DEPARTMENT OF ENGINEERING MANAGEMENT

An iterated local search algorithm for water distribution network design optimisation

Annelies De Corte & Kenneth Sörensen

UNIVERSITY OF ANTWERP
Faculty of Applied Economics
City Campus
Prinsstraat 13, B.226
B-2000 Antwerp
Tel. +32 (0)3 265 40 32
Fax +32 (0)3 265 47 99
www.uantwerpen.be

FACULTY OF APPLIED ECONOMICS

DEPARTMENT OF ENGINEERING MANAGEMENT

An iterated local search algorithm for water distribution network design optimisation

Annelies De Corte & Kenneth Sörensen

RESEARCH PAPER 2014-018
AUGUST 2014

University of Antwerp, City Campus, Prinsstraat 13, B-2000 Antwerp, Belgium
Research Administration – room B.226
phone: (32) 3 265 40 32
fax: (32) 3 265 47 99
e-mail: joeri.nys@uantwerpen.be

The research papers from the Faculty of Applied Economics are also available at www.repec.org (Research Papers in Economics - RePEc)

D/2014/1169/018

An iterated local search algorithm for water distribution network design optimisation

Annelies De Corte* and Kenneth Sörensen

ANT/OR Operations Research Group
University of Antwerp, Belgium

July 2013

Using a structured design methodology, we develop an efficient iterated local search metaheuristic for the water distribution network design optimisation problem. The algorithm outperforms existing algorithms in the literature while being considerably less complex. Our approach is — contrary to algorithms in the literature — also shown to perform well on large, realistic instances.

Keywords: water distribution network design; iterated local search; mixed-integer non-linear optimisation; pipe sizing

1 Introduction

Although a safe supply of clean water is one of the most basic necessities of any human being and any society, large parts of the world remain without access to it. According to the World Health Organisation, 1.1 billion people have no access to a safe source of drinking water. As a consequence, goal 7 of the Millennium Development Goals aims at halving the proportion of people without sustainable access to safe potable water by 2015 (World Health Organisation, 2013).

The most efficient way to transport potable water is through pipe networks. The realisation or reorganisation of these distribution systems, however, requires huge investments. Hence, an efficient *layout, design, and planning* of water distribution networks is crucial, and it is

*Corresponding author: Prinsstraat 13, 2000 Antwerp, Belgium — annelies.decorte@uantwerpen.be

important that appropriate tools are developed to support water distribution companies or governments in achieving this.

Table 1: The different phases in the optimisation of water distribution networks

Phase	Decision level	Decision variables
Layout	Strategic (long term)	Network topology, ...
Design	Tactical (medium and long term)	Pipe sizing, ...
Planning	Operational (short term)	Pump and valve control, ...

Decisions concerning a water distribution network are taken at different levels. This paper focuses on the optimal *design* of water distribution networks. In this design phase, the water distribution network topology (node placement and network connectivity) is assumed to be known. As shown in Table 1, the optimal design problem concerns medium and long term decisions related to the optimal type of pipe connecting the supply, demand and junction nodes in the network. Such decisions are taken when pipes need to be replaced due to aging or to changing demographic settings for existing networks, or when network extensions are built. The design of water distribution networks gives rise to an optimisation problem called the water distribution network design optimisation problem. The aim is to find the least cost design, in terms of optimal pipe types, that satisfies hydraulic laws and customer requirements.

This paper is organised as follows. The next section given a formal definition of this mixed-integer, non-linear problem. In Section 3, an overview of existing literature on water distribution network design optimisation is presented. Section 4 discusses the iterated local search procedure that has been developed to tackle this network design optimisation problem. Iterated local search is a metaheuristic technique that has been successfully applied to a variety of combinatorial optimisation problems. Section 5 describes an extensive experiment used to determine the optimal algorithm configuration together with the optimal settings of all parameters, and presents experimental results. The WDND optimisation problem can be extended in a variety of ways, the most important of which are described in a final section together with the general conclusions of this paper.

2 Model formulation

In this paper, the focus is on the single-objective, single-period, gravity-fed water distribution network design (WDND) optimisation problem, which is in line with previous research in this area, as can be seen in Section 3. Possible extensions of this basic model are formulated in Section 6.

To formulate the mathematical model, the water distribution network is represented as a connected graph with a set of water demand or supply nodes $N = \{n_1, n_2, \ldots\}$, a set of water distribution pipes $P = \{p_1, p_2, \ldots\}$. The set of closed loops in this graph is denoted

$L = \{l_1, l_2, \ldots\}$. The objective of this water distribution network design optimisation problem is to minimise the total investment cost TIC of the network design by selecting the optimal pipes out of a set of available pipe types. The cost of an individual pipe depends on the type t that is chosen for this pipe from the list of commercially available types $T = \{t_1, t_2, \ldots\}$. The type of a pipe determines both its diameter and the material of which it is made, which in turn determine its hydraulic properties. If the cost per meter of a pipe p of type t is represented by IC_t and the length of pipe p is represented as L_p, the objective function of the single period, gravity-fed water distribution network design optimisation problem can be written as:

$$\min TIC = \sum_{p \in P} \sum_{t \in T} L_p \; IC_t \; x_{p,t} \tag{1}$$

where $x_{p,t}$ is a binary decision variable that determines whether pipe p is of type t ($x_{p,t} = 1$) or not ($x_{p,t} = 0$). Since the network layout is assumed to be given, L_p is known. The investment cost IC_t is also given for every commercially available pipe type.

This objective function is limited by physical mass and energy conservation laws, and by minimum head requirements in the demand nodes.

The *mass conservation law* must be satisfied for each node $n \in N = \{n_1, n_2, \ldots\}$. This law states that the volume of water flowing into a node in the network per unit of time must be equal to the volume of water flowing out of this node. Let $Q_{(n_1,n)}$ represent the amount of water flowing from node n_1 to node n, and let S_n be the water supply and D_n the water demand of node n (all expressed in m^3/s) then the following should hold:

$$\sum_{n_1 \in N/n} Q_{(n_1,n)} - \sum_{n_2 \in N/n} Q_{(n,n_2)} = D_n - S_n \qquad \forall n \in N \tag{2}$$

Furthermore, for each closed loop $l \in L = \{l_1, l_2, \ldots\}$, the *energy conservation law* must be satisfied. This law states that the sum of pressure drops in a closed loop is zero. Pressure drops, or head losses, in piping systems are caused by wall shear in pipes and friction caused by piping components such as junctions, valves, and bends. In the basic WDND optimisation problem, only the wall shear in the pipes is taken into account. The energy conservation law can be stated as:

$$\sum_{p \in l} \Delta H_p = 0 \qquad \forall l \in L \tag{3}$$

with ΔH_p representing the head loss in pipe p. Head losses in the pipes of the network are approximated using Hazen–Williams equations, with the parameters set to the values used by EPANET 2.0, the hydraulic solver used in this paper:

$$\Delta H_p = \frac{10.6668 \; y_p \; Q_p^{1.852} \; L_p}{\sum\limits_{t \in T} (x_{p,t} \; C_t^{1.852} \; D_t^{4.871})} \tag{4}$$

which (3) can be rewritten as:

$$\sum_{p \in l} \left[\frac{10.6668 \; y_p \; Q_p^{1.852} \; L_p}{\sum\limits_{t \in T} (x_{p,t} \; C_t^{1.852} \; D_t^{4.871})} \right] = 0 \tag{5}$$

In this equation, y_p is the sign of Q_p, which is the amount of water flowing through pipe p (in m^3/s). This sign incorporates changes in the water flow direction relative to the defined flow directions. L_p is the pipe length (in m), C_t is the Hazen–Williams roughness coefficient of pipe type t (unitless) and D_t is the diameter of pipe type t (in m). Parameters D_t and C_t are determined by the type of a pipe and are assumed given for each available type. Note that y_p and Q_p represent alternative formulations of $Q_{n_1,n2}$ if pipe p connects n_1 and n_2.

Finally, *minimum pressure head requirements* exist for every (demand) node $n \in N$. Let H_n be the pressure head in node n (in m) and H_n^{min} the minimum pressure head in node n (in m). This constraint therefore can be represented as:

$$H_n^{min} \leq H_n \qquad \forall n \in N \tag{6}$$

3 Literature

The water distribution network design optimisation problem described in the previous section is a mixed-integer, non-linear problem (MINLP), and is shown to be NP-hard by Yates et al. (1984). A large number of (meta)heuristic techniques have been developed to solve the problem over the past thirty years, as can be seen in Table 2. A more detailed overview of the developed techniques is given in De Corte and Sörensen (2013).

Table 2: Overview of the metaheuristic techniques developed for the water distribution network design optimisation problem

Local search metaheuristics	
Simulated Annealing	(Loganathan et al., 1995; Cunha and Sousa, 1999, 2001)
Tabu Search	(Cunha and Ribeiro, 2004)
Population-based metaheuristics	
Genetic Algorithm	(Murphy and Simpson, 1992; Simpson et al., 1994)
Improved Genetic Algorithm	(Dandy et al., 1996; Savic and Walters, 1997)
Real Coding Genetic Algorithm	(Gupta et al., 1999; Vairavamoorthy and Ali, 2000; Reca et al., 2008)
Differential Evolution	(Vasan and Simonovic, 2010), (Suribabu, 2010)
Memetic Algorithm	(Baños et al., 2007, 2010)
Cross-Entropy	(Perelman and Ostfeld, 2007)
Scatter Search	(Lin et al., 2007)
Immune Algorithm	(Chu et al., 2008)
Improved Immune Algorithm	(Chu et al., 2008)
Shuffled Frog Leaping Algorithm	(Eusuff and Lansey, 2003)
Constructive metaheuristics	
Ant Colony optimisation	(Maier et al., 2003)
Ant Systems	(Zecchin et al., 2005, 2006)
Max-Min Ant System	(Zecchin et al., 2006)
Particle Swarm Harmony Search	(Geem, 2009)

In De Corte and Sörensen (2013) we discuss several shortcomings of the current state of the art. An important one is that most metaheuristic techniques are tested on only a handful of available benchmark networks: the New York City Tunnels network (Schaake and Lai, 1969), the

Hanoi network (Fujiwara and Khang, 1990) and the two loop network (Alperovits and Shamir, 1977). It is demonstrated in De Corte and Sörensen (2013) that these networks show poor resemblance to real networks. Additionally, nearly every developed technique finds the same optimal solutions, which raises questions with respect to the challenging nature of the default benchmark networks. For these reasons, it is hard to draw solid conclusions on the performance and robustness of earlier developed techniques for water distribution network design optimisation. To overcome this lack of a comprehensive set of benchmark networks, a network generation tool, called HydroGen, was developed (De Corte and Sörensen, 2014). HydroGen is able to generate a wide variety of benchmark networks of arbitrary size and varying characteristics. An extensive library of benchmark networks generated using Hydrogen is available online at `http://antor.ua.ac.be/hydrogen`. This library could be used as a new default benchmark set in the area of water distribution network design optimisation.

4 Iterated Local Search

One of the criticisms levied by De Corte and Sörensen (2013) at the state of the art in the field of algorithm development for WDND optimisation is that most of the developed algorithms are overly complex. This complexity is mainly the result of the metaheuristic frameworks employed in previous research, which is typically of the "nature-inspired" type, with many parameters and a complicated algorithmic structure. One of the main aims of the current paper is to show that the same (if not better) performance can be obtained by an algorithm that is considerably simpler. Algorithmic simplicity has many advantages: first of all, it makes them easier to grasp and therefore, quicker to implement. Simple algorithms tend to have fewer bugs, and take a reduced effort for testing. Moreover, their maintainability is easier and flexibility is higher, which is of crucial importance when used by third parties in their decision support tools.

In this section, an iterated local search (ILS) algorithm is developed to tackle the water distribution network design optimisation problem. ILS is a simple metaheuristic that alternates a local search phase with a random perturbation. Local search is used for intensification purposes, whereas the perturbation is used for diversification, i.e., to escape from a local optimum. Therefore, ILS can be understood as a random walk over local optima. Despite its simplicity, ILS is at the basis of many state-of-the-art algorithms for problems such as travelling salesman problems, scheduling problems, graph partitioning problems, etc. (Lourenço et al., 2003).

In a first phase, the ILS developed in this paper generates an initial solution, that is then sorted. In a next phase, the algorithm iteratively applies a large perturbation to the current solution, on which the local search algorithm is applied afterwards. The resulting local optimum, s'', is compared to the best found solution so far, s^*. This procedure is repeated until a stopping criterion is met. The pseudocode of this algorithm is presented in Algorithm 1. For a more detailed discussion of ILS, we refer to Lourenço et al. (2001).

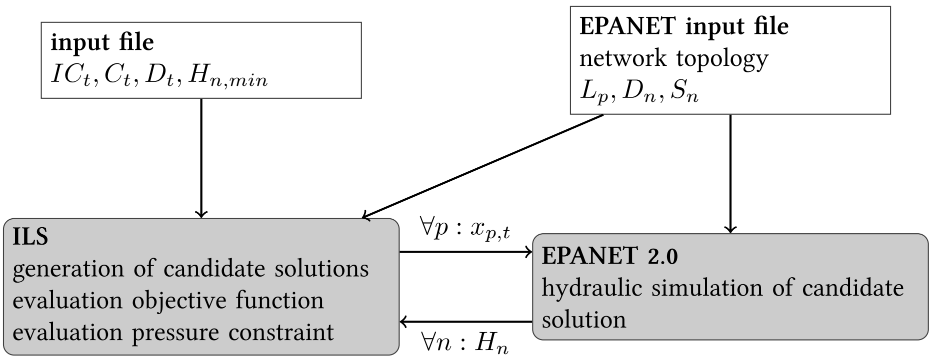
Algorithm 1 Iterated Local Search

s_0 = GenerateInitialSolution;
Sort;
s = LocalSearch(s_0);
while *stopping criterion not met* **do**
    s' = Perturbation(s);
    s'' = LocalSearch(s');
    s = Evaluation(s'', s^*);

4.1 Algorithm implementation

Every candidate solution generated by the ILS has to be checked for hydraulic feasibility. This is done by a hydraulic solver, EPANET 2.0 (Rossman, 2000), which solves mass (equation (2)) and energy (equation (5)) conservation laws. EPANET 2.0 obtains the water flow in each pipe and the pressure head in each node. This pressure head is then used by our ILS algorithm to evaluate the pressure constraint, equation (6), and corresponding feasibility of the candidate solution.

Figure 1: Interaction between ILS and EPANET 2.0.



The interaction between the ILS algorithm and the EPANET simulation engine is established by the use of an extended EPANET toolkit, developed by M. López-Ibáñez (López-Ibáñez, 2009). More information can be found on `http://iridia.ulb.ac.be/~manuel/epanetlinux.html`.

4.2 Algorithm description

Lourenço et al. (2001) state that, in order to achieve a good performance, four components of an ILS algorithm need consideration: the generation of the initial solution, the local search mechanism, the perturbation and the acceptance criterion.

The algorithm developed in this paper is the result of a structured design phase in which different alternatives for each of the aforementioned components are tested and compared in a full-factorial experiment. In this way, it is ensured that the best-performing components are kept and that the parameters of these components are set to their best possible levels. These components are discussed in the following paragraphs. The results of the statistical experiment are reported in Section 5.

Initial solution

In a first step, an initial feasible solution s_0 is generated. Two methods are compared. Method `highestCost` sets all pipe diameters to the largest available diameter in the set of commercially available pipe types. This leads to a feasible, albeit very expensive, solution. The second method, `lowCost`, constructs an initial solution in two steps. First, it sets all pipes to the smallest possible diameter. This solution is almost always infeasible (if it is feasible, it is optimal). In a second step, therefore, pipe diameters are increased by iteratively going through the set of pipes and increasing each pipes diameter with one size at a time, until a feasible solution is attained. This feasible solution will have a lower cost than the one created under `highestCost`.

Sort

In a next step, the set of all pipes is sorted. The order in which they are sorted determines the order in which pipes will be handled in the local search. Two sort orders were tested: decreasing pipe length and random. Pre-testing shows that sorting the pipes according to decreasing pipe length always outperforms sorting pipes randomly.

Local search

In iterated local search algorithms, the local search component is responsible for most of the improvement in solution quality. In the water distribution network design optimisation problem, the cost of a solution can only be decreased by decreasing the diameter of a pipe. Two local search mechanisms are tested to achieve this. The first, `noMemory`, starts from the sorted initial solution s_0 and goes through the set of pipes by applying one type of move. The applied move, *decrease*, reduces the pipe diameter of one pipe with one size at a time. Hence, neighbouring solutions will have equal diameters for all pipes except one (the one that has been decreased). A first-improving strategy is applied, which means that the current solution will be adjusted as soon as a feasible, lower cost solution is encountered. The second mechanism, `memory`, uses the same move and strategy as the first, but a memory structure is added. When, while iteratively decreasing the pipe diameters during the local search, a pipe is encountered for which a decrease in diameter results in a violation of the minimal pressure head requirements, this pipe is added to the memory list, and the corresponding diameter is increased again. This memory list keeps track of pipes that cannot be decreased any further during the current local search. Therefore, the search space of the local search is reduced, which leads to shorter local search procedures and thus shorter algorithm running times. Another implication of adding a memory structure lies in a potential loss in solution quality (or a higher cost end solution). This is due to the fact that the pipes on the memory list cannot be decreased in diameter any further, even though this decrease (that led to infeasible solutions in a specific encountered configuration), could become feasible again in other future configurations. The list is erased after every perturbation in order to start every local search procedure with an empty memory list. This local search procedure generates a new local optimum s''.

Evaluation and acceptance

In a next step, the new local optimum s'' is evaluated. If the cost of the current solution s'' is lower than that of the best solution found so far s^*, s^* is replaced by s''. Otherwise, the current best solution remains unchanged.

$$\text{Evaluation:} \begin{cases} s^* = s'' & \text{if } TC_{s''} < TC_{s^*} \\ s^* \text{unchanged} & \text{otherwise} \end{cases}$$

The acceptance criterion specifies which solution will be used as starting solution s in the next iteration. This does not necessarily have to be the best solution found so far. Two different acceptance mechanisms are compared. The first, called `bestCost`, always uses the best solution found so far s^* as a starting point for the next iteration. The second, `currentCost`, works with the most recently found local optimum s''.

$$\text{Acceptance:} \begin{cases} s = s^* & \Leftrightarrow \texttt{bestCost} \\ s = s'' & \Leftrightarrow \texttt{currentCost} \end{cases}$$

Perturbation

A perturbation is applied to the current solution s to escape from local optima and to explore other parts of the search space. The perturbation operator works as follows. A fixed percentage (`perturbationRate`) of all pipes is randomly selected. The perturbation then increases the diameters of these pipes with one size. This perturbed solution s' is used as input for the local search algorithm.

Termination criterion

One of the properties of the iterated local search method developed in this paper is that the algorithm is able to provide a feasible solution at any given moment. The applied termination criterion is a maximal number of times (`noImprovement`) that the local search procedure can be performed without finding a solution s'' that is of lower cost than the best solution found so far s^* Different values for `noImprovement` are tested.

5 Experimental results

In this section, an extensive statistical experiment is carried out to determine the ideal configuration of the algorithm, i.e., the components that — when combined — achieve the best performance. In the same experiment, the ideal parameter settings for each of these components is determined.

5.1 Algorithm configuration

A full factorial experiment is executed on a set of 25 different test networks (the XXXaX networks in Table 6) to determine the techniques and parameters that yield the best performance. The perturbation phase of the algorithm is a randomised component and the algorithm therefore generally finds different solutions when executed multiple times on the same problem instance. The algorithm is therefore executed 10 times on each benchmark instance and for each combination of parameter values. This leads to a data set of $2 \times 2 \times 2 \times 4 \times 5 \times 25 \times 10 = 40,000$ observations.

The lowest cost found during every execution of the algorithm is taken as the performance measure (dependent variable) for every combination of parameters. An analysis of variance (ANOVA) model is then estimated using the statistical package JMP (`www.jmp.com`). Each model uses a random effect for the instance in order to indicate that all the measurements for the same instance are correlated. Table 4 shows the p-values of the F-tests that establish the significance of each parameter main effect.

Table 3: Parameter values applied in the full-factorial experiment.

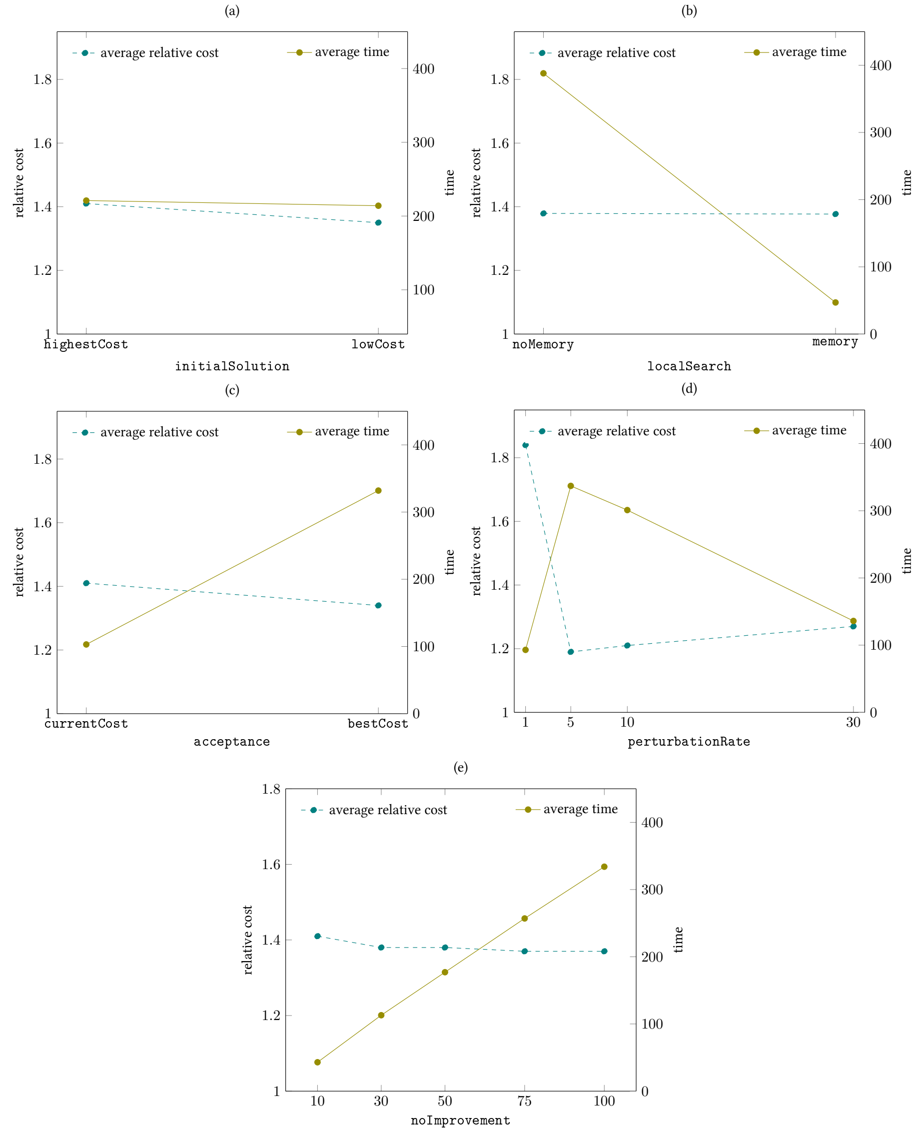
Parameter	Value
`initialSolution`	`highestCost,lowCost`
`localSearch`	`noMemory, memory`
`acceptance`	`currentCost, bestCost`
`perturbationRate`	`1%, 5%, 10%, 30%`
`noImprovement`	`10, 35, 50, 75, 100`

As can be seen from Table 4 and Figure 2, most parameters have a significant impact on both the average relative cost and the average execution time.

Table 4: p-values of the F-tests to determine the significance of each parameter in the ANOVA models for the average cost and average execution time.

Parameter	Avg cost	Time
`initialSolution`	**<0.0001**	0.1501
`localSearch`	0.3542	**<0.0001**
`acceptance`	**<0.0001**	**<0.0001**
`perturbationRate`	**<0.0001**	**<0.0001**
`noImprovement`	**<0.0001**	**<0.0001**

Figure 2: Effect of (a) `initialSolution` (b) `localSearch`, (c) `acceptance`, (d) `perturbationRate` and (e) `noImprovement` on the average relative cost and average execution time (in s).



As reported in Figure 2 (a), the two initial solution construction methods (`initialSolution`) do not affect the average execution times significantly, but `lowCost` leads to slightly lower costs.

From Figure 2 (b), it is clear that adding a memory structure (`memory`) to the local search leads to significant time savings without worsening average cost results.

Figure 2 (c) and Figures 4 and 5 show that `bestCost` gives better results in terms of cost minimisation. When a new, lower cost solution is encountered, `bestCost` continues exploring this region in the solution space, which could be seen as an intensification strategy. For this problem, this strategy generates better results than the `currentCost` strategy, in which knowledge on regions where good solutions are encountered is not exploited in the same way.

In Figure 2 (d) and Table 4 it can be observed that the value of `perturbationRate` that produces the best algorithm performance in terms of average cost is 5 %. This is explained by the fact that smaller perturbation rates (e.g., 1 %) do not allow the algorithm to escape from poor local optima. This also explains why small perturbation rates have significantly shorter running times: the algorithm is not able to escape the local optima, therefore, the stopping criterion (the maximal number of iterations that no improvement can be made) will be reached quickly. On the other hand, perturbation sizes that are too large (e.g. $>$ 30 %) lead to significant loss of information about the current local optimum and its surrounding solution space. This can also be seen in Figure 3: higher perturbation rates have higher amplitudes, which indicate that the cost of the perturbed solutions is much higher than the one before the perturbation, and therefore do not exploit promising areas in the solution space enough. Consecutive s'' solutions when a high perturbation rate is used therefore differ much more than under lower perturbation rates, and the local search algorithm has difficulties recovering the solution quality.

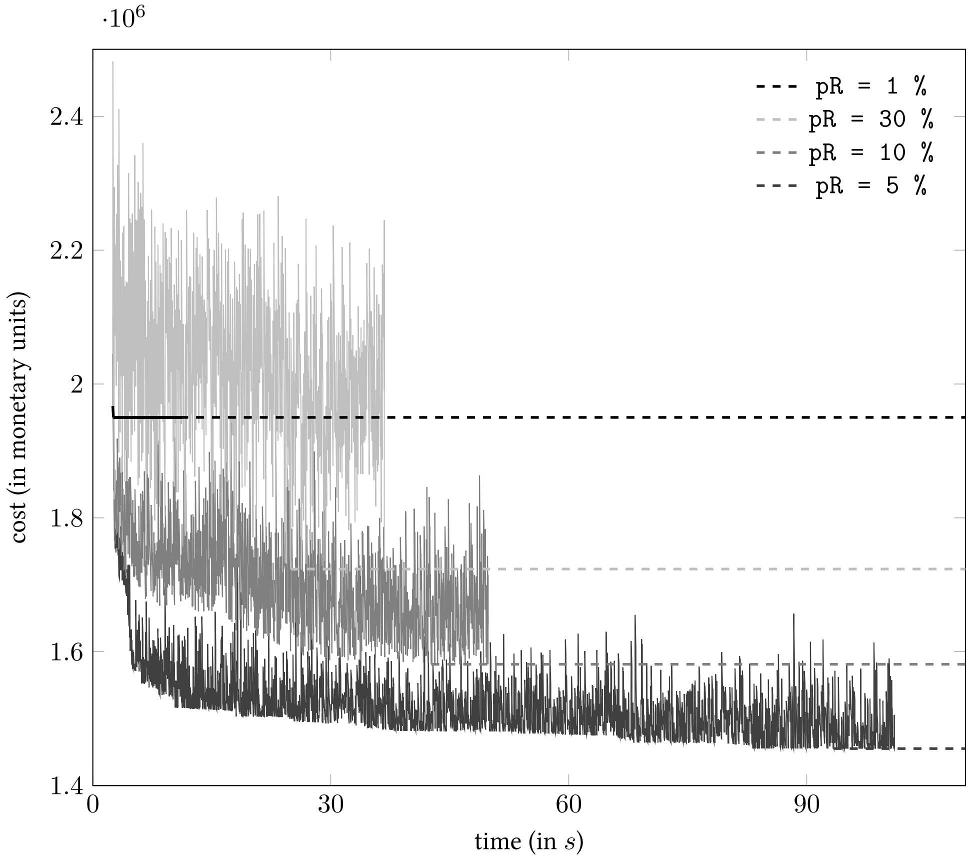
As can be expected, running times increase linearly with an increase in the maximal number of times that no lower cost solution can be encountered during the local search (`noImprovement`). Average relative costs decrease when this maximal number is increased. A clear decrease is made if this number is increased from 10 to 30, but further increases in `noImprovement` do not result in significant additional cost drops.

5.2 Results on benchmark instances

The developed iterated local search algorithm is applied to both the widely-used benchmark networks from the literature and a set of more challenging benchmark instances generated by HydroGen. The previous section clearly showed the conflicting nature of the objective function value and the computation time required by the algorithm: most parameters settings that increase the computation time decrease the solution cost and vice versa.

Rather than choosing a single compromise algorithm configuration, two different configurations are therefore tested. The first configuration, called **cost**, focuses on obtaining least cost solutions while largely disregarding the computation time. The second, called **time**, has its

Figure 3: Effect of varying `perturbationRate (pR)` on the network cost and algorithm execution time: consecutive s'' and s are plotted. Parameter settings: **cost**, applied on network 300a1.



parameters set in order to achieve minimal running times, finding a solution that is perhaps somewhat worse. Specific parameter settings are listed in Table 5

5.2.1 Traditional benchmark networks

When applied to the well-known and widely-used benchmark networks: the New York city tunnels network (Schaake and Lai, 1969), the Hanoi network (Fujiwara and Khang, 1990), and the Two loop network (Alperovits and Shamir, 1977) the ILS developed in this paper consistently (both under **cost** and **time** settings) finds the optimal reported solution values (3.864×10^7 USD , 6.081×10^6 USD and $419,000$ USD respectively for a hydraulic coefficient $w = 10.6668$).

Figure 4: Effect of varying `acceptance` on the network cost and algorithm execution time: s", s^* and s are plotted. Other parameter settings: **cost**, applied on network 300a1. A close-up is given in Figure 5.

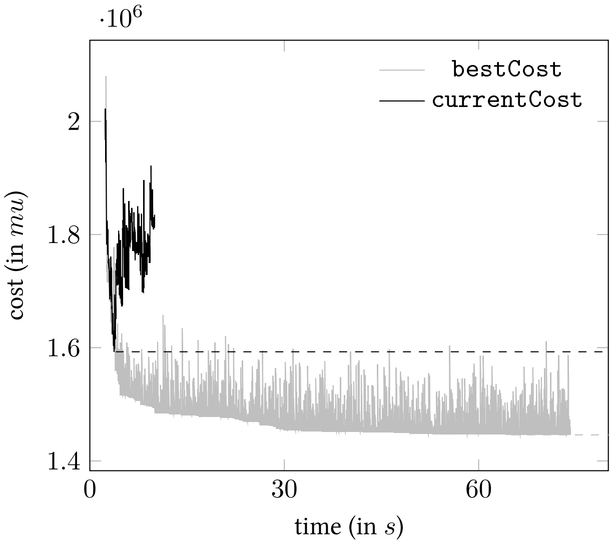


Figure 5: Close-up of effect of (a) `acceptance = bestCost` and (b) `acceptance = currentCost` on solution cost.

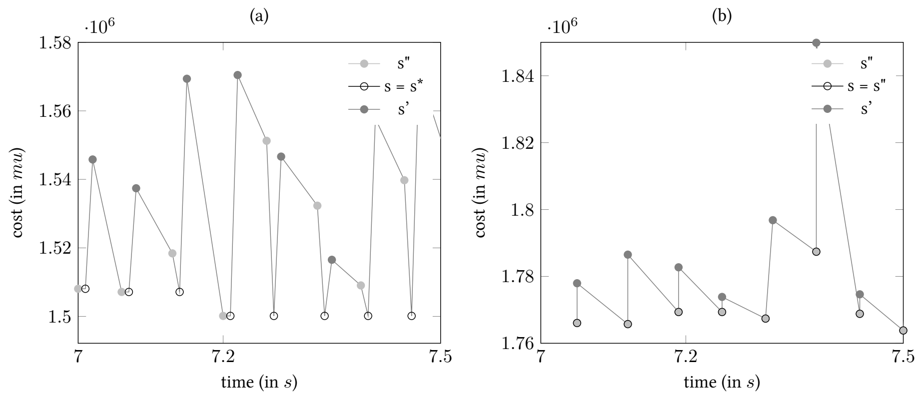


Table 5: Parameter setting under **cost** and **time** settings.

Phase	**Cost**	**Time**
`initialSolution`	`low`	`low`
`localSearch`	`memory`	`memory`
`acceptance`	`bestCost`	`currentCost`
`perturbationRate`	5%	30%
`noImprovement`	100	10

5.2.2 HydroGen benchmark networks

The algorithm is also applied to a set of HydroGen test networks. The input files of these networks are available at http://antor.ua.ac.be/hydrogen. For both settings, **cost** and **time**, 10 runs per network are executed. The algorithm was tested on 50 different networks (100a1 – 500b5). For every network, the number of nodes, including reservoirs, (n) and the number of pipes (p) is mentioned. Both the average and minimal network cost (**Avg c** and **Min c**) and running time (**Avg t** and **Min t**) of these 10 runs can be found in Table 6. As can be expected, under **cost** settings, average and minimal costs are about 20% lower, whereas average and minimal running times are between 5 and 15 times longer compared to the **time** configuration.

A visual example of an optimised network can be seen in Figure 6.

6 Conclusion and future research

Water distribution network design optimisation poses a challenging task for water distribution companies. This medium-to-long term decision problem concerns finding the least-cost pipe configuration using a set of discrete pipe types, while respecting both hydraulic laws and customer requirements. The non-linearity between water flows and corresponding head losses and the limited set of possible pipe types render this problem a complex, non-linear, mixed-integer combinatorial optimisation problem.

Algorithms found in the literature for this problem are typically complex "nature-inspired" metaheuristics. These algorithms have many parameters, which are set in an ad-hoc way without any clear underlying methodology. In a previous paper, we have argued that these algorithms are not properly tested on challenging benchmark instances. This paper demonstrates that a simple algorithm can achieve equal (and, we believe, much better) performance when designed according to a transparent methodology and configured in a methodologically sound way.

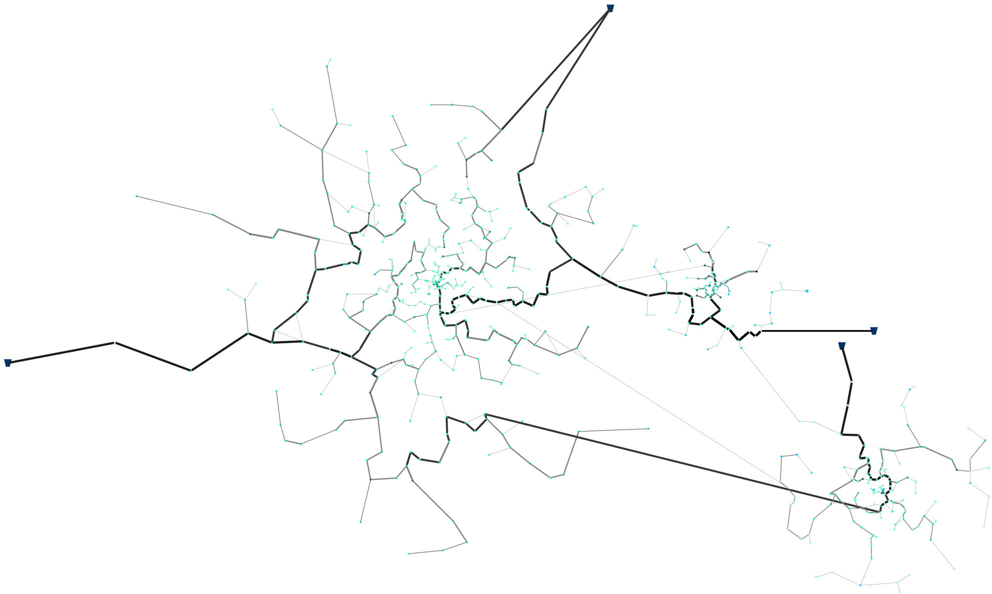
To this end, an iterated local search technique is presented to tackle this basic water distribution network design optimisation problem. A full-factorial experiment is conducted to find the best

Table 6: Results of the ILS algorithm applied on a set of HydroGen test networks.

			Cost
			`initialSolution = low`
			`localSearch = memory`
			`acceptance = bestCost`
			`perturbationRate = 5%`
			`noImprovement = 100`
							Time
							`initialSolution = low`
							`localSearch = memory`
							`acceptance = currentCost`
							`perturbationRate = 30%`
							`noImprovement = 10`			

Network	**n**	**p**	**Avg c**	**Min c**	**Avg t**	**Min t**	**Avg c**	**Min c**	**Avg t**	**Min t**
100a1	101	109	74,072	69,568	2.3	1.5	93,286	81,141	0.4	0.4
100a2	101	109	58,731	56,763	2.9	2.2	69,728	67,357	0.5	0.4
100a3	101	109	101,433	97,915	2.2	1.2	111,091	109,713	0.4	0.4
100a4	101	109	70,184	65,144	2.4	1.4	81,872	77,970	0.4	0.4
100a5	101	109	96,253	94,781	2.1	1.5	105,655	101,700	0.4	0.4
200a1	201	219	515,030	495,296	22.0	10.9	646,809	582,199	1.8	1.8
200a2	201	219	435,708	422,936	16.8	7.7	515,575	466,647	2.0	2.0
200a3	201	219	425,708	411,305	16.3	5.8	518,359	488,139	1.8	1.8
200a4	201	219	733,938	697,267	19.1	10.1	867,690	800,047	1.9	1.8
200a5	201	219	475,975	463,591	20.3	10.7	560,850	524,655	1.9	1.8
300a1	302	330	1,465,742	1,424,783	69.0	38.8	1,833,729	1,763,601	4.5	4.4
300a2	302	330	1,411,663	1,369,022	58.6	34.4	1,847,394	1,746,259	4.5	4.0
300a3	302	330	1,849,662	1,688,332	70.0	41.4	2,060,029	1,970,482	5.5	4.8
300a4	302	330	1,584,277	1,558,914	85.1	53.0	1,977,206	1,807,012	5.3	4.4
300a5	302	330	1,359,832	1,315,723	75.3	38.5	1,846,644	1,723,060	4.7	4.3
400a1	402	440	2,203,678	2,148,597	79.1	35.5	2,662,861	2,449,502	9.1	8.3
400a2	402	440	2,536,652	2,325,390	111.8	37.7	2,839,919	2,701,355	11.2	9.5
400a3	402	440	2,902,426	2,722,449	154.4	87.2	3,980,127	3,845,274	8.8	8.0
400a4	402	440	4,659,084	2,997,028	136.9	64.6	4,821,734	3,939,808	10.7	8.2
400a5	402	440	2,859,680	2,446,748	111.9	62.3	2,925,105	2,683,210	11.0	8.8
500a1	502	550	6,313,431	6,162,037	263.0	113.1	6,878,290	5,437,490	16.7	13.5
500a2	502	550	5,206,649	4,862,934	173.7	59.1	6,108,900	5,112,050	14.1	13.0
500a3	502	550	4,426,193	3,884,712	219.1	53.8	5,498,186	4,952,628	14.1	13.0
500a4	502	550	4,403,306	3,971,958	216.3	64.9	5,306,111	4,655,200	15.3	12.9
500a5	502	550	5,063,995	4,142,606	325.6	140.3	5,234,672	4,600,476	20.1	14.2
100b1	101	100	53,786	52,653	1.7	1.1	57,967	55,978	0.3	0.3
100b2	101	100	95,056	93,710	1.2	0.9	96,613	94,012	0.3	0.3
100b3	101	100	71,486	69,763	1.3	0.9	79,052	73,496	0.4	0.3
100b4	101	100	95,588	94,429	2.0	1.1	97,935	95,956	0.3	0.3
100b5	101	100	73,719	71,778	1.4	1.0	81,423	74,180	0.3	0.3
200b1	201	200	566,988	550,230	13.3	8.1	633,924	588,554	1.4	1.4
200b2	201	200	401,589	383,821	10.3	8.2	442,095	414,887	1.4	1.4
200b3	201	200	339,462	331,950	8.1	6.3	359,933	348,419	1.4	1.4
200b4	201	200	369,347	351,069	9.3	6.0	399,244	359,263	1.4	1.4
200b5	201	200	672,100	629,359	8.7	5.6	755,974	691,216	1.4	1.4
300b1	301	300	2,254,867	2,204,570	20.6	13.4	2,340,978	2,276,432	3.6	3.4
300b2	301	300	1,468,885	1,407,143	24.1	15.6	1,548,259	1,465,696	3.6	3.4
300b3	301	300	1,362,782	1,289,531	25.7	16.1	1,424,273	1,333,938	3.5	3.3
300b4	301	300	1,571,805	1,516,813	21.4	14.3	1,604,164	1,549,792	3.5	3.4
300b5	301	300	1,626,979	1,545,755	30.8	17.4	1,719,841	1,635,173	3.4	3.3
400b1	401	400	3,245,593	3,150,686	49.0	30.9	3,565,780	3,242,113	6.3	6.1
400b2	401	400	2,345,016	2,202,884	42.5	26.8	2,441,772	2,302,471	6.3	6.1
400b3	401	400	2,835,441	2,732,619	38.7	22.2	2,869,374	2,680,146	6.5	6.0
400b4	401	400	2,476,752	2,387,889	38.5	21.7	2,693,404	2,496,172	6.6	6.2
400b5	401	400	2,921,925	2,748,248	38.6	31.1	3,112,441	2,868,215	6.5	6.1
500b1	501	500	4,433,659	4,335,346	79.5	50.0	4,630,904	4,494,752	10.0	9.4
500b2	501	500	4,308,855	4,206,957	86.8	51.9	4,595,831	4,198,528	9.6	9.2
500b3	501	500	3,830,222	3,609,527	69.2	46.4	4,005,423	3,719,080	10.1	9.6
500b4	501	500	3,467,124	3,200,049	73.1	37.5	3,665,580	3,215,396	10.2	9.6
500b5	501	500	5,242,849	5,191,973	98.3	50.4	5,390,328	5,299,424	9.7	9.5

Figure 6: Example of an optimised network containing 3 clusters (of 100, 450 and 155 demand nodes) and 4 reservoirs (marked in thicker symbols at the outer ends of the clusters). There are 778 pipes, with lengths between 65 cm and 1680 m and an average length of 50 m. Optimised pipe diameters are in a range of 40 mm up to 2000 mm: the thicker the line, the bigger the pipe.



algorithm configuration and parameter settings. Two different configurations are compared: one that minimises the solution cost and one that minimises computation time.

Our iterated local search algorithm is applied on both the available benchmark networks and on a set of benchmark instances generated by the network generation tool HydroGen. Since the ILS is currently the only technique applied on the latter instances, it would be interesting that other techniques are applied on the HydroGen instances as well, in order to foster comparison of developed techniques.

The formulated basic water distribution network design optimisation problem can be extended in a number of different ways. Such extensions include dynamic water demands (changes during the day), that tanks (with finite capacity) are used in addition to reservoirs for the water supply and that pumps are added to currently exclusively gravity-fed networks. Another extension is the formulation of the above single-objective optimisation problem as a multi-objective one (where other objectives such as reliability could be added to the cost minimisation objective) or adding extra constraints, such as maximum water velocity constraints, to the problem.

References

A. Alperovits and U. Shamir. Design of optimal water distribution systems. *Water Resources Research*, 13:885–900, 1977.

R. Baños, C. Gil, J. Agulleiro, and J. Reca. A memetic algorithm for water distribution network design. *Soft Computing in Industrial Applications*, 39:279–289, 2007.

R. Baños, C. Gil, J. Reca, and F.G. Montoya. A memetic algorithm applied to the design of water distribution networks. *Applied Soft Computing*, 10:261–266, 2010.

C.-W. Chu, M.-D. Lin, G.-F. Liu, and Y.H. Sung. Application of immune algorithms on solving minimum-cost problem of water distribution network. *Mathematical and Computer Modelling*, 48:1888–1900, 2008.

M. Cunha and L. Ribeiro. Tabu search algorithms for water network optimization. *European Journal of Operational Research*, 157:746–758, 2004.

M. Cunha and J. Sousa. Water distribution network design optimization: simulated annealing approach. *Journal of Water Resources Planning and Management*, 125, 1999.

M. Cunha and J. Sousa. Hydraulic infrastructures design using simulated annealing. *Journal of Infrastructure Systems*, 7(1):32–39, 2001.

G.C. Dandy, A.R. Simpson, and L.J. Murphy. An improved genetic algorithm for pipe network optimisation. *Water Resources Research*, 32:449–458, 1996.

A. De Corte and K. Sörensen. Optimisation of gravity-fed water distribution network design: a critical review. *European Journal of Operational Research*, 228:1–10, 2013.

A. De Corte and K. Sörensen. Hydrogen: an artificial water distribution network generator. *Water Resources Management*, 28:333–350, 2014.

M.M. Eusuff and K.E. Lansey. Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources Planning and Management*, 129, 2003.

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

Z.W. Geem. Particle-swarm harmony search for water network design. *Engineering Optimization*, 41:297–311, 2009.

I. Gupta, A. Gupta, and P. Khanna. Genetic algorithm for optimization of water distribution systems. *Environmental Modelling & Software*, 14:437–446, 1999.

M.D. Lin, Y.H. Liu, G.F. Liu, and C.W. Chu. Scatter search heuristic for least-cost design of water distribution networks. *Engineering Optimization*, 39:857–876, 2007.

G.V. Loganathan, J.J. Greene, and T.J. Ahn. Design heuristic for globally minimum cost water distribution systems. *Journal of Water Resources Planning and Management*, 121:182–192, 1995.

M. López-Ibáñez. *Operational optimisation of water distribution networks*. PhD thesis, School of Engineering and the Built Environment, Edinburgh Napier University (UK), 2009.

Helena R Lourenço, Olivier C Martin, and Thomas Stützle. *Handbook of Metaheuristics*, chapter 11: Iterated local search. Springer US, 2003.

H.R. Lourenço, O. Martin, and T. Stützle. Iterated local search. *arXiv preprint math/0102188*, 2001.

H.R. Maier, A.R. Simpson, A.C. Zecchin, W.K. Foong, K.Y.Phang, H.Y.Seah, and C.L. Tan. Ant colony optimization for design of water distribution systems. *Journal of Water Resources Planning and Management*, 129:200–209, 2003.

L.J. Murphy and A.R. Simpson. Genetic algorithms in pipe network optimisation. Research Report R93, University of Adelaide, 1992.

L. Perelman and A. Ostfeld. An adaptive heuristic cross-entropy algorithm for optimal design of water distribution systems. *Engineering Optimization*, 39:413–428, 2007.

J. Reca, J. Martínez, C. Gil, and R. Baños. Application of several meta-heuristic techniques to the optimization of real looped water distribution networks. *Water Resources Management*, 22:1367–1379, 2008.

L.A. Rossman. *EPANET 2: users manual*. Environmental Protection Agency, 2000.

D. Savic and G. Walters. Genetic algorithms for least-cost design of water distribution networks. *Journal of Water Resources Planning and Management*, 123:67–77, 1997.

J. Schaake and D. Lai. Linear programming and dynamic programming applications to water distribution network design. Report 116, Department of Civil Engineering, MIT, Cambridge USA, 1969.

A.R. Simpson, G.C. Dandy, and L.J. Murphy. Genetic algorithms compared to other techniques for pipe optimization. *Journal of Water Resources Planning and Management*, 120:423–443, 1994.

C Suribabu. Differential evolution algorithm for optimal design of water distribution networks. *Journal of Hydroinformatics*, 12(1):66–82, 2010.

K. Vairavamoorthy and M. Ali. Optimal design of water distribution systems using genetic algorithms. *Computer-Aided Civil and Infrastructure Engineering*, pages 374–382, 2000.

A. Vasan and S.P. Simonovic. Optimization of water distribution network design using differential evolution. *Journal of Water Resources Planning and Management*, 136:279–287, 2010.

World Health Organisation. Mdg 7: ensure environmental sustainability, September 2013. URL http://www.who.int/topics/millennium_development_goals/mdg7/en/.

D.F. Yates, A.B. Templeman, and T.B. Boffey. The computational complexity of the problem of determining least capital cost designs for water supply networks. *Engineering Optimization*, 7:143–155, 1984.

A.C. Zecchin, A.R. Simpson, H.R. Maier, and J.B. Nixon. Parametric study for an ant algorithm applied to water distribution system optimization. *IEEE Transactions on Evolutionary Computation*, 9:175–191, 2005.

A.C. Zecchin, A.R. Simpson, H.R. Maier, M. Leonard, A.J. Roberts, and M.J. Berrisford. Application of two ant colony optimisation algorithms to water distribution system optimisation. *Mathematical and Computer Modelling*, 44:451–468, 2006.

7 Acknowledgements

This research is supported by the Research Foundation - Flanders (FWO) and partially supported by the Interuniversity Attraction Poles (IAP) Programme initiated by the Belgian Science Policy Office (COMEX project). Special thanks to Daniel Palhazi Cuervo for his constructive insights on algorithm testing.