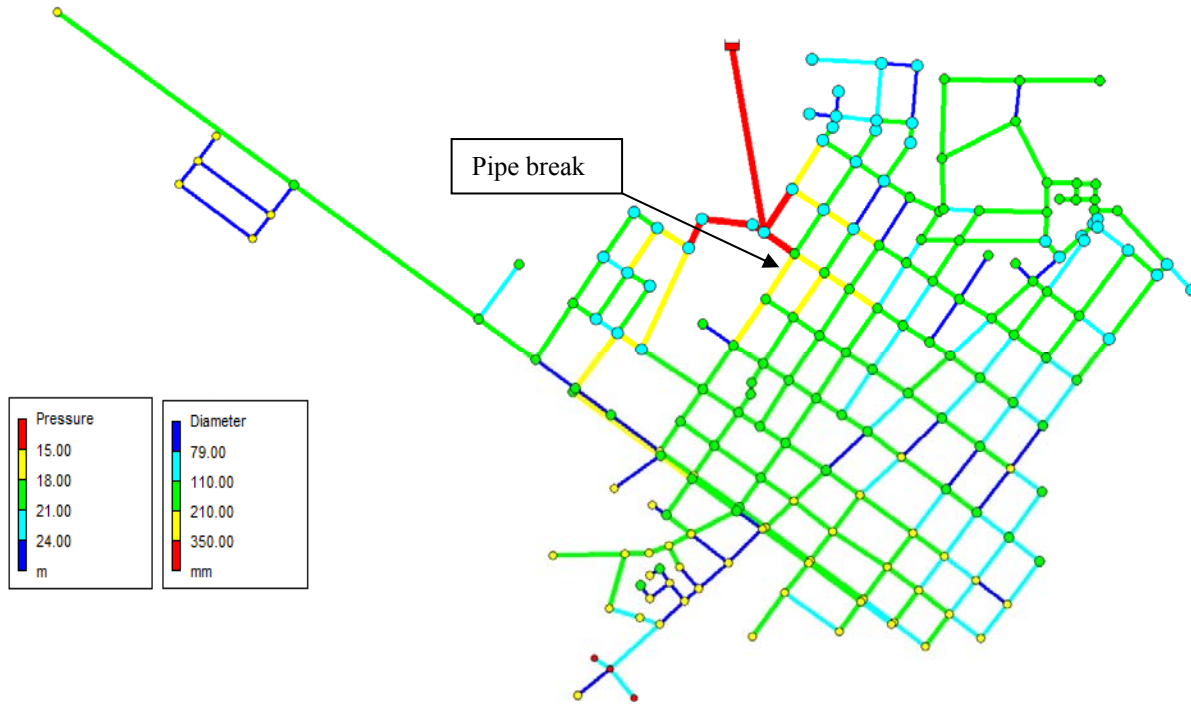
---

[1] www.ingeniousware.net

Figure 4 A Pareto design of the network under a failure scenario (with rules)
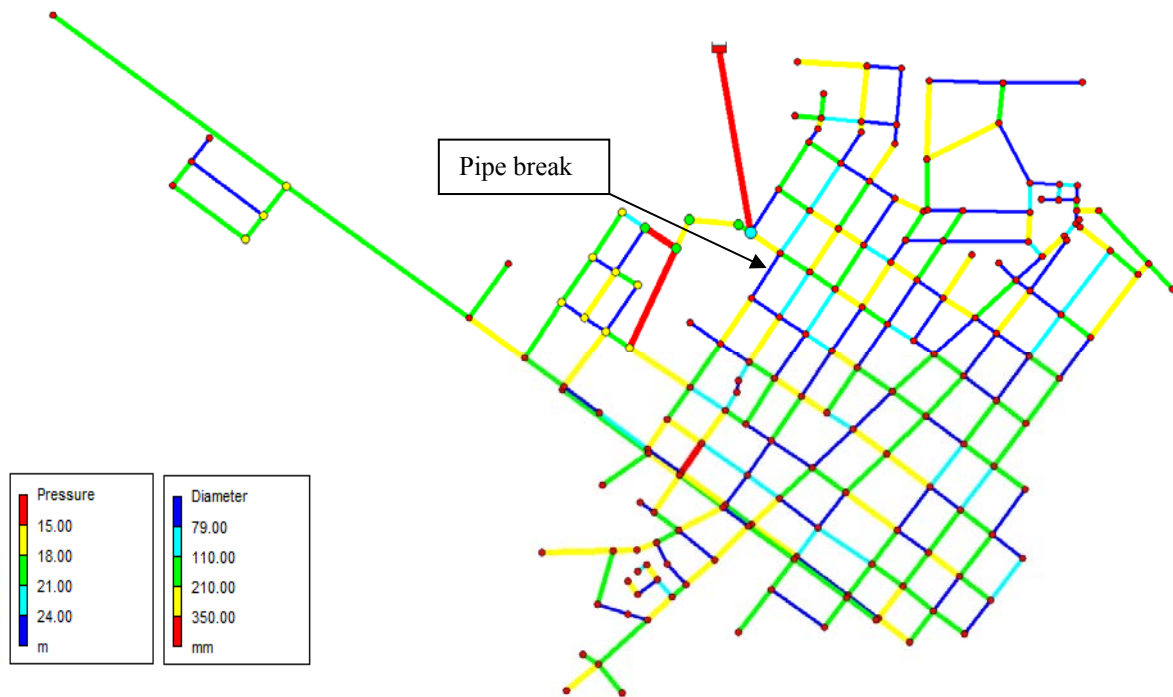

Figure 5 Dominant solution under same failure scenario (without rules)

As a termination condition, we ran the algorithm until 600 iterations were completed without improvement. An improvement is understood as any positive change in the approximated Pareto front obtained by the algorithm. It must be noted that even if the algorithm reaches its own termination condition, it could still be receiving requests from users or other swarms running in parallel; each swarm can, in addition, restart the search by itself when an update in its Pareto front is needed after the interaction with a user or another swarm.

4.2 The system and the solutions provided

This system is a real-world network (with fixed layout) (Fig. 4) fed by a reservoir, and made of 273 pipes and 183 consumption nodes with distinct consumptions amounting to 176 l/s. The design considers the various objectives above described; namely, minimizing the investment cost (the pipe diameters, in this case); minimizing the lack of pressure at demand nodes; and minimizing additional costs

caused by reliability or tolerance problems. We restrict ourselves here to presenting qualitative results that show the differences between designs – depending whether rules have been applied or not.

Fig. 4 corresponds to one of the dominant solutions that showed good tolerance to pipe break failures (the specific design is specified by the pipe colors). Under normal conditions this solution satisfies the minimum required pressure at every demand node (no dark red nodes appear).

However, in the failure scenario represented (marked as 'pipe break'), only three points (demand nodes marked in solid red (dark) at the bottom of the figure) had pressure values under the minimum.

Fig. 5 represents (consider just pipe colors) another dominant solution also able to satisfy the minimum pressure under normal conditions (no dark red nodes appear); but with a cost 16% lower than the solution in Fig. 4.

The major difference between both solutions reflects their tolerance to failure conditions. The network in Fig. 5 is under the same failure condition as the network in Fig. 4; but in the case of Fig. 5, the pressure problems can be found throughout the network (solid dark red points).

Additionally, the use of rules provides the main difference between both solutions related to the way the diameters were selected. In effect, the solution in Fig. 4 was obtained using the rule of decreasing diameters from upstream to downstream, and the result shows a 'smart' layout, one with a 'more logical' diameter distribution. However, this rule was not used to obtain the distribution of diameters in Fig. 5. This distribution makes no sense from an engineering practical perspective, since some pipes can be found with a diameter completely different to the diameter of the neighboring pipes (abrupt reductions or expansions in diameter for no logical reason). From the engineering point of view, the uniformity of diameters represents a clear advantage for the construction, operation, and maintenance of the network.
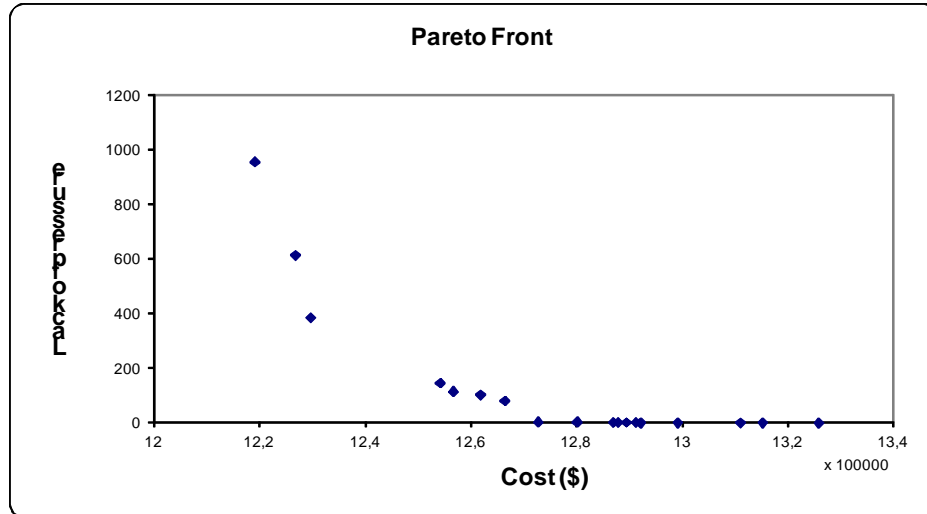
Figure 6 2D view of a Pareto front including cost and lack of pressure

Finally, in Fig. 6, a view of the (dynamic) Pareto front obtained at a certain stage of the evolution of the algorithm is shown. Dominant solutions are represented in a two dimensional format by selecting two desired objectives of the problem (lack of pressure and network cost in the case of Fig. 6). Note that only non-dominated solutions are

represented. At this stage, the number of agents involved was around 120. This representation facilitates an understanding of the overall group of dominant solutions obtained in order to decide the final design variant. In this specific case, three swarms following three different *singular points* were working. In addition to the *singular point* given by the minimum values of both objectives, another *singular point* was attracting solutions close to the low cost branch of this Pareto front, while a third *singular point* attracted higher-cost solutions with lower values for lack of pressure (horizontal right branch).

Plenty of rich 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. Also, the relationship between the initial investment cost and the minimum pressure in the network may help decide, among other factors, which pressure to use for the final solution. In this case (with 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 (impaired) if the budget is increased (shortened) 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

Evolutionary algorithms, in general, represent possible ways of analyzing multi-objective optimization problems. Each algorithm has advantages and disadvantages, and its performance may be better or worse depending on the characteristic of the problem to be solved. Agent Swarm Optimization (ASO), the computer-aided platform proposed in this paper, profits from the best of various algorithms when solving complex real-world WDS optimization problems.

Specifically, in the design of infrastructures such as water distribution systems, using the philosophy of ASO contributes several advantages. Firstly, the problem can be solved with a multi-objective approach. Secondly, various agents with different characteristics may be added, which includes the possibility of making various evolutionary algorithms work together and also the possibility of having rule-based agents participating in the search process. Last but not least, the human interaction with the algorithm offers a special platform for finding solutions as a team. Integrating the search capacity of algorithms and the ability of specialists to redirect the search towards specific interest points – based on their experience in solving problems – results in a powerful collaborative system for finding

solutions to engineering problems. Most artificial intelligent works try to substitute humans in some of their tasks; ASO is not aimed at substituting any human team but at being integrated with the team. Artificial agents can profit from the creativity and ideas of human experts to improve their own solutions; in turn, human experts can profit from the speed and search capabilities of artificial agents to explore broader solution spaces.

Future work should be aimed at introducing new agents with possibly more efficient rules of behavior during solution searches. Additional examples of study designs should also be considered that make use of various ways to approach the objective function. The way in which the objective functions are considered is in itself an evolutionary process that must not stop and must be adapted to the requirements of time (as new requirements appear) and place (according to possible local needs).

The study of the current conditions and the needs of WDS design must continue, and there must be a broad exchange with specialists to add improvements to the algorithm used, and the resulting software application. The implementation of the proposed algorithm must be updated using emerging technologies in parallel and distributed computing.

ACKNOWLEDGMENTS

REFERENCES

Adeli, H. % Kumar, S. (1995), Distributed Genetic Algorithms for Structural Optimization, *Journal of Aerospace Engineering*, **8**:3, 156-163.

Afshar, M. H., Akbari, M. & Mario, M. A. (2005), Simultaneous layout and size optimization of water distribution networks: Engineering approach, *Journal of Infrastructures Systems*, **11**(4), 221–230.

Amini, F., Khanmohamadi Hazaveh, N., & Abdolahi Rad, A. (2013), Wavelet PSO-based LQR algorithm for Optimal Structural Control using Active Tuned Mass Dampers, *Computer-Aided Civil and Infrastructure Engineering*, **28**:7, 542-557.

Arumugam, M. S. & Rao, M. V. C. (2008), 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, *Applied Soft Computing*, **8**(1), 324-336.

Badawy, R., Yassine, A., Heßler, A., Hirsch, B. & Albayrak, S. (2013), A Novel Multi-Agent System Utilizing Quantum-Inspired Evolution for Demand Side Management in the Future Smart Grid, *Integrated Computer-Aided Engineering*, **20**:2, 127-141.

Černý, V. (1985), Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, *Journal of Optimization Theory and Applications*, **45**, 41–51.

Coello Coello, C. A., Lamont, G. B. & Van Veldhuizen, D. A. (2007), *Evolutionary Algorithms for Solving Multi-Objective Problems*, Springer.

Dandy, G. C. & Engelhardt, M. O. (2006), Multi-Objective Trade-Offs between Cost and Reliability in the Replacement of Water Mains, *Journal of Water Resources Planning and Management-ASCE*, **132**(2), 79-88.

Deb, K. (2001), *Multi-objective optimization using evolutionary algorithms*, Wiley, New York.

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

Dorigo, M., Maniezzo, V. & Colorni, A. (1996), The ant system: optimization by a colony of cooperating ants, *IEEE Transactions on Systems, Man and Cybernetics—PartB*, **26**(1), 1–13.

Dridi, L., Parizeau, M., Mailhot, A. & Villeneuve, J. P. (2008), Using Evolutionary Optimisation Techniques for Scheduling Water Pipe Renewal Considering a Short Planning Horizon, *Computer-Aided Civil and Infrastructure Engineering*, **23**(8), 625-635.

Duan, Q., Gupta, V.K. & Sorooshian, S. (1993), A shuffled complex evolution approach for effective and efficient global optimization, *Journal of Optimization. Theory and Applications*, **76**, 501-521.

Duchesne, S., Beardsell, G., Villeneuve, J.P., Toumbou, B. & Bouchar, K. (2013), A Survival Analysis Model for Sewer Pipe Structural Deterioration, *Computer-Aided Civil and Infrastructure Engineering*, **28**:2, 146-160.

Dupont, G., Adam, S., Lecourtier, Y. & Grilheres, B. (2008), Multi-objective particle swarm optimization using enhanced dominance and guide selection, *International Journal of Computational Intelligence Research*, **4**(2), 145-158.

Fougères, A. J. & Ostrosi, E. (2013), Fuzzy agent-based approach for consensual design synthesis in product configuration, *Integrated Computer-Aided Engineering*, **20**:3, 259-274.

Fuggini, C., Chatzi, E., Zangani, D., & Messervey, T.B. (2013), Combining Genetic Algorithm with a Meso-scale Approach for System Identification of a Smart Polymeric Textile, *Computer-Aided Civil and Infrastructure Engineering*, **28**:3, 227-245.

Geem, Z. W., Kim, J. H. & Loganathan, G. V. 2001, A new heuristic optimization algorithm: Harmony search, *Simulation*, **76**(2), 60-68.

Giustolisi, O., Savic, D. & Kapelan, Z. (2008), Pressure-Driven Demand and Leakage Simulation for Water Distribution Networks, *Journal of Hydraulic Engineering-ASCE*, **134**(5), 626-635.

Goldberg, D. E. (1989), *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley, Reading, Ma.

Goulter, I. C. & Bouchart, F. (1990), Reliability-Constrained Pipe Network Model, *Journal of Hydraulic Engineering-ASCE*, **116**(2), 211-229.

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

Gupta, R. & Bhave, R. (1994), Reliability analysis of water distribution systems, *Journal of Environmental Engineering-ASCE*, **120**(2), 447-460.

Gutierrez-Garcia J. O. & Sim, K. M. (2012), Agent-based Cloud Workflow Execution, *Integrated Computer-Aided Engineering*, **19**:1, 39-56.

Hejazi, F., Toloue, I., Noorzaei, J., & Jaafar, M.S., (2013), Optimization of Earthquake Energy Dissipation System by Genetic Algorithm, *Computer-Aided Civil and Infrastructure Engineering*, **28**:10, 796–810.

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

Herrera, M., Izquierdo, J., Montalvo, I. & Pérez-García, R. (2011), Injecting diversity into particle swarm optimization. Application to water distribution system design, *Advances in Computer Science and Engineering*, **6**(2), 159-179.

Hsiao, F. Y., Wang, S. S., Wang, W. C., Wen, C. P. & Yu, W. D. (2012), Neuro-Fuzzy Cost Estimation Model Enhanced by Fast Messy Genetic Algorithms for Semiconductor Hookup Construction, *Computer-Aided Civil and Infrastructure Engineering*, **27**:10, 764-781.

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

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

Izquierdo, J., Montalvo, I., Pérez, R. & Tavera, M. (2008b), Optimization in Water Systems: a PSO approach, in *Proceedings of the 2008 SpringSim, BIS'08*, 239-246, Ottawa, Canada.

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

Izquierdo, J., Montalvo, I., Herrera & M., Pérez-García, R. (2012), A general purpose non-linear optimization framework based on Particle Swarm Optimization, in *Proceedings of the 12th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE2012*, La Manga, Spain.

Jafarkhani, R. & S.F. Masri, S.F. (2011), Finite Element Model Updating Using Evolutionary Strategy for Damage Detection, *Computer-Aided Civil and Infrastructure Engineering*, **26**(3), 207-224.

Janson, S., Merkle, D. & Middendorf, M. (2008), Molecular docking with multiobjective particle swarm optimization, *Applied Soft Computing*, **8**(1), 666–675.

Kalungi, P. & Tanyimboh, T. T. (2003), Redundancy model for water distribution systems, *Reliability Engineering & System Safety*, **82**(3), 275-286.

Keedwell, E. & Khu, S. (2006), Novel cellular automata approach to optimal water distribution network design, *Journal of Computing in Civil Engineering*, **20**(1), 49–56.

Kennedy, J. & Eberhart, R. C. (1995), Particle swarm optimization, in *Proceedings of the IEEE International Conference on Neural Networks*, Piscataway, NJ, 1942-1948.

Khomsi, D., Walters, G. A., Thorley, A. R. D. & Ouazar D. (1996), Reliability tester for water-distribution networks, *Journal of Computing in Civil Engineering-ASCE*, **10**(l), 10-9.

Kim, H. & Adeli, H. (2001), Discrete Cost Optimization of Composite Floors using a Floating Point Genetic Algorithm, *Engineering Optimization*, **33**:4, 485-501.

Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. (1983), Optimization by Simulated Annealing, *Science* **220**(4598), 671–680.

Kleiner, Y., Adams, B. J. & Rogers, J. S. (2001), Water distribution network renewal planning, *Journal of Computing in Civil Engineering*, **15**(1), 15-26.

Martínez-Rodríguez, J. B., Montalvo, I., Izquierdo J. & Pérez-García, R. (2011), Reliability and Tolerance Comparison in Water Supply Networks, *Water Resources Management*, **25**, 1437–1448.

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

Montalvo, I., Izquierdo, J., Pérez, R. & Iglesias, P. L. (2008), A diversity-enriched variant of discrete PSO applied to the design of Water Distribution Networks, *Engineering Optimization*, **40**(7), 655-668.

Montalvo, I., Izquierdo, J., Pérez R. & Herrera M. (2009), Improved performance of PSO with self-adaptive parameters for computing the optimal design of Water

Supply Systems, *Engineering Applications of Artificial Intelligence*, **23**(5), 727-735.

Montalvo, I., Izquierdo, J., Schwarze, S. & Pérez-García, R. (2010), Multi-objective particle swarm optimization applied to water distribution systems design: An approach with human interaction, *Mathematical and Computer Modelling*, **52**, 1219-1227.

Montalvo, I. (2011), *Diseño óptimo de sistemas de distribución de agua mediante Agent Swarm Optimization*, Ph.D. thesis, Universidad Politècnica de València, Valencia, Spain.

Moscato, P. (1989), *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms*. Caltech Concurrent Computation Program (report 826).

Nejat, A. & Damnjanovic, I. (2012), Agent-based Modeling of Behavioral Housing Recovery Following Disasters, *Computer-Aided Civil and Infrastructure Engineering*, **27**:10, 748-763.

Pareto, V. (1896), *Cours d'economie politique*, Université de Lausanne, Lausanne, Switzerland.

Park, H. & Leibman, J. (1993), Redundancy-constrained minimum-cost design of water distribution networks, *Journal of Water Resources Planning and Management-ASCE*, **119**(l), 83-98.

Paya, I., Yepes, V., González-Vidosa, F. & Hospitaler, A. (2008), Multiobjective Optimization of Concrete Frames by Simulated Annealing, *Computer-Aided Civil and Infrastructure Engineering*, **23**(8), 596-610.

Pinto-Praça, T., Vale, Z., Morais, H., Sousa, T.M. (2013), Strategic Bidding in Electricity Markets: an Agent-Based Simulator With Game Theory For Scenario Analysis, *Integrated Computer-Aided Engineering*, **20**:4, 335-346.

Plevris, V. & Papadrakakis, M. (2011), A Hybrid Particle Swarm – Gradient Algorithm for Global Structural Optimization, *Computer-Aided Civil and Infrastructure Engineering*, **26**(1), 48-68.

Putha, R., Quadrifoglio, L. & Zechman, E. (2012), Comparing Ant Colony Optimization and Genetic Algorithm Approaches for Solving Traffic Signal Coordination under Oversaturation Conditions, *Computer-Aided Civil and Infrastructure Engineering*, **27**(1), 14-28.

Raich, A.M. & Liszkai, T.R. (2012), Multi-Objective Optimization of Sensor and Excitation Layouts for Frequency Response Function-based Structural Damage Identification, *Computer-Aided Civil and Infrastructure Engineering*, **27**:2, 95-117.

Reyes-Sierra, M. & Coello Coello, C. A. (2006), Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art, *International Journal of Computational Intelligence Research*, **2**(3).

Rodrıguez-Seda, E. J., Stipanovic, D. M. & Sponga, M. W. (2012), Teleoperation of Multi-Agent Systems with Nonuniform Control Input Delays, *Integrated Computer-Aided Engineering*, **19**:2, 125-136.

Rossman, L. A. (2000), *EPANET 2 User's Manual*, Cincinati (IN), USA, Environmental Protection Agency.

Saldarriaga, J. G., Bernal A. & Ochoa, S. (2008), Optimized design of water distribution network enlargements using resilience and dissipated power concepts. in *Proceedings of the 10th Annual Water Distribution Systems Analysis Conference (WDSA 2008), ASCE*, Reston, Va., 298–312.

Sarma, K. & Adeli, H. (2000), Fuzzy Genetic Algorithm for Optimization of Steel Structures, *Journal of Structural Engineering*, ASCE, **126**:5, 596-604.

Savic, D. A. (2002), Single-objective vs. multiobjective optimisation for integrated decision support integrated assessment and decision support, in *Proceedings of the First Biennial Meeting of the International Environmental Modeling and Software Society*, Vol. 1, A. E. Rizzoli and A. J. Jakeman, eds., Univ. of Lugano, Lugano, Switzerland, 7–12.

Sgambi, L., Gkoumas, K. & Bontempi, F. (2012), Genetic Algorithms for the Dependability Assurance in the Design of a Long Span Suspension Bridge, *Computer-Aided Civil and Infrastructure Engineering*, **27**:9, 655-675.

Shafahi, Y. & Bagherian, M. (2013), A Customized Particle Swarm Method to Solve Highway Alignment Optimization Problem, *Computer-Aided Civil and Infrastructure Engineering*, **28**:1, 52-67.

Sycara, K. P. (1998), Multiagent systems, *American Association for Artificial Intelligence – AI Magazine*, **19**(2), 79-92.

Tanyimboh, T. T., Tabesh, M. & Burrows, R. (2001), Appraisal of source head methods for calculating reliability of water distribution networks, *Journal of Water Resources Planning and Management-ASCE*, **127**(4), 206-213.

Tao, H., Zain, J. M., Ahmed, M. M., Abdalla, A. N. & Jing, W. (2012), A Wavelet-Based Particle Swarm Optimization Algorithm for Digital Image Watermarking, *Integrated Computer-Aided Engineering*, **19**:1, 81-91.

Todini, E. (2000), Looped water distribution networks design using a resilience index based heuristic approach, *Urban Water*, **2**, 115–122.

Vamvakeridou-Lyroudia, L. S., Walters, G. A. & Savic, D. A. (2005), Fuzzy Multiobjective Optimization of Water Distribution Networks, *Journal of Water Resources Planning and Management-ASCE*, **131**(6), 467-476.

Vitins, B. J. & Axhausen, K. W. (2009), Optimization of Large Transport Networks Using the Ant Colony Heuristic, *Computer-Aided Civil and Infrastructure Engineering*, **24**(1), 1-14.

Vrugt, J. A., Gupta, H. V., Bastidas, L. A., Bouten, W., & Sorooshian, S. (2003), Effective and efficient algorithm for multiobjective optimization of hydrologic models, *Water Resources Research*, **39**(8), 1214.

Vrugt, J. A., Nualla´in, B. O´ , Robinson, B. A., Bouten, W., Dekker, S. C., & Sloot, P. M. A., (2006), Application of parallel computing to stochastic parameter estimation in environmental models, *Computational Geosciences*, **32**(8), 1139–1155.

Vrugt, J. A. & Robinson, B. A., (2007), Improved evolutionary search from genetically adaptive multi-search method, *P. Natl. Acad. Sci USA*, **104**(3), 708-711.

Walski, T. M. (Ed.) (2003), *Advanced water distribution modeling and management*, Haestad Press, Waterbury, Conn., USA.

Wooldridge, M. (2002), *An Introduction to Multiagent Systems*, John Wiley & Sons.

Wierzbicki, A. (1998), *Reference Point Methods in Vector Optimization and Decision Support. IR-98-017*, 43 pp, International Institute for Applied Systems Analysis, Laxenburg, Austria.

Wu, Z. Y., Wang, R. H., Walski, T. M., Yang, S. Y., Bowdler, D. & Baggett, C. C. (2006), Eficient pressure dependent demand model for large water distribution system analysis, in *Proceedings of the 8th Annual Water Distribution System Analysis Symposium*, Cincinnati, Ohio, USA.

Xie, C. & Waller, S.T. (2012), Optimal Routing with Multiple Objectives: Efficient Algorithm and Application to the Hazardous Materials Transportation Problem, *Computer-Aided Civil and Infrastructure Engineering*, **27**:2, 77-94.

Xu, C. & Goulter, I. C. (1997), Simulation-based optimal design of reliable water distribution networks, in *Proceedings of the 3rd International Conference on Modeling and Simulation*, Victoria University of Technology, Melbourne.

Xu, C. C & Goulter, I. C. (1999), Reliability based optimal design of water distribution networks, *Journal of Water Resources Planning and Management-ASCE*, **125**(6), 352-362.

Zeferino, J. A., Antunes, A. P. & Cunha, M. C. (2009), An Efficient Simulated Annealing Algorithm for Regional Wastewater Systems Planning, *Computer-Aided Civil and Infrastructure Engineering*, **24**(5), 359-370.