The Algorithm Selection Problem on the Continuous Optimization Domain

Mario A. Muñoz, Michael Kirley, and Saman K. Halgamuge

Abstract. The problem of algorithm selection, that is identifying the most efficient algorithm for a given computational task, is non-trivial. Meta-learning techniques have been used successfully for this problem in particular domains, including pattern recognition and constraint satisfaction. However, there has been a paucity of studies focused specifically on algorithm selection for continuous optimization problems. This may be attributed to some extent to the difficulties associated with quantifying problem "hardness" in terms of the underlying cost function. In this paper, we provide a survey of the related literature in the continuous optimization domain. We discuss alternative approaches for landscape analysis, algorithm modeling and portfolio development. Finally, we propose a meta-learning framework for the algorithm selection problem in the continuous optimization domain.

1 Introduction

A continuous optimization problem is such that, given a function $f : \mathbb{R}^n \mapsto \mathbb{R}$, we want to find $\mathbf{x}^\star = \arg\min f(\mathbf{x})$. When solved in a computer, a *search algorithm* samples from the very large but finite *search set*, $\mathcal{X} \subset \mathbb{R}^n$. Each observation $\mathbf{x}_i \in \mathcal{X}$ has an associated output value $y_i \in \mathcal{Y}$ such that $y_i \approx f(\mathbf{x}_i)$, where $\mathcal{Y} \subset \mathbb{R}$ is the *objective set*. The algorithm aims to find one or more candidate solutions $\mathbf{x}_o \in \mathcal{X}, y_o \approx f(\mathbf{x}_o)$, such that $|y_o - y^\star| \ll \delta$, where $y^\star = f(\mathbf{x}^\star)$ and $\delta \to 0$. It is expected that the algorithm produces a solution of acceptable quality after a bounded number of function evaluations. The opposite is known as *premature convergence*.

Mario A. Muñoz · Saman K. Halgamuge
Department of Mechanical Engineering,
The University of Melbourne, Parkville, Victoria, 3010, Australia
e-mail: mariom@student.unimelb.edu.au, saman@unimelb.edu.au

Michael Kirley
Department of Computing and Information Systems, The University of Melbourne,
Parkville, Victoria, 3010, Australia
e-mail: mkirley@unimelb.edu.au

C. Moewes et al. (Eds.): Computational Intelligence in Intelligent Data Analysis, SCI 445, pp. 75–89.
springerlink.com

Premature convergence is related to the nature of the search algorithm, as each algorithm exploits differently the information obtained by sampling f. Therefore, unless some restrictions are in place, it is optimistic to expect that an algorithm would work well across a wide range of functions [78]. Due to the plethora of available algorithms, it is non-trivial to know which one is able to exploit the information more efficiently [29]. This is an instance of the well known algorithm selection problem. In this paper we propose a framework based on meta-learning for the algorithm selection problem. For this purpose, we review the literature about the different stages of the new framework — namely landscape analysis, meta-learning models and algorithm portfolios. Then, we outline the requirements for implementation of the new framework.

The paper is organized as follow: Section 2 presents the algorithm selection problem for continuous optimization, and the related parameter tuning problem. Section 3 describes the characteristics that make an optimization problem difficult and it reviews different methods for landscape analysis. Section 4 discusses how machine learning techniques have been employed to solve the algorithm and parameter selection problems. Section 5 analyzes the related works in algorithm portfolio design. Section 6 presents our meta-learning based framework for the algorithm selection problem. Finally, Section 7 discusses avenues for further research.

2 Algorithm Selection

Rice [56] defined the algorithm selection framework as a loose methodology that relates problems and solution methods through performance and problem characteristics. This framework did not provide specific methods for implementation, which is one of the reasons it has not been thoroughly explored. However, in the last decades, meta-learning has been favored as implementation method with demonstrated success in different problem domains [62]. Meta-learning exploits data obtained from previous experiments by constructing models that can be used for prediction, using machine learning techniques [28]. Figure 1 presents a summary of this implementation adapted to continuous optimization problems. In this figure, $\mathcal{F}$ is the very large, amorphous, high dimensional and hard to define *function set*, for which $f \in \mathcal{F}$. Let $\mathcal{A}$ be the large and diverse *algorithm set*, and $a \in \mathcal{A}$ be one of the many algorithms capable of searching for $\mathbf{x}_o$ in $\mathcal{X}$. The cost of running a in f can be measured by a function $\rho(f,a)$. Let $\mathcal{P} \subset \mathbb{R}$ be the set of feasible values of $\rho(f,a)$, called the *performance set*. Then, the algorithm selection problem is to find $a_o = \arg\min \rho(f,a)$ with f constant. It is noteworthy to point out that this problem cannot be solved directly. Hence, let $\mathcal{C} \subset \mathbb{R}^m$ be the *set of function characteristics*. This set includes known attributes of f such as the dimension, but also measurements about the occurrence of certain structures known to pose difficulties for a [57, 73]. Characteristics are important as they provide some order and coherence to the complicated problem space by imposing a lower dimensional coordinate system [57]. Characteristics can be calculated through user defined functions known as *landscape analysis* methods, $c(\mathbf{x},y)$. These functions should be designed such that varying complexities are

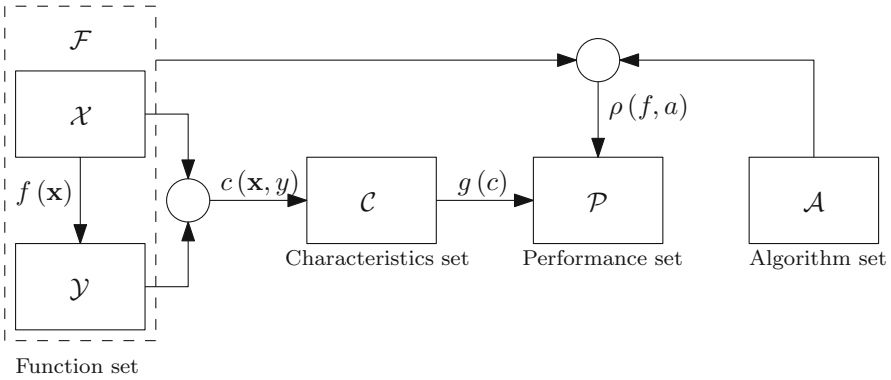


Fig. 1 Summary of the algorithm selection framework for the continuous optimization domain

exposed, structural properties are captured, and advantages and limitations of the different algorithms are related to them.

Let $g : \mathcal{C} \mapsto \mathcal{P}$ be a function that forecasts the performance based on the characteristics. Consider the existence of a subset of functions from $\mathcal{F}$ such that we know the values of $c(\mathbf{x}, y)$ and $\rho(f, a)$ for all $a \in \mathcal{A}$. Then, it is possible to use a machine learning technique to identify the function g. These *empirical performance models* provide a way to forecast the performance of an algorithm when a new problem is presented. The whole process can be automated if the results of several models are compared through an objective procedure.

The algorithm selection framework does not consider the algorithm parameter, θ, which controls the way that the search is carried out. This parameter can potentially adapt a to f if it is properly tuned, and it can appreciably change the overall performance [8]. This implies that an optimal θ for one function might not be appropriate for others [35, 50]. Choosing θ for a given a is a time-consuming and non-trivial task, and considerable effort has gone into developing methods for parameter selection that can be categorized as *parameter tuning* and *parameter control* [16]. Tuning keeps the parameters constant during the run, while control modifies them. Both approaches have advantages and disadvantages that have been thoroughly discussed in the literature [16, 35, 50].

Meta-learning is compatible with both parameter tuning and control [62]. If two instances of the same algorithm differ only in one parameter, we can consider them as two completely different algorithms [57]. This approach was followed by Hutter *et al.* [31, 30, 32] for tuning randomized algorithms in the context of boolean satisfiability problems, and by Muñoz *et al.* [45] for tuning the Covariance Matrix Adapted Evolutionary Strategy. Therefore, parameter selection can be seen as a component of the algorithm selection problem. As such, assume that g is not only dependent of the landscape characteristics c but also from θ. In fact, if we assume that c and θ are representations of f and a respectively, then $\rho(f, a) \equiv g(c, \theta)$.

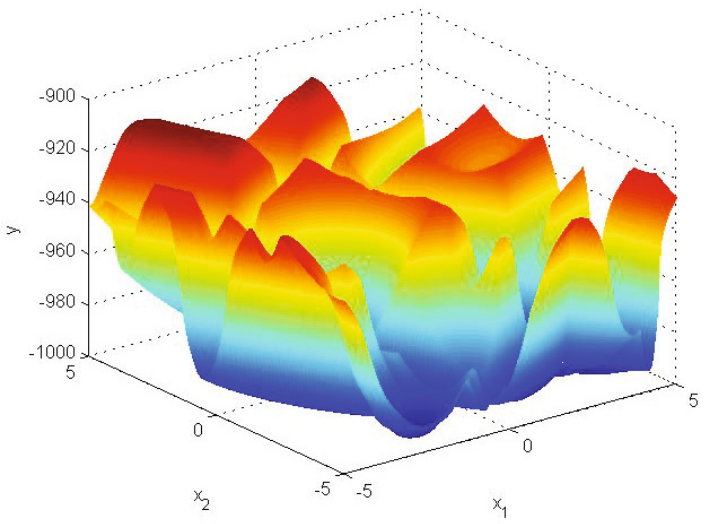
Even though meta-learning provides a clear avenue for implementation, every problem domain has specific issues to be considered, e.g. characteristic quantification, methods for selecting algorithms, and issues with uncertainties. One of the most important issues is the characteristic quantification. It is important to have a good understanding of what makes a continuous optimization difficult. For that purpose, in the next section we will discuss the *search landscape* metaphor and different methods used in landscape analysis of optimization problems.

3 Landscape Analysis

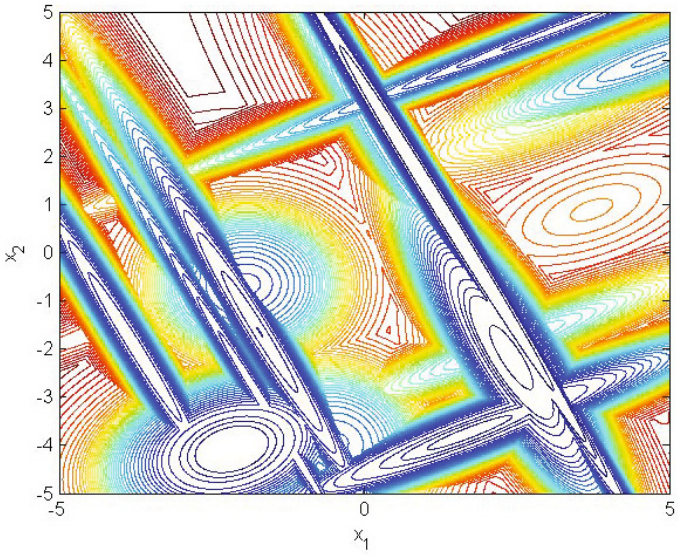
To understand what makes an optimization problem difficult, we employ the search landscape metaphor. Consider a surface in a three dimensional space composed of ridges, valleys and basins, such as the ones shown in Fig. 2. In this surface the highest or deepest areas represent the optimal points. The objective of the search is to navigate the surface until such areas are found. This metaphor helps us to understand what is needed for a successful search [53], and allows us to describe features in the landscape that are influential, even though most practical problems will have a dimension several orders of magnitude larger than two. These features — which are defined on detail in [44] — can be described qualitatively in cases where the knowledge about the function is complete. However, in cases where the only information available are the pairs $(\mathbf{x}_i, y_i)$ these attributes are usually unknown. Therefore, a *landscape analysis* technique is used to provide a measure that quantifies one or several attributes.

A number of landscape analysis methods have appeared in the past two decades. Table 1 presents a summary of some well-known landscape analysis methods and their underlying concepts. The measures have been classified into two groups: global and local. The former takes the whole sample to produce the measure while the later calculates the average of evaluating a condition over each observation and its neighborhood. Global measures have the advantage that samples extracted during an experiment can be reused to calculate different measures. However, they do not provide details about the locality of the landscape. Nevertheless, local measures can become intractable when the sample is too large, as each observation has to be analyzed independently. Other disadvantage of local measures is that samples obtained in one type of experiment are not reusable, e.g. time series measures require that a random walk experiment, while the basin of attraction measures require a local search experiment. Samples could be reusable if an intermediate processing step is placed. In previous work [44] we have demonstrated a procedure to calculate local measures from scattered data —extracted using random sampling— for two dimensional problems. However, this approach is not scalable. This is due to the sensitivity that local measures have to the neighborhood definition.

Other authors have identified limitations in the landscape analysis methods. Their application requires a sufficient number of observations, which grows exponentially as the dimension of the search space increases. This establishes a difference between



(a)



(b)

Fig. 2 Landscape for Gallagher's Gaussian 21-hi Peaks function in two dimensions from the Comparing Continuous Optimization Benchmark. Figure (a) show a three dimensional rendering of the function, while Fig. (b) show a contour plot. This function is multimodal, without global structure, non-separable, homogeneous with medium sized anisotropic basins of attractions.

Table 1 Summary of some well-known landscape analysis methods. Local measures calculate the average of evaluating a condition over each observation and its neighborhood, while global measures use the whole dataset to produce the measure.

Type	Concept	Measure	References
Local	Time Series	Kolmogrov complexity	[9]
		Correlation length	[64]
		Information content of random walk	[71, 72]
		Random walk correlation function	[76, 77]
	Fitness clouds	Negative slope coefficient	[68, 69, 67, 70]
	Evolvability	Fitness distributions	[7]
		Locality	[21]
		Fitness evolvability portraits	[61]
	Landmarking	Basin of attraction distributions	[22, 12, 17, 18, 54]
	Markov Models	Basin of attraction estimations	[2]
	Others	Phase transitions	[1]
		Ruggedness coefficient	[3]
		Information Landscapes	[8]
		Path diversity	[10]
		Motif difficulty	[38]
		Fourier transformations	[59]
Global	Linear correlation	Fitness distance correlation	[34]
		Multiple correlation coefficient	[41]
	Epistasis	Epistasis variance	[15]
		Bit-wise epistasis	[19]
		Walsh Transformation	[27]
		Bit decidability	[48]
		Analysis of variance tables	[55]
		Epistasis correlation	[58]
		Entropic epistasis	[60]
	Other	Dispersion	[39]

theoretical results and empirical estimators, whose precision changes as the number of observations increases to infinity [33]. Hence, a large amount of computation has to be made to obtain precise estimators and theoretically they cannot be calculated in polynomial time [26, 46, 66]. This also explains why statistical analysis can be artificially "fooled" by giving a special weight to insufficiently sampled regions of the landscape [66]. However, statistical measures are by nature approximate. The real question is how much information is actually lost and if its possible to deal with such losses. Also, providing a single global measure to analyze a whole landscape is overly optimistic and several measures may be necessary [6, 46, 61]. There is, without a doubt, another form of the no free lunch theorem [78] at work in this situation.

Our interest is to obtain as much information possible for a reasonable expenditure of effort. However, care must be taken into selecting methods that provide co-linear measurements [75].

Note that even if the analysis is performed, the results could only be applicable to the current representation [40]. To solve this issue, it has been suggested to use the Metropolis algorithm to get an initial sample and extract the measures while the optimization algorithm is running [47]. However, the bias imposed by the search algorithm can produce deceiving results [44]. It is obvious that difficulty of a problem can only be measured relative to the algorithm used to solve it [27]. Hence, it is necessary to relate the landscape features to the search cost. Otherwise, the resulting measures fail to account for much, if any, of the variability on problem difficulty [75]. However, meta-learning provides an avenue to solve various of these difficulties.

We will continue the discussion on how landscape analysis fits into the general framework in Sec. 6. Also, we can find examples on how landscape analysis has been applied to create meta-learning models. The next section reviews this research area.

4 Meta-learning Models for Optimization Algorithms

As we mentioned in Sec. 2, meta-learning uses data obtained from previous experiments by constructing prediction models of the algorithm using a machine learning technique. Table 2 presents a summary related of works to meta-learning in the continuous optimization domain. Unlike *fitness prediction* [11] — where a model of the function is created, so only promising observations are actually evaluated — machine learning is used as a mean to identify relationships among functions and algorithms with the purpose of selecting algorithms, tuning parameters, or simply understanding the algorithm behavior.

Francois and Lavergne [20] suggested that statistical analysis, in particular regression, could be useful to identify trends in the algorithm behavior. Their experiments concluded that performance is a random variable that follows a gamma distribution. This affirmation was confirmed by Yeguas *et al.* [79]. Although in [20] it is proposed to relate algorithm classes to performance, there is not a specific methodology in how to determine such classes. This means that for each problem a new model has to be trained. Hence, the resulting models could not be realistically used for parameter tuning. A similar conclusion can be drawn of the works by Bartz-Beielstein *et al.* [4, 5].

Leyton-Brown *et al.* [36, 37] are one of the first to focus on the algorithm selection in the optimization domain. Their work using combinatorial problems demonstrated the practical application of meta-learning, and how it can be successful in actual applications. However, only deterministic algorithms were studied at this stage. The work of Hutter *et al.* [31, 30, 32] demonstrated that randomized algorithms also can be modeled following this approach. These works provide justification to the exploration of meta-learning into the continuous optimization domain.

Table 2 Summary of the application of meta-learning concepts in the optimization domain. It is noteworthy the paucity of works dealing with algorithm selection using meta-learning concepts for continuous optimization.

Problem	Application	Model type	Reference
Algorithm Selection	Scheduling	Bayesian Classifier	[13, 14]
		Linear regression	[42]
	Program induction	Linear regression	[24]
	Boolean satisfiability	Standard ridge regression	[31, 30]
		Random Forest	[32]
	Combinatorial auctions	Linear regression	[36, 37]
		Multivariate Adaptive Regression Splines	[36]
Parameter Selection	Continuous optimization	Generalized Linear Models	[20]
		Regression tree	[4, 5]
		Linear models	[79]
		Neural networks	[45]
	Program induction	Linear models	[24]
	Boolean satisfiability	Standard ridge regression	[30]

A set of models would allow the user to maintain an empirical database of problem-algorithm relationships in a compact format. The database would be useful to select a single algorithm to run or a group of algorithms that can be run sequentially or concurrently, with or without communication between each other. This type of collection is known as *algorithm portfolio* [23], which we will discuss in the following section.

5 Algorithm Portfolios

The concept behind algorithm portfolios is simple [51]: "Instead of betting the entire time budget in a single algorithm, how do we invest it in multiple algorithms?" In other words, a portfolio aims to improve on the performance of the component algorithms, in terms of expected computational cost and overall risk [23]. This concept has been explored for more than ten years [23], and it is closely related to the developments in *memetic algorithms* [43, 49], *hyper-heuristics* [25] and *hybrid algorithms* [73]. In general, a portfolio contains besides the algorithm set, a procedure called selector, whose purpose is to decide which a is the best for a given f [25, 28, 51, 52, 65, 73]. In some cases, the portfolio provides provides communication among algorithms through a *migration scheme* [51]. The portfolio approach has demonstrated computational advantages over individual algorithms particularly when high-variance methods are combined [23].

The performance of a portfolio depends of both its composing algorithms and the selector [49]. In fact, some portfolios are strictly preferable than others, as they provide a lower risk and also a lower expected computational cost. However, in some cases, these are conflicting objectives [23]. Hence, it is preferable to select algorithms that are mutually complementary so a synergy can develop between them [74]. Therefore, it is crucial to understand the relative strengths and weaknesses of the different algorithms in the portfolio for effective selection [63]. It has been suggested that meta-learning systems infer which and why certain algorithms work for specific classes of functions. As such, the information gained through meta-learning allows to systematize the insights to combine algorithms purposefully and even provide clues for new algorithm designs [28].

A final element to consider in portfolio development is the random nature of the performance measure $\rho(f,a)$. Since it is possible to have large variations of performance over different instances of the same problem [25], the overall performance is quite sensitive to the runtime distributions of the algorithms involved [23]. Fortunately, for many randomized algorithms such distributions closely resemble standard parametric distributions [31], usually gamma as discussed in Sec. 4. Hence, they can be described by certain sufficient statistics. By forecasting such statistics, a prediction of the entire distribution for an unseen instance can be obtained [31].

6 An Extension of the Algorithm Selection Framework

So far we have discussed three main areas of research in the continuous optimization domain: landscape analysis, algorithm modeling and portfolio design. We have also pointed out how these areas are related to the algorithm selection problem and meta-learning. Now, we propose the framework shown in Fig 3, which connects these research areas together. This extended framework is composed by two feedback loops. The first of such loops is the *analysis loop*, which starts at the junction α where the pairs $(\mathbf{x}_i, y_i)$ are fed into the landscape analysis stage. At this stage, different analysis methods work in parallel to produce a vector of estimated characteristics denoted as $\hat{\mathbf{c}} \in \mathcal{C}$. As we pointed out in Sec. 3, global measures have useful computational advantages, particularly the possibility to reuse data from previous experiments. However, there are two important factors to take into account: The level of uncertainty associated with landscape analysis methods in general and the inherent bias of the samples that have been extracted during the run. For the former, a possible avenue is to consider confidence intervals instead of a single value as the result of the analysis. For the later, weighted resampling might provide correction over the bias. This solution was demonstrated in a previous work [44], with promising results.

The vector $\hat{\mathbf{c}}$ is the input for a set of models, each one of them represents an available algorithm. We do not favor any particular machine learning method to model the algorithms, although it is desirable to have a method that recognizes the uncertainty associated with the inputs, and provides a confidence interval for the output. The result is the vector of performance predictions $\hat{\boldsymbol{\rho}} \in \mathcal{P}^m$, where m is

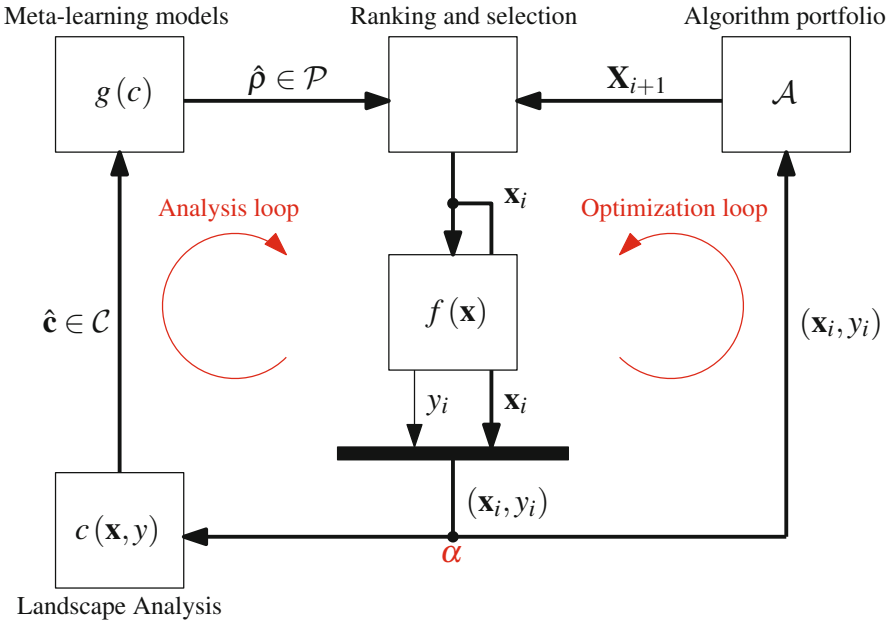


Fig. 3 Proposed extension of the algorithm selection framework. This new framework is composed of two loops: the optimization loop, where the search for promising solutions is carried out; and the analysis loop, where the selection for the best algorithm is made based on the landscape analysis.

the number of available algorithms. The predictions in this vector will be accurate depending on: the diversity in the knowledge base used to train the models, the relevance of the features, and the inherent randomness of the performance function. The results from this stage are used to create a ranking of the likelihood of each algorithm to create a new, potentially useful solution. At this point, it is important to consider the exploration/exploitation balance as well as the propagated uncertainty due to the landscape analysis and performance prediction.

The second loop is the *optimization loop*. It starts from the junction α, where the pairs $(\mathbf{x}_i, y_i)$ are fed into the algorithm portfolio. Besides producing new solutions to be evaluated, the portfolio shares information among constituent algorithms with the objective to improve the chances of producing useful solutions. For that purpose, it must be considered if the algorithms use a type of reinforcement learning, i.e. CMA-ES, or not, i.e. PSO. This is because data that is improperly supplied to the system might disrupt significantly the learning process. The resulting new solutions, $\mathbf{X}_{i+1}$, are transmitted to the ranking and selection mechanism, where the decision is taken into which one of them are fed into f.

In overall, the proposed framework provides different sources of information in order to produce a more extensive and detailed search. It also stores expertise that otherwise must be acquired by long, trial-and-error experiments. The framework

still provides flexibility into the selection of each component, such as which landscape analysis method to use, what type of model to implement, and importantly, which algorithms to select. It provides the opportunity of switching on and off algorithms depending on the case and to run concurrently as many algorithms as desired, unlike other portfolio approaches.

7 Discussion

In this paper we have discussed the relationship between landscape analysis, meta-learning models and algorithm portfolios to the algorithm selection framework as proposed by Rice [56]. We did so by reviewing the existing literature and proposing an extended framework that can be used for the algorithm selection in the continuous optimization domain. The proposed framework has several advantages: First, it enhances the search by providing additional sources of information that can be used to make decisions during the run. Second, it facilitates the storage of expertise that otherwise must be acquired by trial-and-error experiments. Third, it provides flexibility into the selection of each component, e.g. which landscape analysis method to use, what type of model to implement, and importantly, which algorithms to place in the portfolio. Finally, it provides the opportunity to run concurrently as many algorithms as desired, and being able to switch on and off those that are suitable at the time and place.

Our current work is focused in three areas of the framework. The first step is to develop a deeper understanding of some of the analysis methods in Tbl. 1. Our approach is to measure the uncertainty produced by the estimators through non-parametric statistical tests. The second step is to develop the meta-learning models. For this purpose, our approach is to use the confidence intervals of the landscape analysis as input to a machine learning strategy. The third step is to produce a ranking and selection mechanism. Our approach is to consider the uncertainty in the output models as part of the decision process. Algorithms with low uncertainty and high performance are deemed as the best choices, while algorithms with high uncertainty and low performance are deemed as the worst choices. The results so far are encouraging [44, 45].

References

[1] Achioptas, D., Naor, A., Peres, Y.: Rigorous location of phase transitions in hard optimization problems. Nature 435, 759–764 (2005)
[2] Anderson, E.: Markov chain modelling of the solution surface in local search. J. Oper. Res. Soc. 53(6), 630–636 (2002)
[3] Angel, E., Zissimopoulos, V.: On the hardness of the quadratic assignment problem with metaheuristics. Heuristics 8(4), 399–414 (2002)
[4] Bartz-Beielstein, T., Markon, S.: Tuning search algorithms for real-world applications: a regression tree based approach. In: CEC 2004, vol. 1, pp. 1111–1118 (2004)

[5] Bartz-Beielstein, T., Parsopoulos, K., Vrahatis, M.: Analysis of particle swarm optimization using computational statistics. In: ICNAAM 2004, pp. 34–37 (2004)
[6] Beck, J., Watson, J.: Adaptive search algorithms and Fitness-Distance correlation. In: Proc. 5th Metaheuristics Int. Conf. (2003)
[7] Borenstein, Y., Poli, R.: Fitness Distributions and GA Hardness. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 11–20. Springer, Heidelberg (2004)
[8] Borenstein, Y., Poli, R.: Information landscapes. In: GECCO 2005, pp. 1515–1522. ACM (2005)
[9] Borenstein, Y., Poli, R.: Kolmogorov complexity, optimization and hardness. In: CEC 2006, pp. 112–119 (2006)
[10] Boukeas, G., Halatsis, C., Zissimopoulos, V., Stamatopoulos, P.: Measures of Intrinsic Hardness for Constraint Satisfaction Problem Instances. In: Van Emde Boas, P., Pokorný, J., Bieliková, M., Štuller, J. (eds.) SOFSEM 2004. LNCS, vol. 2932, pp. 184–195. Springer, Heidelberg (2004)
[11] Branke, J., Schmidt, C.: Faster convergence by means of fitness estimation. Soft Comput. 9(1), 13–20 (2005)
[12] Brooks, C., Durfee, E.: Using Landscape Theory to Measure Learning Difficulty for Adaptive Agents. In: Alonso, E., Kudenko, D., Kazakov, D. (eds.) AAMAS 2000 and AAMAS 2002. LNCS (LNAI), vol. 2636, pp. 291–305. Springer, Heidelberg (2003)
[13] Carchrae, T., Beck, J.: Low knowledge algorithm control. In: AAAI 2004, pp. 49–54 (2004)
[14] Carchrae, T., Beck, J.: Applying machine learning to low-knowledge control of optimization algorihms. Comput. Intell. 21(4), 372–387 (2005)
[15] Davidor, Y.: Epistasis variance: A viewpoint on GA-hardness. In: Foundations of Genetic Algorithms. Morgan Kaufmann (1991)
[16] Eiben, A., Hinterding, R., Michalewicz, Z.: Parameter control in evolutionary algorithms. IEEE Trans. Evol. Comput. 3(2), 124–141 (1999)
[17] Eremeev, A.V., Reeves, C.R.: Non-parametric Estimation of Properties of Combinatorial Landscapes. In: Cagnoni, S., Gottlieb, J., Hart, E., Middendorf, M., Raidl, G.R. (eds.) EvoIASP 2002, EvoWorkshops 2002, EvoSTIM 2002, EvoCOP 2002, and EvoPlan 2002. LNCS, vol. 2279, pp. 31–40. Springer, Heidelberg (2002)
[18] Eremeev, A.V., Reeves, C.R.: On Confidence Intervals for the Number of Local Optima. In: Raidl, G.R., Cagnoni, S., Cardalda, J.J.R., Corne, D.W., Gottlieb, J., Guillot, A., Hart, E., Johnson, C.G., Marchiori, E., Meyer, J.-A., Middendorf, M. (eds.) EvoIASP 2003, EvoWorkshops 2003, EvoSTIM 2003, EvoROB/EvoRobot 2003, EvoCOP 2003, EvoBIO 2003, and EvoMUSART 2003. LNCS, vol. 2611, pp. 224–235. Springer, Heidelberg (2003)
[19] Fonlupt, C., Robilliard, D., Preux, P.: A Bit-Wise Epistasis Measure for Binary Search Spaces. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 47–56. Springer, Heidelberg (1998)
[20] Francois, O., Lavergne, C.: Design of evolutionary algorithms-A statistical perspective. IEEE Trans. Evol. Comput. 5(2), 129–148 (2001)
[21] Galván-López, E., McDermott, J., O'Neill, M., Brabazon, A.: Defining locality as a problem difficulty measure in genetic programming. Genet. Program Evolvable Mach. 12(4), 365–401 (2011)
[22] Garnier, J., Kallel, L.: Efficiency of local search with multiple local optima. SIAM J. Discrete Math. 15(1), 122–141 (2002)

[23] Gomes, C., Selman, B.: Algorithm portfolios. Artif. Intell. 126(1-2), 43–62 (2001)
[24] Graff, M., Poli, R.: Practical performance models of algorithms in evolutionary program induction and other domains. Artif. Intell. 174, 1254–1276 (2010)
[25] Grobler, J., Engelbrecht, A., Kendall, G., Yadavalli, V.: Alternative hyper-heuristic strategies for multi-method global optimization. In: CEC 2010, pp. 1–8 (2010)
[26] He, J., Reeves, C., Witt, C., Yao, X.: A note on problem difficulty measures in black-box optimization: Classification, realizations and predictability. Evol. Comput. 15(4), 435–443 (2007)
[27] Heckendorn, R., Whitley, D.: Predicting epistasis from mathematical models. Evol. Comput. 7(1), 69–101 (1999)
[28] Hilario, M., Kalousis, A., Nguyen, P., Woznica, A.: A data mining ontology for algorithm selection and meta-mining. In: SoKD 2009, pp. 76–87 (2009)
[29] Hough, P., Williams, P.: Modern machine learning for automatic optimization algorithm selection. In: INFORMS AI/DM Workshop (2006)
[30] Hutter, F., Hamadi, Y., Hoos, H.H., Leyton-Brown, K.: Performance Prediction and Automated Tuning of Randomized and Parametric Algorithms. In: Benhamou, F. (ed.) CP 2006. LNCS, vol. 4204, pp. 213–228. Springer, Heidelberg (2006)
[31] Hutter, F., Hamadi, Y., Hoos, H., Leyton-Brown, K.: Performance prediction and automated tuning of randomized and parametric algorithms: An initial investigation. In: The AAAI Workshop on Learning for Search: Schedule (2006)
[32] Hutter, F., Hoos, H., Leyton-Brown, K.: Tradeoffs in the empirical evaluation of competing algorithm designs. Tech. Rep. TR-2009-21, The University of British Columbia (2009)
[33] Jansen, T.: On classifications of fitness functions. Tech. Rep. CI-76/99, University of Dortmund (1999)
[34] Jones, T., Forrest, S.: Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In: GECCO 1995, pp. 184–192. Morgan Kaufmann Publishers Inc. (1995)
[35] Jong, K.D.: Parameter setting in EAs: a 30 year perspective. In: Parameter Setting in Evolutionary Algorithms, Stud. Comput Intell., vol. 54, pp. 1–18. Springer (2005)
[36] Leyton-Brown, K., Nudelman, E., Shoham, Y.: Learning the Empirical Hardness of Optimization Problems: The Case of Combinatorial Auctions. In: Van Hentenryck, P. (ed.) CP 2002. LNCS, vol. 2470, pp. 556–572. Springer, Heidelberg (2002)
[37] Leyton-Brown, K., Nudelman, E., Shoham, Y.: Empirical hardness models: Methodology and a case study on combinatorial auctions. J. ACM 56(4), 22:1–22:52 (2009)
[38] Liu, L., Abbass, H., Green, D., Zhong, W.: Motif difficulty (MD): a predictive measure of problem difficulty for evolutionary algorithms using network motifs. In: Evolutionary Computation (2011)
[39] Lunacek, M., Whitley, D.: The dispersion metric and the CMA evolution strategy. In: GECCO 2006, pp. 477–484. ACM, New York (2006)
[40] Merkuryeva, G., Bolshakovs, V.: Structural analysis of benchmarking fitness landscapes. Scientific Journal of Riga Technical University: Computer Sciences 42, 81–86 (2010)
[41] Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C., Rudolph, G.: Exploratory landscape analysis. In: GECCO 2011, pp. 829–836. ACM (2011)
[42] Messelis, T., Haspeslagh, S., Bilgin, B., Causmaecker, P.D., Berghe, G.V.: Towards prediction of algorithm performance in real world optimisation problems. In: BNAIC 2009, pp. 177–183 (2009)

[43] Molina, D., Herrera, F., Lozano, M.: Adaptive local search parameters for real-coded memetic algorithms. In: CEC 2005, vol. 1, pp. 888–895 (2005)
[44] Muñoz, M., Kirley, M., Halgamuge, S.: Landscape characterization of numerical optimization problems using biased scattered data. In: CEC 2012 (2012)
[45] Muñoz, M., Kirley, M., Halgamuge, S.: A Meta-Learning prediction model of algorithm performance for continuous optimization problems. Submitted to PPSN XII (2012a)
[46] Müller, C.L., Sbalzarini, I.F.: Global Characterization of the CEC 2005 Fitness Landscapes Using Fitness-Distance Analysis. In: Di Chio, C., Cagnoni, S., Cotta, C., Ebner, M., Ekárt, A., Esparcia-Alcázar, A.I., Merelo, J.J., Neri, F., Preuss, M., Richter, H., Togelius, J., Yannakakis, G.N. (eds.) EvoApplications 2011, Part I. LNCS, vol. 6624, pp. 294–303. Springer, Heidelberg (2011)
[47] Naudts, B., Kallel, L.: A comparison of predictive measures of problem difficulty in evolutionary algorithms. IEEE Trans. Evol. Comput. 4(1), 1–15 (2000)
[48] Naudts, B., Suys, D., Verschoren, A.: Epistasis as a basic concept in formal landscape analysis. In: GECCO 1997, pp. 65–72. Morgan Kaufmann (1997)
[49] Özcan, E., Bilgin, B., Korkmaz, E.E.: Hill Climbers and Mutational Heuristics in Hyperheuristics. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) PPSN 2006. LNCS, vol. 4193, pp. 202–211. Springer, Heidelberg (2006)
[50] Pedersen, M.: Tuning & simplifying heuristical optimization. PhD thesis, University of Southampton (2009)
[51] Peng, F., Tang, K., Chen, G., Yao, X.: Population-Based algorithm portfolios for numerical optimization. IEEE Trans. Evol. Comput. 14(5), 782–800 (2010)
[52] Petrovic, S., Epstein, S., Wallace, R.: Learning a mixture of search heuristics. In: CP 2007 (2007)
[53] Reeves, C.: Fitness landscapes. In: Search Methodologies, pp. 587–610. Springer (2005)
[54] Reeves, C., Eremeev, A.: Statistical analysis of local search landscapes. J. Oper. Res. Soc. 55(7), 687–693 (2004)
[55] Reeves, C., Wright, C.: An experimental design perspective on genetic algorithms. In: Foundations of Genetic Algorithms, vol. 3, pp. 7–22. Morgan Kaufmann (1995)
[56] Rice, J.: The algorithm selection problem. In: Adv. Comput., vol. 15, pp. 65–118. Elsevier (1976)
[57] Rice, J.: Methodology for the algorithm selection problem. In: Proc. IFIP TC 2.5 Working Conference on Performance Evaluation of Numerical Software (1979)
[58] Rochet, S., Venturini, G., Slimane, M., El Kharoubi, E.M.: A Critical and Empirical Study of Epistasis Measures for Predicting GA Performances: A Summary. In: Hao, J.-K., Lutton, E., Ronald, E., Schoenauer, M., Snyers, D. (eds.) AE 1997. LNCS, vol. 1363, pp. 275–285. Springer, Heidelberg (1998)
[59] Rockmore, D., Kostelec, P., Hordijk, W., Stadler, P.: Fast fourier transform for fitness landscapes. Appl. Comput. Harmon Anal. 12(1), 57–76 (2002)
[60] Seo, D., Moon, B.: An Information-Theoretic analysis on the interactions of variables in combinatorial optimization problems. Evol. Comput. 15(2), 169–198 (2007)
[61] Smith, T., Husbands, P., Layzell, P., O'Shea, M.: Fitness landscapes and evolvability. Evol. Comput. 10(1), 1–34 (2002)
[62] Smith-Miles, K.: Cross-disciplinary perspectives on meta-learning for algorithm selection. ACM Comput. Surv. 41(1), 6:1–6:25 (2009)
[63] Smith-Miles, K., Hemert, J.v.: Discovering the suitability of optimisation algorithms by learning from evolved instances. Ann. Math. Artif. Intel. 61(2), 87–104 (2011)

[64] Stadler, P.: Landscapes and their correlation functions. J. Math. Chem. 20(1), 1–45 (1996)
[65] Streeter, M., Golovin, D., Smith, S.: Combining multiple heuristics online. In: AAAI 2007, vol. 2, pp. 1197–1203. AAAI Press (2007)
[66] Tomassini, M., Vanneschi, L., Collard, P., Clergue, M.: A study of fitness distance correlation as a difficulty measure in genetic programming. Evolutionary Computation 13(2), 213–239 (2005)
[67] Vanneschi, L.: Investigating problem hardness of real life applications. In: Genetic Programming Theory and Practice V, Genetic and Evolutionary Computation, pp. 107–124. Springer, US (2008)
[68] Vanneschi, L., Clergue, M., Collard, P., Tomassini, M., Vérel, S.: Fitness Clouds and Problem Hardness in Genetic Programming. In: Deb, K., Tari, Z. (eds.) GECCO 2004. LNCS, vol. 3103, pp. 690–701. Springer, Heidelberg (2004)
[69] Vanneschi, L., Tomassini, M., Collard, P., Vérel, S.: Negative Slope Coefficient: A Measure to Characterize Genetic Programming Fitness Landscapes. In: Collet, P., Tomassini, M., Ebner, M., Gustafson, S., Ekárt, A. (eds.) EuroGP 2006. LNCS, vol. 3905, pp. 178–189. Springer, Heidelberg (2006)
[70] Vanneschi, L., Valsecchi, A., Poli, R.: Limitations of the fitness-proportional negative slope coefficient as a difficulty measure. In: GECCO 2009, pp. 1877–1878. ACM, New York (2009)
[71] Vassilev, V., Fogarty, T., Miller, J.: Information characteristics and the structure of landscapes. Evol. Comput. 8(1), 31–60 (2000)
[72] Vassilev, V., Fogarty, T., Miller, J.: Smoothness, ruggedness and neutrality of fitness landscapes: from theory to application. In: Advances in Evolutionary Computing, pp. 3–44. Springer, New York (2003)
[73] Vassilevska, V., Williams, R., Woo, S.: Confronting hardness using a hybrid approach. In: Proc. 17th Ann. ACM-SIAM Symp. Discrete Algorithms, pp. 1–10. ACM, New York (2006)
[74] Vrugt, J., Robinson, B., Hyman, J.: Self-Adaptive multimethod search for global optimization in Real-Parameter spaces. IEEE Trans. Evol. Comput. 13(2), 243–259 (2009)
[75] Watson, J., Beck, J., Howe, A., Whitley, L.: Problem difficulty for tabu search in job-shop scheduling. Artif. Intell. 143(2), 189–217 (2003)
[76] Weinberger, E.: Correlated and uncorrelated fitness landscapes and how to tell the difference. Biol. Cybern. 63(5), 325–336 (1990)
[77] Weinberger, E., Stadler, P.: Why some fitness landscapes are fractal. J. Theor. Biol. 163(2), 255–275 (1993)
[78] Wolpert, D., Macready, W.: No free lunch theorems for optimization. IEEE Trans. Evol. Comput. 1(1), 67–82 (1997)
[79] Yeguas, E., Joan-Arinyo, R., Luzón, M.V.: Modeling the performance of evolutionary algorithms on the root identification problem: A case study with PBIL and CHC algorithms. Evolutionary Computation 19(1), 107–135 (2011)