Multi-Objective Optimisation of the Pump Scheduling Problem using SPEA2

Manuel López-Ibáñez
Centre for Emergent Computing
School of the Built Environment
Napier University, Edinburgh (UK)
m.lopez-ibanez@napier.ac.uk

T. Devi Prasad
Centre for Emergent Computing
School of the Built Environment
Napier University, Edinburgh (UK)
p.tumula@napier.ac.uk

Ben Paechter
Centre for Emergent Computing
School of Computing
Napier University, Edinburgh (UK)
b.paechter@napier.ac.uk

Abstract- Significant operational cost and energy savings can be achieved by optimising the schedules of pumps, which pump water from source reservoirs to storage tanks, in Water Distribution Networks. Despite the fact that Pump Scheduling Problem involves several conflictive objectives, few studies have considered multi-objective optimisation in terms of Pareto optimality. Our approach links a well-known multi-objective optimiser, SPEA2, with a hydraulic simulator, EPANET, in order to provide a Pareto set of explicit schedules. Since only fixed speed pumps and fixed time intervals are considered, we use a natural binary representation and simple and straightforward initialisation and recombination operators. Unlike earlier studies, feasibility constraints are handled by a methodology based on the dominance relation rather than using penalty functions or reparation mechanisms. We test the proposed approach using a network instance and an assessment of the results is carried out by means of empirical attainment surfaces. The results show that the proposed approach is able to obtain better schedules than the state-of-the-art single-objective algorithm for this network instance and within the same number of function evaluations.

1 Introduction

In Water Distribution Networks, water is pumped from reservoirs into tanks where it is stored during periods of low demand and released during periods of high demand. Costs incurred for the operation and maintenance of pumps constitute the major part of the budget required for management of Water Distribution Networks and can amount to as high as 90% of the operating costs. Therefore, great savings in operational costs can be obtained by carefully scheduling the operations of pumps. Operational costs include cost of the electrical energy consumed during a time period (electrical consumption charge), cost associated with the maximum amount of power consumed during any time interval (demand charge) and maintenance costs due to the wearing on pumps caused by frequent switching of pumps, that is, changing the state of a pump from off to on.

The objective of Pump Scheduling Problem is to minimise the above cost whilst satisfying physical and operational constraints. These constraints include supplying required volume of water at demand nodes with adequate pressure and maintaining water levels inside tanks within maximum and minimum limits. Since pumps are scheduled over a time period, usually 24 hours, we must achieve periodicity between supply and demand. Therefore, another feasibility constraint is that the volume of water in the tanks at the end of the time period is not lower than the volume at the start of time period.

The problem of scheduling the operation of pumps to minimise a single objective has been studied using many approaches. Linear, non-linear, integer, dynamic [1] and mixed [2] programming are some of these approaches. A review of earlier studies was carried out by Ormsbee and Lansey [3].

Complex water distribution networks could not be represented realistically by these methods due to their inherent limitations. Therefore, researchers have considered the application of Genetic Algorithms [4, 5, 6, 7, 8] and other techniques including Particle Swarm Optimisation [9] and Simulated Annealing [10, 11]. The objective in most of these studies was to minimise electricity cost. Other objectives were incorporated as penalties to the objective function. However, little attention has been given to the multi-objective nature of the Pump Scheduling problem.

In this work, we study the multi-objective optimisation of pump scheduling with respect to both electricity cost and number of pump switches. While in single objective optimisation the goal is to find the optimal solution, in multi-objective optimisation defined in terms of Pareto optimality the goal is to find, or at least approximate to, the *optimal Pareto set*, that is, the set of feasible solutions such that none of them is dominated by any other feasible solution. A solution dominates another, if the former is not worse than the later for each objective value and better for at least one objective. When neither of two solutions dominate each other, they are mutually nondominated. As in single objective optimisation, given two solutions with equal objective values, one of them is considered to be dominated by the other.

Savic et al. [12] studied a bi-objective problem: minimisation of both the electricity cost and the number of pump switches. They used a hybrid approach (a genetic algorithm combined with a local search) based on Goldberg's Pareto optimal ranking [13]. Roughly, nondominated solutions in the current population are assigned rank one and then removed from the current population. Then, nondominated solutions in the reduced population are given rank two. This procedure is repeated until a rank has been assigned to all solutions in the current population and the fitness of each solution is calculated according to their rank. In their approach, constraint violations on tank water levels were incorporated into the electricity cost as penalties. This may result in a Pareto set containing infeasible solutions. To prevent this, they assigned all infeasible solutions a rank greater than one.

Sotelo et al. [14] compared the performance of several multi-objective evolutionary algorithms (MOEA) for the pump scheduling problem. They concluded that the Strength Pareto Evolutionary Algorithm (SPEA) produced the best overall performance for the minimisation of four objectives: (*i*) electricity cost, (*ii*) demand charge, (*iii*) number of pump switches, and (*iv*) difference between the initial and final levels of the tank. They incorporated a heuristic into the algorithms for repairing solutions which violate restrictions on the maximum and minimum tank levels. If at some time interval the tank level is above the maximum, then the current scheduling is pumping more water than the amount required, and thus, a number of pumps which were on in the previous intervals are turned off. A similar method was adopted to repair solutions which produced tank levels below the minimum level.

A basic difference between the present study and the previous works which considered multiple objectives is the fact that, in the present work, we use a disaggregated or dual-level methodology, that is, linking an optimisation model with a network simulation model. The use of a simulation model allows to handle more complex network instances, while previous multi-objective approaches tackled simple networks composed of several pumps in parallel and a single tank. The optimisation model chosen is the second version of the Strength Pareto Evolutionary Algorithm (SPEA2) [15] and the network simulator EPANET [16] is used to conduct hydraulic analysis and evaluate pump operation policies. In addition, we use a method based on dominance criteria to handle infeasible solutions [17]. Finally, we assess the quality of the results using attainment surfaces [18] which provide a more robust measure of the quality of multi-objective optimisers than other metrics used in previous studies of multi-objective optimisation for the Pump Scheduling problem.

2 The Pump Scheduling Problem

2.1 Water Distribution Networks

A typical Water Distribution Network, given by Van Zyl et al. [8], is as shown in Figure 1. It consists of a source of potable water (reservoir), three pumps, two tanks and a check valve which prevents water flowing backwards. Water is pumped from the reservoir into the tanks and it is consumed at the demand node. The amount of water which can be pumped is higher than the amount of water consumed, thus pumps do not need to be active all the time. Moreover, water can be stored in tanks to be consumed later in a gradual way. Water demand varies over time and consumption patterns can be estimated using historical data. Therefore, the operation of pumps can be scheduled to minimise the cost of supplying water.

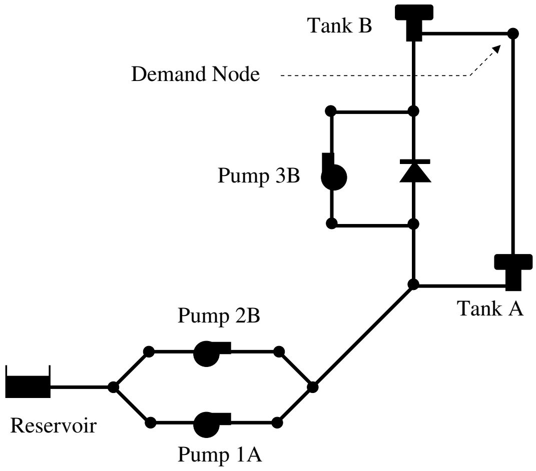

Figure 1: Example of a Water Distribution Network

2.2 The Pump Scheduling Problem

In a Pump Scheduling Problem, operation of N pumps are scheduled over a time period, usually 24 hours. The main goal is to minimise the cost of supplying water, while keeping the physical and the operational constraints within limits [3]. There are two classes of costs associated with the operation of pumps: electrical cost and maintenance costs.

Pump maintenance costs are mainly due to wear and tear of pumps caused by frequent switching them on and off. A *pump switch*, i.e., turning on a pump that was not operating in the previous time period [19], causes a cost of wear on the pump which cannot easily be estimated. However, a safe assumption is that maintenance costs increase with the number of pump switches. Therefore, a surrogate objective of number of pump switches is considered to represent pump maintenance cost.

Total energy cost is composed of an *electrical consumption charge* (£/kW·h), i.e., the cost of electrical energy consumed during a time period, and the *demand charge* (£/kW), i.e., the cost associated with the maximum amount of power consumed (peak energy) within a time interval. The electricity consumption charge usually varies depending on the time of the day, with peak and off-peak electricity tariffs. The energy consumption rate of pumping depends on several factors such as efficiency and power of the pumps, flow of water through the pump and the elevation of tanks.

The *optimal pump policy* is defined as the schedule of pump operations that will result in the lowest total operating cost for a given set of boundary conditions and system constraints [20].

Implicit system constraints define the hydraulic equilibrium state of the system, e.g., conservation of mass at each junction node and conservation of energy around each loop in the network. Implicit system constraints are handled by the network simulator EPANET.

Implicit bound constraints represent system performance criteria. There are two implicit bound constraints that are usually considered: constraints on tank water levels and pressures at demand nodes.

Minimum and maximum tank levels are handled by

EPANET (and in the real world by automatic systems of valves). In order to achieve periodicity between supply and demand, we must ensure that the volume of water in the tanks at the end of the simulation period is not lower than the volume at the start of the simulation period. The difference between the initial volume and the final volume of water in a tank will be called *volume deficit*. If the volume deficit in a tank is higher than a tolerance volume, then the operation policy is considered *infeasible*.

A solution is considered to be *invalid* when EPANET generates warnings during the simulation of a particular operational policy. For example, if during a particular simulation period, the system could not supply required volume of water with the specified minimum pressure at demand nodes, then the objectives of this solution cannot be evaluated and it is considered to be invalid.

3 Solution Methodology

The approach followed in this work considers a natural binary representation for operation policies. Each operation policy is a solution to the problem which is evaluated by EPANET. For each solution, EPANET calculates two objective values, that is, electricity cost and the total number of pump switches. The goal is the multi-objective minimisation of these values. This optimisation is performed by SPEA2. Additionally, EPANET calculates the total volume deficit and pressure deficiency of each solution. These values are used to establish the feasibility of a solution.

3.1 Multi-objective Optimisation

In multi-objective optimisation [21, 22] we do not have a single objective value, but a vector of objective values, that is, an *objective vector*. Given two objective vectors u and v, $u \neq v$, we say that u *dominates* v if u is not worse than v for each objective value and better for at least one objective. When neither u dominates v nor vice versa, we say that the two objective vectors are *nondominated*. Since each solution represents an objective vector, we use the same terminology among solutions. Therefore, given a set of solutions, we can use the dominance relation among their objective vectors to define a subset of solutions which are not dominated by any other solution of that set. This subset is called a *Pareto set* and by definition its elements are nondominated. The elements of a Pareto set define implicitly a partition in the objective space between the region dominated by them and the region not dominated by them. Thus, it is often called *Pareto frontier* or *Pareto surface* in the literature.

In our multi-objective approach to the pump scheduling problem, each operation policy is a solution which represents an objective vector formed by the electricity cost and the number of pump switches of that operation policy. The goal is to find, or at least approximate to, the *optimal Pareto set* of operation policies, that is, the set of feasible operation policies such that any other feasible operation policy has a higher value for the electricity cost or the total number of pump switches. Therefore, the outcome of our optimisation procedure will be a Pareto set of feasible operation policies.

3.2 Bit/Binary Representation

In this study, only fixed speed pumps are considered. Therefore, as shown in Fig. 2, for each pump during a certain time interval, the operation policy can be represented by one bit of a string. The pump is off during that time interval if the bit's value is zero, and the pump is operating at fixed speed if the value is one. The number of pump switches is the number of $\boxed{0\ 1}$ sequences, plus one if the scheduling starts with $\boxed{1}$ and ends with $\boxed{0}$. Given N pumps and T time intervals, the number of possible solutions is $2^{N \cdot T}$ and the maximum number of switches per pump is $T/2$.

	t_0	t_1	t_2	t_3	t_j	t_T
P_i	$0/1$	$0/1$	$0/1$	$0/1$	$\dots$	$0/1$

Figure 2: Binary representation $(0/1)$ for each pump i and a number T of fixed time intervals t_j.

In the present work, we consider 24 fixed time intervals of 1 hour. Then, for $T = 24$ and $N = 3$ the number of possible solutions is 4.72×10^{21} and the maximum number of pump switches is 36.

3.3 SPEA2

The optimiser used in the present work is the second version of the Strength Pareto Evolutionary Algorithm (SPEA2) [15]. The main features of SPEA2 are: (*i*) the fitness of a solution depends on the strength of the solutions by which it is dominated, where the strength of a solution is defined as the number of other solutions in the current population that it dominates; (*ii*) the ties of solutions with the same fitness are broken by a nearest neighbour density estimation technique; (*iii*) the size of the archive of nondominated solutions is a fixed value α, when the actual number of nondominated solutions is lower than this value the archive is filled with dominated solutions and when the actual number of nondominated solutions exceeds α, some of them are discarded by a truncation operator which preserves boundary solutions.

The algorithm schema of SPEA2 as implemented in this work can be summarised as follows. Firstly, an initial population is generated and the archive starts empty. Secondly, the algorithm calculates the fitness of solutions in the current population and all nondominated solutions are added to the archive. If the size of the archive becomes larger than α, the solution which has the minimum distance to another solution (according to the truncation operator) is discarded until archive size is exactly α. In case of the number of nondominated solutions is less than α, the dominated solution with the minimum fitness value is added to the archive until there are α solutions in the archive. Next, a number of solutions are selected as parents using binary tournament selection with replacement. Finally, recombination is applied to parents in order to generate a number of offspring solutions, which become a new population that must be evaluated and merged into the archive. More details on SPEA2 can be found in the original publication [15].

Because the different objective values considered in this work, i.e., electricity cost and number of pump switches, are not comparable, we normalise the distance between two solutions s_i, s_j with respect to objective f_k as:

$$\frac{(f_k(s_i) - f_k(s_j))^2}{(f_k^{\max} - f_k^{\min})^2} \quad (1)$$

where $f_k^{\max}$ and $f_k^{\min}$ are known for each particular objective. The maximum electrical cost corresponds to that schedule where all pumps are operating during the whole simulation period, while the minimum electrical cost is zero. For the total number of pump switches, the maximum value when $T = 24$ hours and $N = 3$ pumps is 36, while the minimum value is always zero.

3.4 Constraint Handling

We handle invalid and infeasible solutions (defined in Section 2.2) following a methodology proposed by Deb and Jain [17], where solutions are partially ordered depending on their feasibility. Concretely, we augment the dominance criteria with the following rules:

- Any invalid solution is dominated by any valid (feasible or infeasible) solution. For two invalid solutions, the one which has lower number of pressure violations during the simulation dominates the other.
- Given two valid solutions, the one with the lower total volume deficit dominates the other. Since the total volume deficit of any feasible solution is always zero, then any feasible solution dominates any infeasible solution.
- Given two valid solutions with equal total volume deficit, the normal dominance criteria between their objective values is applied. That is, one solution dominates other if the electricity cost and the number of pump switches of the first one are not higher than the values corresponding to the second solution, and at least one of these values is lower than the respective value in the second solution.

4 Experiments

The method to generate the initial population and the recombination operator are problem dependent and the alternatives studied in this work are described in the following paragraphs.

With regard to the initial population, in this work we consider two simple, straightforward methods: either the initial population is randomly generated or it is generated from mutations of a particular solution. The mutation procedure changes the value of a random number in $[1, 18]$ of positions per pump. We have tested three different solutions as the solution which is mutated: the *empty solution*, which is the solution where all pumps are off during the whole simulation, that is, all positions of the binary string have value 0; the *complete solution*, which is the solution where all pumps are on during the whole simulation, that is, all positions of the binary string are 1; and a *custom solution* which is known to be feasible. The rationale for the use of this custom solution is that a real network is already working using a feasible operation policy, which is not an optimal solution but has more quality than the average quality of a randomly generated solution.

We tested three types of recombination: one-point crossover, uniform crossover, and a deterministic (turn-based) uniform crossover.

The *one-point crossover* creates one offspring solution by joining a part of the first parent solution from the first position to a crossover point with another part of the second parent from the crossover point to the last position. The crossover point can occur with equal probability between any two adjacent positions.

In *uniform crossover*, the value of each position in the offspring solution is produced by randomly selecting, with equal probability, the value at the same position of one of the parents. Thus, those positions with the same value in both parents will keep that value in the offspring solution. As an alternative, we also tested a *deterministic uniform crossover*, which keeps the value of those positions with the same value in both parents, but assigns alternately the value of each parent for those positions where the values of each parent differ.

Our implementation of SPEA2 is based on original `C` source code[1] from the PISA project [23] but with significant modifications to serve our purposes. We also modified EPANET Toolkit version 2.00.10 but maintaining backward compatibility (under some assumptions).

The test instance shown in Figure 1 is used for studying the various alternatives proposed here. In this instance the demand charge is taken to be zero and the water available at the reservoir is assumed to be infinite. The electricity cost is divided into two periods with a peak electricity tariff period from 7 am to 12 am and a off-peak tariff from 12 am to 7 am. The demand pattern contains two peaks at 7 am and 6 pm. More details about the test instance are provided by Van Zyl et al. [8]

The custom solution used to generate the initial population has an electricity cost of 370.47, a total number of pump switches of 4, a volume deficit for tank A of -0.41% and for tank B of -0.19%, where the negative deficits mean that there is more water at the end than at the start of the simulation.

The volume deficit tolerated per tank was 5%. The archive size of SPEA2 was $\alpha = 200$. The number of solutions selected as parents, the number of offspring solutions and the number of initial solutions were 50. As in the state-of-the-art algorithm for this test instance [8], we ran each experiment for 6000 function evaluations, that is, 6000 calls to the EPANET simulator. Finally, we performed 30 repetitions of each configuration.

Experiments were ran on a Pentium 4 (2.80 GHz) with 1 GB RAM using GNU/Linux (Ubuntu).

[1] Source code available at `http://www.tik.ee.ethz.ch/pisa/selectors/spea2/spea2.html`

5 Results

The attainment function [18] represents the probability of obtaining an arbitrary goal in the objective space during a single run of an arbitrary algorithm. This attainment function can be estimated using data collected from several runs of the particular algorithm. For example, the *median attainment surface* contains objective vectors with an empirical frequency of 50% of being attained. The median attainment surface is a Pareto set because all the objective vectors are nondominated. Therefore, these objective vectors can be connected by a line (or a surface when the number of objectives is larger than two) which defines the partition of the objective space dominated by them. Similarly, the *best attainment surface* connects objective vectors attained by at least one of the runs carried out, and the objective vectors in the *worst attainment surface* were attained in all the runs carried out

We must remark that the best, median and worst attainment surfaces only describe the distribution of the outcomes in terms of location. However, they do not address the dependence structure within each outcome, and thus, they do not show the frequency of two objective vectors being attained in the same run [24].

In our experiments we found that uniform crossover obtained always better results than one-point crossover. Also, results obtained by the deterministic uniform crossover based on alternative turns were slightly worse than results produced by uniform crossover. Therefore, in the following we restrict to the results obtained when using the uniform crossover.

Figure 3 shows the best, median and worst attainment surfaces of the 30 repetitions ran for each of the four methods of generating the initial population and using uniform crossover. For reference, the average solution obtained by the single objective state-of-the-art algorithm for this instance [8] is denoted by the symbol "$\times$", and it has an electricity cost of 348.58 and a number of pump switches of 4.29.

Comparing the top row with the bottom row of Fig. 3, the attainment surfaces obtained using an initial population generated randomly or from a custom solution are better than the attainment surfaces obtained when the initial population was generated by mutations from the empty or from the complete solution. Particularly, the median attainment surfaces corresponding to an initial population generated randomly or from a custom solution (top row) dominate the average solution obtained by the single objective state-of-the-art algorithm.

When the population is generated from mutations of a custom solution (top right), the best, median and worst attainment surfaces are closer to each other compared to the attainment surfaces obtained when the initial population is generated randomly (top left). Moreover, the worst attainment surface in the top right plot is better than the one corresponding to the top left plot. On the other hand, the best attainment surface shows the opposite result, that is, the best result of 30 trials is obtained with an initial population generated randomly. From these results we can conclude that an initial population generated from a custom solution provides robust results but lacks the diversity in solutions obtained when using an initial population generated randomly. This diversity allows us to obtain better results in some of the runs but produces worse results in the worst case.

Finally, Table 1 shows the average computation time required by each run depending on the method used to generate the initial population and the recombination operator. Since no initialisation method or recombination operator is more algorithmically expensive than the others, the differences observed in computation time are only caused by the time required by EPANET for evaluating the solutions. One clear result is that quasi-complete solutions (right-most column), where the pumps are active most of the time intervals, produce longer simulation time and thus longer computation time. Nevertheless, the variability in computation time for the other configurations indicates that the computation time depends in a high manner on the simulation engine.

Recombination	**Initial population**			
	Custom	Random	Empty	Complete
One-point	76.2	107.6	70.6	1043.0
Uniform	75.2	238.9	121.8	945.2
Determ. Unif.	68.9	224.6	95.3	1011.0

Table 1: Average computation time in seconds.

6 Conclusions and Future Work

This paper shows the viability of a multi-objective approach for solving the Pump Scheduling problem, which allows the system operator to examine a range of Pareto-optimal solutions and choose one solution with regard to additional criteria. The importance of this has already been noticed by Ormsbee and Reddy [20].[2]

Although the use of multi-objective algorithms for the Pump Scheduling problem has been studied previously [12, 14], recent improvements on algorithms [15] and performance assessment methodologies [18, 24] have increased the applicability of multi-objective optimisers. We have applied these improvements in this paper.

Moreover, we have considered a disaggregated or dual-level methodology, that is, linking a multi-objective optimisation model (SPEA2) with a network simulation model (EPANET). This methodology, which has not been considered in previous multi-objective approaches to the Pump Scheduling problem, allows to consider complex network instances.

Additionally, we have used a feasibility handling technique designed for multi-objective optimisation and based on the dominance criteria [17] which replaces penalty functions and reparation mechanisms.

[2] The exact quote is "Indeed, not only is an *optimal* solution obtained, but all resulting feasible solutions are available for examinations by the system operator. As a result, the operator is provided with an increased flexibility with regard to selection of alternative solutions that may not be optimal from a purely cost-savings objective but may provide a superior solution based on additional more subjective operational considerations." Ormsbee and Reddy [20]

The use of a well-known multi-objective algorithm (SPEA2) and simple and straightforward initialisation and recombination methods produces solutions of a quality similar to results already published in the literature. In particular, using uniform crossover and an initial population generated randomly produces state-of-the-art results for this Water Distribution Network. If the initial population is generated from mutations of a feasible solution of a certain quality, the results are even better in the median and the worst case, but a randomly generated initial population produces eventually the best Pareto set of pump schedules.

Nevertheless, our results should encourage the study of more advanced techniques, as alternative representations to the binary string and hybridisation with local search methods. Furthermore, since a complete description of the test instance used in this work has been already published [8] and the simulation engine is available[3], the results obtained by different optimisation algorithms can be compared with the results provided in this work.

Although the binary representation is a natural representation for fixed speed pumps and fixed time intervals, it imposes the restriction that pumps can only start or stop at fixed time intervals. We are currently studying other representations which do not have such limitation, and thus, may allow more flexible schedules, leading to better results.

Finally, we have noted that computation time depends greatly on the simulation engine and on the type of solutions evaluated. Therefore, although real-world applications should take into account computation time, future developments on the simulation engine may completely change any conclusions relying on a computation time limit, and thus, the number of function evaluations is a more robust measure when comparing algorithms for the Pump Scheduling problem.

References

[1] V. Nitivattananon, E. C. Sadowski, and R. G. Quimpo, "Optimization of water supply system operation," *Journal of Water Resources Planning and Management, ASCE*, vol. 122, pp. 374–384, Sept. / Oct. 1996.

[2] G. McCormick and R. S. Powell, "A progressive mixed integer-programming method for pump scheduling," pp. 307–313.

[3] L. E. Ormsbee and K. E. Lansey, "Optimal control of water supply pumping systems," *Journal of Water Resources Planning and Management, ASCE*, vol. 120, no. 2, pp. 237–252, 1994.

[4] G. Mäckle, D. A. Savic, and G. A. Walters, "Application of genetic algorithms to pump scheduling for water supply," in *Genetic Algorithms in Engineering Systems: Innovations and Applications, GALESIA'95*, vol. 414, (London, UK), pp. 400–405, IEE Conference Publication, 1995.

[3]It is public-domain software that can be obtained at `http://www.epa.gov/ORD/NRMRL/wswrd/epanet.html`

[5] R. Atkinson, J. E. van Zyl, G. A. Walters, and D. A. Savic, "Genetic algorithm optimisation of level-controlled pumping station operation," in *Water network modelling for optimal design and management*, pp. 79–90, Centre for Water Systems, Exeter, U.K., 2000.

[6] P. F. Boulos, C. H. Orr, W. D. Schaetzen, J. G. Chatila, M. Moore, P. Hsiung, and D. Thomas, "Optimal pump operation of water distribution systems using genetic algorithms," in *AWWA Distribution System Symp.*, (Denver, USA), American Water Works Association, 2001.

[7] M. D. Kazantzis, A. R. Simpson, D. Kwong, and S. M. Tan, "A new methodology for optimizing the daily operations of a pumping plant," in *Proceedings of 2002 Conference on Water Resources Planning*, (Roanoke, USA), ASCE, May 2002.

[8] J. E. van Zyl, D. A. Savic, and G. A. Walters, "Operational optimization of water distribution systems using a hybrid genetic algorithm," *Journal of Water Resources Planning and Management, ASCE*, vol. 130, pp. 160–170, Mar. 2004.

[9] C. Wegley, M. Eusuff, and K. E. Lansey, "Determining pump operations using particle swarm optimization," in *Building Partnerships, Proceedings of the Joint Conference on Water Resources Engineering and Water Resources Planning and Management* (R. H. Hotchkiss and M. Glade, eds.), (Minneapolis, USA), 2000.

[10] F. E. Goldman and L. W. Mays, "The application of simulated annealing to the optimal operation of water systems," in *Proceedings of 26th Annual Water Resources Planning and Management Conference*, (Tempe, USA), ASCE, June 2000.

[11] G. McCormick and R. S. Powell, "Derivation of near-optimal pump schedules for water distribution by simulated annealing," *Journal of the Operational Research Society*, vol. 55, pp. 728–736, July 2004.

[12] D. A. Savic, G. A. Walters, and M. Schwab, "Multiobjective genetic algorithms for pump scheduling in water supply," in *Evolutionary Computing Workshop, AISB'97* (D. Corne and J. L. Shapiro, eds.), vol. 1305 of *Lecture Notes in Computer Science*, pp. 227–236, Springer-Verlag Berlin, 1997.

[13] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[14] A. Sotelo, C. von Lücken, and B. Barán, "Multiobjective evolutionary algorithms in pump scheduling optimisation," in *Proceedings of the Third International Conference on Engineering Computational Technology* (B. H. V. Topping and Z. Bittnar, eds.), Civil-Comp Press, Stirling, Scotland, 2002.

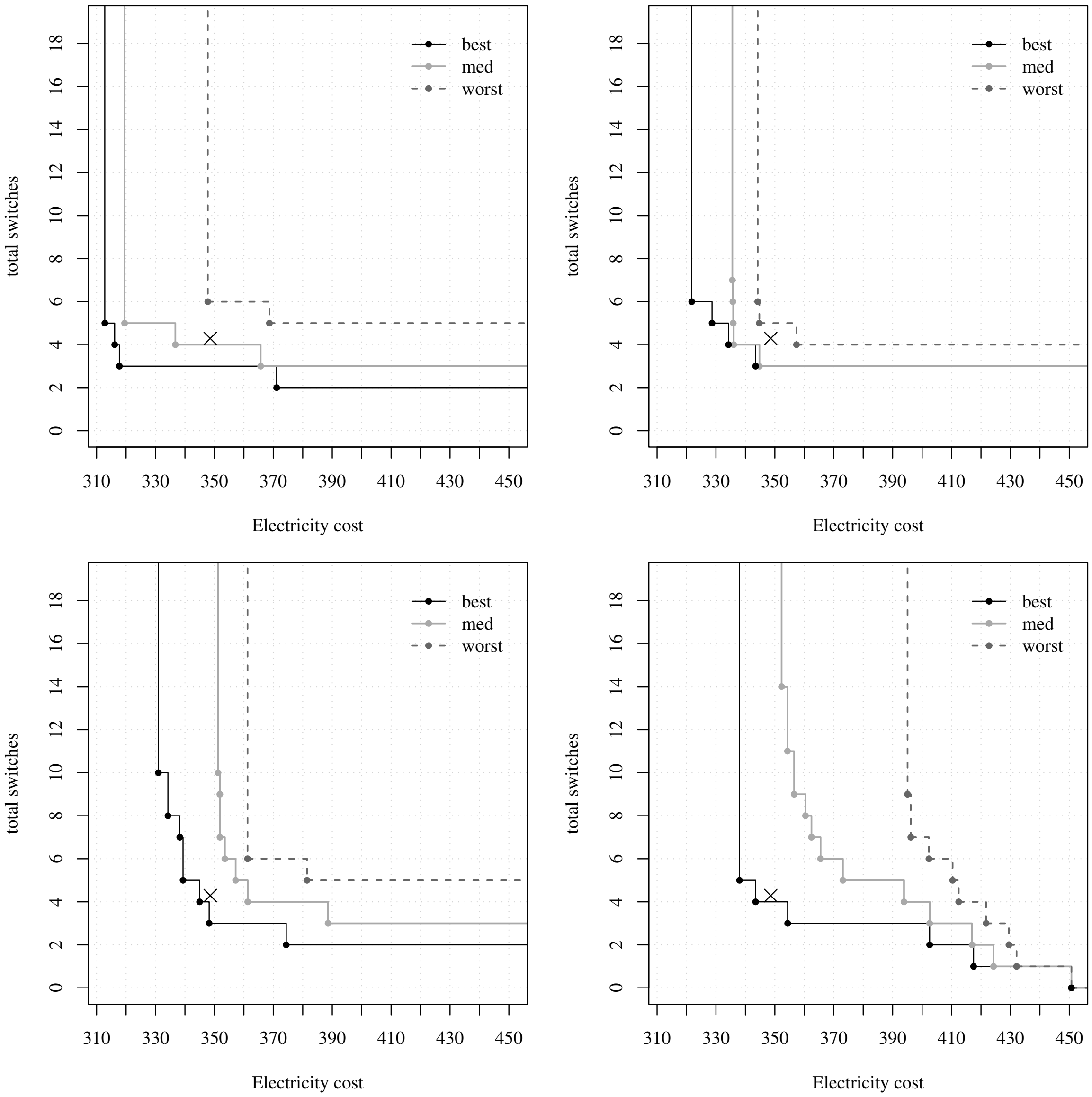

Figure 3: Best, median and worst attainment surfaces obtained by 30 repetitions of SPEA2 using uniform crossover and an initial population randomly generated (*top left*) and generated by mutations from a custom solution (*top right*), from the empty solution (*bottom left*) and from the complete solution (*bottom right*).

[15] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization,” in *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems. Proceedings of the EUROGEN2001 Conference* (K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papaliliou, and T. Fogarty, eds.), pp. 95–100, International Center for Numerical Methods in Engineering (CIMNE), 2002.

[16] L. A. Rossman, “The EPANET Programmer’s Toolkit for analysis of water distribution systems,” in *Proceedings of the Annual Water Resources Planning and Management Conference*, (Reston, USA), ASCE, 1999.

[17] K. Deb and S. Jain, “Multi-speed gearbox design using multi-objective evolutionary algorithms,” Tech. Rep. 2002001, KanGAL, Feb. 2002.

[18] V. Grunert da Fonseca, C. M. Fonseca, and A. O. Hall, “Inferential performance assessment of stochastic optimisers and the attainment function,” in *First International Conference on Evolutionary Multi-Criterion Optimization* (E. Zitzler, K. Deb, L. Thiele, C. A. Coello, and D. Corne, eds.), vol. 1993 of *Lecture Notes in Computer Science*, pp. 213–225, Springer Verlag, 2001.

[19] K. E. Lansey and K. Awumah, “Optimal pump operations considering pump switches,” *Journal of Water Resources Planning and Management, ASCE*, vol. 120, pp. 17–35, Jan. / Feb. 1994.

[20] L. E. Ormsbee and S. L. Reddy, “Nonlinear heuristic for pump operations,” *Journal of Water Resources Planning and Management, ASCE*, vol. 121, pp. 302–309, July / Aug. 1995.

[21] M. Ehrgott, *Multicriteria optimization*, vol. 491 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, Berlin, 2000.

[22] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, Inc., 2001.

[23] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler, “PISA – a platform and programming language independent interface for search algorithms,” in *Evolutionary Multi-Criterion Optimization (EMO 2004)* (C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, eds.), vol. 2632 of *Lecture Notes in Computer Science*, pp. 494–508, Springer Verlag, Berlin, Germany, 2003.

[24] C. M. Fonseca, V. Grunert da Fonseca, and L. Paquete, “Exploring the performance of stochastic multiobjective optimisers with the second-order attainment function,” in *Proceedings of the Evolutionary Multicriterion Optimization (EMO 2005)* (E. Zitzler, C. C. Coello, and A. H. Aguirre, eds.), vol. 3410 of *Lecture Notes in Computer Sience*, pp. 250–264, Springer Verlag, 2005.