Algorithm Selection Using Transfer Learning

Niranjana Deshpande
Naveen Sharma
{nd7896,nxsvse}@rit.edu
Rochester Institute of Technology
Rochester, New York, USA

ABSTRACT

Per-instance algorithm selection has been shown to achieve state of the art performance in solving Travelling Salesman Problems (TSP). By selecting optimization algorithms for each TSP instance, significant time savings have been achieved. In this work, we highlight how recent algorithm selection techniques apply to service composition; which is posed as a TSP problem. However, the service composition environment is highly dynamic, which poses unique challenges for algorithm selection. Chief amongst those is the availability of training data for all algorithms on unseen tasks, which is infeasible to obtain. To address this problem, we propose the use of transfer learning techniques to improve classification accuracy in dynamic settings such as service composition.

KEYWORDS

algorithm selection, transfer learning, service composition, travelling salesman problem

ACM Reference Format:
Niranjana Deshpande and Naveen Sharma. 2021. Algorithm Selection Using Transfer Learning. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3462736

1 MOTIVATION

Metaheuristic algorithms have been used to solve optimization problems in a wide range of applications ranging from software test generation [3] to training neural networks [1]. The advantages of using such algorithms is that they compute near-optimal solutions, fast. However, applying these algorithms to a new problem domain requires extensive empirical evaluation and modeling. Moreover, recent literature has also demonstrated that state-of-the-art performance is achieved not by selecting a single algorithm for an entire problem domain, but by selecting an algorithm for each instance in the problem domain. These approaches have great potential for application in systems that require frequent, quick and accurate execution for optimization problems. Using service composition as a case study, we outline the benefits and challenges associated with online algorithm selection in dynamic environments.

The goal of service composition is to select multiple web services based on non-functional, Quality of Service (QoS) attributes such as response time, throughput etc.. Thousands of web services providing similar functionality may exist, leading to a large search space of possible solutions. As metaheuristic algorithms are capable of searching through large search spaces in a relatively shorter period of time, they are often used as *composition algorithms* to select a set of services that fulfill QoS requirements. Additionally, to avoid increased operational costs, time taken and memory used for composition also needs to be minimized. So, the choice of composition algorithm depends on its ability to meet QoS constraints, minimize time and memory usage. However, certain algorithms can fail to converge to a suitable solution on some problem instances, leading to sub-optimal solutions. To minimize this possibility, we use a portfolio of composition algorithms and leverage the *complementary performance* property to use algorithm selection techniques for service composition.

Composition algorithms need to be executed periodically to ensure that QoS needs are consistently being met. This is because services may evolve internally; leading to unanticipated changes in their QoS values. In such scenarios, service (re-)selections must be carried out using metaheuristic algorithms. To determine which algorithm is suited to compose a request, algorithm selectors are trained on historical data collected for previously composed user requests. These selectors learn to model algorithm behavior on different composition tasks and select a composition algorithm that fulfills QoS requirements while minimizing time and memory. We build on the work in [2] to show that a trained Multi-Layer Perceptron (MLP) model achieves 75% accuracy and F1-score on a dataset consisting of 3840 composition instances labeled with one of three algorithms - Ant Colony System (ACS), Genetic Algorithm (GA) and Multi-Constrained Shortest Path (MCSP). Note that MCSP is an exact algorithm and is also useful for those instances on which ACS and GA do not converge. The final labeled dataset contains 1822 MCSP, 1628 GA and 390 ACS instances. Compared to the Single Best Solver (SBS), which is MCSP, we observe from Table 1 that the MLP selector chooses algorithms that are 44% faster and use 26.9% less memory on average.

While these results hold potential for applying algorithm selection to service composition, the caveat is that labeled training data is available. Gathering training data during deployment/runtime is infeasible due to two factors (1) Changes in services' QoS can happen in unforeseen ways. While QoS attributes over several timesteps can be collected by invoking services to gather real-time information, this is a resource-intensive process. (2) Executing all composition algorithms on every composition task is infeasible, because solutions for composition tasks must be computed fast at run-time. To address these issues, we propose the use of transfer learning to train classifiers on similar problems and use it to select algorithms for service composition at run-time. We consider

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07.
https://doi.org/10.1145/3449726.3462736

a similar problem to be the Traveling Salesman Problem (TSP) because service composition problems are often formulated as TSP instances [2], in which each service is a node to be visited. Specifically, we aim to explore the use of domain-invariant techniques to learn common feature representations given certain labels.

Table 1: Mean Computation Resource Usage

Approach	Mean Time (s)	Mean Memory (kB)
VBS	659.84	110039.11
MLP	481.11	102074.27
SBS	855.38	139655.80

2 TECHNICAL APPROACH

In certain transfer learning tasks, learning a common subset of invariant features for both source and target domains can improve generalization [4, 7]. Invariant features representations are certain intermediate features for which the conditional label distribution does not change across domains. Our goal is to first learn these invariant representations using a classifier on labeled TSP samples and generalize to the service composition domain. We hypothesize that there exist common, invariant features that can be used to model algorithm performance and that this relationship does not change regardless of problem domain. Certain features characterizing the size of the graphs such as number of nodes and distribution of fitness values are important in selecting which algorithm to use. For example, MCSP takes a long time to find solutions on graphs with larger search spaces than GA does. In addition to features indicative of structure and size of graphs, we consider a set of features characterizing the fitness landscape from [5]. At a minimum, we hypothesize that features characterizing graph size and structure will be invariant amongst both TSP and service composition domains, as training samples from both domains will be carefully selected. To test this hypothesis, we formulate the following research questions:

- **RQ1**: How well can a classifier trained on TSP instances generalize to service composition instances using domain invariant learning techniques?
- **RQ2**: Which algorithm behavior features demonstrate an unchanging relationship with the best performing algorithms on TSP and service composition tasks?

Our rationale for this study is based on recent advances using fitness landscapes as a predictor of algorithm performance. For most optimization problems, a fitness function is used to combine and denote the objectives of the problem instance being solved. Fitness functions offer a convenient way to compare different solutions and select one that meets constraints. Additionally, studying the characteristics of a problem function using fitness functions provides insight into how search algorithms may perform [5]. For example, search spaces with several local optima clustered together may lead to poor convergence. Ultimately, the fitness landscape does not depend on the problem instance, but instead depends on the fitness function used and the encoding of the problem. Thus, the choice of fitness function and the way a problem is encoded is important because it affects algorithm performance. Several features are discussed in [5] to characterize fitness landscapes. We use features that can be independently computed without needing to run other algorithms. We consider the following features from [5] for our analysis - autocorrelation function, correlation length, fitness statistics and distance correlation, dispersion metric, information landscape hardness measure, first and second order entropy measures, hamming distance in a level, hamming distance between a level and neutral walk.

To answer our research questions, we propose the study of two techniques: Domain-Invariant Component Analysis (DICA) [6] and the invariant model based on causal learning proposed in [7]. DICA aims to find a set of components whose distributions are common to both source and target domains. The core idea is that while the feature distribution may vary across domains, the functional relationship between the labels and features remains unchanged. In our context, it would mean that DICA is able to accurately model the performance of ACS and GA across different landscapes and use it to predict which might perform better on a certain problem instance. However, this assumes that the TSP and service composition instances we model are similar not only in terms of size, but also in terms of the fitness landscape. To address this, we propose the use of the L_2 distance metric and a discrete encoding across both domains. In addition to this technique, we propose modeling the relationship between the algorithm performance and features as a causal relationship, as proposed in [7]. In [7], the core assumption is that only a subset of features is invariant and can be causally modeled. This is a weaker assumption than in DICA, but powerful because it allows for the consideration of other features and feature-values that may not be present in both domains, even though they do not provide any guarantees.

In conclusion, in this paper we have outlined the challenges associated with using algorithm selection techniques in a dynamic service composition environment. This study is designed to leverage recent advances in understanding metaheuristic behavior, applying it for classification and generalizing the insights learned to a service composition domain. We propose using two transfer learning techniques to model metaheuristic algorithm behavior across varied problem instances in the source(TSP) domain and apply it to the service composition domain to improve classification accuracy and thus improve composition algorithm performance.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Award No. 1943002.

REFERENCES

[1] Edvinas Byla and Wei Pang. 2020. DeepSwarm: Optimising Convolutional Neural Networks Using Swarm Intelligence. In *Advances in Computational Intelligence Systems*. Springer International Publishing, Cham, 119–130.
[2] Niranjana Deshpande and Naveen Sharma. 2020. Composition Algorithm Adaptation in Service Oriented Systems. In *Software Architecture*. Springer International Publishing, Cham, 170–179.
[3] Giovani Guizzo, Federica Sarro, Jens Krinke, and Silvia Regina Vergilio. 2020. Sentinel: A Hyper-Heuristic for the Generation of Mutant Reduction Strategies. *IEEE Transactions on Software Engineering* (2020), 1–1. https://doi.org/10.1109/TSE.2020.3002496
[4] Wouter M. Kouw and Marco Loog. 2019. An introduction to domain adaptation and transfer learning. arXiv:1812.11806 [cs.LG]
[5] Katherine M. Malan and Andries P. Engelbrecht. 2013. A survey of techniques for characterising fitness landscapes and some possible ways forward. *Information Sciences* 241 (2013), 148–163. https://doi.org/10.1016/j.ins.2013.04.015
[6] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. 2013. Domain Generalization via Invariant Feature Representation. arXiv:1301.2115 [stat.ML]
[7] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. 2018. Invariant Models for Causal Transfer Learning. *Journal of Machine Learning Research* 19, 36 (2018), 1–34. http://jmlr.org/papers/v19/16-432.html