A Reinforcement Learning Hyper-heuristic for Water Distribution Network Optimisation

Azza O. M. Ahmed

Department of Computer Science
University of Khartoum, Faculty of Mathematical Sciences
Khartoum, Sudan
azzaomer70@gmail.com

Shahd M. Y. Osman

Department of Computer Science
University of Khartoum, Faculty of Mathematical Sciences
Khartoum, Sudan
shahd.mustafa1996@gmail.com

Terteel E. H. Yousif

Department of Computer Science
University of Khartoum, Faculty of Mathematical Sciences
Khartoum, Sudan
terteelhussein@gmail.com

Ahmed Kheiri

Department of Management Science
Lancaster University Management School
Lancaster, LA1 4YX, United Kingdom
a.kheiri@lancaster.ac.uk

***Abstract*—The Water Distribution Networks (WDNs) optimisation problem focuses on finding the combination of pipes from a collection of discrete sizes available to construct a network of pipes with minimum monetary cost. It is one of the most significant problems faced by WDN engineers. This problem belongs to the class of difficult combinatorial optimisation problems, whose optimal solution is hard to find, due to its large search space. Hyper-heuristics are high-level search algorithms that explore the space of heuristics rather than the space of solutions in a given optimisation problem. In this work, different selection hyper-heuristics were proposed and empirically analysed in the WDN optimisation problem, with the goal of minimising the network's cost. New York Tunnels network benchmark was used to test the performance of these hyper-heuristics including the Reinforcement Learning (RL) hyper-heuristic method, that succeeded in achieving improved results.**

***Keywords*: Hyper-heuristics, Water Distribution Network Design, Pipe Optimisation**

I. Introduction

Throughout the last decades, researches tackling the pipe optimisation in WDNs have grown, as they play major role in the improvement of quality of life. Optimisation of the water distribution network design or rehabilitation is a classic fundamental problem. It is considered a discrete NP-hard difficult combinatorial optimisation problem [1], where the goal is to choose a combination of pipes from a collection of discrete sizes available to construct a minimal cost network taking into consideration the problem's constraints [2].

A number of metaheuristic methods have been successfully applied to the WDN design/rehabilitation problem in the last decades [3]. Relatively, new optimisation methods known as hyper-heuristics were recently applied to solve this problem. Hyper-heuristic methods are general purpose, automated methodologies and unlike metaheuristics, they explore the heuristics search space rather than the solutions search space directly [4]. Hyper-heuristics can be categorised into two groups *generation* or *selection* hyper-heuristics. *Generation* hyper-heuristics build a set of heuristics, while selection hyper-heuristics select from a collection of pre-defined Low Level Heuristics (LLHs) to apply during the search process [5].

In this study, the New York Tunnels (NYT) benchmark was used to examine the application of reinforcement learning and other well-known selection hyper-heuristics to solve the WDN rehabilitation problem.

II. Background

A. WDN Problem

WDNs are complex systems that aim to deliver fresh water to demand points with adequate pressure using least monetary costs. The optimisation of these networks depends on multiple parameters. For example, water age, water supply shortage, and velocity adherence [1]. This research focuses only on pipe sizing as decision variables to reduce the network's cost. Solving the pipe optimisation problem requires finding the combinations of pipe sizes that give the minimum possible cost for a given test network. However, many combinations of these pipes lead to a network failure. For example, it might cause a failure in meeting the hydraulic requirements. This has been addressed by introducing a 'feasibility' cost that is much more heavily penalised than the network's cost [1], [6]. Therefore, a solution is evaluated (Eq 1) in terms of the feasibility (Eq 2) and the network's cost (Eq 3).

$$obj = C + \alpha * F_{headDeficit} \tag{1}$$

where α is used to adjust the weight of the head deficit with respect to the network's cost in the objective function.

The head deficit is the feaibility function:

$$F_{headDeficit} = \sum_{n=1}^{m} ((h_t - h_n) > 0) \tag{2}$$

where n is one of the m demand nodes in the network, h is the hydraulic head at node n, and h_t is the target head for

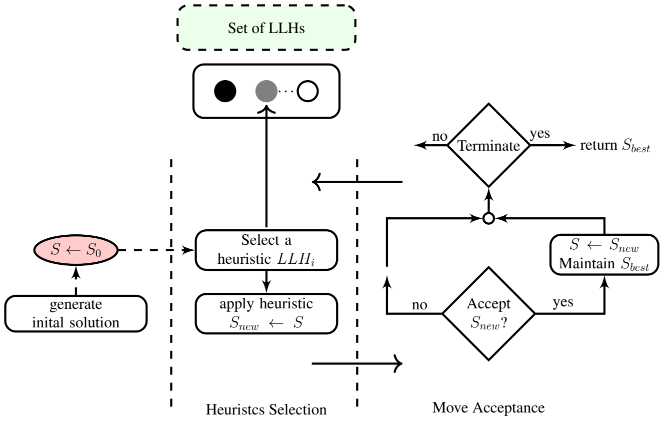

Fig. 1: A selection hyper-heuristic framework

each node, which is a uniform (not always necessary) value for all network nodes.

$$C = \sum_{i=1}^{k} (1.1 d_i^{1.24} \times l_i) \tag{3}$$

where k is the total number of pipes in the network, d is the available diameter of pipe i, and l is its length in feet or meters.

B. Selection Hyper-heuristics

Various number of optimisation methods were used to solve the WDN problem, including single-based metaheuristic methods [7], population-based genetic algorithms [8], and selection hyper-heuristics [9], [1], [10].

Selection hyper-heuristics have shown significant improvement in solving the problem by reducing financial and environmental costs. Selection hyper-heuristics have two main stages: *heuristic selection* and *move acceptance*. An initial solution is passed through these stages for a number of iterations. At each iteration, a heuristic is selected from a set of low level heuristics and applied to the candidate solution to generate a new solution. The generated solution is then accepted or rejected based on the move acceptance method as illustrated in Figure 1.

There is a wide variety of heuristic selection methods described in [11]. Random Permutation (RP) produces a permutation of the LLHs and applies them in the provided order sequentially. On the other hand, Random Descent (RD) first selects a random LLH and applies it until it fails to produce further improvement, then another LLH is randomly selected and so on. Random Permutation Descent (RPD) combines the two methods RP and RD. Greedy (GR) applies all LLHs to a candidate solution, and then chooses the LLH that produces the largest improvement. Reinforcement Learning (RL) is based on the idea of "rewarding and punishing" LLH. Initially, it assigns all LLHs the same score, then at each iteration of the search process, the LLH that results in an improvement to the solution is rewarded (its score is increased), whereas the LLH that results in non-improving or worsening solution is punished (its score is decreased), and at each stage of the problem the LLH with the highest score is chosen [12].

Move acceptance methods can be classified into a *deterministic* or *non-deterministic* approaches, as well as *stochastic* or *non-stochastic*. Accept All Moves is an example of a non-stochastic deterministic approach. Some stochastic non-deterministic acceptance approaches accept worsening solutions such as Simulated Annealing (SA) move acceptance method. SA [13] is inspired by the procedure of annealing in physics where heating with high temperature is applied allowing aggregation of particles as it is cooled. Thus, two parameters need to be tuned based on the problem's domain, the temperature T that regulates the probability of accepting solutions with higher costs, and β is a geometric cooling schedule with $T_{i+1} = T_i\beta$ where, i is the current iteration. In SA, a newly generated solution at each step is accepted if it improves the quality of the previous solution. Worsening solutions are accepted with a certain probability P where, $P = e^{\frac{-\Delta}{T}}$, and Δ is the quality change [14]. These worsening solutions are accepted so as to elude from local optima. Late Acceptance (LA) approach only accepts non-worsening solution with respect to a solution visited L steps previously [15].

III. Methodology

The test network used in this research is the New York Tunnels (NYT) problem instance. It consists of 21 pipes and 16 diameters, with a minimum head required to all nodes of 255 ft. except for the nodes 16, 17, and 1 which are 260, 272.8, and 300 ft. respectively. The optimal solution for this network under standard conditions with zero head deficit is $38.64m [1]. Figure 2 shows the layout of the problem instance.

Epanet2 hydraulic simulator [16] has been employed to provide the information needed to generate the hydraulic values so as to decide to what extent the tested network meets the hydraulic constraints.

A. Low Level Heuristics

Nine LLHs have been implemented as follows:

- LLH1: Change single pipe diameter in a random manner.
- LLH2: Change two pipe diameters in a random manner.
- LLH3: Select two pipes randomly and swap them.
- LLH4: Increase or decrease all pipes by one pipe size.
- LLH5: Select a pipe randomly and increase or decrease its size by one.
- LLH6: Change between one to five pipe diameters in a random manner.
- LLH7: Pick two random pipes, increase one and decrease the other.
- LLH8: Pick four random pipes, increase two by one, and decrease the remaining two by one.
- LLH9: Change all pipes randomly.

B. Selection Hyper-heuristics for WDN

In this study, four selection hyper-heuristics have been implemented as follows: RP-LA, RPD-LA, GR-SA, and RL.

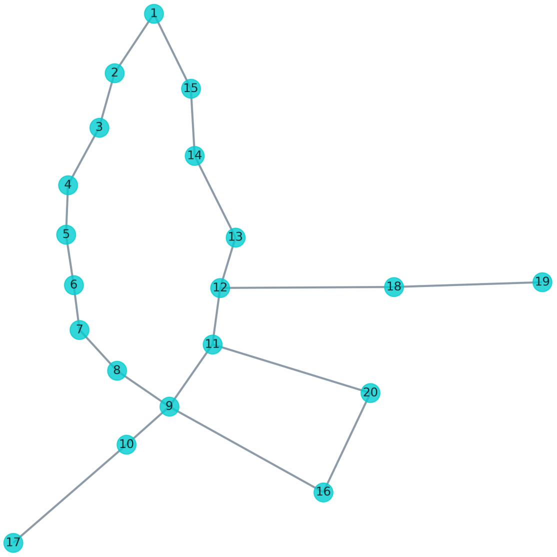

Fig. 2: New York Tunnels problem layout. Node 1 is the reservoir/source of water

At each step, the heuristic selection method selects and applies a LLH from the set of LLHs described above to generate a new solution. The move acceptance method decides whether to accept or reject the newly generated solution. In RL method, initially all LLHs are assigned the same scores, then at each iteration of the search process the LLH that results in an improvement to the solution, is rewarded (its score is increased), whereas the LLH that results in non-improving or worsening solution, is punished (its score is decreased), and at each stage of the problem the LLH with the best score is selected. In this work, and as suggested in [12], the LLHs with worsening solutions are given less penalty cost compared to the reward, in order to give the LLH a second chance. A counter referred to as β is used to manage the acceptance of worsening solutions, it is increased more when infeasible solutions are generated, and α represents the tolerance number for successive non-improvements. In the case $\beta > \alpha$ and the new solution is not much worse than the preceding one, then the new solution will be accepted and all LLH's scores are reassigned to the initial values. Algorithm 1 provides the pseudocode of RL algorithm, where α is set to 100, $Reward$ is set to 0.3, and $penalty_w$ is set to 0.0625.

Algorithm 1: Pseudocode of Reinforcement Learning

```
Let $LLH$ be the set of LLHs
Let $LLHS$ be the scores of LLHs
Let $j$ be the current iteration
Let $\beta$ be the number of iterations without improvement
Let $\alpha$ be a tolerance number of non-improvements
Let $reward$ be the factor for rewarding LLH
Let $penalty_w$ be the factor for penalising LLH generating
  worsening solutions
Let $S, S_{new}, S_{best}$ be the current, new and best solutions
Let $C(S)$ be the objective function value of $S$;
Generate initial solution $S$;
$S_{best} \leftarrow S$;
repeat
    $LLH_i \leftarrow SelectBest(LLHS)$;
    $S_{new} \leftarrow$ApplyLLH$(LLH_i, S)$
    if $C(S_{new}) < C(S)$ then
        $S_{new} \leftarrow S$;
        $LLHS_i \leftarrow LLHS_i + j * reward$;
        $\beta = 0$;
        if $C(S_{new}) < C(S_{best})$ then
            $S_{best} \leftarrow S_{new}$;
        end
    end
    else if $C(S_{new}) <= C(S_{best}) + 0.05 * C(S_{best})$ then
        $LLHS \leftarrow [0.5] * K$        ▷ K is set to the number of LLHs
        $\beta = 0$;
    end
    else
        $LLHS_i \leftarrow LLHS_i - j * penalty_w$;
        $\beta \leftarrow \beta + 1$;
    end
until TerminationCriterionSatisfied();
return $S_{best}$
```

TABLE I: Summary of experimental results. Best values are highlighted in bold

Method	Avg	Std	Min	Max	Itr
RP-LA	38974842.69	490019.799	**38643533.22**	40121269.35	29039.2
RPD-LA	39036747.43	571527.639	**38643533.22**	40020372.28	39924.5
GR-SA	42579057.84	5135995.78	**38643533.22**	55104316.05	56617.9
RL	**38937030.07**	**279147.073**	**38643533.22**	**39246212.35**	**17371.1**

IV. Empirical Results

The device used to conduct the experiments for this work is an Intel Core i7-9750H, with a 16GB RAM. The four selection hyper-heuristic methods described in Section III were run for 10 trails, with 100,000 evaluations for each. Table I presents the results, providing the average, the standard deviation, the minimum, and the maximum objective function values over the ten trials. The table shows that all methods reached the minimum cost of $38.64m, and reveals that the RL method obtained the minimum average with significantly fewer number of iterations in comparison with the other tested methods, with an average of 17371.1 objective function evaluations.

Mann-Whitney-Wilcoxon statistical test was used with a confidence level at 95 percent to compare pairwise variations in performance between the hyper-heuristics statistically. The notations are defined as follows: A versus B, where A is the row and B is the column, $<$ $(>)$ indicates that A (B) is better than B (A) and this performance variance is statistically significant, while $=$ indicates that there is no statistical significance between A and B. Table II shows that RL outperformed RPD-LA, and GR-SA algorithms, and this performance is statistically significant. However, there is no statistical significant between RL and RP-LA hyper-heuristics.

Figure 3 illustrates the average utilisation rate for all LLHs over 10 runs for RP-LA and RL. It is noted that LLH1 and LLH5 have succeeded in producing the best performance in both RL and RP-LA hyper-heuristic methods.

Figure 4 depicts the changes in the objective function value in RL versus the evaluations while solving the problem. It can

TABLE II: Pairwise performance comparison of RP-LA, RPD-LA, GR-SA, and RL, based on the average over 10 trials

	RP-LA	RPD-LA	GR-SA	RL
RP-LA	-	=	<	=
RPD-LA	=	-	<	>
GR-SA	>	>	-	>
RL	=	<	<	-

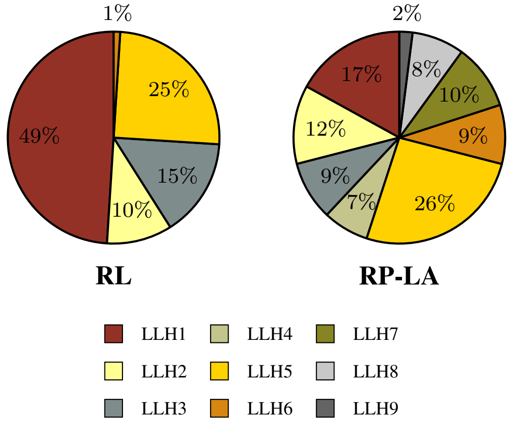

Fig. 3: Utilisation rate of each low level heuristics

be observed that RL continuously improves the quality of the solution.

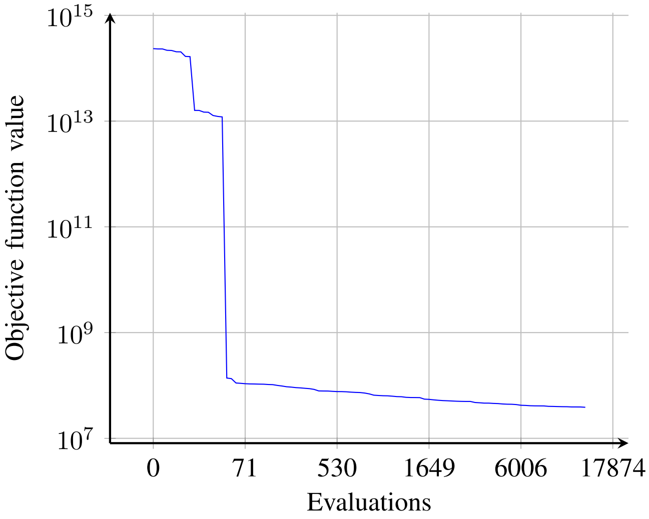

Fig. 4: Improvement of cost in terms of objective function value

V. Conclusion

In this work, four selection hyper-heuristics have been applied to the WDN problem. These methods combine four different heuristic selection methods and two move acceptance methods. New York Tunnels benchmark was used to test the performance of these methods, and their performances were compared against each other to determine the best algorithm. The analysis was made using Mann-Whitney-Wilcoxon statistical test, and it showed the success of RL method. The results showed that in RL, LLH1 (randomly change only one pipe diameter) was the most low level heuristic that contributed to the best solutions produced. The generality level and effectiveness that the selection hyper-heuristics achieve will be further investigated on other WDN problem instances of various sizes.

References

[1] A. Kheiri, E. Keedwell, M. J. Gibson, and D. Savic, "Sequence analysis-based hyper-heuristics for water distribution network optimisation," *Procedia Engineering*, vol. 119, pp. 1269–1277, 2015.

[2] V. V. Vazirani, *Approximation Algorithms*. Springer Publishing Company, Incorporated, 2010.

[3] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM computing surveys (CSUR)*, vol. 35, no. 3, pp. 268–308, 2003.

[4] J. H. Drake, A. Kheiri, E. Özcan, and E. K. Burke, "Recent advances in selection hyper-heuristics," *European Journal of Operational Research*, vol. 285, no. 2, pp. 405–428, 2020.

[5] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and R. Qu, "Hyper-heuristics: A survey of the state of the art," *Journal of the Operational Research Society*, vol. 64, no. 12, pp. 1695–1724, 2013.

[6] K. McClymont, "Multi-objective hyper-heuristics and their application to water distribution network design," 2012.

[7] H. Mala-Jetmarova, N. Sultanova, and D. Savic, "Lost in optimisation of water distribution systems? a literature review of system operation," *Environmental modelling & software*, vol. 93, pp. 209–254, 2017.

[8] D. Savic and G. Walters, "Genetic algorithms for least-cost design of water distribution networks," *Journal of Water Resources Planning and Management-ASCE*, vol. 123, pp. 67–77, 03 1997.

[9] K. McClymont, E. Keedwell, D. Savić, and M. Randall-Smith, "A general multi-objective hyper-heuristic for water distribution network design with discolouration risk," *Journal of Hydroinformatics*, vol. 15, no. 3, pp. 700–716, 2013.

[10] W. B. Yates and E. C. Keedwell, "Offline Learning with a Selection Hyper-heuristic: An Application to Water Distribution Network Optimisation," *Evolutionary computation*, pp. 1–25, 2020.

[11] P. Cowling, G. Kendall, and E. Soubeiga, "A hyperheuristic approach to scheduling a sales summit," in *International conference on the practice and theory of automated timetabling*. Springer, 2000, pp. 176–190.

[12] Y. Pylyavskyy, A. Kheiri, and L. Ahmed, "A reinforcement learning hyper-heuristic for the optimisation of flight connections," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, 2020, pp. 1–8.

[13] D. Abramson, M. Krishnamoorthy, H. Dang *et al.*, "Simulated annealing cooling schedules for the school timetabling problem," *Asia Pacific Journal of Operational Research*, vol. 16, pp. 1–22, 1999.

[14] A. Kheiri and E. Özcan, "An iterated multi-stage selection hyper-heuristic," *European Journal of Operational Research*, vol. 250, no. 1, pp. 77–90, 2016.

[15] A. Kheiri and E. Ozcan, "Constructing constrained-version of magic squares using selection hyper-heuristics," *The Computer Journal*, vol. 57, no. 3, pp. 469–479, 2014.

[16] L. A. Rossman *et al.*, "Epanet 2: users manual," 2000.