

# Agent swarm optimisation, a novel approach in swarm intelligence

I. Montalvo<sup>1</sup>, J. Izquierdo<sup>2</sup>, S. Schwarze<sup>3</sup> & R. Pérez-García<sup>2</sup>

<sup>1</sup>*3SConsult, Germany*

<sup>2</sup>*FluIng-IMM, Universitat Politècnica de València, Spain*

<sup>3</sup>*Institute of Information Systems, University of Hamburg, Germany*

## Abstract

Agent swarm optimisation (ASO) is a new paradigm based on particle swarm optimisation that exploits distributed or swarm intelligence and borrows some ideas from multi-agent based systems. It is aimed at supporting decision-making processes by solving either single or multi-objective optimisation problems.

Classical methods of optimisation have been shown to be poorly suited for many real-world problems since they are unable to deal with highly-dimensional, multimodal, non-linear problems; and process inaccurate, noisy, discrete and complex data. Robust methods of optimisation are often required to generate suitable results. ASO offers robustness through a common framework where a plurality of population-based algorithms co-exist, thereby offering superior performance by dynamically combining the strengths of multiple metaheuristics.

In this work the ASO framework is used to solve a complex problem in water management, namely the optimal design of water supply systems (including sizing of components, reliability, renewal, and rehabilitation strategies) using a multi-objective approach. Conditions for the correct development of the Pareto front are described. In addition, during the solution process, the users, working in parallel with computational algorithms, can force the recruitment of new agents/swarms to the environment and even contribute to the solution process with expert-based personal proposals that are later 'learned' by the algorithms.

*Keywords: water supply systems, engineering design optimisation, computer-human interaction, behavioural rules, swarm intelligence, multi-objective optimisation, intelligent agents.*



# 1 Introduction

In many fields of science and engineering, the optimisation techniques employed have conditioned the way in which optimisation problems (OP) have been approached. Consciously or unconsciously, problems have been adapted to the optimisation techniques in use. However, these techniques have been shown to be poorly suited to many real-world problems since they are unable to deal with highly-dimensional, multimodal, and strongly non-linear problems; while, at the same time, are bound to process inaccurate, noisy, discrete and complex data. Robust methods of optimisation are often required to generate suitable results.

ASO, which stands for agent swarm optimisation, is a newly introduced [1] abstract term that refers to a class of extensible, flexible, and robust algorithms for solving complex optimisation problems where classical techniques and recent evolutionary algorithms become stranded. ASO takes ideas from a multi-agent philosophy that crystallises on a platform where different kinds of agents, including humans, may interact and co-operate in the solution of a given OP.

In evolutionary algorithms (EA), one agent, due to its individual behaviour, is able to find a potential solution for the problem in hand by itself. Nevertheless, the knowledge of the solution space that individual agents have is very limited; effective search of optimal solutions is only possible as a result of interaction among several agents. Associations of interacting agents result in collective structures, called swarms that represent the emergent behaviour of groups of agents. These structures are also considered agents at a higher abstraction level. In turn, swarms are able to interact with other existing swarms.

Nevertheless, the conceptual framework of ASO is not a fixed meta-heuristic but a dynamic environment combining the strengths of multiple meta-heuristics where new algorithms (agents or swarms) can be added in real time to contribute to the solution of the given problem. A precisely defined hierarchical structure is needed to coordinate the efforts in an efficient way.

In addition, agents are provided with sets of problem-dependent rules with two clear objectives, namely tuning agent behaviour to specialise for the OP in hand, while, at the same time, reducing the search space, thus enabling designs to be made with increased reliability and within a reasonable time frame.

This paper considers how these two aspects, namely, the hierarchy of swarms and the addition of problem-dependent rules, can greatly improve the performance of ASO. This is shown in connection with a real-world application, namely the design of a water supply system (WSS). The design optimisation of a large WSS from a multi-objective standpoint within a reasonable time frame remains a challenging and burdensome problem, mainly due to the very high computational complexity involved, since numerous, expensive hydraulic simulations are needed.

The rest of the paper is organised as follows. Firstly, the main features of the ASO multi-objective framework, emphasising the hierarchical structure of the swarms and the agent rules, are presented. Then, the problem of the design of a WSS is concisely presented. A case-study is then fully developed. The conclusions section closes the paper.



## 2 Multi-objective optimisation using ASO

Economic motivations usually condition the difference between a solution and a better solution. Nevertheless, decision-making also needs to fulfil many other technical and non-technical targets and constraints involved. A typical scenario includes the consideration of multiple conflicting objectives and involves finding an acceptable trade-off between them. In this regard, ASO fully incorporates the concept of dominance to compare solutions.

### 2.1 ASO in multi-objective optimisation

The idea behind ASO is a particle swarm optimisation-based environment (developed by the authors) that mimics the judgment of an engineer [2]. It was built by using various prior features and improvements regarding swarm intelligence. In ASO, multi-agent systems, and the necessary adaptation to multi-objective performance, including human interaction, are also integrated.

The first feature derives from the philosophy behind PSO (particle swarm optimisation) [3]. The feature consists of a variant of the standard PSO that can deal with various types of variables [4], including a mechanism for increased diversity [5], and enables the self-management of the parameters involved so that engineers are spared the task of parameter selection and fine-tuning [6]. Although the authors have applied this algorithm mainly to WSS design, it has proven very efficient in solving optimisation problems in other fields [7, 8].

The emergent behaviour of a PSO swarm is strongly reminiscent of the philosophy behind the multi-agent (MA) paradigm [9]. 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 decentralised and computation is asynchronous. Each agent acting alone cannot solve the problem in its entirety, but a group of agents, with the coexistence of different views, is better able to find a solution by interacting together. This idea can be clearly extrapolated to the case of multi-objective optimisation, since the result of the many interactions occurring within an MA system is improved performance.

Taking into account the desirability of solving real-world multi-objective OPs and the benefits offered by MA systems, a departure from the standard behaviour of particles in PSO was performed [1]. 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 normalisation procedure, propose two mechanisms to enrich the Pareto front, incorporate human interaction within the framework, and propose endowing the agents with specific, problem-dependent behavioural rules.

#### 2.1.1 Leadership

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 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 the *singular point*, which is updated with the best values found so far during the evolution of the algorithm [2].

### 2.1.2 Normalisation

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

### 2.1.3 Pareto front enrichment

Arguably, the most interesting solutions are located near the singular point and not too far from the peripheral areas of the Pareto front. Therefore, instead of seeking a complete and detailed Pareto front, we may be more interested in precise details around the singular point. Nevertheless, situations can occur 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 optimisation problems.

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 solving the problem and on the problem itself.

In fact, additional information is always needed for making final decisions in the real-world [10]. This information can be established a priori, for example when objectives are represented in only one weighted expression. 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, additional information can be used during the search process; 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. We describe here one approach based on dynamic population increases to raise the Pareto front density, and another approach based on human computer interaction, that adds swarms in a hierarchical structure to complete poorly represented areas of the front. This is achieved in real time 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 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 agent is placed where the new solution is found, thus increasing the density of

solutions on the Pareto front. Greater densities on the Pareto front must be restricted to the case where the new clone has at least one of its neighbours located further away than some minimal permissible distance in the objective space. It has to be noted that two objective vectors are considered to be neighbours when no other objective vector is located between them in at least one of the objectives considered in the problem.

The incorporation of new agents in real time makes it possible for the algorithm to evolve both in its structure and in its capacity to find good solutions. Nevertheless, the increase in the population makes the determination of the Pareto front harder as more potential solutions are involved. The approach we describe below helps in this issue.

### **2.1.5 Human computer interaction and the hierarchy of swarms**

In the second approach, the concept of a singular point is extended to any desired point in the objective space for particles to search around. This is performed by users, which are allowed to add new swarms for searching in desired regions of the objective space. This is achieved in real-time during the execution of the algorithm. This human interaction with the algorithm in real time also enables the incorporation of human behaviour, so the human turns out to be other members of the swarm 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, the behaviour of other agents may change; it happens, for example, when some agents decide to 'follow' the new solution introduced by the user. Proposed solutions can, then, even become leaders of the swarm.

The combination of various swarms within the same algorithm is efficient because it conducts a neighbourhood search in which each of the swarms specialises, 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. Nevertheless, some practical aspects must still be considered.

A time consuming task for any population-based algorithm used to solve a multi-objective optimisation problem is determining which solutions belong to the Pareto front when a large number of solutions already belong to it. In ASO, a hierarchy of swarms is used for fulfilling this task, which profits from parallel and distributed computing. Different swarms can find 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 of the new non-dominated solution. While swarms are waiting for asynchronic responses from their superiors, solutions are assumed to belong to their Pareto subsets. Swarms are organised hierarchically at different levels. When new swarms are added they have to be

placed in the hierarchy of all swarms already solving the problem. Each swarm has a maximum number of connected swarms at the lower level. New swarms will be connected to any of those existent swarms that still have connection capacity.

At this point, human behaviour begins to have a proactive role during the evolution of the algorithm. This represents the main difference with 'classical' multi-agent systems, where agents are normally considered as part of a software code; here, human beings 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 engineering decision making, where politicians, economists, engineers, and environmental specialists 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 Rule-driven agents**

Within the ASO framework, semantic entities from different heuristics are genuine agents in a multi-agent system. In particular, they can be endowed with specific, problem-dependent behavioural rules purposely designed to heuristically approach the solution process. These rules try to mimic the judgment of a human engineer when approaching the solution of the problem in hand.

Evolutionary algorithms generally have not previously taken advantage of this feature and, as a result, have been bound to analyse a larger solution space than necessary. Thus, including these 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 checking a number of usually simple rules avoids many expensive calculations or simulations (hydraulic simulations in the problem we present in this paper). Finally, the fact that the rules have strong problem-dependent meaning definitely brings the solution nearer to reality. For example, in the problem dealt with in this paper one rule just states that downstream pipe diameters must be smaller than upstream pipe diameters. In the section devoted to the application of ASO we present other examples of problem-specific rules.

## **3 The WSS optimal design problem**

Various objectives may be considered in the WSS optimal design problem. In this section, we briefly describe some of these objectives, namely, cost of components, adherence to hydraulic constraints, satisfaction of minimal pressure in the network, and resilience of the system during stressed conditions.

### **3.1 Cost of components**

Apart from the basic variables of the problem, which are the diameters of the new pipes, additional variables that depend on the design characteristics of the

system may be required: storage volume, pump head, type of rehabilitation to be carried out for various parts of the network, etc. The correct approach to assess the costs for each element 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. For the already existing pipes, the objective may consist in one of various actions: rehabilitation (with several available alternatives with associated costs), replacement, or simply duplication. In addition, it is important that the objective function reflects with utmost reliability the total cost of the system during its lifetime.

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

$$C_{WSS} = \sum_k \left[ P_{wc}^k (a_p \cdot C_p + a_{ps} \cdot C_{ps} + a_v \cdot C_v + a_{tk} \cdot C_{tk} + C_{Op}) \right]$$

includes various individual working conditions that depend on the values adopted by two types of variables – namely, demand models and roughness coefficient values:  $P_{wc}^k$  represents the probability for the  $k$ -th working condition. The function also considers the operational costs of the network,  $C_{Op}$ , along a certain temporal horizon, and this forces the use of the necessary amortization rates,  $a_{xx}$ , to multiply any of the investment costs, namely,  $C_p$ ,  $C_{ps}$ ,  $C_v$ , and  $C_{tk}$ , representing costs for pipes, pumping systems, valves and storage tanks, respectively.

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

### 3.2 Hydraulic constraints

Several formulations are available (see, for example, [11]) that are usually solved using some gradient-like technique. Various tools to analyse water networks using gradient-like techniques have been developed in the past. Among them, EPANET2 [12], is used in a generalised way. We use a modification of the EPANET2 toolkit to support pressure-driven demands [1].

### 3.3 Minimum pressure satisfaction

There are various ways of expressing lack of compliance with pressure, velocity, disinfectant, etc. Expressions for these constraints are straightforward using, for example, the Heaviside function, and suitable coefficients for a meaningful aggregation of different types of violations. These expressions depend on pipe diameters and other variables of the hydraulic model.

### 3.4 WSS resilience

Resilience of a WSS, which may interpreted in terms of the ability of the system to satisfactorily work not only in normal but also in abnormal or stressed conditions, is assessed in terms of various types of properties, such as redundancy, reliability, and tolerance.



Redundancy is closely related to reliability. The concept of reliability was introduced to quantitatively measure the possibility of maintaining an adequate service for a given period. Numerous WSS reliability quantification schemes exist [13]. For example, reliability may represent the expected fraction of the total required demand that can be maintained for a certain time horizon, provided that the network properties used for this reliability calculation are maintained.

As WSSs should behave satisfactorily under normal conditions when there are no failures, it is worthwhile making a separate and specific analysis of their behaviour under only failure states. Accordingly, the concept of *tolerance to failure*,  $T$ , has been introduced to represent the expected total required demand fraction that the network supplies as an average when it is in a state of failure [13]. In other words, this index answers the question of how well the network behaves, on average, when a pipe is removed from service.

### 3.5 Specific rules

A number of rules may be specified to approach the solution process for the problem. For the sizing of pipes in a WSS, 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 increased engineering sense, 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.

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. For example, adjusting diameters if one or more consumption nodes experiment too little or too much pressure as in [14]; increase the diameter of pipes joining at a certain node with too much pressure [15]; increase or reduce diameters depending on the energy dissipation in certain pipes [16]; increase the diameters of pipes experiencing higher unitary losses or reducing the diameters of pipes with lower losses [17], etc. Different new rules may be devised, such as reducing the diameter of a pipe if it undergoes a large reduction of chlorine concentration, among others. These are examples of rules that, obviously, are closely connected to the nature of the problem.

## 4 Case study

This case-study has been completely developed with WaterIng ([www.ingeniousware.net](http://www.ingeniousware.net)). 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 well-known evolutionary algorithms, namely, GA, ACO, and PSO. Some parameters of the algorithms were established a priori for running other case studies: the initial population size was set equal to 20 for the three evolutionary algorithms. Fine-tuning of the



other parameters in PSO is performed by using both adaptive and the self-adaptive techniques, as described in [6].

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.

This system is a real-world network (Figure 1) fed by a reservoir, and made of 273 pipes and 183 consumption nodes that consume 176 l/s. The design considers the various objectives above described; namely, minimising the investment cost; minimising the lack of pressure at demand nodes; and maximising the tolerance to pipe break failures.

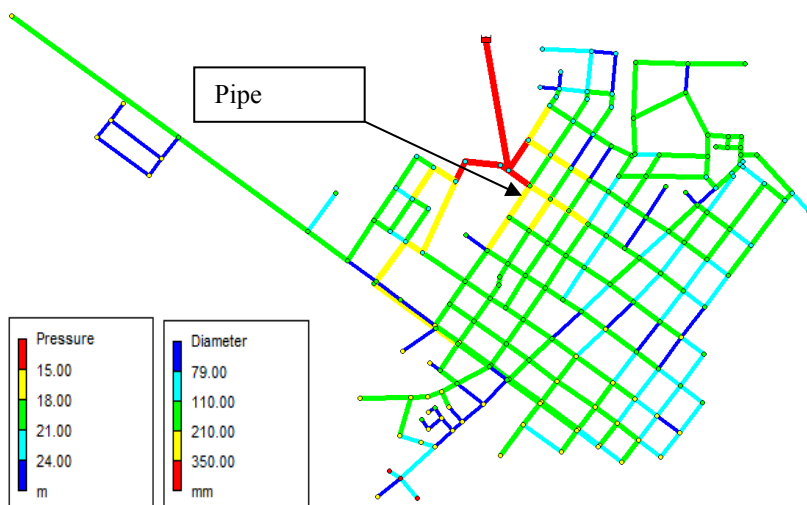


Figure 1: Layout (and a design) of the network.

Figure 1 corresponds to one of the dominant solutions that showed good tolerance to pipe break failures. Under normal conditions this solution satisfies the minimum required pressure at every demand node. In the failure scenario represented (marked as 'pipe break'), only three points (demand nodes marked in red at the bottom of the figure) had pressure values under the minimum.

Figure 2 represents another dominant solution also able to satisfy the minimum pressure under normal conditions; but with a cost 16% lower than the solution in Figure 1. The major difference between both solutions reflects their tolerance to failure conditions. The network in Figure 2 is under the same failure condition as the network in Figure 1; but in the case of Figure 2, the pressure problems can be found throughout the network. Additionally, there is a significant difference between both solutions related to the way the diameters

were selected. The solution in Figure 1 uses the rule of decreasing diameters from upstream to downstream, and the result shows a ‘smart’ layout. Moreover, the distribution of diameters in Figure 2 makes no sense from an engineering perspective; some pipes can be found with a diameter completely different to the diameter of the neighbouring pipes (abrupt reductions or expansions in diameter for no logical reason). The network in Figure 1 was designed by agents holding rules aimed to provide a ‘more logical’ diameter distribution. From the engineering point of view, the uniformity of diameters represents a clear advantage for the construction, operation, and maintenance of the network.

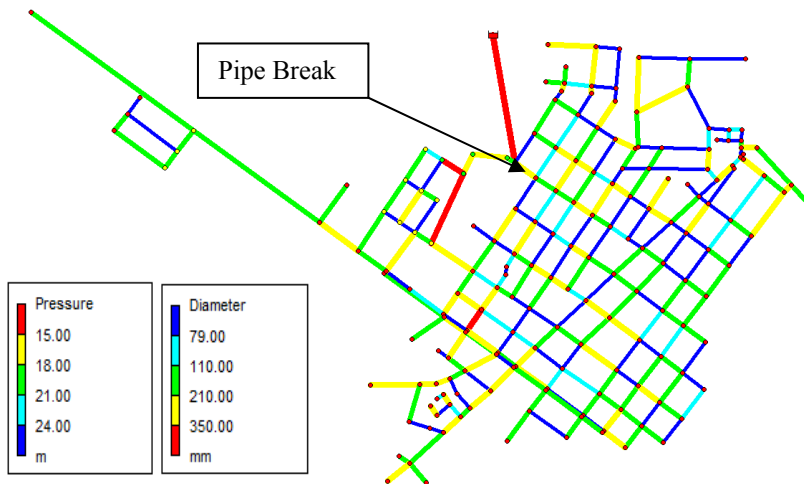


Figure 2: Dominant solution 2 under same failure scenario.

Finally, in Figure 3 a view of the Pareto front obtained during 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 Figure 3). This representation facilitates an understanding of the overall group of dominant solutions obtained in order to decide the final design variant. 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.

It is worth mentioning how the lack of pressure is represented in Figure 3: firstly, the difference between the existent pressure and the minimum required pressure is computed for those nodes where the existent pressure is lower than required. This difference is called lack of pressure at a node. The overall lack of pressure is determined as the sum of the lack of pressure at each node and multiplied by a penalty factor. In the case of the network cost, the estimation is performed as a function of the selected pipe diameters.

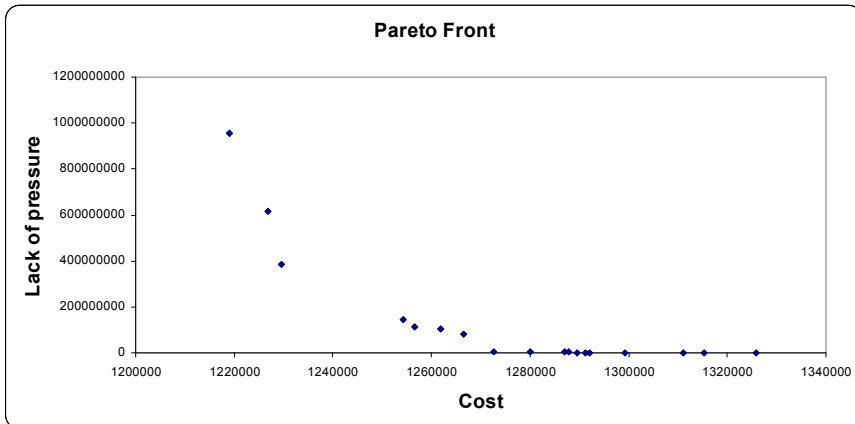


Figure 3: 2D view of a Pareto front including lack of pressure and cost.

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 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 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 WSS.

## 5 Conclusions

In the solution of some complex multi-objective optimisation problems, using the ASO philosophy contributes several advantages. Firstly, the problem can be solved with a multi-objective approach. Secondly, various agents with different characteristics may be added, which include the possibilities of making different 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 – result in a powerful collaborative system for finding solutions to engineering problems. Most artificial intelligence systems try to substitute humans in some of their tasks; however, ASO is aimed at integrating rather than substituting human teams. Artificial agents can profit from the creativity and ideas of human experts for improving solutions; in turn, human experts can profit from the speed and search capabilities of artificial agents when exploring broader solution spaces.

Future work should be aimed at the implementation of the proposed algorithm using emerging technologies in parallel and distributed computing.

## Acknowledgements

This work has been developed with the support of the project IDAWAS, DPI2009-11591, of the Spanish Ministry of Education and Science, and ACOMP/2010/146 of the Education Department of the Generalitat Valenciana. The use of English was revised by John Rawlins.

## References

- [1] Montalvo I. Optimal Design of Water Distribution Systems Using Agent Swarm Optimization. Doctoral dissertation. Universidad Polit cnica de Valencia, Valencia, Spain, 2011.
- [2] I. Montalvo, J. Izquierdo, S. Schwarze, and R. P rez-Garc a, Multi-objective particle swarm optimization applied to water distribution systems design: An approach with human interaction, *Mathematical and Computer Modelling*, 52, 1219-1227, 2010.
- [3] J. Kennedy and R.C. Eberhart, Particle swarm optimization, in *Proceedings of the IEEE International Conf. on Neural Networks*, Piscataway, NJ, 1942-1948, 1995.
- [4] J. Izquierdo, I. Montalvo, R. P rez, and V.S. Fuertes, Design optimization of wastewater collection networks by PSO, *Computer & Mathematics with Applications*, 56(3), 777-784, 2007.
- [5] I. Montalvo, J. Izquierdo, R. P rez, and P.L. Iglesias, A diversity-enriched variant of discrete PSO applied to the design of Water Distribution Networks, *Engineering Optimization*, 40(7), 655-668, 2008.
- [6] I. Montalvo, J. Izquierdo, R. P rez, R., and M. Herrera, 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, 2010.
- [7] J. Izquierdo, R. Minciardi, I. Montalvo, M. Robba, and M. Tavera, Particle Swarm Optimization for the biomass supply chain strategic planning, in *Proceedings of 4th Biennial Meeting, iEMSs 2008: International Congress on Environmental Modelling and Software*, pp. 1272-1280, Barcelona, Spain, 2008.
- [8] M. Herrera, J. Izquierdo, I. Montalvo, J. Garc a-Armengol, and J.V. Roig. Identification of surgical practice patterns using evolutionary cluster analysis, *Mathematical and Computer Modelling*, 50, 705-712, 2009.
- [9] M. Wooldridge, *An Introduction to Multiagent Systems*, John Wiley & Sons, 2002.
- [10] C.A. Coello Coello, G.B. Lamont, and D.A. van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Springer, 2007.
- [11] J. Izquierdo, R. P rez, and P.L. Iglesias, Mathematical models and methods in the water industry, *Mathematical and Computer Modelling*, 39, 1353-1374, 2004.
- [12] Rossman, L.A. *EPANET 2 User's Manual*, Cincinnati (IN), USA, Environmental Protection Agency, 2000.



- [13] J.B. Martínez, Cost and reliability comparison between branched and looped water supply networks, *Journal of Hydroinformatics*, IWA, 12(2), 150-160, 2010.
- [14] Keedwell, E. and Khu, S. Novel cellular automata approach to optimal water distribution network design. *J. Comput. Civ. Eng.*, 20(1), 49–56, 2006.
- [15] Afshar, M.H., Akbari, M. and Mario, M.A. Simultaneous layout and size optimization of water distribution networks: Engineering approach. *J. Infrastruct. Syst.*, 11(4), 221–230, 2005.
- [16] Todini, E. Looped water distribution networks design using a resilience index based heuristic approach. *Urban Water*, 2, 115–122, 2000.
- [17] Saldarriaga, J. G., Bernal, A., and Ochoa, S. Optimized design of water distribution network enlargements using resilience and dissipated power concepts. *Proc., 10th Annual Water Distribution Systems Analysis Conf. (WDSA 2008)*, ASCE, Reston, Va., 298–312, 2008.

