# Injecting problem-dependent knowledge to improve evolutionary optimization search ability

Joaquín Izquierdo*[1], Enrique Campbell*, Idel Montalvo**, Rafael Pérez-García*
* FluIng-IMM, Universitat Politècnica de València, Cno. de Vera s/n, 46022 Valencia (Spain)
{jizquier, encamgo1, rperez}@upv.es
**3SConsult GmbH, Albtalstrasse 13, 76137 Karlsruhe, Germany
montalvo@3sconsult.de

## ABSTRACT

The flexibility introduced by evolutionary algorithms (EAs) has allowed the use of virtually arbitrary objective functions and constraints – even when evaluations require, as for real-world problems, running complex mathematical and/or procedural simulations of the systems under analysis. Even so, EAs are not a panacea. Traditionally, the solution search process has been totally oblivious of the specific problem being solved, and optimization processes have been applied regardless of the size, complexity, and domain of the problem. In this paper, we justify our claim that far-reaching benefits may be obtained from more directly influencing how searches are performed. We propose using data mining techniques as a step for dynamically generating knowledge that can be used to improve the efficiency of solution search processes. In this paper, we use Kohonen SOMs and show an application for a well-known benchmark problem in the water distribution system design literature. The result crystalizes the conceptual rules for the EA to apply at certain stages of the evolution, which reduces the search space and accelerates convergence.

**Keywords**: non-standard optimization problem, evolutionary algorithm, knowledge-based system, SOM, water distribution

## 1 INTRODUCTION

Optimization permeates every human endeavor, in particular, science and technology. The main interest is usually placed in solving real-world problems. However, the closer a problem is to reality, the more complex it becomes. Complexity derives from a number of facts: coexistence of various (in general, conflicting) objectives; objectives defined by complex mechanisms (not only functions, but also procedures); sensitive constraints that are difficult to meet (perhaps needing simulation to be represented); nonlinear expressions (frequently associated with lack of smoothness and even continuity); dependence of many decision variables (multi-dimensionality); coexistence of various types of decision variables (mixed Boolean-integer-real); uncertainty (both for the model and the problem data); multi-modality (coexistence of many good non-optimal solutions), etc. Classical optimization techniques (including classical numerical methods for optimization) have shown an obvious inability to meet their objectives. During the last two decades a plethora of new derivative-free approaches based on various natural (social, biological, etc.) principles have shown better performances when tackling some categories of real-world problems. Sometimes they are grouped together under the general umbrella of evolutionary algorithms (EAs) and include: genetic algorithms (GA) [1]; ant colony optimization (ACO) [2]; particle swarm optimization (PSO) [3]; simulated annealing [4,5]; shuffled complex evolution [6]; harmony search [7]; and memetic algorithms [8].

Unlike most of the classical optimization algorithms, evolutionary algorithms enable the use of any form of quantitative (numerical) assessment of the desired objectives without conditioning the approach to the problem [9, 10]. The flexibility introduced by EAs has allowed the use of virtually any objective function, even when evaluations require, as is the case of many real-

[1] *Corresponding author: Cno. de Vera s/n, 46022,Valencia (Spain) Tel: +34 628 028 804

world problems, running complex mathematical and/or procedural simulations of the systems under analysis. There is an extensive literature of examples within all fields of engineering and science, and more specifically in the water industry, and in particular urban hydraulics (the field of expertise of the authors) regarding design, calibration, energy saving, etc. See, among many other references in the water industry [11, 7, 12-19].

Typically, an EA considers a population of candidate solutions and applies algorithm-specific rules that are iterated through generations in an attempt to improve the fitness of at least one individual (which will hopefully hit the optimum). Despite its virtues, each EA has its own drawbacks and is better adapted to some problems than to others. The heuristics behind a certain evolutionary algorithm endow its elements with specific capabilities for efficiently solving some types of problems, while being clearly inefficient with problems of a different nature. This fact indicates that, firstly, their rules apply better to certain problems than to others; and, secondly, that even if the population of solutions does evolve, the way it evolves is somehow static and does not dynamically adapt to the specific search process. This is one of the reasons why many researchers develop variants of the basic forms of some EAs that adapt better to different problems. Another reason derives from the parameters on which every EA is based. These parameters condition the way an EA works. Fine-tuning those parameters to obtain better results from evolutionary algorithms is, in many cases, part of a hand-made meta-process where specialists, using their experience or recommendations from the literature, start changing parameters, testing algorithm performances, perhaps performing sensitivity analyses, and eventually keeping the best parameter set of values. There are methods that skip these cumbersome processes by using adaptive and self-adaptive parameters; for example, algorithms ASO [20] and TRIBES [21] are based on free-parameter versions of PSO; in [22], a support vector machine was trained to generate PSO parameters while the solution space of a problem was explored. Other self-tuning algorithms have also been developed [23-26]. Despite these attempts, many recent optimization methods (including their variants) still use parameters that are adjusted a priori, frequently undergoing very expensive processes.

However, better adjustments to the parameters of an EA is not the final solution. We suggest that this solution will come from influencing more directly the way a search is performed. In effect, EAs have been frequently accused of using solution search processes that completely ignore the specificities of the problem being solved. As a result, optimization processes have been insensitively applied and ignore the size, complexity, and domain of the problem.

In this paper, we justify our claim that far-reaching benefits will be obtained from more directly influencing the way the search is performed, since algorithms that adapt their behavior to the problems they are intended to solve will have more chances to succeed. This can be achieved by combining EA performances with the introduction of knowledge based on the domain of the problem being solved. Specifically, this paper proposes using Kohonen self-organizing feature maps (SOMs) [27] on sets of solutions evaluated after batches of generations from a single run of an EA in order to extract knowledge intended initially to be used by the following generations. This approach is applied to a very important optimization problem in hydraulics, namely, the optimum design of water distribution networks (WDNs).

The paper is organized as follows. After this introduction we present the problem of the optimum design of a WDN, emphasizing its inherent complexities, which are used to exemplify the approach we propose. A short description of the evolutionary approach used is then presented. We then motivate and describe our approach. Finally, we demonstrate the approach performance on a very well-known benchmark of the WDN design literature, namely the Hanoi problem. The paper concludes with conclusions and references.

## 2 MODEL PROBLEM: OPTIMUM DESIGN OF A WATER DISTRIBUTION NETWORK

A mathematical description of a general simulation-based multi-objective optimization problem, considering uncertainties derived from changing environmental and operating conditions (see [28] for an overview of the state of the art in the field of robust optimization) may take the following form:

$$\text{Optimize } F(x, \varepsilon) = (f_1(x, \varepsilon), \ldots, f_m(x, \varepsilon))^t$$

subject to

$$g_i(x,\varepsilon) > 0,\ i = 1,2, \ldots, k$$
$$h_j(x,\varepsilon) = 0,\ j = 1,2, \ldots, l$$

where $f_i$, $i = 1, \ldots, m$, are the objectives, and $g_j$ and $h_j$ are $k$ and $l$ inequality and equality constraints, respectively, which depend on the vector $x$ of decision variables and ε, the uncertainty state vector. Decision vectors belong to the decision or search space $S \subseteq \mathbb{R}^d$ of decision variables, $x_1, \ldots, x_d$, and the uncertainty state vector belongs to certain state uncertainty set, $U_s$. The vector function, $F(x, \varepsilon)$ takes its values on the objective space $F(S,U_s) \subset \mathbb{R}^m$, its $m$ components representing the various objective functions considered. The symbol $^t$ is the matrix transposition operator. Both the optimization criteria $f_k$, and the constraint functions $g_i$ and $h_j$ may require multi-level computer simulations.

To gain specificity we now present the specific problem addressed in the case study of this paper.

### 2.1 Optimum design of a water distribution network

Water supply system design is a wide and open problem in hydraulic engineering that may involve the addition of new elements in a system: the rehabilitation or replacement of existing elements; decision-making on operation, reliability, and protection of the system; among other actions. Designs are necessary to carry out new configurations, or to enlarge or improve existing systems to meet new conditions [29-31, 9].

For the sake of simplicity we limit ourselves here to consider just one objective, namely, the cost of the network components, and two constraints, namely a form of satisfying water demand quality and the compulsory adherence to hydraulic equations. We deliberately leave aside other important aspects in WDN design such as the various aspects related to the resilience of the system during stressed conditions, aspects that the authors have dealt with before [32,33].

#### *2.1.1 Cost of components*

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

$$C_{\text{WSN}} = \sum_{k=1}^{N_{\text{wc}}} \left[ P_{\text{wc}}^k \left( a_{\text{pipe}} C_{\text{pipe}} + a_{\text{pump}} C_{\text{pump}} + a_{\text{valv}} C_{\text{valv}} + a_{\text{tank}} C_{\text{tank}} + C_{\text{Oper}} \right) \right] \quad (1)$$

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

In general, $C_{\mathrm{WSN}}$ is a non-linear, partially stochastic function that is dependent on continuous, discrete, and binary variables.

Although a wide range of decision variables may be considered, for the sake of simplicity the basic variables of the problem that we consider here are the diameters of the new pipes, together with a number of options in terms of rehabilitation of pipes (with several available alternatives, such as replacement, simply duplication, or no action, with their associated costs),

$$C(D) = \sum_{i=1}^{L} c(D_i) l_i \,, \tag{2}$$

where $L$, as said, is the number of pipes, which includes the cost of new pipes and the cost of rehabilitating existing pipes. This function, besides contributing most of the total cost, exhibits the characteristics we are interested in underlining.

It is a function of $D$, the vector of diameters of the $L$ pipes (both new and rehabilitated) in the network; $c(\cdot)$ represents the cost per meter, which depends on the diameter, $D_i$, of pipe $i$, and $l_i$ is the length of pipe $i$. Note that $D_i$ is chosen from a discrete set of commercially available diameters, and $c(\cdot)$ is a non-linear (discrete) function of diameter. There are various rehabilitation options (no rehabilitation, relining, duplication, and replacement being the most common). Rehabilitation costs are also non-linear.

Costs corresponding to rehabilitation options and to other elements (tanks, pumps, valves and operation) are also typically non-linear (see [34], for example).

### *2.1.2 Satisfaction of water demand quality*

WDN design is typically performed subject to several performance constraints in order to achieve an adequate service level. The most used constraint requires a certain minimum pressure level at each node of the system. Other constraints may include minimum or maximum pipe flow velocities, and minimum concentrations of chlorine, for example.

There are various ways of expressing a lack of compliance with conditions of pressure, velocity, disinfectant, etc. A general weighted expression for penalties takes the form

$$P = \sum_{k=1}^{n} \alpha_k P_k, \tag{3}$$

where each $P_k$ accounts for the global lack of compliance for any of the $n$ considered problem magnitudes with an associated constraint:

$$P_k = \sum_{j=1}^{K} (u_{\text{ref}}^{(k)} - u_j^{(k)}) H(u_{\text{ref}}^{(k)} - u_j^{(k)}) . \quad (4)$$

Here, $u^{(k)}$ is the vector of values of a certain problem magnitude $u$ (node pressure, pipe flow velocity, sensor disinfectant concentration, etc.) related to the demand nodes, the lines or any other points (mainly sensors) used to sample specific variables. In the case of pressure, $K$ equals $N$, the number of demand nodes; in the case of flow velocity $K = L$, the number of pipes, etc. For variables with values greater than some reference value, $u_{\text{ref}}^{(k)}$, the associated individual terms vanish, and the Heaviside step function $H$ is used in this explicit expression for this purpose.

Thus, (3) represents a weighted sum of lack of compliance for various variables associated with the WDN. Parameters $\alpha_k$ help normalize the importance of the different scales between the various variables, and this enables a more meaningful aggregation of different types of constraint violation and can also be used to balance the importance among them. Extensions of (4) may be provided to consider maximum bounds for the variables. Expressions such as (3) enable the consideration of any objective, the most common being minimum nodal pressure, minimum pipe velocity, and minimum level of chlorine at specific points if water quality is included in the optimization. Expression (3) is a function of the selected pipe diameters through the hydraulic model presented in the next subsection.

For many years nodal and pipe constraints were considered as hard constraints in the sense that they should be strictly satisfied. Nevertheless, the possibility of violating by a small degree some of these constraints opens the door to various strategies for adopting sub-optimal designs or soft solutions that may be more acceptable from other (global, strategic, or political, for example) perspectives. This is a new source of uncertainty, in this case, making the borders between feasible and non-feasible areas fuzzy. Here we only consider hard constraints.

### *2.1.3 Adherence to hydraulic equations*

Several formulations are available (see, for example, [35]) to mathematically model pressurized water systems, especially for the large systems found in even medium-sized cities, since it involves solving the continuity and energy equations associated with the system. These equations constitute a coupled system of many simultaneous nonlinear equations. One formulation considers the $N - 1$ continuity equations, which are linear, plus the $L$ energy equations, typically non-linear:

$$\begin{aligned} \sum_{j \in N_i} q_{ij} &= Q_i, & i &= 1,..,N-1 \\ H_{k1} - H_{k2} &= R_k q_k |q_k| & k &= 1,...,L \end{aligned} \quad . \quad (5)$$

As already stated, $N$ is the number of demand junctions, and $L$ is the number of lines in the system. In addition, $N_i$ is the number of nodes directly connected to node $i$; $Q_i$ is the demand associated to node $i$; $k1$ and $k2$ represents the end nodes of line $k$, which carries an unknown flowrate $q_k$ and is characterized by its resistance $R_k$, which depends on $q_k$ through the Reynolds number (the non-linearity of the energy equations arises not only from the quadratic term, but also from the function $R_k$). $H_{k1}$ and $H_{k2}$ piezometric heads at nodes $k1$ and $k2$ are unknown for consumption nodes and are given for fixed head nodes. The complete set of equations may be written, by using block matrix notation such as

$$\begin{pmatrix} A_{11}(q) & A_{12} \\ A_{12}^{\text{t}} & 0 \end{pmatrix} \begin{pmatrix} q \\ H \end{pmatrix} = \begin{pmatrix} -A_{10} H_f \\ Q \end{pmatrix} \quad (6)$$

where $A_{12}$ is the connectivity matrix describing the way demand nodes are connected through the lines. Its size is $L \times N_p$, $N_p$ being the number of demand nodes; $q$ is the vector of the flowrates through the lines; $H$ the vector of unknown heads at demand nodes; $A_{10}$ describes the way fixed head nodes are connected through the lines and is an $L \times N_f$ matrix, $N_f$ being the number of fixed head nodes with known heads in the components of $H_f$, and $Q$ is the $N_p$-dimensional vector of demands. Finally, $A_{11}(q)$ is an $L \times L$ diagonal matrix, with elements $a_{ii} = R_i q_i + B_i + A_i / q_i$, with $R_i = R_i(q_i)$ being the line resistance, and $A_i$ , $B_i$ coefficients characterizing a pump potential in the line.

System (6) is a non-linear problem, whose solution is the state vector $x = (q^t, H^t)^t$ (flowrates through the lines and heads at the demand nodes) of the system.

Since most water systems involve a huge number of equations and unknowns, system (6) is usually solved using some gradient-like technique. Various tools to analyze water networks using gradient-like techniques have been developed. Among them, EPANET2 [36] is used in a generalized way by hydraulic engineers around the world. It is clear that this complex set of constraints constitutes a process that must be built-in within the optimization problem, and implemented through a sophisticated tool that develops the hydraulic simulation.

## 3 SOME WEAKNESSES OF EVOLUTIONARY APPROACHES

Problems like the one described in the previous section may be categorized among the so-called non-standard optimization problems [37], which are being increasingly used in engineering optimization and, in general, in real-world optimization problems. One of the main characteristics of this category of problems is that relevant optimization criteria, as well as prescribed constraints, can only be evaluated by virtue of computer simulations (simulation-based optimization). Although it is universally accepted that no general strategy to solve different types of problems in an equally efficient manner exists, or even can be expected to be designed (see the *No Free Lunch Theorem* in [38]), EAs offer a number of advantages [9] that make them suitable for tackling general problems. The main advantages of various EAs have been gathered by the authors into a software platform called ASO[2], for agent swarm optimization [19]. To put it in a nutshell: multi-agent systems, and the necessary adaptation to multi-objective performance (including human interaction) have been built in ASO; in addition, ASO integrates various algorithms in runtime. The mixture of different algorithms and the incorporation of new agents in runtime within ASO are possible because ASO makes use of parallel and distributed computing to enable the incorporation of new agents, as well as the asynchronous behavior of agents. Moreover, self-adaptive parameter control is also implemented in ASO, meaning that the parameters of various metaheuristics are incorporated into the representation of the solution and, thus, the parameter values evolve together with the population solutions. As a result, the combination of self-adaptive algorithms is better able to find the best solution to the problem in hand. This approach is used as a high level concept for flexibly and consistently hybridizing various strategies of optimization– so that better solutions can be found by cooperation. We call ASO an evolutionary hybridized platform of self-adaptive algorithms (EHPSA).

Nevertheless, a better adjustment of the parameters of an EA is not the ultimate solution in itself, and the best solution will come from influencing more directly the way searches are performed. In effect, each EA uses a solution search process that completely ignores the specificities of the problem being solved. As a result, optimization processes are applied insensitively – no matter the size, complexity, and domain of the problem. Although a powerful

[2] Agent Swarm Optimization (ASO) was developed in C# and works for the Microsoft .Net Framework 4.0 or superior.

hybridization of EAs, as implemented in ASO, injects increased diversity into the solution possibilities, it is not enough.

To solve a simulation-based optimization problem efficiently, interaction between the optimization expert, who must be endowed with proven experience and an extensive knowledge of the problem, and the implemented optimization mechanism is critical.

EAs have not generally taken advantage of this characteristic and, as a result, have been bound to analyze a larger solution space than necessary. Including expert know-how may reduce the search space and, as a result the solution is not only more efficient, but also closer to reality. Efficiency stems from the fact that just checking a number of usually simple rules embodying expert knowledge avoids many expensive calculations or simulations (hydraulic simulations in the application we present in this paper). Moreover, the fact that the rules have strong problem-dependent meaning definitely takes the solution closer to reality. For example, in the application considered in this paper one rule just states that downstream pipe diameters must not be larger than upstream pipe diameters – while this may be reasonable from an engineering point of view, it is not necessarily reasonable for the random assignment approach typically used by EAs.

To illustrate these ideas we succinctly show, by using a simple example, namely the benchmark network shown in Figure 1 [39], the reduction of the search space achieved when applying this rule. We omit here most of the specific data for this problem, which may be found elsewhere, since it is not relevant to our current purpose. Let us just mention that the problem specifications present a set of 14 candidate diameters for any of the 8 pipes (numbered P1 to P8) of the network. As a consequence, an algorithm facing the analysis of all the possibilities should be confronted with a number of 8^14 = 1 475 789 056 ≈ $1.476\times10^9$ candidate solutions. Meanwhile, the solution space when the rule is applied embraces just 1.46 % of this number, that is to say, $2.15\times10^7$ candidate solutions. Given the simplicity of the example, it is clear from an engineering point of view that the process will be able to find the best solutions.

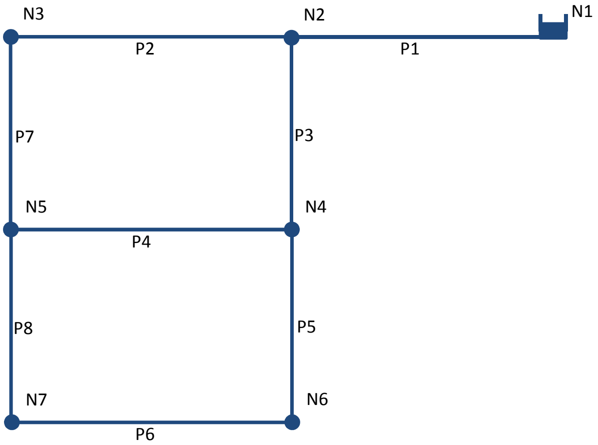

Figure 1. Alperovits and Shamir’s benchmark network

Other rules may be considered to further facilitate the process of finding solutions, (see [19] and the references therein). These are examples of rules that, obviously, reflect the nature of the problem in hand.

The attractiveness of this approach is endangered by several disadvantages. For example, it is not trivial to convey expert knowledge to the algorithm; and no changes can be enforced without changing the existing source code or adding more code to the software supporting the algorithms. In addition, injecting problem-related knowledge to be combined with evolutionary techniques requires the active participation of specialists from the problem domain. For a specific problem, it may be hard to suitably devise knowledge to be included in order to improve the solution search process without a good understanding of the problem specificities.

Moreover, even having a deep understanding of the problem domain it may be hard to shape specific knowledge to work efficiently in combination with evolutionary techniques. The proposal of this paper is to include (automatic) data mining techniques as a step for dynamically generating knowledge that can be used to improve the efficiency of solution search processes.

## 4 THE RATIONALE BEHIND THE PROPOSAL

The rationale behind our proposal is the following. During the execution of EAs, typically the number of solutions evaluated represents quite a small percentage of the total solution space corresponding to the problem being solved. Nevertheless, the number of solutions evaluated is still considerable, and most evolutionary techniques use just a small number of them at a time. Many of the solutions evaluated during the search process are "forgotten" after one generation, and the 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 simply glimpsed and rapidly disregarded because they were dominated by better solutions during an ephemeral moment in the evolution process. Our claim is that by exploring 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.

The proposal of this paper is to use DM techniques as a step for dynamically and automatically generating knowledge that can be used to improve the efficiency of solution search processes.

Thus, in this paper we explore the idea of combining 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.

The description of the process, summarized in Figure 2, is the following.

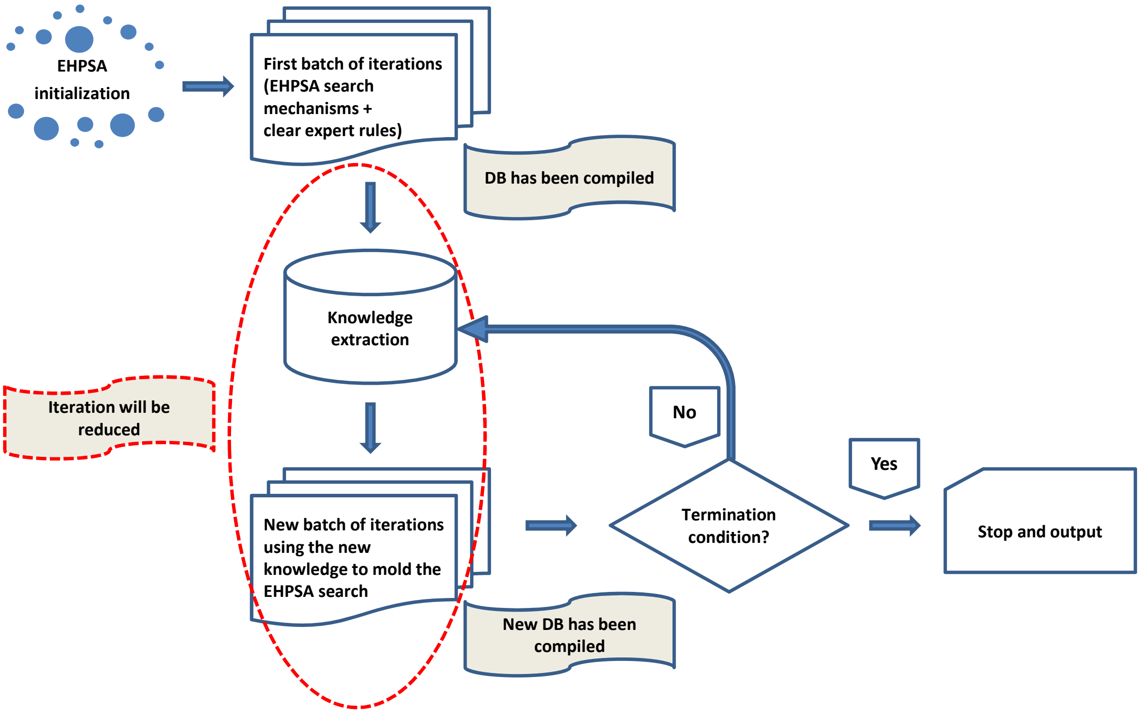
EHPSA initialization
First batch of iterations (EHPSA search mechanisms + clear expert rules)
DB has been compiled
Knowledge extraction
Iteration will be reduced
No
Yes
New batch of iterations using the new knowledge to mold the EHPSA search
Termination condition?
Stop and output
New DB has been compiled

Figure 2. Putting knowledge extraction and EAs to work together

The typical operation of an EHPSA aided by the knowledge extraction we propose is the following. When initializing the EHPSA only random solutions are available. As a result, there is no possibility of knowledge extraction and the EHPSA, using its own search mechanisms (and perhaps some clear expert rules), must work during several iterations to produce and collect new solution candidates. After this number of iterations, a database (DB) must have been created. The EHPSA will then stop the search, and the knowledge extraction algorithm will be launched to work on the DB. Hopefully, a number of pieces of knowledge will be obtained that will be used to guide the EHPSA during a new batch of iterations. Then a new DB of candidate solutions will be available. Again the knowledge extraction algorithm will probably produce new knowledge that, in its turn, will be used during the subsequent process of iteration. Assuming that injecting this knowledge will accelerate the convergence of the EHPSA, and taking into account that the EHPSA is controlled by a certain termination condition, it is expected that only a limited number of knowledge extraction processes will be eventually performed. When to stop the EHPSA and launch a knowledge extraction process is a matter that will need further insight and the target should be automatic execution.

Various DM techniques that handle large volumes of data, as well as scanning available information and tracking down understandable and useful knowledge have been devised. In this paper we explore Kohonen SOMs [27].

The map of Kohonen is known as an important paradigm of an unsupervised neural network for analyzing data [27]. A Kohonen map is a two dimensional array of neurons that are fully connected with the input vector and organized in a square or hexagon. Hexagonal arrangement is advised because at the end of the learning process it provides better visualization of the structure of the input space [40].

The topology preserving property is obtained by a learning rule involving the winning neuron and its neighbors in the update process. Therefore, nearby neurons learn to activate when presented with similar patterns. The learning algorithm follows the pattern of competitive models, but the update rule produces an output layer in which the topology of the input patterns is preserved. This means that if two patterns are close in the input space (in the sense of some similarity measure, such as measures used in winner-take-all strategies) their corresponding active neurons are also topologically close in the output layer. A network that performs this function is called a map of characteristics. These maps not only group the input patterns in clusters but also visually describe the relationship between the clusters of the input space.

During training, the network allocates a position to the neurons on the map based on the effect of the dominant feature of the input pattern. For this reason, Kohonen maps are called self-organizing maps (SOMs).

If the input space is highly dimensional, as is the case of real-world optimization objectives, Kohonen maps can be interpreted as projectors of neurons onto a two-dimensional array that takes into account the probability density of the data and preserves the topology of the original input pattern. Preserving the topology of the original input pattern is a great advantage for visualizing results.

In this paper we use the implementation of SOMs in R, through the xyf function [41]. The R's *xyf* function obtains SOMs from data consisting of a set of independent variables and a dependent variable. It is also said that it is a type of supervised SOM. For the set of independent variables, a network is first trained in an unsupervised manner and, then, on the same network, the values of the dependent variable are projected. This allows better identification of characteristic patterns. In this study, the independent variables correspond to the diameters of the pipes and other decision variables, and the dependent variable is the cost of each solution.

## 5 APPLICATION TO THE HANOI NETWORK

The Hanoi water distribution problem is a very well-known benchmarking problem in the WDN design field and has been often approached in the literature, see [42, 34, 43-46] among many others. To gauge the effectiveness of our proposal, we will consider this same problem. Figure 3 contains a representation of the network. We now describe the characteristics of this network.

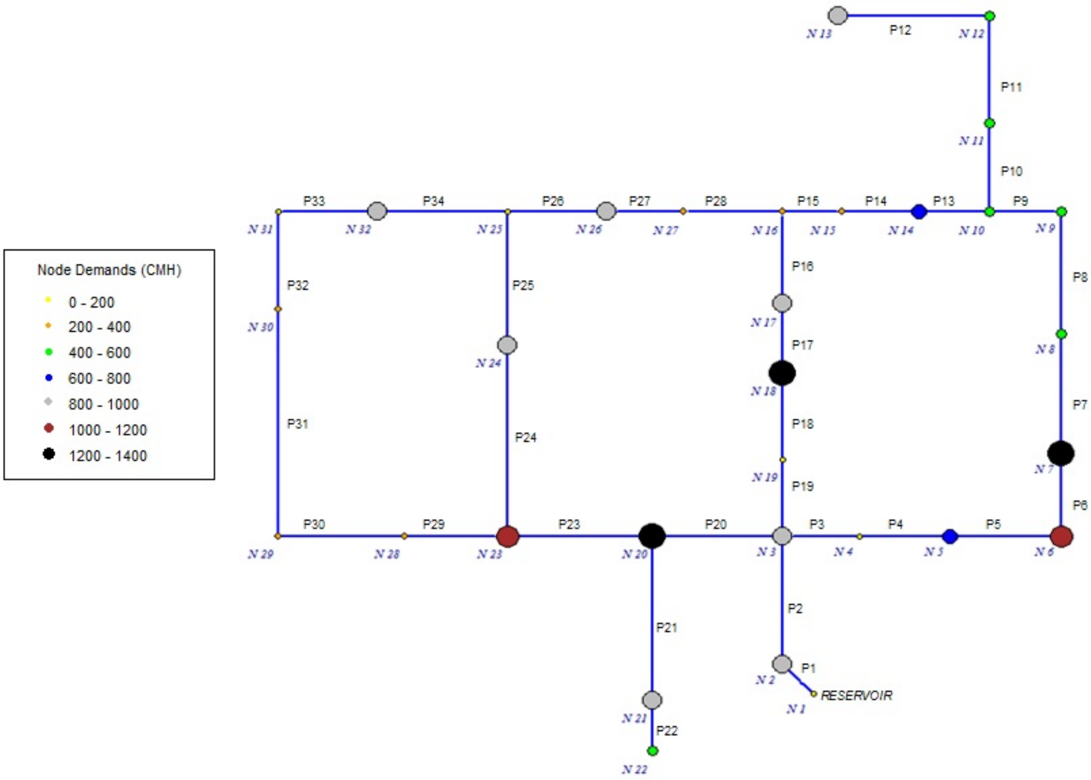

Figure 3. Hanoi network

### 5.1 Problem description

The network consists of one fixed head source (reservoir), 34 pipes numbered P1 to P34, and 31 demand nodes, numbered *N2* to *N31*, subject to one load condition given by the demand associated with the nodes. Tables 1 and 2 include, respectively, the candidate diameters, and the pipe and node data (node thickness and color in Figure 3 represents the associated node demand). Furthermore, the network has three grids and two ramified branches. One has to find the diameters for the 34 pipes such that the total cost of this network is minimal and the pressure at each consumption node is at least 30 m. The complete setting can be found in [47].

Table 1. Commercial diameter for the design of Hanoi network

| **Diameter (mm)** | **Diameter (inches)** | **Unit Cost ($/m)** |
|---|---|---|
| 304.8 | 12 | 45.726 |
| 406.4 | 16 | 70.4 |
| 508 | 20 | 98.378 |
| 609.6 | 24 | 129.333 |
| 762 | 30 | 180.748 |
| 1016 | 40 | 278.28 |

Table 2. Pipe and node data for the Hanoi network

| Pipe | Length (*m*) | Pipe | Length (*m*) | Node | Demand ($m^3/h$) | Node | Demand ($m^3/h$) |
|---|---|---|---|---|---|---|---|
| 1 | 100 | 18 | 800 | 1 | -19940 | 17 | 865 |
| 2 | 1350 | 19 | 400 | 2 | 890 | 18 | 1345 |
| 3 | 900 | 20 | 2200 | 3 | 850 | 19 | 60 |
| 4 | 1150 | 21 | 1500 | 4 | 130 | 20 | 1275 |
| 5 | 1450 | 22 | 500 | 5 | 725 | 21 | 930 |
| 6 | 450 | 23 | 2650 | 6 | 1005 | 22 | 485 |
| 7 | 850 | 24 | 1230 | 7 | 1350 | 23 | 1045 |
| 8 | 850 | 25 | 1300 | 8 | 550 | 24 | 820 |
| 9 | 800 | 26 | 850 | 9 | 525 | 25 | 170 |
| 10 | 950 | 27 | 300 | 10 | 525 | 26 | 900 |
| 11 | 1200 | 28 | 750 | 11 | 500 | 27 | 370 |
| 12 | 3500 | 29 | 1500 | 12 | 560 | 28 | 290 |
| 13 | 800 | 30 | 2000 | 13 | 940 | 29 | 360 |
| 14 | 500 | 31 | 1600 | 14 | 615 | 30 | 360 |
| 15 | 550 | 32 | 150 | 15 | 280 | 31 | 105 |
| 16 | 2730 | 33 | 860 | 16 | 310 | 32 | 805 |
| 17 | 1750 | 34 | 950 |  |  |  |  |

According to the objective and the constraints adopted in this study, which coincide with the benchmark problem requirements, the optimization problem may be stated as

$$\text{Minimize } F(D)=\sum_{i=1}^{34} c(D_i)l_i+\sum_{j=2}^{32}\alpha H(p_{\min}-p_j)\cdot(p_{\min}-p_j)\,,$$

s.t. the hydraulic conditions given by (6).

The first summation corresponds to the cost of the pipes, and the second to the penalty for lack of pressure condition satisfaction. The factor $\alpha$ that multiplies with the pressure difference $\Delta p_i = p_{\min} - p_i$ represents a fixed value, which becomes effective (by using the Heaviside function $H$) whenever the minimal pressure requirement is not met. Note that in this model the individual penalties grow linearly with $\Delta p_i$. The variables in the problem are the diameters pertaining to the new pipes of the network or those of the rehabilitated pipelines. One therefore deals with determining the values which minimize the total cost of the pipelines – while complying with the minimal pressure requirements of the network.

Furthermore, this simple variant for the design of a water supply system forms an NP-complete problem; the solution space is so large that in practice analysis of all the possibilities is not feasible due to the huge amount of computational time required.

### 5.2 Application of the proposed synergy between EHPSA and SOMs

To apply the process described in Figure 2, one run of ASO with a population of 100 individuals was launched for 150 generations. This generated a database with 15000 registers. Thirty-five columns constituted 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, following engineering criteria derived from knowledge on the specific field of the case study, we discretized *a priori* the objective field into four categories to obtain a qualitative cost attribute. The first category included the current excellent solutions, corresponding to registers with objectives between 0 and 3% over the cheapest solution in the database; then good solutions included those registers with objectives between 3% and 5% over the cheapest solution; poor solutions were those with a cost between 5% and 15% over the cheapest solution; and, finally, the remaining solutions constituted the class of bad solutions.

A network with a hexagonal topology of 5×8 neurons (Figure 4) was then trained and based on this DB, and using the *xyf* function, the pipe diameters were independent variables and the qualitative cost was the dependent variable.

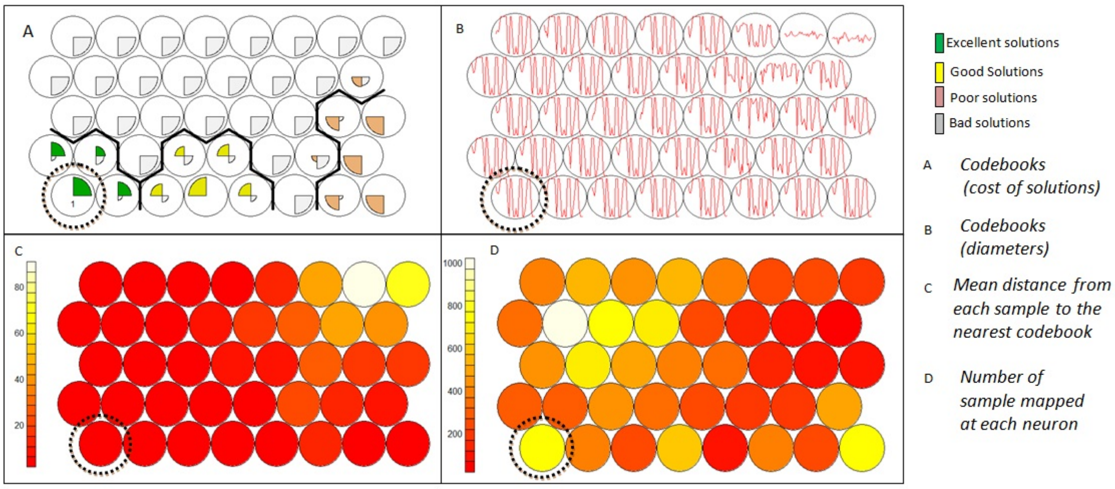

Figure 4. SOM after 150 iterations. In inset A the solutions in the database are clustered; graphs in inset B provide a rough visual representation of the neuron codebooks; inset C shows, for each neuron, the average distance between the samples mapped into the neuron and the codebook of the neuron; finally, the number of samples mapped in each neuron are represented in inset D.

Figure 4 shows the trained network. The red curves in the neurons of inset B are representations of the codebooks of the neurons. High points in these curves represent big diameters and low points small diameters. In this SOM, current economic solutions are found in neurons on the bottom left part of the SOM, as shown on inset A, which gathers almost all the current excellent and good solutions in the DB. Note how the codebooks of these solutions are comprised of diameters lower than those of other solutions. This is clearly evident in inset B when comparing, for example, neurons on the bottom left (such as the one surrounded by a circle) with neurons on the upper right of the map, corresponding to more expensive solutions.

The codebook of the neuron that gathers most of the current good solutions is now obtained. It corresponds to the bottom leftmost neuron (marked with a circle in all the insets): observe, looking at inset D, the large number of excellent solutions concentrated in this neuron. Also observe the quality of this clustering. In effect, inset C provides this interesting qualitative information. This inset represents, for each neuron, the mean distance between each sample mapped into the neuron and the neuron codebook: the lower the associated value, the better the identification of the mapped samples with the codebook. We can observe how the bottom leftmost neuron, containing most of the excellent solutions, exhibits a low value for that mean distance – meaning that the corresponding codebook suitably represents those samples. The codebook of this neuron is given by the values in the shaded squares in Figure 5.

The codebook corresponding to this neuron is then used to extract straightforward rules regarding the pipes analyzed. Specifically, these pipes are assigned the values given by the codebook of the winning neuron. In any case, a small amount of randomness must be considered instead of taking the codebook values as hard rules to apply.

After implementing these rules, the EHPSA continues with the iteration. In the specific run we are describing convergence for the EHPSA+SOM occurred at iteration 223. The optimum was then obtained. It corresponds to a network with a cost of 6.545325 million dollars and pipe diameters as noted in Figure 5 (non-shaded squares). It is worth noting that a new SOM was obtained using the new database collected during this last part of the iteration process. This map is represented in Figure 6. The codebook for the neuron gathering most of the good solutions completely coincides with the diameters associated with the best solution.

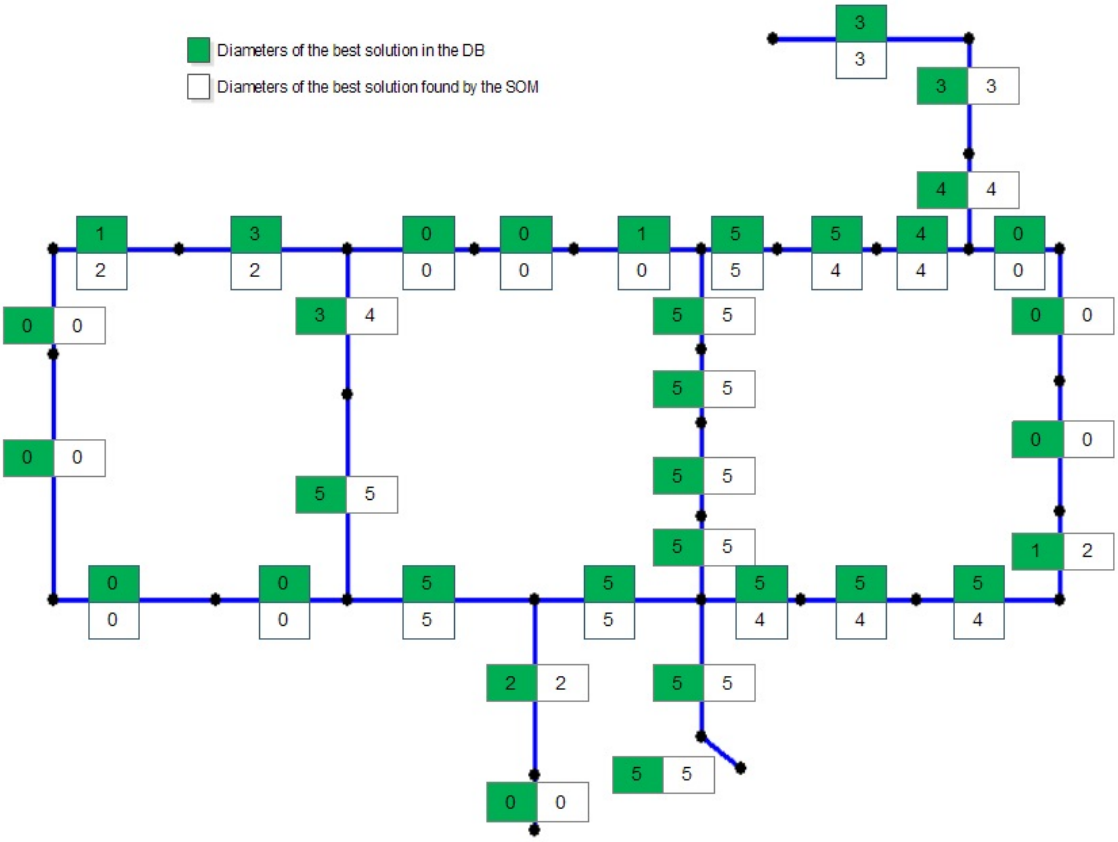

Figure 5. Hanoi network with codebooks for the first and second SOMs

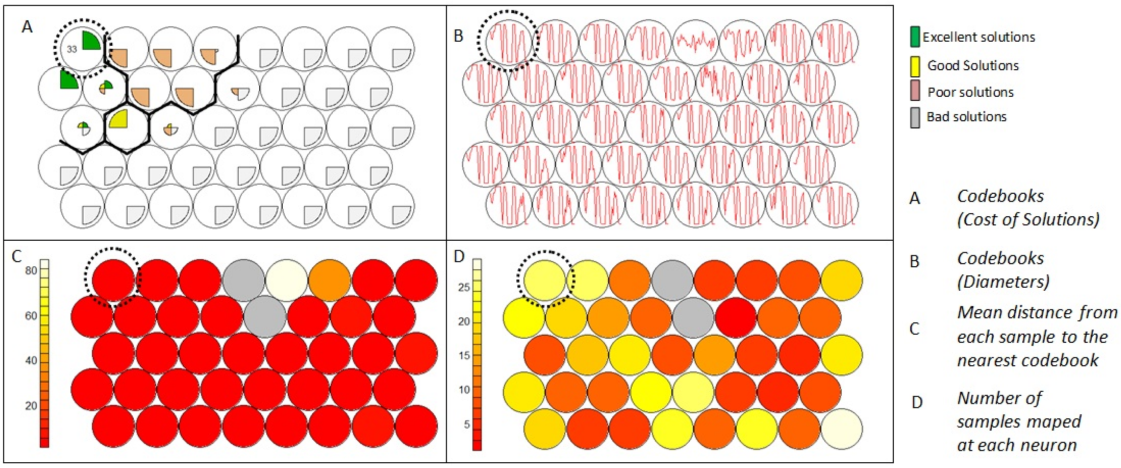

Figure 6. SOM portraying the last iteration steps (see caption of Figure 4 for more detailed information).

A second aspect is worth mentioning: the ease with which a SOM can be generated, even for a relatively large DB. The SOMs presented in Figures 4 and 6 converged in about 15 seconds with an Intel(R) core computer with 2.70 GB of usable RAM. As a consequence, rule extraction through SOMs is much cheaper than running the EPANET simulations because of the avoided iterations. We can here conclude that rules may be easily generated that have not been evaluated using expert knowledge. Their evaluation is only based on data and structure of patterns found. We claim that, in addition to expert knowledge, these rules may be of great interest in helping EAs restrict the search to more promising areas and so reducing the size of the search space.

A third interesting point is the following. When comparing the SOMs of Figures 4 and 6 the different distribution of the various types of solutions is easily observable. In the SOM in Figure 4, just after iteration has started, the diversity of the current solutions is low (which can be seen by the relatively large number of neurons that have captured (current) excellent and good solutions). In contrast, the SOM of Figure 6 shows the final portrait of the evolutionary history – diversity has increased, and as a result, excellent and good solutions are relatively scarce and concentrated in few neurons.

To provide an extra support to the approach herein presented, we now describe a comparison performed with and without the use of SOMs in terms of the number of iterations. In fact, in the same run a replica of the swarm at iteration 150 was launched in a different computational thread to continue performing EHPSA iteration without adding the rules obtained from the SOM after the stop at iteration 150. Curves for evolution of costs with iteration are shown in Figure 7. We can observe how after injecting the knowledge obtained from the SOM built after iteration 150, the algorithm with rules rapidly reduces the cost, thus converging faster, while the conventional algorithm takes longer to converge.

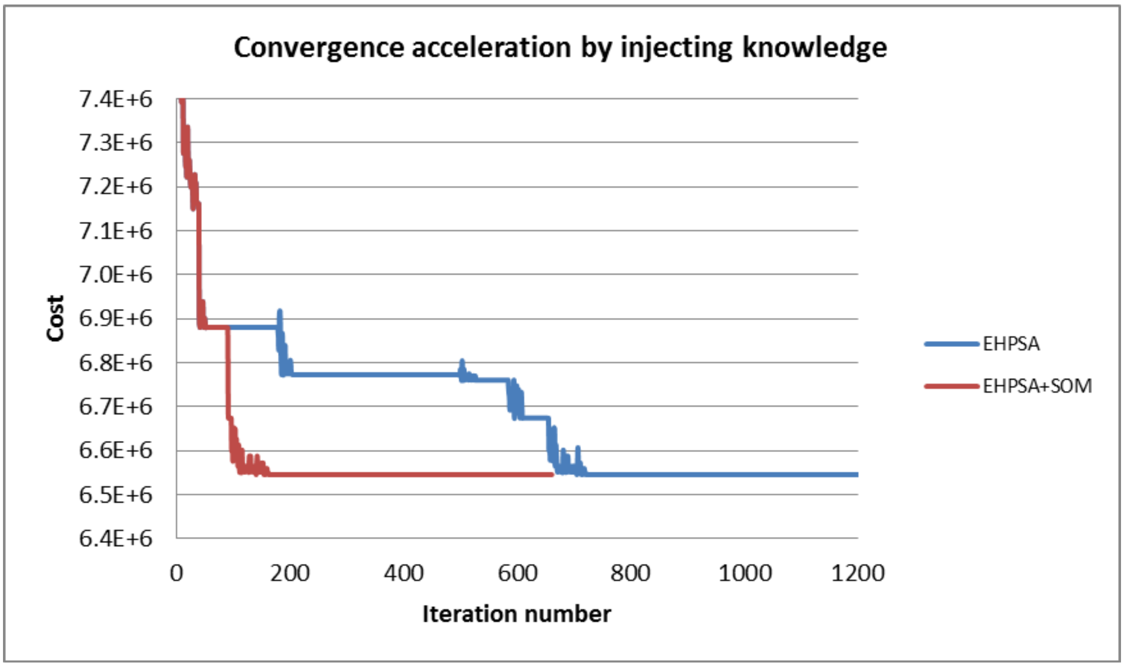

Figure 7. Evolution histories with and without knowledge injection for the Hanoi problem

## 6 CONCLUSIONS

This paper explores the use of data mining techniques (obtained from a suitable database of solutions generated by an EA) to guide the search towards more problem-relevant solutions that improve search efficiency. We have specifically applied SOMs to a database obtained when trying to obtain the best design for a water distribution network used in the literature as a benchmark, namely, the Hanoi network.

The results are very promising and show a very attractive approach to improving evolutionary search in real-world optimization problems. To further and efficiently develop 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. Incorporating any additional objective or reliability assessment to deal with real cases does not require major effort from the implementation point of view [14]. The algorithm is prepared to discover the insides of problems independently of the objective function under analysis. One of the largest advantages of this evolutionary approach is that it can work practically with any objective function. Nevertheless, depending on the case, the injection of problem-dependent knowledge directly from specialists in the field could help the algorithm significantly reduce its computational effort in finding good solutions.

Secondly, updating techniques should be developed for maintenance of the discovered knowledge during the whole optimization process, and so avoid completely re-mining the data on the whole updated database every time new and 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 should not invalidate existing knowledge.

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

## 7 REFERENCES

[1] D.E. Goldberg, Genetic algorithms in search, optimization and machine learning, Addison-Wesley, Reading, Ma, 1989.

[2] M. Dorigo, V. Maniezzo, A. Colorni, The ant system: optimization by a colony of cooperating ants, IEEE Trans. Syst. Man Cybern.—PartB 26(1) (1996) 1–13.

[3] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, Piscataway, NJ, 1995, pp. 1942-1948.

[4] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by Simulated Annealing. Sci. 220 (4598) (1983) 671–680.

[5] V. Černý, Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, J. Optim. Theory Appl. 45 (1985) 41–51.

[6] Q. Duan, V.K. Gupta, S. Sorooshian, A shuffled complex evolution approach for effective and efficient global optimization, J. Optim. Theory Appl. 76 (1993) 501-521.

[7] Z.W. Geem, Optimal cost design of water distribution networks using harmony search. Eng. Optim. 38(3) (2006) 259-280.

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

[9] I. Montalvo, Diseño óptimo de sistemas de distribución de agua mediante Agent Swarm Optimization, Ph.D. thesis, Universitat Politècnica de València, Valencia, Spain, 2011.

[10] J. Izquierdo, I. Montalvo, R. Pérez-García, A. Matías, On the Complexities of the Design of Water Distribution Networks, Math. Probl. Eng., Vol. 2012 (2012) 1-25.

[11] S.Y. Liong, M. Atiquzzama, Optimal design of water distribution network using shuffled complex evolution, J. Inst. Eng. Singap. 144(1) (2004) 93-107.

[12] J. Izquierdo, I. Montalvo, R. Pérez, M. Tavera, Optimization in water systems: a PSO approach, in: Proc. Business and Industry Symposium (BIS), Ottawa, Canadá, 2008.

[13] X. Jin, J. Zhang, J.L. Gao, W.Y. Wu, Multi-objective optimization of water supply network rehabilitation with non-dominated sorting Genetic Algorithm-II, J. Zhejiang Univ. Sci. A 9(3) 2008 391-400.

[14] I. Montalvo, J. Izquierdo, S. Schwarze, R. Pérez-García, R., Multi-objective particle swarm optimization applied to water distribution systems design: An approach with human interaction, Math. Comput. Model. 52 (2010) 1219-1227.

[15] H. Shean, E. McBean, Hydraulic calibration for a small water distribution network, in: Proc. 12th Water Distribution Systems Analysis Symp, Tucson, Arizona: K. Lansey, C. Choi, A. Ostfeld, and I. Pepper, 2010.

[16] W. Bei, G.C. Dandy, Retraining of metamodels for the optimization of water distribution systems, in: Proc. Water Distribution System Analysis Conference, Adelaide, Australia, 2012, pp. 36-47.

[17] L. Berardi, D. Laucelli, O. Giustolisi, A decision support tool for operational optimization in WDNETXL, in: Proc. Water Distribution System Analysis Conference, Adelaide, Australia, 2012, pp. 48-65.

[18] Z.Y. Wu, M. Behandish, Real-time pump scheduling using genetic algorithm and artificial neural network based on graphics processing unit, in: Proc. Water Distribution System Analysis Conference, Adelaide, Australia, 2012, pp. 1088-1099.

[19] I. Montalvo, J. Izquierdo, M. Herrera, R. Pérez-García, Water supply system computer-aided design by Agent Swarm Optimization, Comput. Aided Civ. Infrastruct. Eng., 29(6) (2014) 433-448.

[20] I. Montalvo, J. Izquierdo, R. Pérez-García, M. Herrera, Improved performace of PSO with self-adaptive parameters for computing the optimal design of water supply systems, Eng. Appl. Artif. Intell. 23(5) (2010) 727-735.

[21] M. Clerc, Particle Swarm Optimization, ISTE Ltd., 2006.

[22] S. Lessmann, M. Caserta, I. Montalvo, Tuning metaheuristics: A data mining based approach for particle swarm optimization, Expert Syst. Appl. Int. J., 38(10) (2011) 12826-12838.

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

[24] J.A. Vrugt, B.A. Robinson, J.M. Hyman, Self-adaptive multimethod search for global optimization in real-parameter spaces, IEEE Trans. Evol. Comput., 13(2) (2009) 243-259.

[25] D. Hadka, P. Reed, Borg: An auto-adaptive many-objective evolutionary computing framework, Evol. Comput. 21(2) (2013) 231-259.

[26] K. McClymont, E.C. Keedwell, D. Savic, M. Randall-Smith, A General Multi-objective Hyper-Heuristic for Water Distribution Network Design with Discolouration Risk, J. Hydroinf. 15(3) (2013) 700-716.

[27] T. Kohonen, Self-Organizing Maps, Springer-Verlag, Berlin, Heidelberg, 2001.

[28] D. Bertsimas, O. Nohadani, K.M. Teo, Robust optimization for unconstrained simulation-based problems, Oper. Res. 58 (1) (2010) 161–178.

[29] I.C. Goulter, A.V. Coals, Quantitative approaches to reliability assessment in pipe networks, J. Transp. Eng. 112(3) (1986) 287-301.

[30] I.C. Goulter, F. Bouchart, Reliability-Constrained Pipe Network Model, J. Hydr. Eng. ASCE 116(2) (1990) 211-229.

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

[32] J.B. Martínez-Rodríguez, I. Montalvo, J. Izquierdo, R. Pérez-García, Reliability and Tolerance Comparison in Water Supply Networks, Water Resour. Manag. 25 (2011) 1437–1448.

[33] I. Montalvo, J.B. Martínez-Rodríguez, J. Izquierdo, R. Pérez-García, Water Distribution System Design using Agent Swarm Optimization, in: Proc., 12th Water Distribution Systems Analysis Symp, Tucson, Arizona: K. Lansey, C. Choi, A. Ostfeld, and I. Pepper, 2010.

[34] A.S. Matías, Diseño de redes de distribución de agua contemplando la fiabilidad, mediante algoritmos genéticos, Ph.D. thesis, Universidad Politécnica de Valencia, Valencia, Spain, 2003.

[35] J. Izquierdo, R. Pérez, P.L. Iglesias, Mathematical models and methods in the water industry, Math. Comput. Modell. 39 (2004) 1353–1374.

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

[37] V.V. Nguyen, D. Hartmann, M. König, A distributed agent-based approach for simulation-based optimization, Advanced Engineering Informatics, 26 (2012) 814–832.

[38] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1(1) (1997) 67–82.

[39] E. Alperovitz, U. Shamir, U., Design of optimal water distribution systems, Water Resour. Res. 13(6) (1977) 885-900.

[40] T. Kohonen, Essentials of the self-organizing map. Neural Netw. 37 (2013), 52–65.

[41] R. Wehrens, L.M.C. Buydens, Self- and Super-organizing Maps in R: The kohonen Package, J. Stat. Softw. 21(5) (2007) 1-19.

[42] M.C. Cunha, J. Sousa, Water distribution network design optimization: simulated annealing approach, J. Water Resour. Plan. Manag. 125(4) (1999) 215-221.

[43] D.A. Savic, G.A., Walters, Genetic operators and constraint handling for pipe network optimization, in: Evolutionary Computing, AISB Workshop 1995, 154–165.

[44] A.C. Zecchin, A.R. Simpson, H.R. Maier, J.B. Nixon, Parametric study for an ant algorithm applied to water distribution system optimization, IEEE Trans Evol. Comput. 9(2) (2005)175-191.

[45] I. Montalvo, J. Izquierdo, R. Pérez, M.M. Tung, Particle Swarm Optimization applied to the design of water supply systems, Comput. Math. Appl. 56(3) (2008) 769–776.

[46] I. Montalvo, J. Izquierdo, R. Pérez, P.L. Iglesias, A diversity-enriched variant of discrete PSO applied to the design of Water Distribution Networks, Eng. Optim. 40(7) (2008) 655–668.

[47] Z.Y. Wu, A.R. Simpson, Competent genetic-evolutionary optimization of water distribution systems, J. Comput. Civil Eng. 15(2) (2001) 89-101.