Chapter 1

Recent Developments of Automated Machine Learning and Search Techniques

Rong Qu

Abstract

The recent successes of artificial intelligence, in particular machine learning, for solving real-world problems have motivated the advances towards automated design of algorithms and systems with less human involvement. In machine learning and meta-heuristic search algorithms, different lines of relevant research are now emerging, with findings feeding into each other. This book presents a selection of some recent advances across automated machine learning (AutoML) and automated algorithm design (AutoAD), where the effectiveness and efficiency of techniques and algorithms has been enhanced with the support of new taxonomies, models, theories, as well as frameworks and benchmarks. The emerging new lines of exciting research directions in AutoML and AutoAD present new challenge across multiple research communities in machine learning, evolutionary computation and optimisation research.

1.1 Introduction

With the recent fast developments of artificial intelligence in tackling practical problems comes an increasing demand of easy-to-use and general tools and intelligent methods with less human involvement for solving new problems. These include automated machine learning (AutoML) and automated algorithm design (AutoAD) in a broad range of application domains. This book selects some of the latest developments across AutoML and AutoAD, and presents key challenges and research issues, calling for and encouraging further advances towards automated intelligent algorithms and systems for solving more new real-world problems.

The wide range of recent developments range from the automated design of heuristics [12, 10, 20] and control software [2] using meta-heuristics and genetic programming [8] to automated design of neural architectures [23] and classifier algorithms [16] using evolutionary algorithms. Application problems include mainly combinatorial optimisation [12, 10, 20, 8] and classification problems [23, 16, 3].

Within the context of automated design of machine learning and meta-heuristics search algorithms discussed in Section 3.2, some interesting research issues are emerging, posing challenges on future advances across different disciplines of machine learning, evolutionary computation and optimisation research as discussed in Section 1.3. The chapter concludes in Section 3.6 by summrising developments and challenges, and encouraging future research on several directions.

1.2 Automated Algorithm Design and Machine Learning

Until recently, the success of most intelligent algorithms and systems heavily replies on the extensive human expertise, which often highly depends on experts' skills on making various decisions. These include, at a lower level, how to fine-tune the parameters or settings of the chosen algorithms or models; and at a higher level, how to select the most appropriate algorithms or system architecture for solving the problems at hand. The algorithms or systems designed manually in an ad-hoc manner are also often problem-specific, requiring a large amount of effort adapting existing algorithms or re-designing new algorithms. These algorithms are often discarded after problem solving, wasting extensive human resources.

With the fast developments in machine learning, there is now evidence that AutoML [5] is achievable [3] in both research and practice. There exist highly effective machine learning methods ready for use for non-experts with limited knowledge. Based on the definitions of components in AutoML, the brief review in Chapter 2 on the AutoML methodologies in supervised learning provides a nice complimentary introduction to the field [3]. Of particular interest is that evolutionary computation naturally plays an important role in *Optimizer*, one of the three important components in AutoML methods. Some of the challenges identified in Chapter 2, e.g. large scale optimisation and transfer of learning in AutoML, require further collaboration and integration of machine learning and computational search algorithms.

Chapter 8 provides an excellent example of the latest developments in AutoML integrating machine learning and evolutionary algorithms [23]. The computational expensive offline optimisation in neural architecture search (NAS) is extended to federated learning within distributed real-time systems for edge devices. With the focus on reducing computational costs, two NSGA-II based multi-objective evolutionary algorithms have been investigated to automatic online NAS for image classification. Much more research remains to be addressed with evolutionary computation for NAS, including the model averaging aggregation for tasks of different features and security of privacy leakage.

Research in AutoAD comes along another line of developments in optimisation research. In building a standard towards automated algorithm design, Chapter 3 presents a new taxonomy to categorise relevant research into three streams, namely automated algorithm configuration, algorithm selection and algorithm composition [18]. With different decisions of parameters, algorithms and algorithm components, automated algorithm design can be defined as an optimisation problem exploring a search space of these decisions. A new model named General Combinatorial Optimisation Problem (GCOP) is defined [19], where elementary algorithm components are considered as decision variables in the search space of algorithms. Chapter 3 also demonstrates that various meta-heuristics and selection hyper-heuristics can be defined with this unified general GCOP model.

In designing control software for robot swarms, Chapter 5 presents a modular principle in AutoMoDe to automatically assemble predefined parametric modules [2]. By exploring a search space of all possible low-level individual behaviors of robots, AutoMoDe optimises the performance of their collective high-level behaviors. This modular principle presents an interesting contrast to that of the GCOP model in Chapter 3, where the design of meta-heuristic algorithms is defined as an optimisation problem upon the search space of elementary algorithmic components [18]. AutoMoDe automatically selects, combines and fine-tunes predefined modules to design control software offline by using Iterated F-race [1], a highly successful framework for automated configuration of meta-heuristics for combinatorial optimisation problems.

Chapter 4 presents an analysis and overview of time complexity and learning in selection hyper-heuristics for function optimisation [12]. It is evidenced that mixing multiple low-level heuristics and switching acceptance criteria are necessary to achieve optimal performance. Furthermore, adaptive learning is crucial in selecting low-level heuristics at different stages of hyper-heuristics. The importance of comprehension in automated selection of low-level heuristics is highlighted in Chapter 7, where powerful tools in machine learning might be of great support. Time complexity is less studied in the existing literature but is a fundamental issue to underpin algorithm design. With analysis on time and landscapes, theoretical studies can lead to more insights and knowledge on the algorithm behavior and performance.

While Chapter 4 focuses on selection hyper-heuristics [12], Chapters 6, 9 and 10 concern generation hyper-heuristics, presenting interesting findings for both combinatorial optimisation [8, 20] and classification [16] problems. Compared to the extensively studied selection hyper-heuristics, generation hyper-heuristics are relatively less studied. They focus on automatic generation of heuristics themselves, thus removing human involvement in providing problem-specific low-level heuristics. Problem attributes need to be provided instead, in addition to a set of general operators / grammars. Genetic programming and grammatical evolution have been employed, addressing interesting research issues of common solution representation [20] in Chapter 6, knowledge transfer [8] in Chapter 9, and fitness landscape [16] in Chapter 10.

Chapter 10 concerns the automated design of classifier algorithms, where chromosomes in grammatical evolution consist of design decisions for different classification tasks [16]. In contrast to the automated search of neural architectures in Chapter 8, this presents a different perspective of integrating evolutionary computation and machine learning in AutoML. With the fitness analysis on genetic algorithm and grammatical evolution, it is also interesting to reveal the different features of the landscapes for multi-class classification and binary classification. In the literature, theoretical analysis such as fitness landscape or time complexity received less research attention, however, is crucial in sustaining the fundamentals towards effective AutoML [3] and AutoAD [18].

In Chapter 6, an ”intermediate” graph-based solution representation is studied for designing constructive and perturbative heuristics for highly different combinatiorial optimisation problems [20]. The research makes an important step towards further removing expert’s involvement in defining solution encoding, to which the problem-specific heuristics are applied. Defining common problem encoding is often neglected in the literature, however, is highly important in AutoAD. Further advances on extending the scope of cross-domain general solution encoding is crucial in sharing and retaining knowledge of automatically designed effective algorithms addressing different problem domains of common structures.

Chapter 9 highlights the importance of knowledge transfer [8], which also receives less attention in the literature of hyper-heuristics, and also meta-heuristics. In guiding the search directions, it is shown that reusing subtrees in the initialisation is more effective than feature importance evolved automatically by genetic programming. Challenging issues remain, including the lack of understanding on truly useful subtrees and building blocks, and the handling of redundant branches (i.e. issue of bloating) in genetic programming.

Chapter 7 presents an insightful overview of hyper-heuristics within the context of autonomous problem solvers [10]. One interesting research issue discussed is on the set of low-level heuristics, which is still hand-picked by human experts for particular problems; while a set of strong problem-specific heuristics does not always warranty strong performance of hyper-heuristics. This echos the definition of a set of elementary algorithm components in GCOP in Chapter 3, which could also deliver strong performance if simple components are composed effectively [19]. The observations are also supported by the theoretical proof in Chapter 2, where multiple low-level heuristics are necessary to achieve optimal performance of hyper-heuristics.

1.3 Challenges in Automated Design of Algorithms and Machine Learning

Recent developments in AutoML have achieved great success in solving various real-world problems based on advanced research, see surveys in [5] and Chapter 4. In AutoAD, different lines of research emerge on automated configuration [21] and automated selection [5] of algorithms. Based on the new taxonomy of algorithm design, another line of research on automated algorithm composition has also been defined [18], where hyper-heuristics represent a subset of such methods. In the autonomy of algorithms and techniques in artificial intelligence, it is interesting to see that different lines of developments integrating machine learning and evolutionary computation are emerging, underpinning each other to address wider range of problems.

It is difficult to review exclusively the latest advances in AutoML and AutoAD across multiple disciplines, however, some of the key challenges and research issues can be identified based on the blend of the latest developments presented in this book. These include in particular the compreshension and interpretability of the algorithms/techniques [12, 8] and theoretical studies [10, 3, 18], which are still neglected although frequently mentioned in the literature. Research addressing these challenges all underpin further advances in AutoML and AutoAD.

Theoretical Fundamentals

Theoretical analysis is relatively less concerned in AutoAD. The overview of time complexity in Chapter 2 on selection hyper-heuristics proves the necessity of multiple low-level heuristics and acceptance criteria, as well as adaptive learning in selecting effective low-level heuristics [6]. In generation hyper-heuristics, it is highly interesting that the fitness landscapes for genetic algorithms and grammatical evolution present different ruggedness for binary and multi-class classifications [16]. Hyper-heuristics haven shown to achieve to some extent free lunches [22], and are more general than some algorithms [10, 17]. More findings with rigorous theoretical analysis will further underpin the fundamentals and better understanding in AutoAD.

While well-defined models and architectures exist in AutoML, in meta-heuristic algorithms, there is a lack of common models or frameworks. Most frameworks in AutoAD are defined descriptively, lacking a fundamentally consistent structure and standard. More research is needed towards modeling and standardising algorithm design [18]; otherwise many of the research findings remain local [10] for specific problems, or are discarded in the rich but scattered literature. Some progress has been made, including the highly successful Iterated F-race framework for automated algorithm configurations [1] and the widely adapted HyFlex platform [11] and EvoHyp [13] in hyper-heuristics [14]. In [19], a new taxonomy is defined based on the decisions considered in AutoAD, and supports the development of common models such as GCOP in Chapter 3 in automated design of general search algorithms.

The idea of modularity presents an interesting principle in designing both control software in Chapter 5 and meta-heuristic algorithms in Chapter 3. In the two distinctive domains, with the search spaces of behaviors and algorithmic components, respectively, the automated design of control software and search algorithms can both be defined as optimisation problems, where modules and components are automatically assembled and optimised. Existing powerful optimisation platforms and frameworks including Iterated F-Race [1] and HyFlex [11] can also be adapted to quickly implement the optimisation, and potentially sustain knowledge sharing across different disciplines and application domains.

In AutoAD, establishing general and common solution encoding presents another challenge for different combinatorial optimisation problems. In Chapter 6, a graph-based encoding shows to be successful representing multiple different problems. With common encoding, knowledge and expertise in AutoAD could be accumulated and retained in a consistent structure for comparable and transferable investigations in different domains.

Interpretability, Reusability and Generality

AutoML demands extensive search, generating a vast amount of information [3]. Extensive information also exists in AutoAD on designing or generating effective algorithms. The relatively well-structured machine learning models can support analysis on designing effective systems thus to extract reusable knowledge. However, this is not the case in search algorithms, where there is a lack of common frameworks or unified models, based on which systematic analysis could be conducted to identify explainable or reusable knowledge. In both AutoML and AutoAD, the extensive information is yet to be collected in consistent structures and analysed to extract transferable and reusable knowledge in designing effective search algorithms and machine learning systems.

While search algorithms have been criticised with lacking theoretical support, explainability and interpretability are often known as unsolved challenges in machine learning. Some attempts have been made to transfer the information evolved in genetic programming [8, 20] into knowledge of generating heuristics. With the consistent data structures in genetic programming, potential knowledge in grammars [20], subtrees and feature importance [8] could be retained and reused in designing effective algorithms addressing similar or even different tasks or problems. More collaborative efforts across disciplines are needed in both AutoML and AutoAD to reveal new transferable knowledge thus to enhance the reusability and generality of algorithms.

In retaining and reusing knowledge evolved automatically, the issue of common solution encoding is understudied, although the search of algorithms are highly dependant on solution representation. The scope of general encoding for multiple domains remains an interesting research issue; while the common graph-based representation in Chapter 6 presents a promising step towards reusing some general properties and knowledge in the automatically generated grammars for highly different combinatorial optimisation problems, and potentially a diverse range of other problems.

Generality of algorithms, although widely mentioned in hyper-heuristics [14], is still often neglected in the literature. Chapter 7 considers generality of algorithms at a higher level with three criteria, namely across multiple problems, distinctive heuristic sets, and varying experimental conditions. Based on a new taxonomy, a four-level assessment of algorithm generality [15] has been defined upon problem domain, problems, instances and benchmark set from a multi-objective perspective. In operational research, the well-established benchmarks (e.g. the OR Library[1]) provide excellent problem sets for generality assessments in designing effective algorithms.

With further advances of interpretability, reusability and generality, the reuse the methods in AutoML and AutoAD with less human involvement will lead to enormous savings of human effort and accumulate continuous and consistent research developments.

---

1 http://people.brunel.ac.uk/~mastjjb/jeb/info.html

Integration of Machine Learning and Optimisation Research

The integration of research outcomes across machine learning and optimisation research has enhanced the efficiency of algorithms, advances fed into each other to address various issues in AutoML and AutoAD. Evolutionary computation has been successfully applied in the intensive search in machine learning, enhancing the optimisation in AutoML. For example, in AutoML, powerful evolutionary algorithms have been used in federated learning to reduce computational costs to improve neural architecture search [23].

In AutoAD, feature engineering in machine learning may contribute to identifying the key attributes of search spaces or problems when designing search algorithms e.g. genetic programming. In hyper-heuristics, the selection of low-level heuristics, parametric modules or algorithmic components can be naturally supported by offline or online learning [9]. The great success recently in machine learning means efficient models can be easily adapted in learning the knowledge in AutoAD, enhancing the comprehension of algorithm performance.

Benchmarking and Competitions

As reviewed in [3], the series of AutoML challenges, from the *Prediction challenge [4]* since 2006 to the most recent *AutoDL* competition [7] in 2020, not only led to some highly effective and popular AutoML methods, but also set consistent standards, boosting advanced research in AutoML. In AutoAD, platforms and frameworks such as Iterated F-Race [1], HyFlex [11] and Evo-Hyp [13] have also been widely adapted by more researchers. However, significant standalone research presumably still stays local and hidden, and has made limited contributions to the literature [10] without benchmarking and standardising the outcomes.

1.4 Conclusions

With the recent successes in machine learning and optimisation research, researchers are now exploring the scope of designing effective algorithms or intelligent methods with less human involvement, towards automated machine learning (AutoML) and automated algorithm design (AutoAD). Promising findings have emerged at the interface of different disciplines, and outcomes have fed into each other, addressing a broad range of research issues and leading to new challenges in AutoML and AutoAD.

With the well-structured pipelines and models in machine learning, powerful optimisation algorithms have been successfully adapted in evolutionary computation to enhance the efficiency of search for either hyperparameters or neural architectures in AutoML. With the frameworks and large amount of datasets, outcomes can be effectively accumulated, establishing further comprehension of machine learning. Challenges in AutoML now remain to be on the interpretability of the models, which also represents a key issue of explainable AI in machine learning communities.

In AutoAD, different streams of research advances have resulted into automatically designed algorithms which outperform some manually designed algorithms. However, there is still a lack of theoretical studies, for example on general standards and models, as well as common problem encoding for different problems. The establishment of these fundamentals are important so that research findings are accessible across the different communities with common structures, and not remain hidden or locally. Some efforts have been made in building new taxonomies and models, although there is still a scope of further collaboration in machine learning and evolutionary computation, impacting on real-world problems.

References

1. M. Birattari, Z. Yuan, P. Balaprakash, and T. Stützle. F-race and iterated F-race: An overview. In *Experimental Methods for the Analysis of Optimization Algorithms,* pages 311–336. Oct. 2010.
2. Mauro Birattari, Antoine Ligot, and Gianpiero Francesca. Automode: a modular approach to the automatic off-line design and fine-tuning of control software for robot swarms. In *Chapter 5, Automated Design of Machine Learning and Search Algorithms.* 2020.
3. Hugo Jair Escalante. Automated machine learning - a brief review at the end of the early years. In *Chapter 2, Automated Design of Machine Learning and Search Algorithms.* 2020.
4. Isabelle Guyon, Amir Reza Saffari Azar Alamdari, Gideon Dror, and Joachim M. Buhmann. Performance prediction challenge. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2006),* pages 1649–1656. Vancouver, BC, Canada, July, 2019.
5. F. Hutter, L. Kotthoff, and J. Vanschoren (eds.). *Automated Machine Learning: Methods, Systems, Challenges.* Springer, 2019.
6. A. Lissovoi, P. S. Oliveto, and J.A. Warwicker. Simple hyper-heuristics can control the neighbourhood size of randomized local search optimally for leading ones. *Evolutionary Computation,* in press, 2019.
7. Zhengying Liu, Isabelle Guyon, Julio Jacques Junior, Meysam Madadi, Sergio Escalera, Adrien Pavao, Hugo Jair Escalante, Wei-Wei Tu, Zhen Xu, and Sebastien Treguer. Autocv challenge design and baseline results. In *In CAp 2019 - Conference sur l ´Apprentissage Automatique,.* July, 2019.

8. Yi Mei, Mazhar Ansari Ardeh, and Mengjie Zhang. Knowledge transfer in genetic programming hyper-heuristics. In *Chapter 9, Automated Design of Machine Learning and Search Algorithms.* 2020.
9. W. Meng and R. Qu. A survey of learning in automated design of search algorithms. *IEEE Computational Intelligence Magazine,* under review.
10. Mustafa Misir. Hyper-heuristics: Autonomous problem solvers. In *Chapter 7, Automated Design of Machine Learning and Search Algorithms.* 2020.
11. G. Ochoa, M. Hyde, T. Curtois, J.A. Vazquez-Rodriguez, J. Walker, M. Gendreau, G. Kendall, B. McCollum, A.J. Parkes, S. Petrovi, and E.K. Burke. HyFlex: A benchmark framework for cross-domain heuristic search. In *Proc. of Evolutionary Computational Combinatorial Optimization,* pages 136–147, Málaga, April 11-13, 2012.
12. Pietro S. Oliveto. Rigorous performance analysis of hyper-heuristics. In *Chapter 4, Automated Design of Machine Learning and Search Algorithms.* 2020.
13. N. Pillay and D. Beckedahl. EvoHyp - a Java toolkit for evolutionary algorithm hyper-heuristics. In *Proc. of IEEE Congress on Evolutionary Computation,* pages 2707–2713. San Sebastian, June 5-8, 2017.
14. N. Pillay and R. Qu. *Hyper-heuristics: Theory and Applications.* Springer Nature, 2018.
15. N. Pillay and R. Qu. Assessing hyper-heuristic performance. *Journal of Operational Research Society,* accepted, 2020.
16. Nelishia Pillay and Thambo Nyathi. Automated design of classification algorithms. In *Chapter 10, Automated Design of Machine Learning and Search Algorithms.* 2020.
17. R. Poli and M. Graff. There is a free lunch for hyper-heuristics, genetic programming and computer scientists. In *European Conf. on Genetic Programming,* pages 195–207, Tubingen, April 15-17, 2009.
18. R. Qu, G. Kendall, and N. Pillay. The general combinatorial optimisation problem - towards automated algorithm design. *IEEE Computational Intelligence Magazine,* 15(2):14–23, April, 2020.
19. R. Qu, G. Kendall, and N. Pillay. The general combinatorial optimisation problem - towards automated algorithm design. *IEEE Computational Intelligence Magazine,* 15:14–23, May, 2020.
20. Christopher Stone, Emma Hart, and Ben Paechter. A cross-domain method for generation of constructive and perturbative heuristics. In *Chapter 6, Automated Design of Machine Learning and Search Algorithms.* 2020.
21. T. Stützle. Automated algorithm configuration: Advances and prospects. In *Camacho D., Braubach L., Venticinque S., Badica C. (eds) Intelligent Distributed Computing VIII. Studies in Computational Intelligence vol 570.* Springer, Cham, 2015.
22. D. H. Wolpert and W. G. McReady. No free lunch theorems for optimisation. *IEEE Transactions on Evolutionary Computation,* 1(1):67–82, April, 1997.
23. Hangyu Zhu and Yaochu Jin. Towards real-time federated evolutionary neural architecture search. In *Chapter 8, Automated Design of Machine Learning and Search Algorithms.* 2020.