Human-Augmented Hyper-heuristics for Water Distribution Optimisation

W. B. Yates, M. B. Johns, H. A. Mahmoud, E. C. Keedwell

Computer Science, College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, EX4 4QF, UK

Abstract

In this work we explore the potential for human expertise to enhance optimisation performance through the execution of a hyper-heuristic algorithm tested on difficult real-world problems taken from the water industry. The novel combination of domain knowledge and hyper-heuristic learning raises the prospect of a powerful method of human-machine teaming to solve large and complex problems. The method is demonstrated on a set of water distribution network design and rehabilitation problems and the results show both the efficacy of the method and the useful extra insight possible with human-augmented hyper-heuristic approaches.

Keywords: Hyper-heurstics, machine learning, water distribution network optimisation.

1. Introduction

Metaheuristic optimisation techniques such as evolutionary algorithms (EAs) continue to enjoy significant success in delivering efficient and improved solutions to problems across a wide variety of sectors, such as operations research, engineering, chemistry, and bioinformatics [6]. An evolutionary algorithm consists of a set of possible solutions to a given problem, a set of low level heuristic operators that select and modify the solutions, and a set of objective functions [12]. Such algorithms treat a problem as a black box, and iteratively apply the operators to improve the solutions until a convergence criteria defined on the objective functions is met. Typically, the operators are generic, that is they make no use of problem specific knowledge. More recently, problem specific operators that encode domain expertise have been included in the optimisation process to produce better performing, but less generalisable algorithms [6] [5] [7]. These problem specific operators can be derived from known characteristics of the problem encoded as heuristics (e.g. first-fit or best-fit heuristics in bin packing and stock cutting), through the communication of human expertise from expert interviews, or through the direct involvement of the expert in the optimisation process, often described as “human-in-the-loop” or ”active learning” methods.

What characterises these approaches is that the expertise is usually included in the metaheuristic as a new perturbation or selection operator, or as a penalty term within the objective function. These methods often work better than their non-expert counterparts, but are relatively inflexible and are likely to perform poorly when applied to other problem domains. Selection hyper-heuristics [9] offer a new way in which to consider the incorporation of expertise through the use of online learning. This process determines which of a set of low-level heuristics to apply to a problem during its optimisation. Although the learning process incurs additional computational cost, it significantly increases algorithmic flexibility by allowing the algorithm to only use those heuristics that are effective for the current problem and state of the optimisation process.

In this paper, we explore the potential for the integration of domain expertise derived from human experts with the power and flexibility of adaptive optimisation techniques in the form of hyper-heuristics. Our approach is applied to the Water Distribution Network (WDN) optimisation problem. This problem is an NP-hard combinatorial problem with significant availability of industrial expertise, and a wide range of benchmark and real-world problem instances. Three domain-specific heuristics are considered, two are generated from direct communication with experts resulting in "pipe smoothing" and "bottleneck elimination" heuristics, while the third operator is derived from expert-model interactions to create a "human derived heuristic".

The experiments presented here demonstrate the improved performance that can be achieved by combining traditional search, and domain specific expertise, within the sequence-based hyper-heuristic learning approach of SSHH introduced in [23]. It also demonstrates the additional insight into the optimisation process that can be derived from such methods by exploring the effectiveness of the low level heuristics selected during the search. This analysis describes which heuristics are most useful and at which point in the search process, helping to better understand the process of human-machine teaming for use in optimisation problems.

The remainder of the manuscript is organised as follows. Section 2 provides an overview of previous research concerning the heuristic optimisation of water distribution networks. Section 3 presents the experimental methodology employed in this study, while Section 4 contains the results of five experiments which explore the performance characteristics of the three domain specific heuristics. Finally, Section 5 presents the conclusions.

2. Previous Work

The problem of optimising water distribution networks using metaheuristics has received significant attention in the literature which is summarised here, along with a description of the problem itself. Hyper-heuristics have not received the same level of attention in the literature, but the approaches that have been trialled are also summarised here along with a treatment of the use of domain expertise in search and optimisation.

Section 2.1 presents a description of the 12 WDN optimisation problems employed in this study, while Section 2.2 defines the objective functions to be minimised. Section 2.3 provides a summary of previous research concerning the application of metaheuristics and hyper-heuristics to WDN optimisation, while Section 2.4 describes previous research with interactive evolutionary methods.

2.1. Water Distribution Networks

The provision of clean, safe, drinking water to hundreds of thousands or millions of customers at a suitable pressure across cities and other urban areas is a significant challenge that is becoming more difficult due to climate change and increasing populations. Water distribution networks use pipes, pumps, tanks, valves and other infrastructure to deliver water from sources such as reservoirs and water treatment works to customers. These systems represent significant capital investment, and operational costs to provide services are high. Given these costs and the social requirement for the sustainable provision of sufficient volumes of clean water, the optimisation of water distribution network design is an established field of research. Typically, the network design problem is characterised as a discrete combinatorial optimisation problem where the decision variables are the diameters of the pipes in a network, and the objectives are to minimise the network's overall cost and maximise the networks hydraulic characteristics as far as possible. Feasible network designs must meet patterns of consumer demand (or loading) conditions, and maintain the minimum required head (pressure) throughout the network. Hydraulic characteristics have previously been expressed as a single measure such as the mean pressure or head deficit across the network. More recent studies have adopted composite measures such as network resilience [44] (which is employed here) that are comprised of a number of metrics of network performance. Network resilience is specified in more detail in Section 2.2. As a widely studied problem, there are a range of benchmark networks available, and in this paper we consider the 12 WDN problems taken from [44] to test our approach[1]. These represent a diverse range of problems, with significant ranges of size (e.g. small, medium, intermediate, large), network types (e.g. dendritic vs looped networks) and optimisation problem scenarios (e.g. least cost design vs rehabilitation). Table 1 summarises these problems and illustrates the range of loading conditions, water sources, decision variables, pipe diameter options, and problem sizes considered.

The problems differ from one another in a number of respects. Of the 12 problems, 11 are based on real world networks, while the TLN network [1] is an example of a hypothetical network. The TRN [19], BAK [26], NYT [36] and EXN [15] networks are expansion problems, where the task is to extend an existing network by modifying *some* of the pipes in the network. In addition to selecting pipe diameters such problems can sometimes make use of extra

[1]Implementations of the problems together with their best known Pareto fronts can be downloaded from: `https://emps.exeter.ac.uk/engineering/research/cws/resources/benchmarks/` under the Two-Objective Design/Resilience tab.

Table 1: The problem name, acronym, the number of loading conditions (LC), number of water sources (WS), number of decision variables (DV), number of pipe diameter options (PD), the search space size, and size classification for the water distribution network problems. For the TRN problem, three existing pipes have eight diameter options for duplication and the two extra options of cleaning or leaving alone.

Problem	Acronym	LC	WS	DV	PD	Search Space	Size
Two-Reservoir	TRN	3	2	8	8*	3.28×10^{7}	S
Two-Loop	TLN	1	1	8	14	1.48×10^{9}	S
BakRyan	BAK	1	1	9	11	2.36×10^{9}	S
New York	NYT	1	1	21	16	1.93×10^{25}	M
Blacksburg	BLA	1	1	23	14	2.30×10^{26}	M
Hanoi	HAN	1	1	34	6	2.87×10^{26}	M
GoYang	GOY	1	1	30	8	1.24×10^{27}	M
Fossolo	FOS	1	1	58	22	7.25×10^{77}	I
Pescara	PES	1	3	99	13	1.91×10^{110}	I
Modena	MOD	1	4	317	13	1.32×10^{353}	L
Balerma	BIN	1	4	454	10	1.00×10^{455}	L
Exeter	EXN	1	7	567	11	2.95×10^{590}	L

options such pipe cleaning, pipe duplication, or "leaving alone". The remaining problems TLN, BLA [37], HAN [17], GOY [25], FOS, PES, MOD [8], and BIN [32], are pure design problems where the diameters of any or all of the pipes in a network can be modified.

Each problem has minimum head pressure requirements for the demand nodes, expressed as constraints in the problem, and defined as an objective in the optimisation algorithm (see Equation 2 below). The BLA, FOS, PES, and MOD networks also have maximum pressure requirements, and upper bounds on water velocities in the pipes. The TRN network differs from the other problems in that it has three sets of loading conditions, while the BIN network, unlike a typical WDN, has a fixed level of water consumption across all demand nodes. There is not enough space here to specify explicitly all the constraints required for each network. The constraints can be found in the problem implementations[1].

Figure 1 shows example schematics of the NYT, FOS, MOD, and BIN distribution networks.

2.2. Objective Functions

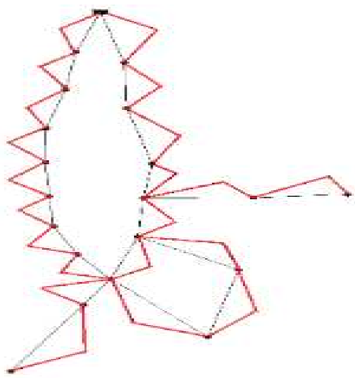
In this study, the water distribution network design problem is specified by three functions: the cost C, the head pressure deficit H, and the network's *resilience* I_n. The cost and head pressure deficit are to be minimised while the resilience is to be maximised.

The monetary cost is usually expressed in millions and is defined by

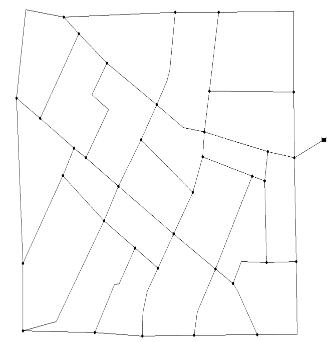
$$C = \sum_{i=1}^{np} U_c(D_i) L_i \tag{1}$$

Figure 1: The layout of the distribution network for the NYT, FOS, MOD, and BIN problems.

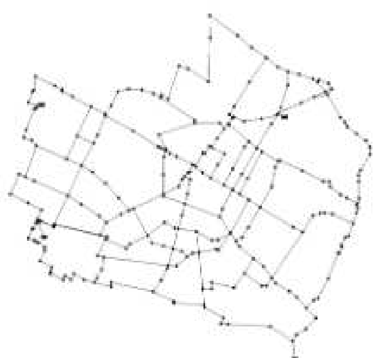
(a) NYT



(b) FOS



(c) MOD



(d) BIN

where U_c is the unit pipe cost which depends on the diameter D_i selected, and the length L_i of pipe $i = 1, \ldots, np$.

The head pressure deficit is defined by

$$H = \sum_{j=1}^{nn} \Big(\max(H_j - H_j^{\max},\, 0) + \max(H_j^{\text{req}} - H_j,\, 0) \Big) \tag{2}$$

where H_j is the actual head pressure, H_j^{req} is the minimum required head pressure, and $H_j^{\max}$ is the maximum required head pressure (if any) for each demand node $j = 1, \ldots, nn$.

A network resilience index measures the redundancy of a pipe network [29]. Maximising a resilience indicator can improve network reliability by reducing the occurrence of nonviable networks, and produces solutions that are more robust under pipe failure conditions [30]. There are many network resilience measures in the literature, and each has its own particular advantages and disadvantages [3]. In this paper, following [44], a network's resilience is defined by

$$I_n = \frac{\sum_{j=1}^{nn} C_j Q_j (H_j - H_j^{\mathrm{req}})}{\left(\sum_{k=1}^{nr} Q_k H_k + \sum_{i=1}^{npu} \frac{P_i}{\gamma}\right) - \sum_{j=1}^{nn} Q_j H_j^{\mathrm{req}}}. \tag{3}$$

where Q_j is the demand, nr is the number of reservoirs, Q_k is the discharge, and H_k is actual head of reservoir k, npu is the number of pumps, P_i is the power of pump i (if any), and γ is the specific weight of water. The term C_j is the *uniformity* which is defined by

$$C_j = \frac{\sum_{i=1}^{npj} D_i}{npj \max\{D_i\}} \tag{4}$$

where npj is the number of pipes connected to node j, and D_i is the diameter of pipe i connected to node j.

2.3. Heuristic Optimisation of WDN Problems

Metaheuristics [6] and hyper-heuristics [9] are general purpose, high level, heuristic methods that are used to solve computationally hard problems for which no known tractable algorithmic solutions exist. Typically such problems are presented as optimisation problems where the goal is to minimise an *objective function* defined on a space of solutions.

A number of heuristic methods have been successfully used to optimise water distribution networks, such as simulated annealing [10], shuffled frog leaping algorithms [13], harmony search [18], honey-bee mating optimisation [28], differential evolution [48], particle swarm optimisation [14], multi-objective evolutionary algorithms [44], ant colony optimisation [38], and hyper-heuristics [46].

Metaheuristics typically apply a small number of operators (e.g. mutation and crossover) in a fixed iterative pattern. Selection hyper-heuristics differ in that they use machine learning to learn the most appropriate sequences of heuristics to use for a given problem from a larger set of potential operators known as low level heuristics, and in doing so tailor themselves to a particular problem domain or problem instance. Such methods have proved effective when applied to a number of real world problems [9].

More broadly, hyper-heuristics may be classified as either *generation* or *selection* hyper-heuristics [9]. A generation hyper-heuristic generates new heuristics by discovery, or by modifying or combining existing low level heuristics. A selection hyper-heuristic, as described above, chooses heuristics from a given set of low level heuristics and applies them sequentially to optimise a particular problem and is the subject of this work.

2.4. Interactive Approaches

An interactive evolutionary algorithm is an optimisation algorithm that allows a human expert to become an active contributor to the optimisation process, and thereby provides a means to include qualitative expert knowledge within the search process. Expert interaction can take a number of forms. For example, the expert can manipulate the solutions directly, identify the best produced solutions, or more generally, rank a set of produced solutions. Direct manipulation can be seen as replacing the crossover and mutation operators with a manual perturbation, while selecting the best, or ranking solutions can be seen as replacing the selection operator. These methods have previously been used to include qualitative objectives within the optimisation process for a number of design applications such as groundwater monitoring design [2], complex river-reservoir system management [34], and groundwater inverse modelling [39]. However as in a traditional EA, in order to successfully optimise a given problem an interactive EA must execute thousands of iterations before the final solution or solution set is discovered [43, 47]. Interaction with this number of network designs is not feasible, leading to user fatigue, and poorer decisions being made within the interactive EA. This, in turn, may degrade the algorithm's search ability. Instead, as demonstrated in [21] a human-derived heuristic (HDH) can be generated by a machine learning algorithm from interaction data (collected over a number of problems and sessions) that can be used to mimic the expert interactions and prolong their useful period of interaction with the EA optimiser.

Every approach that seeks to embed human expertise into an EA must have a mechanism to learn from the expert. In previous studies this has been achieved through the expression of "rules of thumb" which are embedded in the EA as heuristics [22, 21]. In practice this is difficult to achieve as many decisions made by an expert will be based on intuition and "feel" rather than explicit rules. [41] proposes a methodology for capturing and integrating human engineering expertise into an EA by using an interactive visualisation and a decision tree based machine learning algorithm. The results demonstrate that the overall performance of the hybrid EA increases as a result of learning from these interactions. However, this method depends on training the decision tree using hydraulic data from the entire network, resulting in a very large number of input features. In [21] the authors address this issue by creating a smaller, more "generalisable" input schema for the HDHs based on the local features of a selected pipe. Here, the interaction data is recorded on, and then applied to, a particular network. Such an approach limits the potential for application of this method to larger WDNs because it is almost impossible for an expert to effectively optimise a large network through manual intervention, a process that is likely to lead to user fatigue. [27] expanded the work of [21] by developing generalisable transferable model features to enable the creation of "universal" heuristics from an expert's interaction on small WDNs which can then be applied to larger, unseen WDN problems.

3. Methodology

This section presents the experimental methodology employed throughout this study. Section 3.1 contains a description of the sequenced-based selection hyper-heuristic SSHH, and the low level-heuristics that are used to optimise the

WDN problems. Section 3.2 describes, in more detail, the bottleneck elimination and pipe smoothing domain specific heuristics, while Section 3.3 outlines the process used to generate a human-derived heuristic. Finally, Section 3.4 details the methodology used to visualise the effectiveness of the low level heuristics over an optimisation run.

3.1. A Sequenced-based, Selection Hyper-heuristic SSHH

The SSHH hyper-heuristic is a sequence-based selection hyper-heuristic with online learning [23, 46]. It uses a hidden Markov model (HMM) [31] to generate sequences of heuristic selections, their parameters, and acceptance check decisions. The HMM consists of a set of hidden states, and four probability matrices: a state transition matrix to determine the probability of moving from one hidden state to another, a heuristic emission matrix to determine which low level heuristic to apply, a parameter emission matrix to determine the parameter for a heuristic, and an acceptance check emission matrix to determine whether a solution should be evaluated and checked for acceptance or not.

In the absence of *a priori* knowledge regarding a given problem, the number of hidden states is set to be the number of low level heuristics in the domain, and the state transition, parameter, and acceptance matrices are set to be equiprobable. The low level heuristic emission matrix is set to the identity matrix. This ensures that, initially, each equiprobable hidden state emits a single low level heuristic together with an equiprobable choice of heuristic parameter and acceptance check decision.

The SSHH hyper-heuristic has online learning capabilities. During optimisation, SSHH keeps a record of the heuristic selections, parameters, and acceptance checks produced by the HMM. If, following an acceptance check, a new, best solution is found, the online learning algorithm steps through the record, increasing the probabilities of the accepted state transitions and emissions that led to the new minima. Thus the probability that the HMM produces the sequence of heuristic selections and emissions contained in the record is now higher. After the acceptance check, the record is erased and the optimisation process is resumed.

The SSHH hyper-heuristic employs a number of low level heuristics (LLHs). The seven low level heuristics used in this work are as follows:

- $\mathtt{M}_0$ - Mutate - change one pipe diameter randomly
- $\mathtt{S}_1$ - Shuffle - swap several pipe diameters at random, where the number of diameters to be exchanged is a parameter that takes a value in the range [1,5],
- $\mathtt{X}_2$ - Crossover - the two-point crossover of two vectors of network pipe diameters,
- $\mathtt{C}_3$ - Creep mutate - increase or decrease a randomly selected pipe diameter by one pipe size,
- $\mathtt{B}_4$ - Bottleneck elimination - increase diameter of a pipe which is downstream from a randomly selected junction in deficit,

$\mathtt{P}_5$ - Pipe smoothing - modify diameter of down stream pipe to ensure a "smooth" transition from large to small diameters, and

$\mathtt{H}_6$ - Human Derived Heuristic (HDH) - a heuristic derived from human expertise via a machine learning algorithm.

The generic heuristics $\mathtt{M}_0$, $\mathtt{S}_1$, and $\mathtt{X}_2$ are taken from [24] and [46], $\mathtt{C}_3$ is taken from [11]. Here, the term generic means that these heuristic operators make no use of domain specific knowledge. In contrast, the remaining three heuristics all make use of domain specific knowledge. The bottleneck elimination and pipe smoothing knowledge based heuristics $\mathtt{B}_4$ and $\mathtt{P}_5$ are taken from [20], and the human derived heuristic $\mathtt{H}_6$ is taken from [42].

A flowchart illustrating the operation of the SSHH optimiser, and its interactions with the low level heuristics is shown in Figure 2. The *EPANET2*[2] software library [35] is used to run all the hydraulic simulations necessary to obtain the objective function values defined in Section 2.2.

3.2. Domain Specific Heuristics

The bottleneck elimination and pipe smoothing heuristics are based on expert knowledge obtained from water engineers. These are described in more detail in [20] and readers are directed to this publication for more information as only a brief summary is presented here.

The bottleneck elimination heuristic is designed to identify situations where an undersized pipe introduces a hydraulic head (pressure) deficit in the network. The operator achieves this by identifying nodes in deficit, and increasing the size of the pipe diameter that is contributing to the deficit. Nodes are selected at random using a roulette wheel biased towards those nodes in deficit, and once selected, the heuristic "crawls" the network upstream until it finds the offending pipe, this is then randomly mutated to a larger diameter.

The concept that underpins the pipe smoothing heuristic is that in almost all gravity fed pipe networks, the diameters of the pipes are seen to smoothly transition from larger to smaller diameters with the direction of flow. This indeed makes sense from a hydraulic performance perspective, larger trunk mains are required close to water sources and smaller distribution pipes can be placed towards the extremities of the network. In practice this can be achieved through the application of a simple rule. For any given junction, the sum of the diameters of downstream pipes is always less than or equal to the sum of the diameters of the pipe directly upstream. This heuristic implements this by first allowing random selection of a pipe to mutate but applying an upper bound to random mutations as the (sum of) the diameter(s) upstream.

These two heuristics faithfully replicate operations that expert engineers perform when presented with a similar network state. As will be shown later, the effectiveness of each heuristic is dependent on the individual network character-

[2]The EPANET software (build 2.00.12) and manual can be downloaded from: `https://www.epa.gov/water-research/epanet/`

istics, and the extent of optimisation progress for example, due to the number of loops and bottlenecks within the networks.

3.3. Human Derived Heuristic Capture

The HOWS interactive visualisation system framework [42] is used to capture expertise for the interactive optimisation heuristic (HDH). In this system, the human expert is provided with a randomly generated solution to the required WDN problem and asked to optimise this manually by reducing design costs (C) and maximising resilience index (I_n) whilst satisfying the minimum pressure head constraints. The expert interacts by selecting the nodes and pipes in a 3D model of the network that incorporates visualisation of hydraulic data. When selecting a node, basic hydraulic and problem specific information such as pressure and head deficit is displayed to the expert to aid in the decision making process. When selecting a pipe (as a decision variable), relevant hydraulic information is presented along with a dialog displaying a list of commercially available diameters and their respective cost is shown. In addition, the client employs a "look-ahead" system which evaluates the effect that changing the selected pipe would have on the solution's objectives and constraints for each available diameter. This information is displayed to the expert in a visually intuitive manner aiding the decision making process when selecting a new diameter. After the expert changes the diameter of a pipe, the change is sent to a server which logs the change and runs a hydraulic simulation on the new network configuration and computes the new objective function and constraint values. This information is then sent to the client which updates the network visualisation to reflect the recent intervention, and allowing further intervention if required.

Experiments are conducted on the data provided from the interactions of five water engineers from the Centre for Water Systems[3] on the TLN, HAN, BLA, FOS, and PES problems (see Table 1). Each engineer was tasked to manually optimise each network through minimising the cost, maximising the resilience index, and satisfying the head deficit constraints to the best of their ability from which the interactions are recorded. Each time an engineer changes the diameter of a pipe the server logs the state of the network, these interaction logs can then be used to train machine learning models to predict an engineer's behaviour given a selected pipe and network state.

The engineer is allowed to change the diameter of any pipe in the network repeatedly until they find the suitable diameter. This is acceptable as long as the engineer makes the decision based on the local properties of the selected pipe. However, the engineers were not allowed to randomly interact, i.e. randomly change the diameter for pipes without considering local pipe or network conditions until the desired value for objective function achieved. In this way, the number of bad decisions can be minimised. This, in turn, enables the machine-learning algorithm to find patterns in the interaction data even if some

[3]University of Exeter - `http://emps.exeter.ac.uk/engineering/research/cws/`

interactions have an instantaneous negative effect. This data was then passed through a decision tree machine learning algorithm to replicate the experts' behaviour as far as possible. The tree is then used as a replacement for the mutation operator, selecting the most likely expert decision for a given network state and thus creating the human-derived heuristic (HDH).

3.4. Visualising Heuristic Effectiveness

The effectiveness of a low level heuristic can be estimated in a number of ways (see for example [16, 40, 45]). In this study, heuristic effectiveness is measured by the *contribution* of the heuristic to the optimisation process, which is defined to be the proportion of the decrease (or increase) in the objective function value due to that heuristic.

It has long been hypothesised that some heuristics are better suited to certain stages or phases of the optimisation process than others, although this is only now starting to be proved in the field of hyper-heuristics. For example, the work of [33] on workforce scheduling, and that of [40] on course timetabling, demonstrates that some low level heuristics which are ineffective at the start of the search process prove to be highly effective at the end, and *vice versa*, while others heuristics are mainly used at the beginning, middle or end of the process.

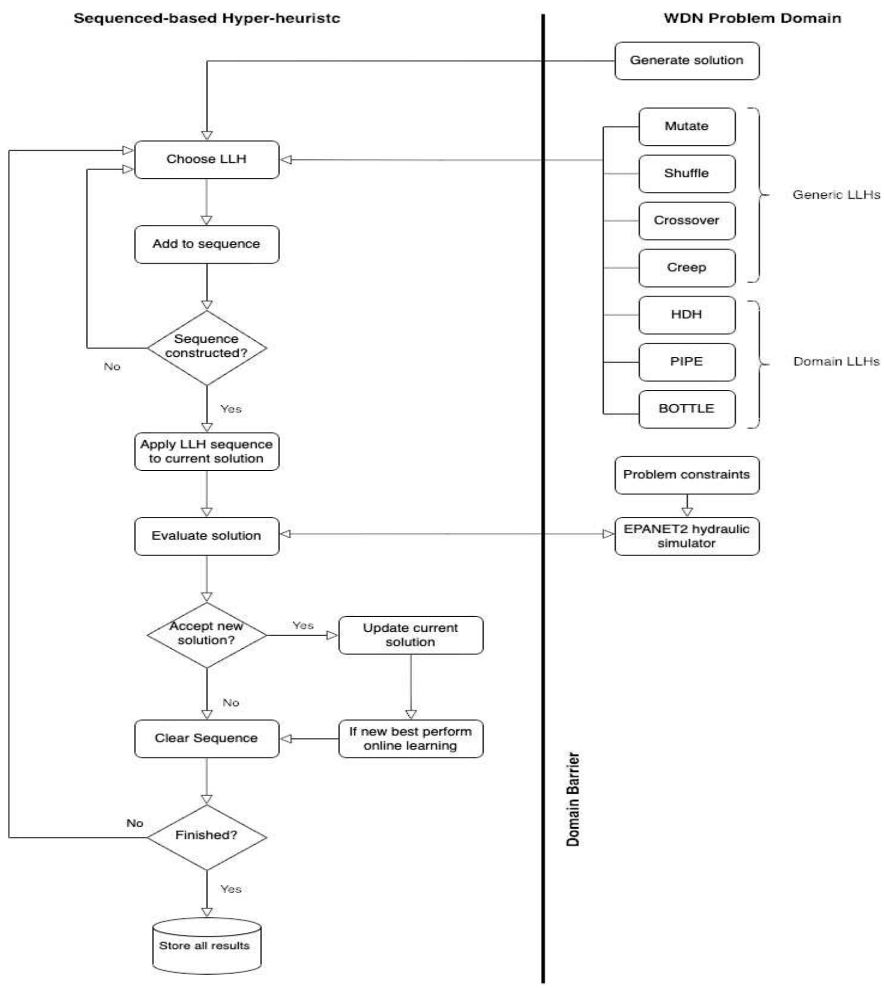
This section introduces a method for visualising the differences in heuristic effectiveness that occur during distinct periods in an optimisation run.

Consider a sequence of low level heuristics selections, and the associated objective function values generated when optimising a given problem. Let the *current objective function* value for a low level heuristic h_t be o_t, where o_t is the objective function value at iteration t, *before* applying the low level heuristic h_t to generate a new solution at iteration $t+1$. The set $\{h_t \mid o_t > P^r_{90}\}$ consists of all those instances of a heuristic h_t which have current objective function values greater than P^r_{90} where P^r_{90} is the 90th percentile. This set contains the 10% of heuristic instances with the highest current objective function values. The set $\{h_t \mid o_t < P^r_{10}\}$ contains the 10% of heuristics with the lowest current objective function values. These sets contain heuristic occurrences that occur at the "beginning" of the optimisation process, when objective function values are relatively high, and at the "end" of the optimisation process, when objective function values are relatively low, although in the absence of elitism in the optimisation process the relationship between time and objective function value is not necessarily linear. Calculating the percentile values over the runs or sequences of all 12 problems can lead to heuristic occurrences from a few problems dominating the sets; those that produce very high or very low objective function values. Here the percentiles P^r are calculated locally over the objective function values of *each* run $r = 1, \ldots, 40$ of each problem. This ensures that heuristic instances from all the problems are included in the sets. The heuristic instances can be further separated by their current objective function values into 10 sets

$$P_{10} = [P^r_{90}, P^r_{100}],\ P_9 = [P^r_{80}, P^r_{90}], \ldots, P_1 = [P^r_0, P^r_{10}].$$

By calculating the contribution of each low level heuristic for each run and each percentile P_i it is possible to visualise the relative effectiveness of each heuristic during the optimisation process.

Figure 2: A flowchart illustrating the operation of the SSHH optimiser, and its interactions with the low level heuristics for a given WDN problem.



4. Results

The SSHH hyper-heuristic is evaluated on the 12 water distribution network problems with six distinct combinations or parameterisations of the low level heuristics described in Section 3.

The BASE parameterisation consists of the three generic heuristics mutate $\mathtt{M_0}$, shuffle $\mathtt{S_1}$, and crossover $\mathtt{X_2}$, and provides a baseline for comparing the results of the other parameterisations. The HDH parameterisation consists of the three generic heuristics in BASE and the HDH heuristic $\mathtt{H_6}$. Similarly, the CREEP, BOTTLE, and PIPE parameterisations consists of the three BASE heuristics and the heuristics $\mathtt{C_3}$, $\mathtt{B_4}$, and $\mathtt{P_5}$, respectively. Finally, the ALL parameterisation consists of all seven low level heuristics. The objective is to investigate the changes in optimisation performance that are due to each of these heuristics.

The hyper-heuristic parameterisations are executed 40 times on each WDN problem. The number of iterations used by SSHH varies with the problem size. Specifically, SSHH is executed for 10,000, 20,000, 50,000, and 100,000 iterations for the small, medium, intermediate and large problems, respectively (see Table 1). The number of iterations for each size were chosen for computational feasibility. The number of 40 runs was chosen so as to ensure that robust statistics could be calculated for each problem instance. Only solutions that satisfy all head (and velocity) constraints are considered feasible. However, if the constraint violations are small, and spread evenly across a network, the solution can be viewed as semi-viable, or approaching viability. In this study, solutions that have a head pressure violation of less than 10m are included in the results.

4.1. Hyper-volumes

Table 2 shows the mean hyper-volumes of the approximate Pareto fronts generated by each parameterisations on the WDN problems.

Table 2: The mean hyper-volume for each hyper-heuristic parameterisation, calculated over 40 runs for each of the WDN problems. The best hyper-volumes are shown in bold.

Prob.	BASE	BOTTLE	HDH	CREEP	PIPE	ALL
TRN	**20.3319**	20.0035	19.3794	20.0464	20.1682	19.6305
TLN	33.2099	34.4958	31.7218	33.9391	**34.8509**	32.7324
BAK	8.1225	**8.3706**	8.0622	8.1137	8.2539	8.3027
NYT	1064.4150	**1082.0670**	1060.7091	1063.9472	1050.2112	1069.6833
BLA	11.4369	11.4117	11.5216	11.2402	11.5034	**11.5995**
HAN	12.6709	13.9609	14.6555	14.8846	**15.8017**	14.9885
GOY	7.4271	7.4426	7.4100	7.4772	7.4528	**7.5757**
FOS	22.9454	22.5862	23.1928	23.0257	22.7974	**23.7384**
PES	134.4144	132.5204	135.1278	**135.7379**	134.4918	134.1947
MOD	172.0159	172.1330	172.1351	177.2181	177.8247	**180.8407**
BIN	115.3284	119.4461	114.6075	121.7054	119.7385	**126.4296**
EXN	66.3723	62.1352	71.4233	67.3898	70.7002	**72.9487**

Clearly there are a range of results here, with each parameterisation achieving a best hyper-volume with the exception of HDH. This is surprising as the $\mathtt{H_6}$

heuristic has previously been found to significantly improve the performance of a standard evolutionary algorithm, although this might reflect a bias towards resilience by the experts, who appear to favour this objective over cost. The most successful parameterisation ALL involves the use of all heuristics which achieves the best hyper-volume on half of the problems tested. When the hyper-volumes for the BASE parameterisation are compared with ALL using a one tailed Wilcoxon signed-rank test, the improvements in optimisation for BLA, GOY and FOS are statistically significant with 99% confidence.

4.2. Rankings

The parameterisations are ranked according to the hyper-volumes of the resulting approximate Pareto fronts computed on each run of each problem. Table 3 shows the rankings based on the mean hyper-volume of each parameterisation calculated over the 40 runs of the 12 WDN problems.

Table 3: The ranks for each parameterisation, calculated using the mean hyper-volume averaged over 40 runs for each of the 12 WDN problems. The best ranks are shown in bold.

Name	TRN	TLN	BLA	NYT	BLA	HAN	GOY	FOS	PES	MOD	BIN	EXN
BASE	**0**	3	3	2	3	5	4	3	3	5	4	4
HDH	5	5	5	4	1	3	5	1	1	3	5	1
BOTTLE	3	1	**0**	**0**	4	4	3	5	5	4	3	5
CREEP	2	2	4	3	5	2	1	2	**0**	2	1	3
PIPE	1	**0**	2	5	2	**0**	2	4	2	1	2	2
ALL	4	4	1	1	**0**	1	**0**	**0**	4	**0**	**0**	**0**

The results of comparing the hyper-volumes of the best and worst ranking parameterisation for the PIPE, BOTTLE, CREEP, and HDH heuristics, on each problem, using the Wilcoxon signed-rank test are shown in Table 4.

Table 4: The sample median difference $\hat{d}$, the sample mean difference $\bar{d}$, the standard deviation SD, the p-value, and the 99% confidence interval for (Best – Worst). Statistically significant results are shown in bold.

Prob.	Best	Worst	$\hat{d}$	$\bar{d}$	SD	p-value	Conf. Int.
TRN	PIPE	HDH	0.7845	0.7888	1.1091	**0.0000**	$[\ 0.3812, \infty]$
TLN	PIPE	HDH	3.2500	3.1291	2.9786	**0.0000**	$[\ 2.0508, \infty]$
BAK	BOTTLE	HDH	0.1049	0.1008	0.2447	**0.0041**	$[\ 0.0140, \infty]$
NYT	BOTTLE	PIPE	25.9265	32.6821	110.9272	0.0607	$[-12.6545, \infty]$
BLA	HDH	CREEP	0.1802	0.3095	0.7029	**0.0000**	$[\ 0.0695, \infty]$
HAN	PIPE	BOTTLE	0.7524	0.6913	1.9709	0.0174	$[-0.0579, \infty]$
GOY	CREEP	HDH	0.0011	0.0672	0.2846	0.4973	$[-0.0606, \infty]$
FOS	HDH	BOTTLE	0.4928	0.5106	2.1616	**0.0053**	$[\ 0.0469, \infty]$
PES	CREEP	BOTTLE	1.7430	3.3598	8.9861	0.0155	$[-0.1640, \infty]$
MOD	PIPE	BOTTLE	4.8347	5.5652	16.4422	**0.0000**	$[\ 1.5070, \infty]$
BIN	CREEP	HDH	3.9338	3.8200	13.3555	**0.0000**	$[\ 2.4730, \infty]$
EXE	HDH	BOTTLE	0.6510	6.3135	27.9935	0.3136	$[-3.2826, \infty]$

These results demonstrate that the differences in performance between the individual domain heuristic parameterisations is statistically significant, with 99% confidence, in seven out of the 12 problems.

Table 5: The total rank for each parameterisation, calculated using the mean hyper-volume averaged over 40 runs for each of the 12 WDN problems. The best rank is shown in bold.

Name	BASE	HDH	BOTTLE	CREEP	PIPE	ALL
Total	39	39	37	27	23	**19**

Table 5 shows the total rankings over all 12 problems. The results show that the pipe smoother heuristic $\mathtt{P}_5$ produces the best individual improvement in optimisation performance, closely followed by the creep operator $\mathtt{C}_3$, while BASE, predictably, produces the worst. The differences between the hyper-volumes generated by each parameterisation, over all 12 problems, are not statistically significant. This is because the performance of the parameterisations varies considerably by problem. It is interesting to note that HDH has the most volatile performance, achieving good performance on some problems and very poor performance on others. This behaviour is further investigated in later sections. Overall, the combination of all heuristics gives the best performance, a result that is perhaps expected given that this adds additional flexibility for SSHH's online learning algorithm to exploit.

In order to test the results presented above, the parameterisations are also ranked using the mean *inverted generational plus* distance IGD+ [4] between the generated approximate Pareto front, and the best known Pareto front for each problem[1].

Table 6: The total rank for each hyper-heuristic parameterisation, calculated using IGD+ averaged over 40 runs for each of the 12 WDN problems. The best rank is shown in bold.

Name	BASE	HDH	PIPE	BOTTLE	CREEP	ALL
Total	40	34	34	29	26	**17**

The results, shown in Table 6, demonstrate that the overall rank order is broadly similar; the main difference being that PIPE has dropped from second best to forth (tied with HDH). The data used to generate this ranking are presented in Appendix A.

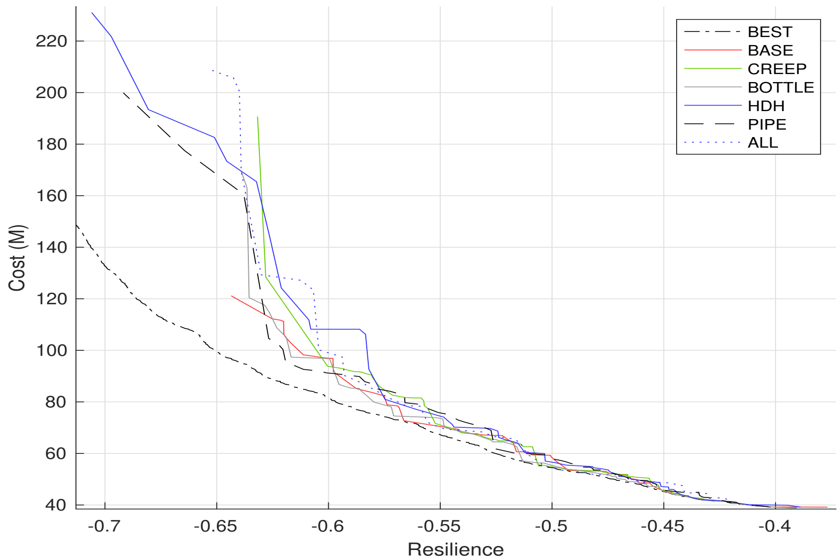
4.3. Approximate Pareto Fronts

Figures 3 and 4 show the approximate Pareto fronts for the NYT, FOS, MOD, and BIN problems which correspond to the best ranked performance by the BOTTLE, HDH, PIPE, and CREEP parameterisations. Note that in these figures the preferred direction of criteria values is "less than" and so the negative resilience, which is to be maximised, is used.

The approximate Pareto fronts were constructed from the population of solutions with *no* constraint violations, produced by all 40 runs on each WDN problem. The best known Pareto fronts BEST are taken from [44] and were generated using five multi-objective evolutionary algorithms, executed for 100,000, 600,000, 1,000,000, and 2,000,000 iterations for the small, medium, intermediate and large problems, respectively (see Table 1).

Figure 3: The approximate population Pareto front generated by each hyper-heuristic parameterisation for the NYT and FOS problems. BEST denotes the current best known solutions.

(a) NYT - BOTTLE is the best parameterisation.



(b) FOS - HDN is the best parameterisation.

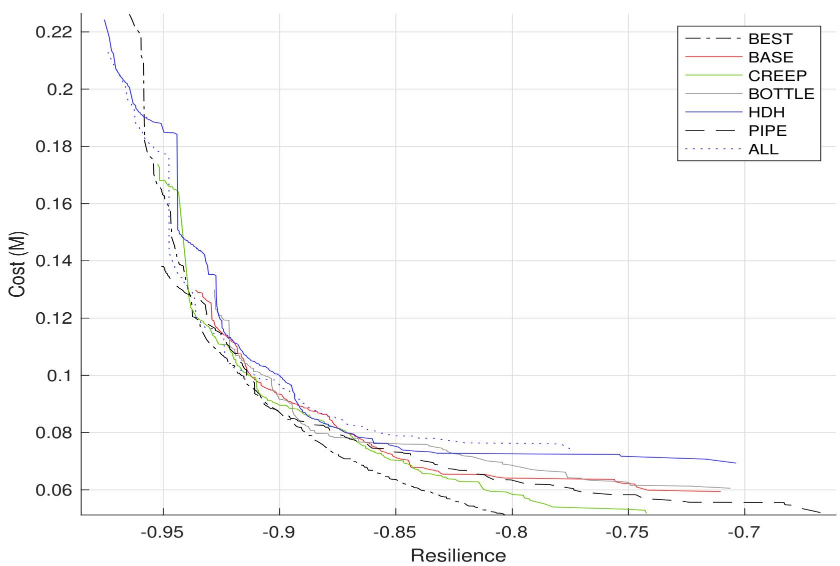
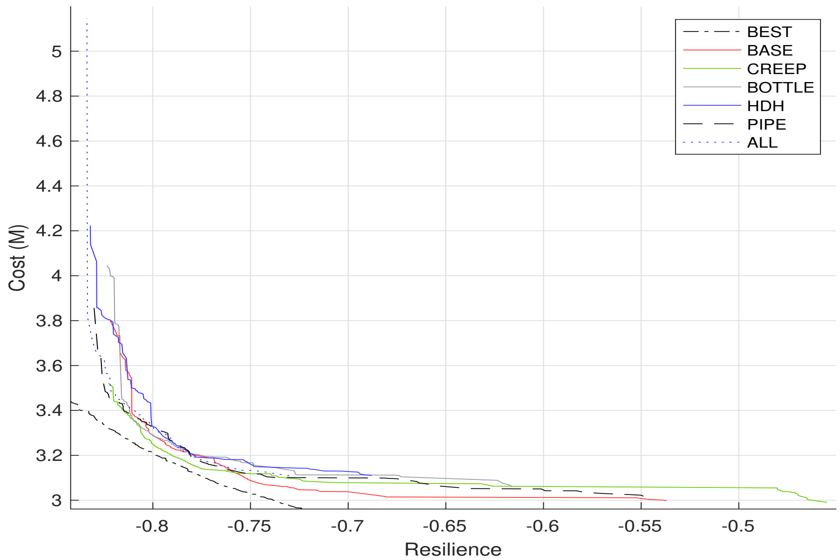
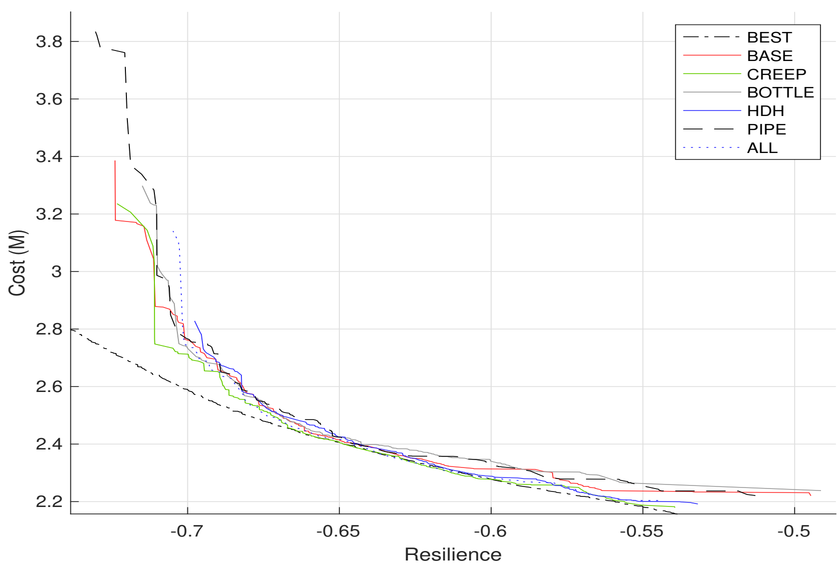


Figure 4: The approximate population Pareto front generated by each hyper-heuristic parameterisation for the MOD and BIN problems. BEST denotes the current best known solutions.

(a) MOD - PIPE is the best parameterisation.



(b) BIN - CREEP is the best parameterisation.



The generated approximate Pareto fronts are close to the best known solutions, particularly at the "knee" of the curve for the FOS, MOD, and BIN problems. The generated fronts deviate from the best known towards the extremities of the Pareto front which is to be expected given that many fewer objective function calculations have been conducted to achieve them. The HDH parameterisation shows a tendency to generate solutions with high resilience and high cost whilst still achieving good alignment with the best-known around the knee area. This suggests that the main driver for the expert on these problems is to generate a network that performs well hydraulically and is robust. The bottleneck parameterisation BOTTLE works well on gravity-fed rehabilitation tasks such as NYT, while the pipe smoothing parameterisation PIPE functions well on large, more dendritic problems such as the MOD network.

4.4. New Solutions

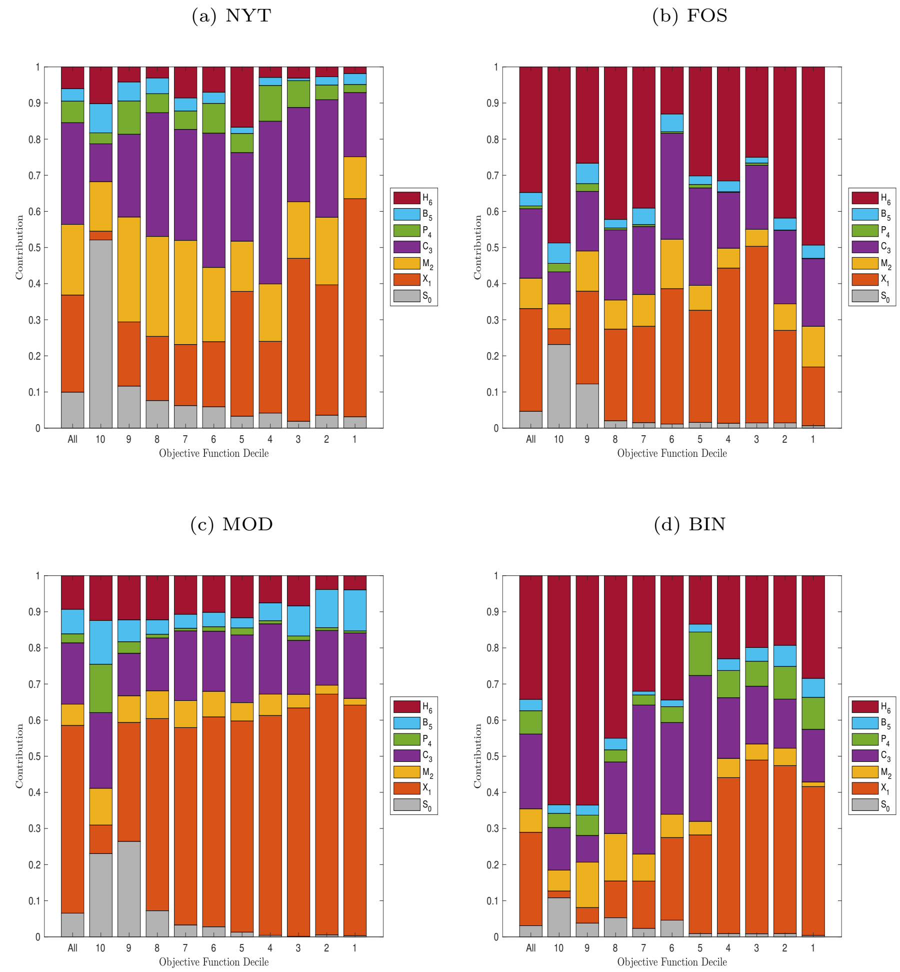
Figure 3b shows that for the FOS problem, the HDH and PIPE parameterisations have produced solutions that dominate points on the best known Pareto front BEST. The PIPE parameterisation discovers three new best-known solutions, while HDH discovers 103 new best-known solutions. The PIPE solutions and two examples of the HDN solutions are shown in Table 7.

Table 7: Five new best solutions for the FOS problem.

Name	Run	Solution	New Point		Dominates	
			C	I_n	C	I_n
PIPE	8	30	0.1332	0.9465	0.1382	0.9414
PIPE	8	44	0.1381	0.9508	0.1454	0.9452
PIPE	8	50	0.1403	0.9520	0.1403	0.9520
HDH	18	45	0.2070	0.9704	0.2197	0.9603
HDH	18	48	0.2060	0.9697	0.2188	0.9596

The generation of new best-known solutions for this complex real-world problem demonstrates the effectiveness of including human knowledge within the optimisation process. The pipe smoothing heuristic generates a small number of these points, but the HDH approach, incorporating human expertise derived from interaction generates significantly more. This result suggests that optimisation techniques with the right domain-specific expertise can achieve state-of-the-art results with many fewer objective function evaluations than generic methods alone.

Figure 5: The contribution of each low level heuristic in the ALL parameterisation for the NYT, FOS, MOD, and BIN problems.



4.5. Heuristic Effectiveness

The objective of this experiment is to visualise the interactions and differences in heuristic effectiveness that occur during distinct periods of an optimisation run. The results of plotting the heuristic contributions to decreases in the objective function value for the NYT, FOS, MOD, and BIN problems from the ALL parameterisation are shown in Figure 5.

The figure demonstrates that some heuristics are indeed better suited to certain problems and certain stages of the optimisation process as expected. For example, the $\mathtt{H_6}$ heuristic makes a relatively small contribution to the NYT and MOD problems, and this diminishes as the objective function tends towards optimality. For both of these problems, crossover is the most effective operator, particularly towards more optimal solutions. However, $\mathtt{H}_6$ makes a much larger contribution to the FOS and BIN problems, and is more effective early on in the optimisation problem for the BIN problem. It should be noted that although $\mathtt{H}_6$ makes the largest contribution to decreases in the objection function value for BIN (see Figure 5d), when considered in isolation in the HDH parameterisation it is the worst performing heuristic on this problem (see Table 3). This illustrates that the individual performance of a heuristic does not always predict its performance when combined with other heuristics. It is also interesting to note that the two domain-specific heuristics, $\mathtt{B}_4$ and $\mathtt{P}_5$, generally speaking make relatively small contributions to the four problems considered here, with $\mathtt{P}_5$ in particular contributing very little on the FOS and MOD problems. This is likely due to the significant number of loops present in these two networks. In contrast, BIN, which is a larger problem, has relatively few loops and a significant number of "dead-ends" (see Figure 1). This makes the network more prone to infeasibility, and therefore BIN benefits more when applying domain-specific heuristics. It is to be expected that a heuristic such as $\mathtt{P}_5$ which aims to smooth transitions from source to nodes at the extremities will perform relatively poorly where the network is highly looped. A surprise here is the extent to which the shuffle heuristic $\mathtt{S}_1$ contributes to the search in all networks, particularly early on. This operator is not particularly suited to this non-permutation-based problem and yet it generates useful randomisation early on in the search process, an effect which diminishes to almost zero when a near-optimal solution is found.

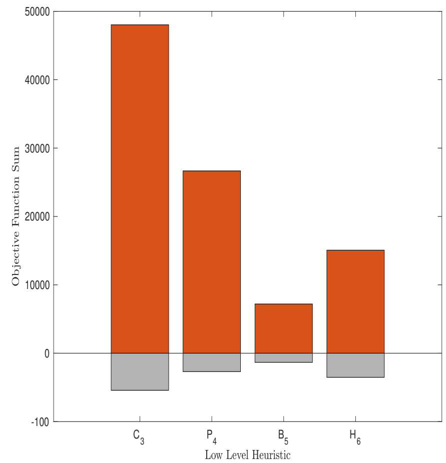
Figure 6 shows a comparison of the *individual* contributions of the low level heuristics $\mathtt{C}_3$, $\mathtt{P}_4$, $\mathtt{B}_5$, and $\mathtt{H}_6$, calculated for the CREEP, PIPE, BOTTLE, and HDH parameterisations, for the NYT, FOS, MOD, and BIN problems. In this case, there is no interaction between the heuristics. It should also be noted that the y-axis scale below 0 is 100 times smaller than the scale above 0. This scaling is necessary as heuristics generally produces more increases in the objective function than decreases.

When applied to the BIN problem, the $\mathtt{H}_6$ heuristic makes the largest contribution to decreases in the objective function value (as noted above). It also contributes the largest increases (see Figure 6d). This offers an explanation as to its poor ranking when evaluated on BIN. This concept of *heuristic variance* raises the question as to whether heuristics that make smaller contributions, both negatively and positively to the search (and therefore have low heuristic variance) should be preferred to those that can make significant improvements, but also significant backward moves. In fact, the best heuristic on BIN is $\mathtt{C}_3$, which produces the next best decrease in objective function value, while producing the smallest increase.

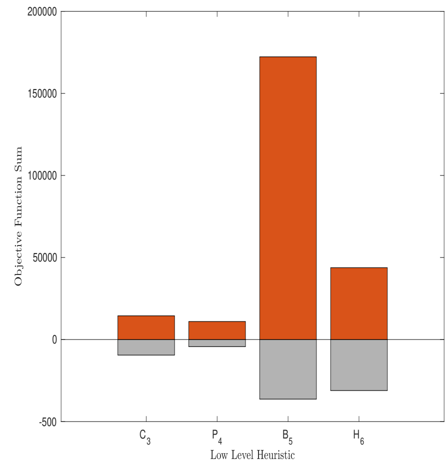
For the NYT problem, the bottleneck heuristic $\mathtt{B}_5$ attains the best individual rank (see Table 3). From Figure 6a it can be seen that the bottleneck heuristic $\mathtt{B}_5$ produces the smallest variance in objective function value. In this case, the low variance heuristic appears to be the most effective. However when all the domain heuristics are used together, Figure 5a shows that both $\mathtt{H}_6$ and $\mathtt{B}_5$ do not contribute a great deal to the optimisation process.

Figure 6: The contribution of the low level heuristics $\mathtt{C}_3$, $\mathtt{P}_4$, $\mathtt{B}_5$, and $\mathtt{H}_6$ in the parameterisations CREEP, PIPE, BOTTLE, and HDH, respectively, for the NYT, FOS, MOD, and BIN problems.

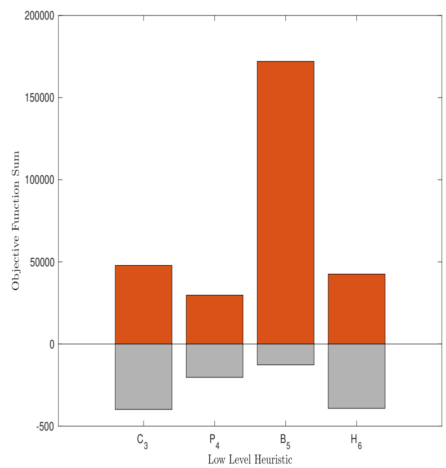
(a) NYT



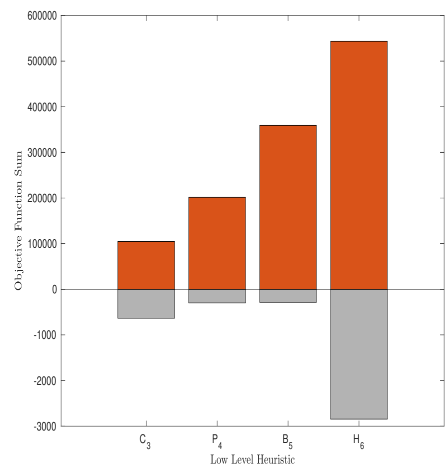
(b) FOS



(c) MOD



(d) BIN



For the FOS problem, the pipe smoothing heuristic $\mathtt{P}_4$ which has rank 4, and the bottleneck heuristic $\mathtt{B}_5$ which has rank 5 on this problem, perform very differently when considering individual contributions. Figure 6b shows that $\mathtt{P}_4$ produces the smallest contribution to the decrease in objective function value, while $\mathtt{B}_5$ produces the largest decrease. However $\mathtt{B}_5$ also produces the largest increase. The best ranked heuristic on FOS is $\mathtt{H}_6$ (see table 3) which produces the second best decrease in objective function value, but a significantly smaller increase than $\mathtt{B}_5$. When all the domain heuristics are used together on FOS, Figure 5b shows that that $\mathtt{P}_4$ (and $\mathtt{B}_5$) do not contribute a great deal to the optimisation process.

These results suggest that the highest ranked heuristics tend to produce smaller increases in the objective function value than other heuristics, even when, in some cases, those operators produce larger decreases. This effect of variance on the effectiveness of a low level heuristic has been acknowledged in the literature. For example [40] employ the principle that large (but possibly infrequent) decreases in the objective function value are likely to be more effective than small frequent decreases [16]. In this case, the hyper-heuristic is clearly having to balance the probability for improving moves with the probability of detrimental moves by a heuristic as shown in Figure 6, and that these probabilities change over time as shown in Figure 5, which supports the use of an adaptive approach such as this.

5. Conclusion

The work here is among a small group of research papers that has attempted to combine hyper-heuristic learning with domain expertise in order to improve performance on difficult search and optimisation problems. The addition of domain specific heuristics has been shown to improve performance over a baseline hyper-heuristic algorithm, with the $\mathtt{P}_5$ and $\mathtt{H}_6$ heuristics discovering a number of new "best" solutions.

The human derived $\mathtt{H}_6$ low level heuristic is particularly interesting as this heuristic was generated by a machine learning algorithm from the interactions between human experts and a (hydraulic) network model. The effectiveness of the $\mathtt{H}_6$ heuristic raises the prospect of a general method for generating novel problem solving heuristics from human expertise. Although individually $\mathtt{H}_6$ did not perform well in the ranking exercise (see Table 5), it is shown in Figure 6 that this is due to its high heuristic variance, that is, it is capable of generating significantly better solutions but also significantly worse solutions, than the other operators across an optimisation run. This raises the question as to whether variance should be considered alongside mean performance when considering heuristic selection, and further work is required to understand how this changes when a set of interacting heuristics is considered. Overall, the best performance was obtained when all knowledge-based heuristics were included in the heuristic pool. This demonstrates that an approach employing a number of potentially diverse problem solving strategies can generate high quality solutions across a range of problem instances and types.

Significant relationships were also observed between the low level heuristics and certain problem types. Heuristics adapted for dendritic networks performed poorly on the looped networks but better on the larger, more skeletal networks as would be expected. It has been possible to observe these differences in heuristic contribution, particularly the knowledge-based heuristics, and their mapping to certain problem types through the use of an online hyper-heuristic. We believe this to be the first time that the contribution of domain expertise has been visualised in this way and hope that it provides an insight into the contribution that human expertise can make to optimisation when combined with a sufficiently sophisticated heuristic selection technique.

Appendix A. Alternative Rankings

The results of ranking the hyper-heuristic parameterisations using the mean *inverted generational plus* distance IGD+ [4] are shown in Table A.8. They demonstrate that, as before, the performance of each parameterisation varies considerably according to the problem, with each parameterisation achieving a best IGD+ distance (including HDH).

Table A.8: The mean IGD+ distance between each parametrisation and the best known Pareto front, calculated over 40 runs for each of the WDN problems.

Prob.	BASE	BOTTLE	HDH	CREEP	PIPE	ALL
TRN	**0.0577**	0.0613	0.0761	0.0704	0.0628	0.0668
TLN	0.0880	0.0660	0.0787	0.0672	**0.0650**	0.0683
BAK	0.0704	**0.0094**	0.0126	0.0127	0.0382	0.0106
NYT	0.8474	0.3267	0.4500	0.3483	0.8185	**0.2234**
BLA	0.0381	0.0375	**0.0214**	0.0511	0.0344	0.0231
HAN	0.0549	0.0563	**0.0527**	0.0543	0.0780	0.0675
GOY	0.0203	0.0171	0.0197	0.0167	0.0169	**0.0141**
FOS	**0.0509**	0.0540	0.0770	0.0511	0.0590	0.0636
PES	0.3024	0.3667	0.2701	**0.2247**	0.3088	0.2552
MOD	0.6906	0.5621	0.6122	0.4052	0.5048	**0.3021**
BIN	0.3244	0.2459	0.2503	0.2434	0.3692	**0.1683**
EXN	3.5806	3.8061	3.0794	3.1081	3.0217	**2.6485**

Table A.9: The ranks for each parameterisation, calculated using the mean IGD+ between each parametrisation and the best known Pareto front, averaged over 40 runs for each of the 12 WDN problems. The best ranks are shown in bold.

Name	TRN	TLN	BLA	NYT	BLA	HAN	GOY	FOS	PES	MOD	BIN	EXN
BASE	**0**	5	5	5	4	**0**	5	**0**	3	5	4	4
HDH	5	4	2	3	**0**	1	4	5	2	4	2	2
BOTTLE	1	1	**0**	1	3	2	3	2	5	3	3	5
CREEP	4	2	3	2	5	3	1	1	**0**	1	1	3
PIPE	2	**0**	4	4	2	5	2	3	4	2	5	1
ALL	3	3	1	**0**	1	4	**0**	4	1	**0**	**0**	**0**

The results of comparing the IGD+ distance of the best and worst ranking parameterisation for the PIPE, BOTTLE, CREEP, and HDH heuristics, on each problem, using the Wilcoxon signed-rank test are shown in Table A.10. These results demonstrate that the differences in performance between the individual domain heuristics is statistically significant, with 99% confidence, in five out of the 12 problems.

Table A.10: The sample median difference $\hat{d}$, the sample mean difference $\bar{d}$, the standard deviation SD, the p-value, and the 99% confidence interval for (Best – Worst). Statistically significant results are shown in bold.

Prob.	Best	Worst	$\hat{d}$	$\bar{d}$	SD	p-value	Conf. Int.
TRN	BOTTLE	HDH	-0.0159	-0.0148	0.0334	**0.0000**	$[-\infty, -0.0063]$
TLN	PIPE	HDH	-0.0224	-0.0136	0.0644	**0.0000**	$[-\infty, -0.0095]$
BAK	BOTTLE	PIPE	-0.0014	-0.0287	0.1753	0.0872	$[-\infty, 0.0009]$
NYT	BOTTLE	PIPE	-0.0032	-0.4919	2.3917	0.4445	$[-\infty, 0.1364]$
BLA	HDH	CREEP	-0.0199	-0.0297	0.0460	**0.0000**	$[-\infty, -0.0135]$
HAN	HDH	PIPE	0.0047	-0.0258	0.1848	0.8929	$[-\infty, 0.0128]$
GOS	CREEP	HDH	0.0000	-0.0030	0.0120	0.2381	$[-\infty, 0.0018]$
FOS	CREEP	HDH	-0.0226	-0.0259	0.0426	**0.0000**	$[-\infty, -0.0010]$
PES	CREEP	BOTTLE	0.0043	-0.1420	0.5883	0.7660	$[-\infty, 0.0226]$
MOD	CREEP	HDH	-0.1171	-0.2070	0.7159	**0.0012**	$[-\infty, -0.0362]$
BIN	CREEP	PIPE	-0.0242	-0.1282	0.3824	0.0189	$[-\infty, 0.0044]$
EXN	PIPE	BOTTLE	-0.0090	-0.7844	2.0755	0.1353	$[-\infty, 0.0078]$

References

[1] Alperovits, E. and Shamir, U. (1977). Design of optimal water distribution systems. *Water Resources Research*, 13(6):885–900.

[2] Babbar-Sebens, M. and Mukhopadhyay, S. (2009). Reinforcement learning for human-machine collaborative optimization: Application in ground water monitoring. In *2009 IEEE international conference on systems, man and cybernetics*, pages 3563–3568. IEEE.

[3] Baños, R., Reca, J., Martínez, J., Gil, C., and Márquez, A. (2011). Resilience indexes for water distribution network design: A performance analysis under demand uncertainty. *Water Resources Management*, 25(10):2351–2366.

[4] Bezerra, L. C. T., López-Ibáñez, M., and Stützle, T. (2017). An empirical assessment of the properties of inverted generational distance on multi- and many-objective optimization. In Trautmann, H., Rudolph, G., Klamroth, K., Schütze, O., Wiecek, M., Jin, Y., and Grimme, C., editors, *Evolutionary Multi-Criterion Optimization*, pages 31–45. Springer International Publishing.

[5] Bianchi, L., Dorigo, M., Gambardella, L. M., and Gutjahr, W. J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, 8(2):239–287.

[6] Blum, C. and Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35:268–308.

[7] Boussaïd, I., Lepagnot, J., and Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences*, 237:82–117.

[8] Bragalli, C., D'Ambrosio, C., Lee, J., Lodi, A., and Toth, P. (2008). An MINLP solution method for a water network problem. Technical Report RC24495, IBM Research.

[9] Burke, E. K., Hyde, M. R., Kendall, G., Ochoa, G., Özcan, E., and Woodward, J. R. (2019). *A Classification of Hyper-Heuristic Approaches: Revisited*, pages 453–477. Springer International Publishing.

[10] Cunha, M. and Sousa, J. (1999). Water distribution network design optimization: Simulated annealing approach. *Journal of Water Resources Planning and Management*, 125(4):215–221.

[11] Dandy, G. and Hewitson, C. (2000). *Optimizing Hydraulics and Water Quality in Water Distribution Networks Using Genetic Algorithms*, pages 1–10. ASCE.

[12] Eiben, A. and Smith, J. (2003). *Introduction to evolutionary computing*. Springer.

[13] Eusuff, M. A. and Lansey, K. E. (2003). Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources Planning and Management*, 129(3):210–225.

[14] Ezzeldin, R., Djebedjian, B., and Saafan, T. (2014). Integer discrete particle swarm optimization of water distribution networks. *Journal of Pipeline Systems Engineering and Practice*, 5(1):04013013.

[15] Farmani, R., Savić, D. A., and Walters, G. A. (2004). EXNET benchmark problem for multi-objective optimization of large water systems. In *Modelling and Control for Participatory Planning and Managing Water Systems, IFAC Workshop, Venice, Italy*.

[16] Fialho, A., Da Costa, L., Schoenauer, M., and Sebag, M. (2008). Extreme value based adaptive operator selection. In Rudolph, G., Jansen, T., Beume, N., Lucas, S., and Poloni, C., editors, *Parallel Problem Solving from Nature – PPSN X*, pages 175–184, Berlin, Heidelberg. Springer Berlin Heidelberg.

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

[18] Geem, Z. W. (2006). Optimal cost design of water distribution networks using harmony search. *Engineering Optimization*, 38(3):259–277.

[19] Gessler, J. (1985). Pipe network optimization by enumeration. In *Speciality Conference on Computer Applications / Water Resources*, pages 572–581. ASCE.

[20] Johns, M. B., Keedwell, E. C., and Savić, D. A. (2020). Knowledge-based multi-objective genetic algorithms for the design of water distribution networks. *Journal of Hydroinformatics*, 22(2):402 – 422.

[21] Johns, M. B., Mahmoud, H. A., Walker, D. J., Ross, N. D. F., Keedwell, E. C., and Savić, D. A. (2019). Augmented evolutionary intelligence: combining human and evolutionary design for water distribution network optimisation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1214–1222.

[22] Johns, M. B., Walker, D. J., Keedwell, E. C., Savić, D. A., Loggia, G. L., Freni, G., Puleo, V., and Marchis, M. D. (2018). Interactive visualisation of water distribution network optimisation. *EPiC Series in Engineering*, 3:995–1003.

[23] Kheiri, A. and Keedwell, E. C. (2015). A sequence-based selection hyper-heuristic utilising a hidden Markov model. In *Proceedings of the Genetic and Evolutionary Computation*, pages 417–424. ACM.

[24] Kheiri, A., Keedwell, E. C., Gibson, M. J., and Savić, D. A. (2015). Sequence analysis-based hyper-heuristics for water distribution network optimisation. *Procedia Engineering*, 119:1269–1277.

[25] Kim, J. H., Kim, T. G., Kim, J. H., and Yoon, Y. N. (1994). A study on the pipe network system design using non-linear programming. *Journal of Korea Water Resources Association*, 27(4):59–67.

[26] Lee, S. C. and Lee, S. I. (2001). Genetic algorithms for optimal augmentation of water distribution networks. *Journal of Korea Water Resources Association*, 34(5):567–575.

[27] Mahmoud, H. A., M. B. Johns, M., Keedwell, E. C., and Savić, D. A. (2019). Generalising human heuristics in augmented evolutionary water distribution network design optimisation.

[28] Mohan, S. and Babu, K. S. J. (2010). Optimal water distribution network design with honey-bee mating optimization. *Journal of Computing in Civil Engineering*, 24(1):117–126.

[29] Prasad, T. D. and Park, N.-S. (2004). Multiobjective genetic algorithms for design of water distribution networks. *Journal of Water Resources Planning and Management*, 130(1):73–82.

[30] Raad, D. N., Sinske, A. N., and van Vuuren, J. H. (2010). Comparison of four reliability surrogate measures for water distribution systems design. *Water Resources Research*, 46(5):W05524.

[31] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

[32] Reca, J. and Martínez, J. (2006). Genetic algorithms for the design of looped irrigation water distribution networks. *Water Resources Research*, 42(5):W05416.

[33] Remde, S., Cowling, P., Dahal, K., and Colledge, N. (2007). Exact/heuristic hybrids using rVNS and hyperheuristics for workforce scheduling. In *Evolutionary Computation in Combinatorial Optimization*, pages 188–197. Springer Berlin Heidelberg.

[34] Rieker, J. D. and Labadie, J. W. (2012). An intelligent agent for optimal river-reservoir system management. *Water Resources Research*, 48(9).

[35] Rossman, L. A. (2000). *EPANET2 users manual*. U.S. Environment Protection Agency.

[36] Schaake, J. C. and Lai, F. H. (1969). Linear programming and dynamic programming application to water distribution network design. Technical report, M.I.T. Hydrodynamics Laboratory.

[37] Sherali, H. D., Subramanian, S., and Loganathan, G. V. (2001). Effective relaxations and partitioning schemes for solving water distribution network design problems to global optimality. *Journal of Global Optimization*, 19(1):1–26.

[38] Shirzad, A. (2017). Shortening the search time in optimization of water distribution networks. *Urban Water Journal*, 14(10):1038–1044.

[39] Singh, A., Walker, D. D., Minsker, B. S., and Valocchi, A. J. (2010). Incorporating subjective and stochastic uncertainty in an interactive multi-objective groundwater calibration framework. *Stochastic Environmental Research and Risk Assessment*, 24(6):881–898.

[40] Soria-Alcaraz, J. A., Ochoa, G., Swan, J., Carpio, M., Puga, H., and Burke, E. K. (2014). Effective learning hyper-heuristics for the course timetabling problem. *European Journal of Operational Research*, 238(1):77 – 86.

[41] Walker, D. J., Johns, M. B., and Keedwell, E. C. (2018a). Generating heuristics to mimic experts in water distribution network optimisation. In *WDSA/CCWI Joint Conference Proceedings*, volume 1.

[42] Walker, D. J., Johns, M. B., Keedwell, E. C., and Savić, D. A. (2018b). Generating heuristics to mimic experts in water distribution network optimisation. In *WDSA / CCWI 2018 Joint Conference*. Morgan Kaufmann.

[43] Walker, D. J., Johns, M. B., Keedwell, E. C., and Savić, D. A. (2018c). Towards interactive evolution: A distributed optimiser for multi-objective water distribution network design. *EPiC Series in Engineering*, 3:2259–2267.

[44] Wang, Q., Guidolin, M., Savić, D. A., and Kapelan, Z. (2015). Two-objective design of benchmark problems of a water distribution system via MOEAs: Towards the best-known approximation of the true Pareto front. *Journal of Water Resources Planning and Management*, 141(3):04014060.

[45] Yates, W. B. and Keedwell, E. C. (2019). An analysis of heuristic subsequences for offline hyper-heuristic learning. *Journal of Heuristics*, 25(3):399–430.

[46] Yates, W. B. and Keedwell, E. C. (2021). Offline learning with a selection hyper-heuristic: An application to water distribution network optimisation. *Evolutionary Computation*, 29(2):187–210.

[47] Zhang, Q., Sun, J., and Tsang, E. (2005). An evolutionary algorithm with guided mutation for the maximum clique problem. *IEEE transactions on evolutionary computation*, 9(2):192–200.

[48] Zheng, F., Zecchin, A. C., and Simpson, A. R. (2013). Self-adaptive differential evolution algorithm applied to water distribution system optimization. *Journal of Computing in Civil Engineering*, 27(2):148–158.