Jul 1st, 12:00 AM

Agent Swarm Optimization: a paradigm to tackle complex problems. Application to Water Distribution System Design

Idel Montalvo

Joaquín Izquierdo

Silvia Schwarze

Rafael Pérez-García

Follow this and additional works at: https://scholarsarchive.byu.edu/iemssconference

Agent Swarm Optimization: a paradigm to tackle complex problems. Application to Water Distribution System Design

Idel Montalvo[a], **Joaquín Izquierdo**[a], **Silvia Schwarze**[b], **Rafael Pérez-García**[a]

[a] *imontalvo@gmmf.upv.es, jizquier@gmmf.upv.es, rperez@gmmf.upv.es, Instituto de Matemática Multidisciplinar, Universidad Politécnica de Valencia Camino de Vera s/n, 46022, Valencia, Spain*

[b] *schwarze@econ.uni-hamburg.de, Institute of Information Systems, University of Hamburg, Von-Melle-Park 5, 20146, Hamburg, Germany*

Abstract: Agent Swarm Optimization (ASO) is a generalization of Particle Swarm Optimization (PSO) orientated towards distributed artificial intelligence, taking as a base the concept of multi-agent systems. It is aimed at supporting decision-making processes by solving either single or multi-objective optimization problems. ASO offers a common framework for the plurality of co-existent population-based algorithms and other heuristics. A particle from a PSO swarm, an ant from an ACO (Ant Colony Optimization) system, and a chromosome from a GA (Genetic Algorithm) structure do exhibit different behaviour. Yet, they all share a common feature: each represents a potential solution for the problem to be solved. In a combined environment, a PSO particle could help reinforce pheromone on the ants' paths; an ant could be reproduced with a chromosome; a chromosome could be the leader of a particle swarm, and so on. This framework is a dynamic environment where new agents/swarms can be added in real time to contribute to the solution of the problem. During the solution process, the own user can add new agents/swarms to the environment and even contribute to the solution process with problem-based personal proposals. In this work the ASO framework is described, and used to solve a complex problem in water management, namely the optimal design of water distribution systems (including, sizing of components, reliability, renewal and rehabilitation strategies, etc.) using a multi-objective approach.

Keywords: Agent Swarm Optimization, multi-objective optimization, distributed artificial intelligence.

1. INTRODUCTION

Agent Swarm Optimization (ASO) is a generalization of Particle Swarm Optimization, Kennedy and Eberhart [1995], orientated towards distributed artificial intelligence and considering ideas from multi-agent systems. It is aimed at supporting engineering decision-making by solving either single or multi-objective optimization problems. In ASO, one agent is able to find by itself a potential solution for the problem. Nevertheless, the knowledge related to the solution space that agents have is very limited; effective search of optimal solutions is only possible as a result of the interaction among several agents.

Every agent has its own individual behaviour. Associations of agents interacting among them result in a collective structure, called swarm, which represents the collective behaviour of a group of agents. This structure can also be considered as an agent in a higher abstraction level. In its turn, each swarm has its own behaviour and is able to interact with other existent swarms. One main difference with multi-agent systems is that

normally agents are considered as part of a software code, while here, human beings are also considered as agents actively involved in the solution search process. Particularly, in the case of multi-objective optimization problems, when there are conflicting objectives, Coello et al. [2007], additional information from humans to make the final decision is always needed. 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. But also, additional information can 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 for deciding which regions of the Pareto front are more interesting and for proposing solutions that can lead or enrich the way other agents behave.

The participation of several human agents with different perspectives of a problem is very close to what happens in practice in engineering decision-making, where politicians, economists, engineers, environment specialists are all involved in making the final decision. Most of the artificial intelligent works try to substitute humans in some of their tasks; ASO is not aimed at substituting a work team but at being integrated with it.

ASO, by definition, has objectives to meet and expressions to evaluate to what extent these objectives are fulfilled by the solutions already found. The algorithm evolves both in its structure and in the quality of the solutions. It has been conceived to distribute different swarms in more than one computer and to bring the possibility to agents for executing actions in parallel. Agents are autonomous entities and the interaction among them happens asynchronously. These characteristics make it possible the incorporation, in real time, of new agents that are able to reinforce or enrich the search process.

Another important capability of ASO is to offer a common framework for the plurality of existent population-based algorithms and others heuristics. A particle from a PSO Swarm, an ant from an Ant Colony System, Dorigo et al. [1996], or a chromosome from a Genetic Algorithm, Montesinos et al. [1999], have different behaviours, but all of them share a common point: they provide a potential solution for the problem to be solved. In a combined environment a particle could help to reinforce pheromone over the paths of ants, an ant could be reproduced with a chromosome; and a chromosome could be the leader of a swarm. This framework is not a fix meta-heuristic but a dynamic environment where a new algorithm (agent) can be added in real time to help solve a problem.

The possibility of assembling general search algorithms with heuristic rules extracted from the problems is one of the major advantages of ASO, when searching in considerable large solutions spaces. In Water Distribution System Design for example, various instances of evolutionary algorithms for searching optimal solutions have been described. Nevertheless, engineer's rules for the problem are not commonly applied. As an example, evolutionary algorithms themselves do not consider the fact that pipes should not increase in size following the direction of flow; nevertheless it is a simple engineer rule that can be applied in water distribution system design.

In this work, the ASO philosophy is described and some results are shown regarding two problems related to Water Distribution System Design. Only one breed of agent is used in this work. Nevertheless, the inclusion of other kinds of agents is straightforward.

2. AGENT SWARM OPTIMIZATION

The kind of artificial agent used in this work is based on the behaviour of particles in Particle Swarm Optimization (PSO). But, in addition, the agents in this work, when moving from one position to another in the solution space, consider the application of the specific rule that pipes should not increase in size in the direction of the flow. The rest of the movement process is similar to the movement of particles in PSO for solving problems of Water Distribution System Design, Montalvo et al. [2008].

Different from the behaviour of particles in single objective problems, when deciding what a better position is in multi-objective problems, agents will use the dominant solution concept. A solution *A* is said to dominate another solution *B* when *A* is better than *B* in at least one objective, and not worse in the others. Two solutions are called indifferent or incomparable if neither dominates the other. The aim of ASO when solving multi-objective optimization problems is to find the Pareto-optimal set or front (or an approximation sample), Montalvo et al. [2010a], defined as the set of all non-dominated solutions.

Also, the leadership in a swarm will be determined in a different way compared to single-objective PSO. The most natural option is to select the leader as the closest particle to the so-called utopia point in the search space. This utopia point is defined as the point in the search space whose components give the best values found for every single objective. In this problem, the utopia point is an unknown dynamical point since the best value for every objective is something unknown at the very beginning (and even during the whole process). Accordingly, we will use an approximation of this utopia point, which we call *singular point*, which is updated during the evolution of the algorithm, Montalvo et al. [2010a]. Additional to the concept of singular point, swarms could have different views for selecting their leaders. One possibility that has also been used by these authors establishes a weighted sum of objectives first and then the leader is selected as the particle whose weighted sum is maximal (or minimal depending on nature of the optimization problem). When using several swarms working in parallel, each may have a different leader selection concept.

Since each objective may be expressed in different units, it was decided to enforce some regularization for evaluating distances in the objective space. Coordinates were regularized in terms of percentage, considering that at every component, the worst and best value of the corresponding objective are 0% and 100% respectively; the percentage corresponding to any other value is calculated using linear interpolation. To know the distance between any two objective vectors, their components are first regularized in terms of percentage and then the Euclidean distance between them is calculated. Both worst and best objectives values are not usually known a priori; they are updated while the solution space is explored.

The movement towards the leader reinforces the search of the Pareto front in a zone near the singular point. Despite that zone may be quite interesting for decision-making, it may also be interesting to know what is happening on other zones of the Pareto front. That is why new swarms may be added to follow different singular points created with some desired values corresponding to the zone where the search should be reinforced. In this sense, the concept of singular point could be extended to any desired point in the objective space where the search of agents is devised as interesting.
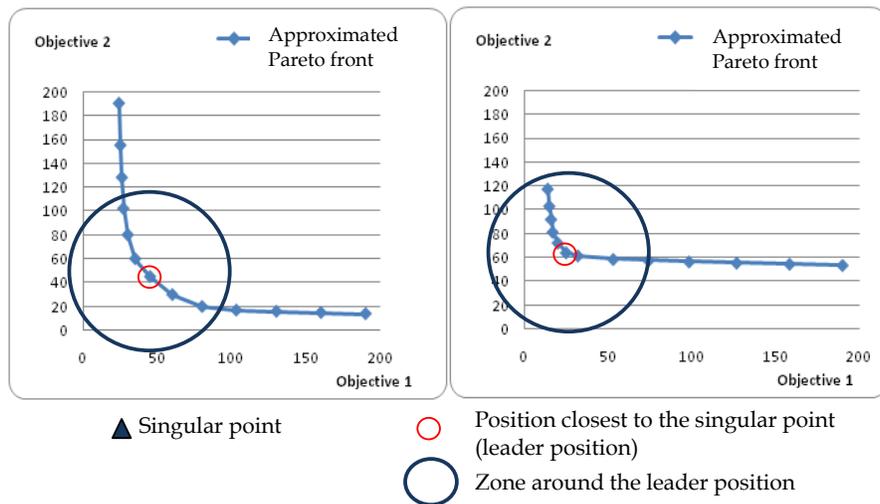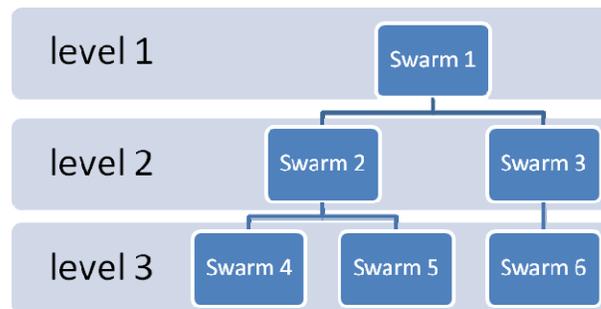

▲ Singular point   ○ Position closest to the singular point (leader position)

⬤ Zone around the leader position

Figure 1. Singular point in an approximated Pareto front

In Figure 1 two examples of approximated Pareto fronts, singular points and leaders of the swarms are represented. The Pareto front on the right is clearly non symmetric. Perhaps it would be interesting to add additional swarms to have symmetric exploration.

Swarms can also increase their population automatically when needed. Also, agents corresponding to best solutions already belonging to the Pareto front can find another solution belonging to the Pareto front. In this situation, a new clone of the agent will be placed at the newly found solution. The initial population of agents in a swarm may not be enough to have agents distributed all along the whole Pareto front. The possibility of increasing the population helps solve this problem. To avoid unlimited increase of the population, cloned agents are allowed to appear only at points located further than a minimum distance of any other solution in the Pareto front. This minimum distance is established a priori for every single objective.

2.1   Construction of the Pareto front

A time consuming task for any population-based algorithm used to solve multi-objective optimization problems is to determine which solutions belongs to the Pareto front when there is already a high number of solutions belonging to it, Deb [2002]. In ASO a hierarchy of swarms is used for fulfilling this task profiting from parallel and distributed computing. Various swarms can search a subset of the approximated Pareto front. To know if a solution belongs to the approximated Pareto front, one swarm 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 ask asynchronously its upper level swarm to check if the solution is dominated or not. The process is repeated at upper hierarchical level if the solution is found to be non-dominated; in that case all the swarms involved in the process will have the information of the new non-dominated solution. While swarms are waiting for asynchrony responses from their superiors, solutions are assumed to belong to their Pareto subsets. When a swarm receives a request to check if a solution belongs to its own subset of the Pareto front, it only uses those solutions belonging to its own Pareto subset that do not come from the swarms requesting 2 hierarchy of swarms.

Figure represents a

Figure 2. Hierarchy of swarms

2.2   The algorithm

In the approach presented in this work, agents based on the behaviour of particles in PSO are used. Agents with different behaviours can be included in a straightforward manner. Also, swarms running in parallel can be distributed in different computers, ensuring the communication among them; a peer to peer scenario may be a good choice. The steps of the algorithm can be summarized as follows.

For each swarm, in parallel, do:
1. Get connected to the hierarchical structure.
2. Run in parallel the awareness to external requests.
3. Set up parameters and initialize the number of iterations, k, to zero.
4. Generate a random population of M particles (agents): $\left\{X_i(k)\right\}_{i=1}^{M}$
5. Evaluate the fitness of the particles (agents) and set the local best location for every particle equal to its current location.
6. Form the approximate Pareto front and make a list of particles (agents) belonging to it.
7. Build the singular point.
8. Find the closest particle (agent) to the singular point and establish it as the swarm leader.
9. While not termination-condition, do the following:
   a. Execute asynchronously from $i = 1$ to number of particles (agents).
      Start
      i. Update the position of the particle (agent):
         (For PSO, determine the inertia parameter $\omega(k)$ - see section 3-, calculate the new velocity and set the new position for particle i).
      ii. Calculate the new fitness vector for particle (agent) i at its new position.
      iii. If the new fitness vector for particle (agent) i dominates the fitness vector that the particle (agent) had before moving to the new position, then set the new position as the best position found up to now by particle i.
      iv. If particle (agent) i is in the list of particles belonging to the Pareto front then:
            if the new fitness vector could also be a point at the Pareto front and this new position has at least one of its neighbours located further than the minimal permissible distance from any of the objectives, then add a new particle j (a clone of i) located at the current position of i;
         else
            try to add (if possible) the particle (agent) i (at its new position) to the Pareto front; if the particle (agent) is added, remove from the list any dominated solution; dominated clones are eliminated from the swarm. This step involves the interchange of swarms located at different hierarchical levels.
      v. If particle (agent) i is closer to the singular point than any other particle in the swarm then set particle i as the leader of the swarm with regard to the singular point.
      vi. If particle (agent) i is not currently the leader of the swarm, but coincides in position with the leader, then re-generate particle i randomly.
      End
   b. Increase the iteration number.
10. Show the Pareto front and related results.

These steps may be understood also for the general case in which agents are completely different from particles in PSO. In that case, the main changes would happen at step (i),

where the agent updates its position. It is convenient to use step (vi) only for agents that behave similarly to particles in PSO. It helps enrich the search, Montalvo et al. [2008].

It is not easy to find a general heuristic rule for deciding which parts of the Pareto front should be more detailed and how much detail the representation of the Pareto front should contain. These decisions are strongly dependent on the people solving the problem and on the problem itself. In this work, the user can specify additional points where the algorithm should focus the search, and how much detail a region should contain. This is achieved in real time during the execution of the algorithm. Human interaction with the algorithm in real time also enables the incorporation of human behaviour, so that the human turns out to be another member of the swarm. This means that a new solution can be proposed to the algorithm at any time and the algorithm should be able to fit it on the Pareto front, if appropriate. Proposed solutions could even become leaders of the swarm if they are good enough. At this point, human behaviour begins to have a proactive role during the evolution of the algorithm. In addition, humans could also add in real time new different swarms to the solution process. These swarms would enrich the search by having different ways to select their own leaders.

3.   APPLICATION TO WATER DISTRIBUTION SYSTEM DESIGN

Two case studies, related to sizing pipes in a water network, were used for testing the proposed algorithm. The first is a benchmarking problem in the hydraulic literature known as the Hanoi network, Fujiwara and Khang [1990]. In this problem it was needed to find the minimum investment cost for a water network, constrained to have at least a minimum pressure value at the demand nodes. This constrain was transformed in a new objective within the multi-objective approach: to find solutions minimizing the investment and minimizing the lack of pressure at demand nodes. The lack of pressure at a demand node is understood as the difference between the minimum required pressure and the calculated pressure at the node. When the calculated pressure is bigger than the minimum required pressure then it is assumed that the lack of pressure is equal to zero. Solutions for the benchmarking problem were compared to solutions obtained by different authors. Good known solutions were found to be part of the obtained Pareto front.

The other case study correspond to a real world design (a sector of Lima's network) where three objectives were considered: minimizing the investment cost, minimizing the lack of pressure at demand nodes and minimizing additional costs because of reliability issues. Details about this network may be found in Montalvo et al. [2010a].

A two dimensional representation of the Pareto front for each case is shown in Figure 3. In both cases it is seen that after some point, the rate at which the minimum pressure could be increased in the network is much lower than the rate at which initial investment costs have to be increased for having the desired pressure level.
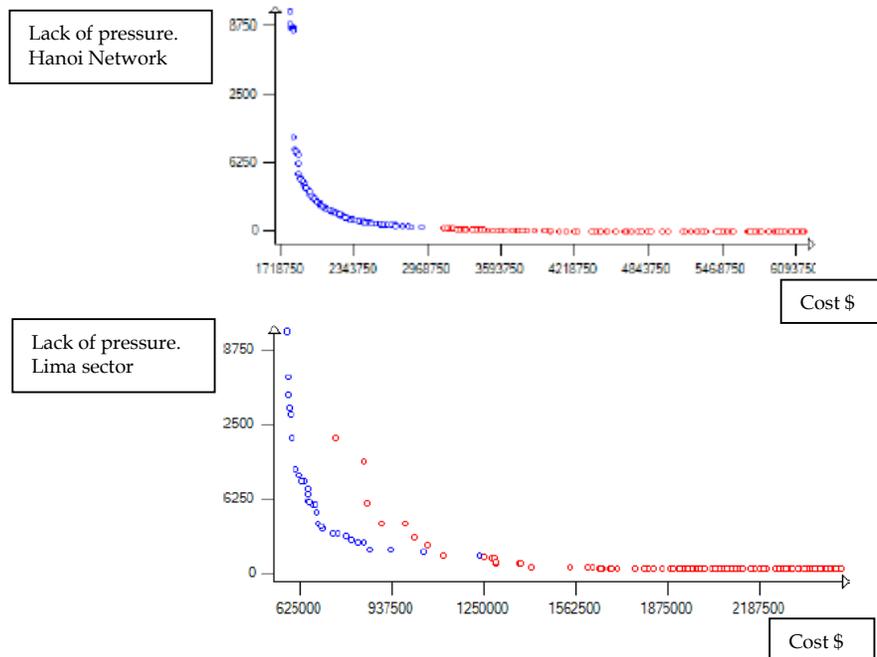
Figure 3. Pareto fronts for the studied cases

The relationship between initial investment cost and minimum pressure in the network that can be obtained for a water supply network may help decide, among other factors, which pressure would be more convenient to use for the final solution. It is an added-value of the multi-objective approach used in this work for sizing pipe networks.

To evaluate the hydraulic performance of the solutions EPANET2, Rossman [2000], a package for water network analysis in steady state, has been used. Additional analyses of quasi-steady state or various transient analyses can be performed without any change in the core of the algorithm. Agent Swarm Optimization and its connection with EPANET2 has been implemented in a software called WaterIng, Montalvo et al. [2009], developed for water distribution system design and analysis.

Some parameters of the algorithm were established a priori for running the two examples. The initial population size was set equal to 20. The maximum velocity was set equal to 50% of the variable range, Montalvo et al. [2008]. The minimum velocity was set equal to minus maximum velocity. Finally, the inertia weight was calculated by using the expression:

$$\omega = 0.5 + \frac{1}{2(\ln(k)+1)} \, .$$

This expression was proposed in Jin et al. [2007] and has shown to produce very good results. Fine tuning of parameters can be performed somehow or, alternatively, some self-adaptive techniques could be used, as proposed in Montalvo et al. [2010b].

4.   CONCLUSIONS

Evolutionary algorithms represent, in general, a good choice to solve complex multi-objective optimization problems. Each algorithm has advantages and disadvantages, and its performance depends on the characteristics of the problem at hand. Agent Swarm Optimization (ASO), the algorithm proposed in this work, can profit from the best of various algorithms when solving complex multi-objective optimization problems.

Accordingly, ASO can be considered as a common framework for population-based algorithms in general, and can be used for solving general multi-objective optimization problems. Specifically, we have checked that ASO produces excellent results when applied to water distribution system design. It is a very complex, mixed discrete-continuous, multi-objective, NP-hard problem to which much effort has been devoted in the literature. The results we obtain by using the proposed approach are well within the best results found in the literature when solving various benchmarking problems. But, in addition, the multi-objective approach allows people involved in the process to make their decision according to a number of criteria.

Also, among the most important capabilities of the algorithm, the inclusion of a high human interaction during the solution process – an approach not used so far, to the authors knowledge –, must be enhanced. Users can focus the search in some desired part of the Pareto front and can also influence the behaviour of the agents by proposing potential solutions to the algorithm.

The use of parallel and distributed computing makes it possible for ASO to run in various computers involving several swarms at the same time. The hierarchical mechanism used for constructing the approximated Pareto front helps make this process more efficient. Additionally, the algorithms can increase populations automatically when needed in order to have enough agents for covering the whole Pareto front. These characteristics help obtain good approximations of the Pareto front for the addressed problem.

Last but not least it should be remarked that in the case of ASO, rule-based agents can also be added to the solution process. The use of rule-based agents increases the probability of finding good solutions for a problem because those agents are closer to the essence of the problem. ASO makes it possible to have rule-based agents and evolutionary algorithms working together for solving the same problem.

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. It is important to note that the way on which objectives are evaluated and the determination of which objectives must be used are key points for obtaining good results. Extensions to this work are perfectly possible for solving successfully other complex engineering problems.

ACKNOWLEDGMENTS

REFERENCES

Coello Coello, C. A., G. B. Lamont and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Second Edition, Springer, 800 pp., 2007.

Deb, K., *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley, 515 pp., 2002

Dorigo, M., V. Maniezzo and A. Colorni, The ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics 26(1): 29-41, 1996.*

Fujiwara, O. and D. B. Khang, A two-phase decomposition method for optimal design of looped water distribution networks, *Water Resources Research* 26(4): 539-549, 1990.

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

Kennedy, J. and R. C. Eberhart, Particle swarm optimization, paper presented at IEEE International Conference on Neural Networks, Perth, Australia, IEEE Service Center, Piscataway, NJ, 1995.

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

Montalvo, I., J. Izquierdo, S. Schwarze, and R. Pérez-García, Software para el dimensionamiento óptimo de tuberías en los sistemas de distribución de agua: una aproximación multi-objetivo, IX SEREA Seminario Iberoamericano sobre Planificación, Proyecto y Operación de Sistemas de Abastecimiento de Agua, Valencia, Spain, November 24-27, 2009.

Montalvo, I., J. Izquierdo, S. Schwarze, and R. Pérez-García, Multi-objective Particle Swarm Optimization applied to water distribution systems design: an approach with human interaction, Mathematical and Computer Modelling, doi:10.1016/j.mcm.2010.02.017, 2010a.

Montalvo, I., J. Izquierdo, R. Pérez-García 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*, doi:10.1016/j.engappai.2010.01.015, 2010b.

Montesinos, P., A. García-Guzmán and J. L. Ayuso, Water distribution network optimization using a modified genetic algorithm, *Water Resources Research* 35(11): 3467-3473, 1999.

Rossman, L. A., *EPANET2, User's Manual*. Cincinati (IN), USA, Environmental Protection Agency, 2000.