Genetic Programming Hyper-Heuristic with Knowledge Transfer for Uncertain Capacitated Arc Routing Problem

Mazhar Ansari Ardeh
Victoria University of Wellington
mazhar.ansariardeh@ecs.vuw.ac.nz

Yi Mei
Victoria University of Wellington
yi.mei@ecs.vuw.ac.nz

Mengjie Zhang
Victoria University of Wellington
mengjie.zhang@ecs.vuw.ac.nz

ABSTRACT

Uncertain Capacitated Arc Routing Problem (UCARP) is an important combinatorial optimisation problem. Genetic Programming (GP) has shown effectiveness in automatically evolving routing policies to handle the uncertain environment in UCARP. However, when the scenario changes, current routing policy can no longer work effectively, and one has to retrain a new policy for the new scenario which is time consuming. On the other hand, knowledge from solving the previous similar scenarios may be helpful in improving the efficiency of the retraining process. In this paper, we propose different knowledge transfer methods from a source scenario to a similar target scenario and examine them in different settings. The experimental results showed that by knowledge transfer, the retraining process is made more efficient and the same performance one can obtained within a much shorter time without having any negative transfer.

CCS CONCEPTS

• **Theory of computation** → *Routing and network design problems*;

KEYWORDS

Uncertain Capacitated Arc Routing Problem, Genetic Programing, Transfer Learning

ACM Reference Format:
Mazhar Ansari Ardeh, Yi Mei, and Mengjie Zhang. 2019. Genetic Programming Hyper-Heuristic with Knowledge Transfer for Uncertain Capacitated Arc Routing Problem. In *GECCO '19: Genetic and Evolutionary Computation Conference Companion, July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3205651.3205661

1 INTRODUCTION

Capacitated Arc Routing Problem (CARP) is a famous combinatorial optimisation problem. CARP is based on a connected and undirected graph $G(V, E)$, where each node $v \in V$ may be an intersection, and each edge $e \in E$ may be a street segment with a non-negative demand to be served by a set of vehicles with limited capacity. The vehicles are located at the depot (a special node in the graph). Serving and traversing through an edge incurs a serving or deadheading cost. The goal of CARP is to find a set of minimum-cost routes for each vehicle to serve the demand of the edges subject to a set of predefined constraints [7]. To tackle uncertain environments, the Uncertain CARP (UCARP) model was proposed in [6], which contains four types of stochastic factors. Routing policy is a promising technique which can generate the routes for UCARP (e.g. [4]). Genetic Programming Hyper-Heuristic (GPHH) has shown success in automatically evolving effective routing policies for given scenarios. However, the existing GPHH approaches evolve routing policies for different scenarios separately without considering the relation between them. In real world, different problem scenarios may be correlated and it is reasonable to expect that the knowledge gained from evolving a routing policy can help improve the effectiveness of the retraining process of similar ones.

The overall goal of this paper is to propose new knowledge transfer methods for evolving routing policies of UCARP efficiently, given that some related problem scenarios have already been considered before. Specifically, it aims to develop knowledge transfer methods based on sub-tree and terminal importance transfer; design proper mechanisms regarding (1) knowledge representation; (2) knowledge extraction; and (3) knowledge usage in the target domain for each knowledge transfer method.

GECCO '19, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-5764-7/18/07.
https://doi.org/10.1145/3205651.3205661

2 THE METHOD

In this paper, we consider two knowledge representations for GPHH based on (1) (sub-)Tree Transfer (GPHH-TT), and (2) Feature Importance Transfer (GPHH-FIT).

2.1 GPHH with (sub-)Tree Transfer

The general idea of GPHH-TT is that an important (sub-)tree in the source domain tends to be also important in the related target domain, and thus should be used more often. Based on this idea, GPHH-TT is described as follows.

2.1.1 Knowledge Extraction. For knowledge extraction, we consider the following different mechanisms to extract (sub-)trees. (1) *BestGen*: select the best individuals of each generation [1]; (2) *Full-tree*: select the $k\%$ best individuals in the final population [1]; (3) *Subtree*: select a random subtree of the $k\%$ best individual in the final population [1]. (4)*GTLKnow*: select the best and median individuals from each generation [3]; (5) *TLGPCriptor*: select random sub-trees of the individuals that are better than average in the final population [2].

2.1.2 Knowledge Usage. For using the transferred (sub-)trees, we directly imports them into the initial population in the target domain. For TLGPCriptor, during the initialization phase and with a probability of 50%, root children are selected from transferred subtrees or generated from terminals and functions. This sub-tree creation procedure is utilized during mutation too.

2.2 GPHH with Feature Importance Transfer

The proposed GPHH-FIT first calculates the feature importance for each terminal in the source domain, then modifies and runs the GPHH for the target domain by the calculated feature importance in the target domain.

2.2.1 Knowledge Extraction. For each feature, the importance is represented as a weight reflecting its contribution to the routing policies in the source domain. First, the final population is cleared from duplicated individuals. Then, if there are multiple individuals with the same fitness, the individual with the smallest depth is kept, while all the remaining individuals are penalised by setting their fitness to ∞. Then, the top 50% individuals in the final population is selected and the contribution of each terminal to each of its individual is calculated by the shuffle test proposed in [5]. Specifically, the contribution $\zeta(\tau, x)$ of a terminal τ to an individual x is defined as the difference between the fitness of the tree with and without fixing the terminal τ to 1. Feature weights are calculated by a weighted voting process. Each individual votes for the terminals that contribute to it. Finally, the weight of each terminal is defined as the total votes it receives. Here, the voting power of each individual depends on its fitness. Since UCARP is a minimisation problem, an individual with a smaller fitness should have a larger voting power. The voting power of each individual is calculated by (1) inverting the original fitness: $g(x) = \frac{1}{1+fit(x)}$, $\forall x \in \Omega$, where Ω is the set of all the individuals evaluated in the source domain; (2) setting $g_{\min} = \min\{g(x)|x \in \Omega\}$, $g_{\max} = \max\{g(x)|x \in \Omega\}$ and (3) normalising the inverted fitness: $pow(x) = \frac{g(x)-g_{\min}}{g_{\max}-g_{\min}}$.

2.2.2 Knowledge Usage. Intuitively, a more important feature should be used more often in the GP tree. Therefore, by transferring the feature importance to the target domain, rather than uniform selection, we can modify the probability of selecting each terminal so that more important features are more likely to be used.

3 EXPERIMENTAL STUDIES

To investigate the effectiveness of the proposed methods, we design a number of source and target domain settings based on the same medium and large-sized UCARP instance that differ from each other in terms of the number of vehicles.

Overall, we report that there is no statistical difference between the GPHHs with and without knowledge transfer in terms of the final performance. Although the problem instances are complex the heuristic space is fixed and defined by the GP parameter settings which may lead to a relatively small search space for GP to find a good result, eliminating the need for transfer knowledge. Also, it is likely that the source and target domains are actually very different that the transferred knowledge are not reusable.

A major goal of knowledge transfer is to improve the retraining efficiency in the target domain. To investigate this, we plotted the convergence curves of the compared algorithms at each generation of the retraining process in the target domain (e.g. Fig. 1) and noticed that saw that the GPHH-TT approaches have much better starting points and convergence speed (GPHH-TT algorithms converged at around 20 generations earlier) than GPHH-FIT and without transfer. This indicates that the transferred sub-trees are effective in the target domain. The GPHH-FIT algorithms had some slightly better starting points than the standard GPHH but it was quickly caught up by it. This shows that the transferred feature importance is weaker in terms of reusability, which is as expected, since the feature importance is a higher-level knowledge than sub-trees.

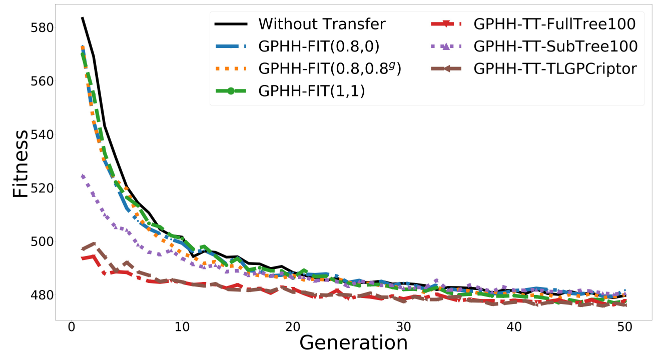

Figure 1: Convergence curves of the compared algorithms on val9D, from 10 to 11 vehicles

In our experiments, the feature weights vary from one dataset to another which indicates that the transferred knowledge is domain specific. Another interesting observation is that the standard deviation of the weights are very high which means the feature weighting calculation is unstable and a weight can receive different weights in different runs. This may be caused by the loss of diversity of the individuals in the final population of the source domain.

4 CONCLUSIONS AND FUTURE WORKS

In this paper, proposed GPHH approaches with knowledge transfer to improve the efficiency of the retraining process for the routing policies of UCARP with two types of knowledge transfer and examined them for extracting and reusing the knowledge.

Our experimental results showed that the knowledge transfer, especially the sub-tree transfer, can greatly improve the training speed in the target domain, and can achieve the same performance within a much shorter time. The algorithms performed statistically comparable, suggesting there is no negative transfer and demonstrates the potential of improving the retraining process of GPHH for UCARP by knowledge transfer.

REFERENCES

[1] T. T. H. Dinh, T. H. Chu, and Q. U. Nguyen. Transfer learning in genetic programming. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 1145–1151, May 2015.

[2] M. Iqbal, M. Zhang, and B. Xue. Improving classification on images by extracting and transferring knowledge in genetic programming. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 3582–3589, July 2016.

[3] Baris Kocer and Ahmet Arslan. Genetic transfer learning. *Expert Systems with Applications*, 37(10):6997 – 7002, 2010.

[4] Yuxin Liu, Yi Mei, Mengjie Zhang, and Zili Zhang. Automated heuristic design using genetic programming hyper-heuristic for uncertain capacitated arc routing problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '17, pages 290–297, New York, NY, USA, 2017. ACM.

[5] Y. Mei, S. Nguyen, B. Xue, and M. Zhang. An efficient feature selection algorithm for evolving job shop scheduling rules with genetic programming. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 1(5):339–353, Oct 2017.

[6] Y. Mei, K. Tang, and X. Yao. Capacitated arc routing problem in uncertain environments. In *IEEE Congress on Evolutionary Computation*, pages 1–8, July 2010.

[7] Sanne Wøhlk. *A Decade of Capacitated Arc Routing*, pages 29–48. Springer US, Boston, MA, 2008.