



Jun 17th, 2:00 PM - 3:20 PM

Problem-specific Rule Extraction for Better Performance of Evolutionary Algorithms


Joaquin Izquierdo
Universitat Politècnica de València, jizquier@upv.es

Idel Montalvo
3SConsult, montalvo@3sconsult.de

Enrique Campbell
Universitat Politècnica de València, encamgo1@upv.es

Rafael Pérez-García
Universitat Politècnica de València, rperez@upv.es

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

 Part of the [Civil Engineering Commons](#), [Data Storage Systems Commons](#), [Environmental Engineering Commons](#), [Hydraulic Engineering Commons](#), and the [Other Civil and Environmental Engineering Commons](#)

Izquierdo, Joaquin; Montalvo, Idel; Campbell, Enrique; and Pérez-García, Rafael, "Problem-specific Rule Extraction for Better Performance of Evolutionary Algorithms" (2014). *International Congress on Environmental Modelling and Software*. 18.

<https://scholarsarchive.byu.edu/iemssconference/2014/Stream-G/18>

This Event is brought to you for free and open access by the Civil and Environmental Engineering at BYU ScholarsArchive. It has been accepted for inclusion in International Congress on Environmental Modelling and Software by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Problem-specific Rule Extraction for Better Performance of Evolutionary Algorithms

Joaquín Izquierdo^a, Idel Montalvo^b, Enrique Campbell^a, Rafael Pérez-García^a

^a *FluIng-IMM, Universitat Politècnica de València*
jizquier@upv.es, encamgo1@upv.es, rperez@upv.es
^b *3SConsult, Karlsruhe, Germany*
montalvo@3sconsult.de

Abstract: Evolutionary algorithms (EAs) are ubiquitous to many environmental issues. Typically, most evolutionary techniques use just a small part of the solutions evaluated at a time. Thus, many of the solutions evaluated during the search process are “forgotten” after one generation, and combined experience of several generations is typically not well exploited. Data mining (DM) techniques can enable deeper insight into the many “good” solutions that have been just simply glimpsed and have been rapidly disregarded because they were dominated by better solutions during an ephemeral moment in the evolution process. Based on a database obtained by suitably recording certain of those disregarded solutions, data mining techniques can help better understand and describe how a system could react or behave after the introduction of changes. This paper proposes applying DM techniques to the set of solutions evaluated after several generations of a single run of an EA in order to extract rules intended initially to be used by the following generations.

Keywords: Evolutionary algorithms; data mining; rule extraction; search space reduction.

1. INTRODUCTION

Optimization is central to most human endeavors. Most decision-making processes in almost any field can be stated as optimization problems. Optimization is also ubiquitous to many environmental issues.

Since most real-world processes are very complex, classical techniques of optimization, depending on too strict mathematical properties, have shown not to be suitable for optimization purposes on some real-world, complex problems. Evolutionary algorithms (EAs) are stochastic optimization techniques that avoid various mathematical complications. EAs handle a number of variable populations of solutions for the problem in hand, in pursuance of the best individual(s), thus representing the best solution(s) of the problem. The flexibility introduced by evolutionary algorithms has allowed the use of virtually any objective function for evaluating solutions, even when these evaluations require running complex mathematical and/or procedural simulations of the systems under analysis. Many applications have been developed in the environmental field including description, planning, design, operation, maintenance, forecasting, etc. (Kim et al., 2012; Ni et al., 2012; Zhang et al., 2012; Jain, 2012; Taghavifar et al., 2013; Wendt et al., 2013; Montalvo et al., 2014), among many others.

Typically, during an iteration step, most evolutionary techniques use just a small part of the evaluated solutions, in an attempt to preserve the best among the explored solutions. This perhaps may seem natural in single-objective optimization. However, taking into account the multimodal character of most real-world problems, suboptimal solutions are sometimes rejected that are close to good – not necessarily corresponding to the global optimum – solutions. Moreover, in multi-objective optimization, in which the concept of best solution is broadened to the concept of Pareto front, it is frequent that suboptimal solutions may be disregarded when they are dominated by other dominant solutions.

As a result, many of the good solutions evaluated during the search process are “forgotten” after one generation, and combined experience of several generations is typically not well exploited.

In addition, some of those disregarded solutions would have been able to obtain a higher score if the constraints and objectives of the problem had been somehow slightly different. Since many parameters, variables, bounds, etc., in environmental processes are of uncertain nature, perhaps some of the neglected solutions would have been better in a slightly different setting. Even more, since some of those parameters or properties are changing with time, perhaps some of those neglected solutions would be the best for some specific states of the studied system.

Data mining (DM) techniques can enable deeper insight into the many “good” solutions that have been just simply glimpsed and have been rapidly disregarded because they were dominated by better solutions during an ephemeral moment in the evolution process. Based on a database obtained by suitably recording certain of those disregarded solutions, data mining techniques can help better understand and describe how a system could react or behave after the introduction of changes.

This paper proposes applying DM techniques to the set of solutions evaluated after several generations of a single run of an EA in order to extract rules intended initially to be used by the following generations, so that the search efficiency is significantly improved. Improvement derives from a number of facts: rules are intended to better describe specific characteristics of the problem being solved, and rules are supposed to narrow search intervals for certain variables, thus globally reducing the search space.

2. KNOWLEDGE DISCOVERY ENRICHED SEARCH

Evolutionary algorithms have brought great flexibility for evaluating objectives in a way hardly achievable by other methods. Running mathematical simulations of many environmental settings under different scenarios can be now part of a global decision-making process where solutions evolve based on some “rules” defined in the evolutionary algorithm in use. These rules strictly focus on how transforming a population of (hopefully good) solutions into another population of better solutions during an evolutionary process.

Each algorithm has its own set of rules and most of them are based on the emulation of a paradigm. Genetic Algorithms (GA) (Goldberg, 1989), for example, use rules based on the theory of natural evolution. In Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995), rules are based on the way birds in a flock find their destination, which is more inspired on social evolution than on a genetic evolution. Ant Colony Optimization rules (Dorigo et al., 1996) follow the foraging behavior of ants. Simulated annealing (Kirkpatrick et al., 1983) takes its name and inspiration from annealing in metallurgy. There are many other examples with different sources of inspiration using a wide variety of rules.

The selection of the most appropriate algorithm to solve a specific problem is not straightforward. Some algorithms perform better than others in some problems and worse or very poorly in some others. This fact indicates that their rules apply better to certain problems than to others.

Also the selection of suitable parameters for a given EA is a key issue. Various attempts to dynamically change the parameters of an EA have been explored (Krasnogor and Gustafson, 2004; Eusuff et al., 2006; Haupt and Werner, 2007; Al-Anzi and Allahverdi, 2007; Mezura-Montes and Palomeque-Ortiz, 2009; Montalvo et al., 2010), among others. Nevertheless, the real big step in improving the performance of EAs in real-world problems will derive from directly influencing the way the search of solutions is performed.

Good results can be obtained for relative small and simplified problems – there are many examples in the literature –, but good solutions are harder to obtain for larger problems and, in addition, those solutions can differ from what engineering good practice and sense could suggest (Montalvo, 2011). Now the question is: how the solution search process is done when the size of problems increase and the interrelations among variables and objectives get more complex?

Traditionally, the solution search process has been totally ignorant of the specific problem being solved. The process has been the same no matter the size, the complexity and the problem domain. In these days of cloud computing, authors can be tempted to use more and more computing resources for solving “real world problems”. It is not a completely wrong idea, but taking it literally and without carefully thinking on the improvement of the search process itself would lead researchers closer to brute force approaches to find the best objective than to clever ways of exploring and exploiting the solution space, thus investing a reasonable amount of resources in the search process. This is especially crucial for many environmental optimization problems, which have shown to be NP-hard problems.

We argue that the search process should be as close as possible to the specific problem being solved. Algorithms adapting their behaviors to the problem in hand will have more chances to succeed. A way to approach this idea could combine the way evolutionary algorithms work with the introduction of knowledge discovery from a suitable database of solutions visited during previous steps of the optimization process. We develop this idea further in the following section, taking into account that the database thus generated may be of considerable size.

3. RULES TO TAILOR THE SEARCH PROCESS

Various data mining techniques have been successfully applied to various areas that have to handle large volumes of data, as tools to scan the available information and thus track down understandable and useful knowledge (Bouguessa et al., 2009; Hsu and Chen, 2009). IF-THEN rules are a common way of passing knowledge, since they are both easy to understand and to implement in software programs. Moreover, adaptive techniques allow adding new rules and removing or modifying existing ones. This is why rules are used by most existing techniques to produce knowledge (Gonçalves et al., 2006; Kamwa et al., 2009; Hasperué et al., 2012).

Rules based on the domain of the problem can be enforced and amalgamated with evolutionary techniques. The use of rule-based agents is, for example, one of the principles in Agent Swarm Optimization (ASO) (Montalvo et al., 2014), which includes rules for taking the EA in use closer to the problem being solved.

In the following paragraph we explore a process that endows an EA with the ability of dynamically generating rules aimed at being used to improve the efficiency of the solution search process.

3.1 Mining rules in the evolutionary history

In many applications, interesting associations among variables often occur at a relatively high concept level. For example, design patterns in a database made out of solutions for a water distribution network obtained during an EA run may not show any substantial regularity at the primitive data level, such as the individual solution fitness level. Nevertheless, the same designs may show some interesting regularities at some high concept level(s), for example, relationships among certain variables, such as pipe diameters. Therefore, it is important to study association rules at a generalized abstraction level (Srikant and Agrawal, 1995), or at multiple levels (Han and Fu, 1995).

Given a database of solutions obtained during a run of an EA, it is of interest to discover associations among decision variables showing that the presence of some values for certain variables in a solution implies the presence of other specific values for other variables in the same solution. A mathematical model was proposed by Agrawal et al. (1993) to address the problem of mining association rules.

The basic mathematical structure to address the problem of mining association rules is the following. Let $I = \{i_1, i_2, \dots, i_n\}$ be a set of n binary attributes called items. Let D be a set of transactions, where each transaction T is a set of items, which is included in I . Let X be a set of items. A transaction T is said to contain X if $X \subseteq T$. An association rule is an implication of the form $X \rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. The set of items X is called antecedent or left-hand-side (LHS) of the rule, and Y is called consequent or right-hand-side (RHS) of the rule. The rule $X \rightarrow Y$ holds in the set D with

confidence c if $c\%$ of registers in D that contain X also contain Y . The rule $X \rightarrow Y$ has *support* s in the transaction set D if $s\%$ of registers in D contain $X \cup Y$. Confidence denotes the strength of implication, and support indicates the frequencies of the occurring patterns in the rule. It is often desirable to pay attention to only those rules which may have reasonably large support and confidence. Such rules with high confidence and strong support are referred to as strong rules in (Agrawal et al., 1993) and (Piatetsky-Shapiro, 1991). The task of mining association rules is essential to discover strong association rules in large databases. The problem of mining association rules may be decomposed into the following two steps:

- 1) Discover the large itemsets, i.e., the sets of itemsets that have transaction support above a predetermined minimum support s .
- 2) Use the large itemsets to generate the association rules for the database.

In this paper, to perform these two steps and to extract rules from a database of solutions generated by an EA during the search process, we propose using the so-called Apriori algorithm (Agrawal and Srikant, 1994). The algorithm works over a so-called transaction matrix which is a binary matrix where rows represent transactions and there is one column for each of the possible values that might be assumed by each item. The presence of a given item in a given transaction is indicated by 1 and the absence by 0. In addition, we also use another measure called *lift*, obtained by dividing the support of $X \cup Y$, $\text{supp}(X \cup Y)$ by the product of supports of the antecedent X and the subsequent Y , $\text{supp}(X) \times \text{supp}(Y)$. Measures are used to establish limits for minimal frequency or topological characteristics when generating rules.

3.2 Putting rule extraction and EA to work together

The typical operation of a given EA aided by rule extraction we propose is the following. When initializing the EA only random solutions are available. As a result, there is no possibility of rule extraction and the EA, using its own search mechanisms, must work during a number of iterations to produce and collect new solution candidates. After this number of iterations a DB must have been created. The EA will then stop the search, and the rule extraction algorithm will be launched to work on the DB. Hopefully, a number of rules will be obtained that will guide the EA during a new batch of iterations. Then a new DB of candidate solutions will be available. Again the rule extraction algorithm will produce probably new rules that, in their turn, will be used during the subsequent process of iteration. Assuming that the use of rules will accelerate the convergence of the EA and taking into account that the EA is controlled by a certain termination condition it is expected that only a limited number of rule extraction processes will be eventually performed. When to stop the EA to launch a rule extraction process is a matter that will need further insight and the target should be towards automatic execution. The following pseudo-code presents a simple flow diagram for the combined work.

- 1- EA initialization
- 2- First batch of iterations (with no rules; only EA search mechanism apply)
- 3- Rule extraction
- 4- New batch of iterations using the new rules to mold the EA search
- 5- If 'no termination condition' then go to 3
- 6- End

4. CASE STUDY

In this paragraph we apply the above ideas to the problem of the design of a water distribution system. This is a very well-known NP-hard problem in hydraulic engineering with mixed (continuous and discrete) variables defined by various non-linear objectives and constraints (Izquierdo et al., 2012). The EA we use is PSO, specifically a variant introduced by the authors (Montalvo, 2011; Izquierdo et al., 2012). In this specific case we have controlled manually the size of the first batch of iterations, and the termination condition was set to stop the search after 800 iterations without improvement. We also present and discuss the results obtained.

4.1 Problem description: the Hanoi network

The Hanoi water distribution problem is a very well-known benchmark problem in the literature attacked many times before; see (Cunha and Sousa, 1999; Matías, 2003; Savic and Walters, 1995; Zecchin et al., 2005; Montalvo et al., 2008a,b), among many others. Figure 1 contains a representation of the network, which consists of one fixed head source, 34 pipes and 31 demand nodes subject to a load condition (figures associated to the nodes are the demands in cubic meters per hour, also visualized by the nodes' size). Furthermore, the network has three grids and three ramified branches. One has to find the diameters, within a pool of six available commercial diameters, namely, 304.8, 406.4, 508.0, 609.6, 762.0 and 1016.0mm (identified in the sequel with numbers from 0 to 5, respectively), for the 34 pipes such that the total cost of this network is minimal and the pressure at each node of consumption is at least 30m. The complete setting can be found in (Wu and Simpson, 2001). To gauge the effectiveness of our proposed algorithm, we apply it to the Hanoi network problem.

4.2 Results

After launching one run of the optimization algorithm with a population of 100 individuals, a database with 13100 registers is obtained corresponding to the first 131 iterations. Thirty five columns constitute the fields of the database, which correspond to values for the diameter for each of the 34 pipes in the network, plus the objective value, corresponding to the cost of the network summed with the penalty incurred for not satisfying the minimum pressure value of 30m.

In this approach we discretize the objective field into four categories to obtain a *qualitative cost* attribute. The first category includes the *excellent* solutions, corresponding to registers with objective between 0 and 3% over the cheapest solution in the database; then *good* solutions include those registers with objective between 3% and 5% over the cheapest solution; *fair* solutions are those with a cost between 5% and 15% over the cheapest solution; and, finally, the rest of solutions constitute the class of *bad* solutions. Specific knowledge of the problem from an engineering point of view led us to choose these specific four categories, which in this particular problem are aimed at being very restrictive with the concepts of excellent and good solutions. Of course, typically, excellent solutions are sought. Nevertheless, excellent (cheapest in this case) solutions not always are the best in engineering practice. Moreover, good solutions may well represent the system under different load conditions and, as a consequence, it is worth to take them into consideration, especially if static design is not the only target. Solutions more expensive than 5% of the best known solution are deemed not to be desirable solutions anyway. In general, since objectives in optimization problems are very varied and strongly depend of the problem at hand our proposal includes the use of expert knowledge working in close synergy with the computer.

Applying the Apriori algorithm a transaction matrix of 13100 rows and 108 items is built. This last figure corresponds to the total number of the possible values assumed by all the items. In this case, it represents an average value of about 3 different diameters for each pipe in the database. The algorithm has to extract a number of rules from this transaction matrix. Nevertheless, it may happen, as in the database used, that some of the variables are not relevant to discover new rules since they exhibit one specific value in a very high (very close to 100%) percentage of registers. At a given stage of the evolution, these 'fixed' values may correspond either to optimal (target) values or to variables that have not been completely explored. A decision must be made according to the evolution stage. In early evolution stages those values will be just disregarded since, with high probability, they correspond to solutions not well explored so far, thus corresponding to local minima, from where the EA, using its stochastic abilities, should escape. On the other hand, in advanced evolution stages, these values would be directly transformed into fixed rules for those variables, if the solutions correspond to so-far excellent solutions. In any case, these variables are (temporarily) eliminated from the database for the current situation.

By eliminating these variables (pipes) a new database with the same 13100 registers and 17 fields (16 pipes plus the qualitative cost) is obtained. We apply Apriori over this new database. The new transaction matrix contains 94 items. In the next sub-paragraph we present the results obtained.

4.3 Analysis of results

Results are analyzed according to the three measures we use: support, confidence and lift. For any of those parameters a subset of 10 rules is extracted for each type of solution (excellent, good, fair and bad), and the rules are ordered increasingly according to the selected parameter. Table 1 shows an example of a mixed subset of rules ordered according to parameter support. Paper length constraints prevent us to include the complete set of rules generated in the process.

Table 1. Example of a subset of obtained rules ordered by parameter support

Rules	support	confidence	lift
{objective=excellent} => {PIPES15=0}	0.07183206	1	1.13351216
{objective=excellent} => {PIPES4=3}	0.07091603	0.98724761	1.39048959
{objective=good} => {PIPES15=0}	0.04091603	1	1.13351216
{objective=good} => {PIPES4=3}	0.03931298	0.9608209	1.35326887
{objective=fair} => {PIPES15=0}	0.1621374	0.97030608	1.09985373
{objective=fair} => {PIPES14=2}	0.14045802	0.84056647	1.18593654

When using the support parameter, only four pipes appear in the set of 10 rules, and diameters for these pipes appear throughout excellent, good, fair and bad solutions (see Table 1). Thus, this parameter is not suitable, in this situation, to determine reasonable rules. If the confidence parameter is used, some diameters appear for the excellent and good solutions, which clearly differ from the diameters for fair and bad solutions. This is also the situation, when the lift parameter is used. In these two cases, the rules obtained allow a soft characterization of the involved pipes.

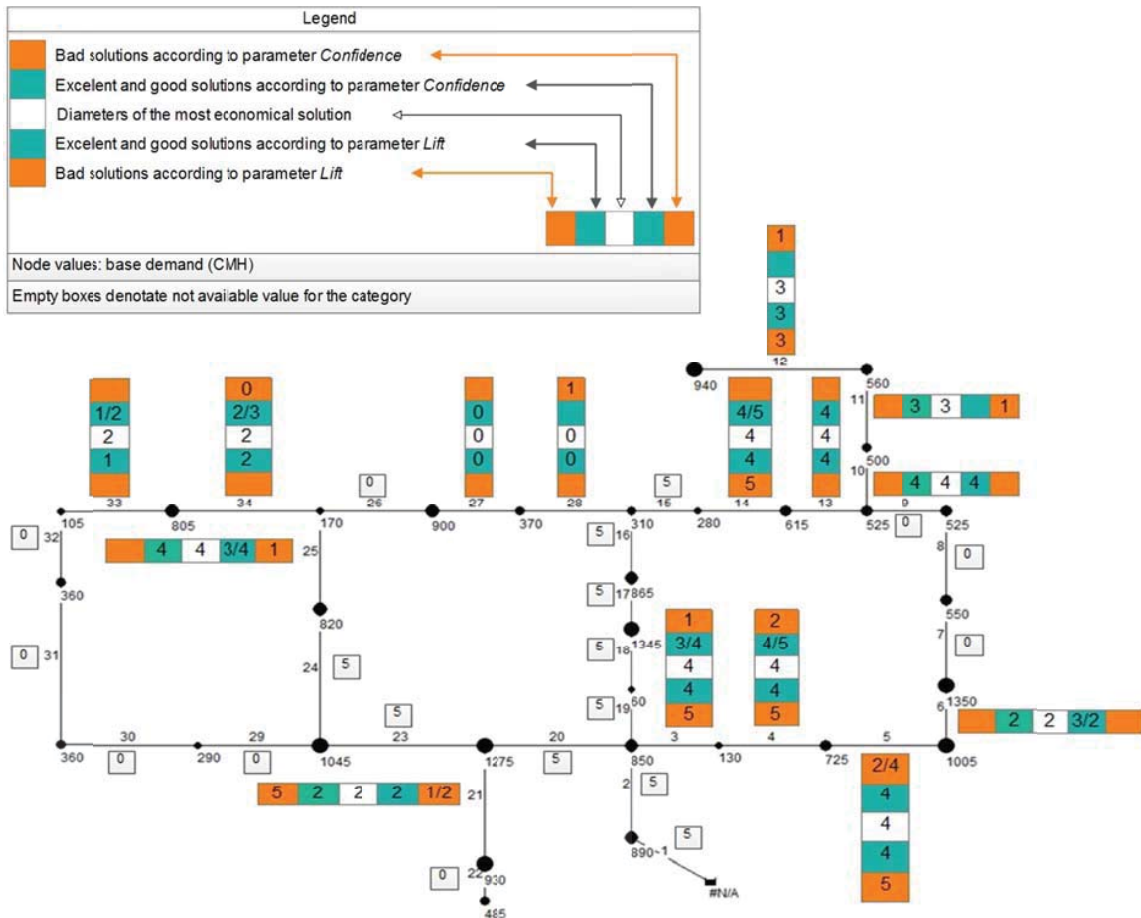


Figure 1. Conceptual map of diameter rules obtained after the described analysis (pipes with just one box correspond to ‘fixed’ diameter pipes in the DB, coincidental with most economical solution values).

For the sake of brevity, we do not include these rules. Instead, the relevant set of results is shown in Figure 1, which can be seen as a conceptual map of rules for the EA to apply at the present stage of the evolution. Information in this map is perfectly consistent from an engineering viewpoint, including hydraulic conditions and demand satisfaction. The rules obtained enable the reduction of the number of candidate diameters for some pipes, thus reducing the search space and the number of iterations.

5. CONCLUSIONS, RECOMMENDATIONS AND FUTURE WORK

In this paper, we have explored the use of data mining techniques, specifically, rules obtained from a suitable database of solutions generated by an EA to guide the search towards more problem-relevant solutions to improve the search efficiency. We have applied a classical rule-extraction algorithm to a database obtained when trying to get the best design for a water distribution network used in the literature as a benchmark, namely, the Hanoi network. The results are very promising, showing a productive way to enhance EA search in real-world optimization problems. To develop further and efficiently the ideas described in this paper a number of research lines should be explored.

Firstly, the scalability of the approach herein presented should be demonstrated by applying it to bigger problems, closer to what is understood as real-world problems.

Secondly, updating techniques should be developed for maintenance of the discovered association rules during the whole optimization process, to avoid complete redoing data mining on the whole updated database every time new better solutions are obtained by the EA. The database must undergo periodic updates (at least during the first stages of the search), and such updates may not only invalidate some existing strong association rules but also turn some weak rules into strong ones. It is nontrivial to maintain such discovered rules in large databases. A good idea would be to reuse the information of the old large itemsets and to integrate the support information of the new large itemsets in order to substantially reduce the pool of candidate sets to be re-examined.

Finally, since the whole process, in general, will require progressive knowledge collection and revision, achieving efficient parallel computing is deemed a necessity, since, data transmission required for reaching global decisions can be prohibitively large, thus significantly compromising the benefit achievable from parallelization.

6. REFERENCES

- Agrawal, R., Imielinski, T., Swami, A., 1993. Database Mining: A Performance Perspective, *IEEE T Knowl Data Eng*, 5(6), 914-925.
- Agrawal, R., Srikant, R., 1994. Fast Algorithms for Mining Association Rules in Large Databases, *Proc. 20th Int'l Conf. Very Large Data Bases*, 478-499.
- Al-Anzi, F.S., Allahverdi, A., 2007. A self-adaptive differential evolution heuristic for two-stage assembly scheduling problem to minimize maximum lateness with setup times. *Eur J Oper Res*, 182(1), 80-94.
- Bouguessa, M., Wang, S., 2009. Mining Projected Clusters in High-Dimensional Spaces. *IEEE T Knowl Data Eng*, 21(4), 507-522
- Cunha, M.C., Sousa, J., 1999. Water distribution network design optimization: simulated annealing approach. *J Water Res Pl-ASCE*, 125(4): 215-221.
- Dorigo, M., Maniezzo, V., Coloni, A., 1996. The ant system: optimization by a colony of cooperating ants, *IEEE T Syst Man Cy B*, 26(1), 1-13.
- Eusuff, M., Lansey, K., Pasha, F., 2006. Shuffled frog-leaping algorithm: a memetic metaheuristic for discrete optimization. *Eng Optimiz*, 38(2), 129-154.
- Goldberg, D.E., 1989. *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley, Reading, Ma.
- Gonçalves, L. B., Vellasco, M. M. B. R., Pacheco, M. A. C., de Souza, F. J., 2006. Inverted Hierarchical Neuro-Fuzzy BSP System: A Novel Neuro-Fuzzy Model for Pattern Classification and Rule Extraction in Databases. *IEEE T Syst Man Cy C*, 36(2), 236-248.
- Han, J., Fu, Y., 1995. Discovery of Multiple-Level Association Rules from Large Databases, *Proc. 21th Int'l Conf. Very Large Data Bases*, 420-431.

- Hasperué, W., Lanzarini, L., De Guisti, A., 2012. Rule Extraction on Numeric Datasets Using Hyper-rectangles. *Comput Inform Sci*, 5(4), 116-131.
- Haupt, R.L., Werner, D.H., 2007. *Genetic Algorithms in Electromagnetics*. John Wiley & Sons, New Jersey.
- Hsu, C.-M., Chen, M.-S., 2009. On the Design and Applicability of Distance Functions in High-Dimensional Data Space. *IEEE T Knowl Data Eng*, 21 (4), 523-536.
- Izquierdo, J., Montalvo, I., Pérez-García, R., Matías, A., 2012. On the Complexities of the Design of Water Distribution Networks. *Math Probl Eng*, Volume 2012, Article ID 947961, 25 pages.
- Jain, S.K., 2012. Modeling river stage-discharge-sediment rating relation using support vector regression. *Hydrol Res*, 43(6), 851-861.
- Kamwa, I., Samantaray, S.R., Joos, G., 2009. Development of Rule-Based Classifiers for Rapid Stability Assessment of Wide-Area Post-Disturbance Records. *IEEE T Power Syst*, 24(1), 258-270.
- Kennedy, J., Eberhart, R.C., 1995. Particle swarm optimization, in *Proceedings of the IEEE International Conference on Neural Networks*, Piscataway, NJ, 1942-1948.
- Kim, D.K., Jeong, K.S., McKay, R.I.B., Chon, T.S., Joo, G.J., 2012. Machine Learning for Predictive Management: Short and Long term Prediction of Phytoplankton Biomass using Genetic Algorithm Based Recurrent Neural Networks. *Int J Environ Res*, 6(1), 95-108.
- Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by Simulated Annealing. *Sci*, 220 (4598): 671-680.
- Krasnogor, N., Gustafson, S., 2004. A study on the use of "self-generation" in memetic algorithms. *Nat Comput Int J*, 3(1), 53-76.
- Matías, A.S., 2003. *Diseño de redes de distribución de agua contemplando la fiabilidad, mediante Algoritmos Genéticos*. Universitat Politècnica de València. Tesis doctoral. Valencia, Spain.
- Mezura-Montes, E., Palomeque-Ortiz, A.G., 2009. Self-adaptive and deterministic parameter control in differential evolution for constrained optimization. In: Efrén Mezura-Montes (Ed.), *Constraint-Handling in Evolutionary Optimization*, Springer-Verlag, Studies in Computational Intelligence, vol.198, 95-120.
- Montalvo, I., Izquierdo, J., Pérez, R., Tung, M.M., 2008a. Particle Swarm Optimization applied to the design of water supply systems. *Comput Math Appl*, 56(3): 769-776.
- Montalvo, I., Izquierdo, J., Pérez, R., Iglesias, P.L., 2008b. A diversity-enriched variant of discrete PSO applied to the design of Water Distribution Networks. *Eng Optimiz*, 40(7): 655-668.
- Montalvo, I., Izquierdo, J., Pérez-García, R., Herrera, M., 2010. Improved performance of PSO with self-adaptive parameters for computing the optimal design of water supply systems. *Eng Appl Artif Intel*, 23(5): 727-735.
- Montalvo, I., 2011. *Diseño óptimo de sistemas de distribución de agua mediante agent swarm optimization*, Ph.D. thesis, Universitat Politècnica de València, Valencia, Spain.
- Montalvo, I., Izquierdo, J., Herrera, M., Pérez-García, R., 2014. Water supply system computer-aided design by Agent Swarm Optimization. *Comput Aided Civil Infrastr Eng*, DOI: 10.1111/mice.12062.
- Ni, Q.W., Wang, L., Zheng, B.H., Sivakumar, M., 2012. Evolutionary Algorithm for Water Storage Forecasting Response to Climate Change with Small Data Sets: The Wolonghu Wetland, China. *Environ Eng Sci*, 29(8), 814-820.
- Piatetsky-Shapiro, G., 1991. Discovery, Analysis, and Presentation of Strong Rules, in: G. Piatetsky-Shapiro and W. J. Frawley, eds., *Knowledge Discovery in Databases*, 229-238. AAAI/MIT Press.
- Savic, D.A., Walters, G.A., 1995. Genetic operators and constraint handling for pipe network optimization. In: *Evolutionary Computing, AISB Workshop 1995*, 154-165.
- Srikant, R., Agrawal, R., 1995. Mining Generalized Association Rules, *Proc. 21th Int'l Conf. Very Large Data Bases*, 407-419.
- Taghavifar, H., Mardani, A., Taghavifar, L., 2013. A hybridized artificial neural network and imperialist competitive algorithm optimization approach for prediction of soil compaction in soil bin facility. *Meas*, 46(8), 2288-2299.
- Wendt, K., Cortés, A., Margalef, T., 2013. Parameter calibration framework for environmental emergency models. *Simulat Model Pract Theory* 31, 10-21.
- Wu, Z.Y., Simpson, A.R., 2001. Competent genetic-evolutionary optimization of water distribution systems. *J Comput Civil Eng*, 15(2): 89-101.
- Zecchin, A.C., Simpson, A.R., Maier, H.R., Nixon, J.B., 2005. Parametric study for an ant algorithm applied to water distribution system optimization. *IEEE T Evolut Comput*, 9(2):175-191.
- Zhang, X.S., Zhao, K.G., 2012. Bayesian Neural Networks for Uncertainty Analysis of Hydrologic Modeling: A Comparison of Two Schemes. *Water Resour Manag*, 26(8), 2365-2383.