Planning Landscape Analysis for Self-Adaptive Systems

Tao Chen
Loughborough University
Loughborough, United Kingdom
t.t.chen@lboro.ac.uk

ABSTRACT

To assure performance on the fly, planning is arguably one of the most important steps for self-adaptive systems (SASs), especially when they are highly configurable with a daunting number of adaptation options. However, there has been little understanding of the planning landscape or ways by which it can be analyzed. This inevitably creates barriers to the design of better and tailored planners for SASs. In this paper, we showcase how the planning landscapes of SASs can be quantified and reasoned, particularly with respect to the different environments. By studying four diverse real-world SASs and 14 environments, we found that (1) the SAS planning landscapes often provide strong guidance to the planner, but their ruggedness and multi-modality can be the major obstacle; (2) the extents of guidance and number of global/local optima are sensitive to the changing environment, but not the ruggedness of the surface; (3) the local optima are often closer to the global optimum than other random points; and (4) there are considerable (and useful) overlaps on the global/local optima between landscapes under different environments. We then discuss the potential implications to the future work of planner designs for SASs.

CCS CONCEPTS

• Software and its engineering → Software performance; Software configuration management and version control systems.

KEYWORDS

Self-adaptive system, configuration tuning, planning, performance optimization, search-based software engineering, landscape

ACM Reference Format:
Tao Chen. 2022. Planning Landscape Analysis for Self-Adaptive Systems. In 17th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS ’22), May 18–23, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3524844.3528060

1 INTRODUCTION

Modern software systems are often engineered as highly configurable for handling different performance needs [13, 14, 26, 32, 33], such as a good latency and throughput, even as the system runs [30]. Therefore, self-adaptation — the ability to find a better adaptation plan of the configurations that improves the performance on the fly — is on high demand for highly configurable systems, which is the type of self-adaptive systems (SASs) we consider in this paper.

For almost any type of SASs, planning is a key step in achieving self-adaptation and it mainly seeks to address one question: what is the best adaptation plan to take under a new environment (or when needed)? To this end, the community has relied on various different planning algorithms (or search algorithms), particularly the stochastic ones, to design a planner [2, 9, 12, 19, 28, 43]. This equips a SAS with the ability to reason about the better or worse of the adaptation plans in the planning landscape, and hence ideally choose the one that has the best performance without the need to traverse the entire search space. From that regard, planning for SASs resembles a search and optimization process [21], which is complementary to many other techniques that are widely-used for SASs, such as control theory [3], machine learning [4, 8, 18, 25], and formal verification [35].

Despite the importance, designing an effective planner is nontrivial, because the search space can be too large; the budget for a planner to conclude is limited; and there may be a complex relationship between adaptation plans and their performance, as well as the changing environment (e.g., there may be difficult local optima¹). This becomes even more challenging when we have little understanding of the planning landscape of SASs or ways by which we can utilize to study it. Indeed, there have been some works on the planning algorithms for SAS [2, 9, 12, 19, 28, 43], it however remains unclear about what characteristics of the SAS planning landscape have enabled their success (or failure).

Figure 1 shows an example of the projected planning landscapes for STORM under two environments. As can be seen, even with a clear visualization of such a simplified version (which is what has been commonly used in existing work [18, 24]), it is not always

¹Some sub-optimal points that have better or identical performance to all of their surrounding points.
straightforward to quantify and conclude useful information/properties about the landscapes and their relative differences across environments. Of course, simply trying the planner on the SAS and examining the outcome is insufficient, even if we can do so. Rather, we need to be able to answer questions like whether the correlation between performance and plans in the landscape can provide strong guidance for a planner; how rugged the surface is; and what properties of the landscapes have been changed (or still unchanged) with the time-varying environment. We envisage that such a better understanding of the planning landscape is important, as it will enable us to derive planner designs that are explicitly tailored to the observed characteristics of SASs, promoting more effective and efficient planning which would otherwise be difficult.

To close the above gap, in this paper, we show how the metrics and notions from the domain of fitness landscape analysis [41, 47] in the optimization community can be derived to study the planning landscape for SAS, particularly in relation to the different environments. This is achieved by an empirical analysis of four real-world SASs with 14 environments from the literature [14, 26, 36, 39, 44], which are of different domains, languages, scales, and search spaces. The results reveal some interesting patterns:

- The SAS planning landscapes often provide useful information to guide the search process in a planner, but their ruggedness and multi-modality can pose a barrier.
- The extents of guidance and number of global/local optima are sensitive to the changing environment, but not the ruggedness of the surface in the planning landscape.
- Local optima are often closer to the global optimum than other random points.
- Planning landscapes under different environments of a SAS often share a good amount of global/local optima. In particular, preserving the local optima of an environment into the newly changed one can be beneficial, as they may immediately become the global optimum therein.

We then discuss the implications of our results for future planner design on SASs. To promote open science, we release the code and data in this work at: https://doi.org/10.5281/zenodo.5866808.

In what follows, Section 2 introduces the background. Section 3 elaborates on the notions and the metrics chosen for our landscape analysis. Section 4 presents the methodology of study and analyzes the results. Key implications are discussed in Section 5 and threats to validity are presented in Section 6. Sections 7 and 8 analyze the related work and conclude the paper, respectively.

2 BACKGROUND

When self-adapting highly-configurable systems, there are \( n \) adaptation options such that the \( i \)th option is denoted as \( x_i \), which can be a binary/integer variable. The search space of all plans, \( \mathcal{X} \), is the Cartesian product of the possible values or all the \( x_i \). Without lose of generality, the ultimate goal of the SAS planning\(^3\) is to achieve the following in a given environment:

\[
\arg\min f(x), \; x \in \mathcal{X} \tag{1}
\]

where \( x = (x_1, x_2, ..., x_n) \) and \( f \) measures the performance achieved by a plan, e.g., [3, 1] for the \{num_counters, numSplitters\} on

\(^{3}\text{We assume minimizing the performance objective.}\)

3 LANDSCAPE ANALYSIS OF SAS PLANNING

Fitness landscape is a concept initially coined by Wright [47] and then extended to study the possible behaviours of algorithms in the optimization process, which fits precisely the needs of our planning analysis for SASs. In a nutshell, fitness landscape analysis concerns with understanding the relationships between the multi-dimensional encoding of the solutions (genotype) and their goodness (fitness) by means of various metrics and procedure [41, 47].

Formally, the landscape for SAS planning under an environment can be represented as a tuple \( \mathcal{F} = (\mathcal{X}, f, \mathcal{N}_K) \), such that:

\[
\mathcal{N}_K(x) = \{ y \in \mathcal{X} : D(x, y) \leq k \} \tag{2}
\]

whereby \( \mathcal{X} \) is a set of points (adaptation plans); \( f \) is the same performance table as that in Equation 1, \( \mathcal{N}_K \) is the neighborhood defined over set \( \mathcal{X} \) according to a distance metric \( D \) of size \( k \) (which may be the bound for some operators to transform one plan into another). Clearly, when \( k \) covers all the neighboring plans in the search space, we obtain a complete planning landscape for the SAS.

3.1 Distance, Neighborhood and Local Optima

Quantifying the distance between adaptation plans (and hence the neighborhood) is the fundamental step in our planning landscape analysis. In this work, we use Hamming distance \( D_H \) to measure two adaptation plans because of three reasons:

- The sparse comparison in Hamming distance fits well with the landscape nature of configurable system [14, 24, 39].
- Hamming distance does not quantify the magnitude of difference on an adaptation option, which fits with the need of most categorical options in configurable systems.
- It is widely used in many real-world problems [38, 40, 46].

Figure 2: Overview of Storm under changing environments.

**Storm.** Since the environment can change as the SAS executes, the planning will run continuously. In the SAS literature, \( f \) has been realized by different ways [10], such as analytical models [22], machine learning models [6, 12, 18], simulation [21], or even digital twins [1], the details of which is outside the scope of this paper and hence we assume that there is a readily available resolution.

Figure 2 illustrates Storm, which is a system that handles data streaming process under periodically incoming micro-batch of jobs (environments), e.g., RollingCount and WordCount. In this case, simply using a fixed configuration can be problematic: Jamshidi and Casale [24] have shown that using the default setting can lead to 480× slower than the best in some environments. This motivates the need for self-adaptation, where the aim is to optimize the latency by searching the right adaptation plan over changing environments.
In this work we set \( D_H = 1 \) hence the neighbors are the plans that differ exactly on one option. A point is a local optimum if it is no worse than all of its neighbors, as shown in Figure 3.

### 3.2 Fitness Distance Correlation

Generally speaking, Fitness Distance Correlation (FDC) examines how close is the relation between fitness value and distance to the nearest optimum in the search space [27], which quantifies the overall guidance that the planning landscape can offer for a planner [46]. Formally, FDC (denoted as \( \varphi \)) is computed as:

\[
\varphi(f, d) = \frac{1}{\sqrt{p}} \sum_{i=1}^{p} (f_i - \bar{f})(d_i - \bar{d})
\]

where \( p \) is the number of points considered in FDC; in this work, we set \( p \) as the total number of adaptation plans in the space and hence the FDC reflects the complete planning landscape. \( f_i = f(x_i) \) is the performance value for the \( i \)th adaptation plan and \( d_i = d_{opt}(x_i) \) is the shortest Hamming distance of such a plan to a global optimum. \( \bar{f} \) and \( \bar{d} \) are the mean and standard deviation, respectively.

Intuitively, FDC is in fact the Pearson correlation between \( f \) and \( d \), hence it ranges on \([-1, 1]\) where 1 and \(-1\) imply the strongest monotonically positive and negative correlation, respectively; 0 indicates no correlation can be detected. Since in our case we prefer a smaller performance value, when \( 0 < \varphi \leq 1 \), the adaptation plan turns better (smaller performance value) as the shortest distance to a global optimum reduces. This means that, when FDC becomes closer to 1, the guidance provided to a planner is stronger and it is more likely to exist a path towards a global optimum via adaptation plans with decreasing performance values, hence the planning can be reasonably solved. In contrast, \(-1 \leq \varphi < 0\) indicates the opposite.

### 3.3 Landscape Structure

In this work, we also explicitly assess the structure of the planning landscape, i.e., multi-modality and ruggedness.

#### 3.3.1 Multi-modality

In general, the multi-modality, as opposed to the uni-modality with one global optimum and no local optimum, refers to a special property of the landscape where there is more than one global/local optimum [23]. To quantify such, we can count the percentage of global/local optima in the complete planning landscape as a global metric of its structure [23, 37]. A landscape with a high degree of multi-modality is an indication that it contains many “troughs” (for minimizing objectives), which may both be a challenge and an opportunity. On one hand, multi-modality implies a complex structure (at least globally) and hence can raise the additional difficulty for the planner. On the other hand, the presence of different “troughs”, together with the ability to locate them, can be beneficial for a planner to eventually reach a global optimum.

#### 3.3.2 Ruggedness

Measuring multi-modality by the number of global/local optima still cannot account for the local paths between local optima, and the other related points. As a result, we additionally measure the Correlation Length (\( \ell \)) of the landscape [45] — a local metric that indicates the local ruggedness property.

To be more specific, the Correlation Length is the results of randomly sampled adaptation plans in the landscape, and hence it models the local surface of traversal that a planner would likely to explore. Formally, \( \ell \) is calculated as below:

\[
\ell(p, s) = -\ln\left(\frac{1}{\sigma_f^2(p - s)} \sum_{i=1}^{p-s} (f_i - \bar{f})(f_{i+s} - \bar{f})\right)^{-1}
\]

\( \ell(p, s) \) is essentially a normalized autocorrelation function of neighboring points’ performance values explored and the notations are the same as that for Equation 3. In this work, we conduct sampling with random walk [46], thus \( s \) denotes the step size and \( p \) is the walk length. We use \( s = 1 \) in this work, which means that we target the most restricted form where the autocorrelation is calculated on adaptation plans sampled from adjacent steps (Note that the correlation cannot be 0, as this is what has been widely followed [38, 40, 46]. The higher the value of \( \ell \), the smoother the landscape, as the performance of adjacent sampled adaptation plans are more correlated. Otherwise, it indicates a more rugged surface [45], which means the easier to trap a planner.

### 4 METHODOLOGY AND RESULTS

In this work, we seek to answer the following research questions:

- **RQ1:** Do the planning landscapes offer useful guidance to a SAS planner under different environments?
- **RQ2:** What are the general structural properties of planning landscapes for SASs under different environments?
- **RQ3:** Do the local optima closer to global optimum than other non-optimal points in the SAS planning landscape?
- **RQ4:** Is it possible to share some information on the planning landscapes for SAS across different environments?

To that end, as shown in Table 1, we consider 3-4 environments that can change arbitrarily at runtime for four real-world SAS and use the same setting as previous work. We exploit the readily available dataset of those systems [14, 24, 26, 36, 39] which contain the samples for the entire planning landscape of each environment.

For interpreting the FDC and Correlation Lengths (RQ1 and RQ2), we adopt Fisher’s transformation [20] to find the z-score, which is then interpreted using Zou’s confidence interval [49] under a significance level of 0.05. We leverage the non-paired Wilcoxon rank-sum test at \( \alpha = 0.05 \) for comparing the distance in RQ3.

#### 4.1 RQ1: Fitness Guidance in Planning

##### 4.1.1 Method

To answer RQ1, we leverage the FDC coefficient to measure the extents of guidance that a planning landscape offers to the planner, considering all SASs and their environments studied.
Table 1: Real-world self-adaptive systems and their environments studied. We use the deep neural network in Keras. Reference shows the work that also uses the same systems.

| Subject SAS | Performance Environments | # Options | Search Space |
|-------------|--------------------------|-----------|--------------|
| STORM [14, 24] | Latency | E1: Speed-Of-Light | 1 | 2,048 |
|              |               | E2: ROLLINGSORT     | 12 | 2,048 |
|              |               | E3: WorkCount       |     |     |
|              |               | E4: RollingCount    |     |     |
| KERAS [26, 36] | Inferred Time | E1: ShapesAll       | 12 | 4,096 |
|              |               | E2: DSR             |     |     |
|              |               | E3: Adac            |     |     |
|              |               | E4: Coffee          |     |     |

To further interpret the FDC coefficients in detail, we adopt the classification concluded by Jones and Forrest [27] (the values are reversed as we focus on minimizing the performance objectives):

- **Misleading** \( (\phi \leq -0.15) \). The landscape can drive the search to move away from the global optimum.
- **Difficult** \( (-0.15 < \phi < 0.15) \). The correlation is insignificant to guide the planner on any particular direction.
- **Straightforward** \( (\phi \geq 0.15) \). The landscape provides useful guidance for a planner to reach a global optimum.

4.1.2 Result. The FDC coefficients have been plotted in Figure 4 with statistical test results between each pair of environments for a SAS in Table 2. As can be seen, we obtain a few interesting findings:

1. Surprisingly, the planning landscapes on all SASs and environments are classified as “Straightforward”, implying that they offer a good degree of useful guidance for the planner.
2. The FDC coefficients tend to differ across the SASs (with the landscapes of KERAS showing the strongest guidance), which is as expected since they all come from different domains and are implemented in different languages.
3. Although the FDC coefficients under different environments of a SAS may be seen as similar, most of their differences are statistically significant, i.e., in 15 out of 18 cases (Table 2).

Therefore, for RQ1, we say:

**To RQ1**: Yes, the planning landscapes offer useful and strong guidance to the SAS planner in general, but the environmental change can influence the guidance provided.

### Table 2: Statistical test \((p\) value) on the FDC coefficient \((\phi)\) and Correlation Length \((\ell)\) between all pairs of environments. Statistically significant ones are highlighted in bold.

| Subject SAS | Performance Environments | # Options | Search Space |
|-------------|--------------------------|-----------|--------------|
| STORM       | Latency                  | E1: Speed-Of-Light | 1 | 2,048 |
|              |                          | E2: ROLLINGSORT  | 12 | 2,048 |
|              |                          | E3: WorkCount   |     |     |
|              |                          | E4: RollingCount |     |     |

**4.2 RQ2: Planning Landscape Structure**

4.2.1 Method. The structure in RQ2 is measured in two ways: (1) the % of global/local optima in the landscape (for multi-modality) and (2) the Correlation Length (for ruggedness), as discussed in Section 3.3, for which we set the random walk length of 50 \((p = 50)\) from Equation 4) with 50 repeats and report the mean values.

While we cannot find a general standard to classify the degree of ruggedness similar to that for FDC, to aid our interpretation, we turned into the literature from the general optimization community. We hence use the calculated Correlation Length for some common problems with well-acknowledged challenges on local optima as the baselines in our discussion3, such as Multidimensional Knapsack Problem (MKP) [46], Quadratic Assignment Problem (QAP) [38], and Timetabling Problem (TP) [40]. This is possible as Correlation Length is a scale- and unit-agnostic metric.

4.2.2 Result. From the results in Figure 5 and the statistical tests of \(\ell\) between environments in Table 2, we observe that:

1. The percentage of global/local optima in SAS planning landscape indicates a reasonable sign of multi-modality in general. For some SASs, such as x264, it can go over 50%.
2. Different SASs and their environments often significantly affect the percentage of global/local optima in the planning landscape, but the overall multi-modal property is unaffected. The planning landscapes of x264 exhibit a much higher degree of multi-modality than the other SASs and appears to be insensitive to environmental change.

3Those problems can have different instances of the landscape; in this work, we use the smallest mean Correlation Length for each (most rugged surface) as reported.
(3) The Correlation Lengths of all SASs/environments are lower than that of MKP (which has the smallest value), suggesting that the ruggedness of SAS planning landscape is non-trivial.

(4) The Correlation Length differs considerably across the SASs (with landscapes for STORM being the most rugged ones), but between the environments for the same SAS, the differences are often insignificant. This has also been evidenced in Table 2 where only one case has \( p < 0.05 \).

In summary, we answer RQ2 as:

**To RQ2:** The planning landscapes for SASs show a good sign of multi-modality and they are more rugged than some other widely-studied problems. Yet, the ruggedness is insensitive to the changing environment but the multi-modality does.

### 4.3 RQ3: Distance to Global Optimum

#### 4.3.1 Method
To study RQ3, we report on the overall Hamming distance (\( d_{local} \)) between local optima and the closest global optima, together with that (\( d_{others} \)) between the rest of non-optimal points and the corresponding closest global optimum.

#### 4.3.2 Result
As we can see from Figure 6, the overall distances of \( d_{local} \) is clearly shorter than \( d_{others} \) regardless of the SASs and their environments; the extents of difference differ depending on the environment for a SAS though. The comparisons also come with statistical significance in general (\( p < 0.05 \)). This is strong evidence that the local optima in the planning landscapes, although may trap the planner, can be useful for serving as the "stepping stones" which may eventually lead to a global optimum.

Therefore, for RQ3, we say:

**To RQ3:** The local optima are indeed generally closer to the global optimum than the other non-optimal points in SAS planning landscape, meaning that preserving then jumping out from them is more likely to reach a global optimum.

### 4.4 RQ4: Information Between Environments

#### 4.4.1 Method
In RQ4, we study three aspects to understand the information of the planning landscapes that can be shared between environments, e.g., when changing from environment \( E_x \) to \( E_y \):

- \( A_1 \): Whether there is at least one global optimum in \( E_x \) that is also a global optimum under \( E_y \).
- \( A_2 \): Whether there is at least one local optimum in \( E_x \) that would become a global optimum under \( E_y \).
- \( A_3 \): The percentage of global/local optima in \( E_x \) that are also global/local optima under \( E_y \).

#### 4.4.2 Result
We demonstrate the results in Table 3, from which we can disclose some interesting observations:

1. From \( A_1 \) and \( A_2 \), there is only one case (\( E_4 \rightarrow E_1 \) for STORM) where the answers for both \( A_1 \) and \( A_2 \) are "No". This, together with the findings for RQ3, suggests that either the global or local optima (sometimes both) in one environment often helps the planner to find a global optimum in another.

2. From \( A_1 \) and \( A_3 \), we see that generally, the global optimum in one environment can directly serve as the global optimum after the environment change, as in 28 out of 36 cases there is at least one global optimum that satisfies \( A_1 \) (with Keras and Spear having overlapped global optimum over all environments.) and over half of the cases with more than 50% global/local optimum overlap. x264 also exhibit particularly high overlapping between environments.

3. From \( A_2 \) and \( A_3 \) under a considerable overlap of global/local optima, there are 21 out of 36 cases in which at least one local optimum of an environment would become a global optimum after changing to another. This implies that preserving the local optima can be useful for a new environment.

At this point, we conclude RQ4 as:

**To RQ4:** Yes, preserving both the global and local optima of the planning landscape in one environment can be useful for SAS planning under a changing environment.

### 5 IMPLICATIONS

Our findings can excite a few research directions for SAS planning.

RQ1 suggests that, for future research on selecting proper planning algorithm, SAS planning landscape is suitable for those algorithms guided by the fitness (performance), and jump from one adaptation plan to another by taking the distance to the currently reached adaptation plans into account, e.g., change one (or some
more) option each time such as the GA with neighborhood-based mutation [46, 47]. In contrast, a planner that alters the adaptation plan in a random number of options each time will likely lose the valuable guidance of the landscape. When exploited fully, such guidance is not only useful when the search space is intractable, but also helps to find global optimum quicker in a tractable space [47], which is attractive for SAS planning. However, it is important to inspect the impact caused by environmental change.

For **planner component design** from RQ2, we show that a mechanism which helps the planner to escape from local optima is indeed necessary. These mechanisms are readily available, such as larger radius of changes [48], random restarting [34], accepting inferior plans [17], and multi-objectivization [14]. Albeit the number of local optima from the landscapes may differ, a mechanism that works for one environment will likely work for the others too.

Finally, from RQ3 and RQ4, we confirm again the importance of **seeded planning and plan reuse** for SASs under changing environments, which has recently attracted attention in the community [15, 16, 28]. Additionally, we provide initial evidence on what should be seeded and shared. The most surprising finding is that the local optima of an environment are also helpful for the planning under the new environment (in addition to the global optimum), as they may still be the local optima or one of them may immediately become the new global optimum therein. This, together with the finding that local optima are very much closer to the global optimum than other random points and hence more helpful (if the planner can escape from them), raises an interesting topic of multi-modal planning for SAS: in addition to finding the global optimum, we are also interested in preserving as many local optima as possible during a planning run [5].

### 6 THREATS TO VALIDITY

Threats to **internal validity** can be related to the setting of one option for defining local optimum and the step of 50 for the random walking that computes Correlation Length, which were decided pragmatically based on the needs. Larger values may change the absolute figures but are unlikely to invalidate the conclusion.

The metrics and evaluation used may possess threats to **construct validity**. In this work, the most common metrics from fitness landscape analysis are used [27, 41] and those are related to SAS planning (e.g., those in RQ4). Statistical significance is also measured. However, we acknowledge that examining more metrics for the properties of landscapes and using alternative baselines may reveal more insights, which we will plan to do in future work.

The SASs and environments studied may be subject to the threats of **external validity**. We mitigated this by using four commonly studied SAS that is of different domains, scales, and performance attributes, together with 14 environments, as used in prior work [14, 24, 26, 36, 39]. Nonetheless, we agree that studying additional systems/environments, even other types of SASs, may prove fruitful.

### 7 RELATED WORK

We now discuss the work related to the landscape analysis for SASs. **Planning for SASs**: Over the last decade, various planning algorithms have been proposed/adopted for SASs, including the ones that rely on exact [2, 7, 19, 29] and stochastic planning [12, 28, 42, 43]. However, those planners were designed under certain hypotheses instead of understanding/evidence of the planning landscapes. We advocate that future planner design for SASs should be evidence-driven, supported by not only hypotheses but also clear insights about the properties and characteristics of the planning landscape [11] — the key point that this paper trying to make.

**Landscape Analysis for SASs and Configurable Systems**: Landscape analysis is a fairly new topic for configurable systems and SASs. Jamshidi and Casale [24] have briefly showcased the local optima of a configurable system, but this is achieved via visualizing the projected landscape, which does not provide a comprehensive summary. Likewise, Donckt et al. [18] also contribute to a simple analysis via 3D visualization. Frederick et al. [21] empirically compare the planners, but they do not comment about the landscape. Recently, Li et al. [31] apply local optima network — a special type of visualization graph — to study the landscape of configurable system. Our work differs from the above in that we demonstrate how quantifiable metrics and notions from the domain of fitness landscape analysis can be used to study SASs with respect to different environments while revealing interesting insights.

**Performance Analysis for SASs and Configurable Systems**: Another different but related topic is performance analysis for SASs and configurable systems, which concerns modeling the correlation between adaptation plan and performance, such as Jamshidi et al. [25] and Chen [4]. Those studies are orthogonal to this work as they are complementary to each other. For example, Jamshidi et al. [25] state that the model can be linearly transferred between workloads (the environment in this work). This matches with our finding that there is often a significant overlap on global/local optima of the planning landscapes between environments (RQ4) — a possible explanation on why they are linearly transferrable.

### 8 CONCLUSION AND FUTURE WORK

In this paper, we demonstrate how the metrics and notions from the domain of fitness landscape analysis can be derived for analyzing the planning landscapes of SASs. We study four real-world SASs under 14 different environments. Our results reveal several findings that can hint on the future research for SAS planner design.

We hope that this work can serve as a good starting point to raise the importance of planning landscape analysis for SASs and spark a dialog on a set of relevant future research directions for SAS planning. As such, the next stage on this research thread is vast, including building a dedicated methodology for SASs and fully exploring the implications as we discussed in the paper.

### REFERENCES

[1] Chris Anderson, Timothy Gordon Walmsley, and Panos Patros. 2021. A Self-Learning Architecture for Digital Twins with Self-Protection. In IEEE International Conference on Autonomic Computing and Self-Organizing Systems, ACSOS 2021, Washington, DC, USA, September 27 - Oct. 1, 2021. IEEE, 291–292.

[2] Radu Calinescu, Lars Grünne, Marta Z. Kwiatkowska, Raffaela Miranda, and Giordano Tamburrelli. 2011. Dynamic QoS Management and Optimization in Service-Based Systems. *IEEE Trans. Software Eng.* 37, 3 (2011), 387–409.

[3] Javier Cámara, Alessandro Vittorio Papadopoulos, Thomas Vogel, Danny Weyns, David Garlan, Shihong Huang, and Kenji Terui. 2020. Towards bridging the gap between control and self-adaptive system properties. In SEAMS '20: IEEE/ACM 15th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, Seoul, Republic of Korea, 29 June - 3 July, 2020. ACM, 78–84.

[4] Tao Chen. 2019. All versus one: an empirical comparison on retrained and incremental machine learning for modeling performance of adaptable software.
