Instance space analysis of combinatorial multi-objective optimization problems

Estefania Yap, Mario A. Muñoz, Kate Smith-Miles, Arnaud Liefooghe

Instance Space Analysis of Combinatorial Multi-objective Optimization Problems

Estefania Yap*, Mario A. Muñoz*, Kate Smith-Miles* and Arnaud Liefooghe†
*School of Mathematics and Statistics, The University of Melbourne, Parkville, VIC 3010, Australia
estefaniay@student.unimelb.edu.au, {munoz.m,smith-miles}@unimelb.edu.au
† JFLI – CNRS IRL 3527, University of Tokyo, 113-0033 Tokyo, Japan
arnaud.liefooghe@univ-lille.fr

***Abstract*—In recent years, there has been a continuous stream of development in evolutionary multi-objective optimization (EMO) algorithms. The large quantity of existing algorithms introduces difficulty in selecting suitable algorithms for a given problem instance. In this paper, we perform instance space analysis on discrete multi-objective optimization problems (MOPs) for the first time under three different conditions. We create visualizations of the relationship between problem instances and algorithm performance for instance features previously identified using decision trees, as well an independent feature selection. The suitability of these features in discriminating between algorithm performance and understanding strengths and weaknesses is investigated. Furthermore, we explore the impact of various definitions of "good" performance. The visualization of the instance space provides an alternative method of algorithm discrimination by showing clusters of instances where algorithms perform well across the instance space. We validate the suitability of existing features and identify opportunities for future development.**

***Index Terms*—Multi-objective optimization, black-box combinatorial optimization, landscape analysis, feature-based performance prediction.**

I. Introduction

Many real world problems consist of two or more, often conflicting, objectives. These are known as multi-objective optimization problems (MOPs), whose solution consists of a set of optimal trade-offs between the objectives. Evolutionary Multi-objective Optimization (EMO) algorithms are used to solve MOPs, due to their ability to maintain a diverse population of solutions spanning the trade-offs in a single simulation run.

Since the pioneering work of Schaffer [1], there has been substantial interest in the development of EMO algorithms. The now large number of available algorithms leads to the question of which one to choose, since the quality of the solution is affected by the choice of algorithm. This challenge is known as the algorithm selection problem [2] which aims to predict which algorithm will perform well with minimized cost to solve a given problem, where cost is measured in the resources used such as time. However, understanding the relationship between algorithm performance and problem instances is not a simple task. This is because problem difficulty is affected by many factors, such as multi-modality or ruggedness [3]. If the optimization problem can be stated analytically, such factors can be measured, but typical MOPs are black-box problems described only by sample input (decision variables) and outputs (objective function values). Given that no *a priori* information is available for black-box problems, how can we identify the relationship between instance features and algorithm performance?

The fitness landscape of black-box optimization problem instances offers an analysis framework, extracting features that characterize instance topology. These features quantify the challenges encountered by algorithms, independent of problem classes. Landscape analysis in single-objective optimization has receiving growing interest [4], with new features being developed. However, while they can be extended to multi-objective optimization, there may be features that are not relevant due to additional challenges incurred by the conflicting nature of the objectives. To counter this, features that are multi-objective in nature are necessary to characterize these problems. Such research exists [5]–[9] but many require enumeration of all Pareto optimal solutions and are therefore impractical for the algorithm selection problem which is attempting to predict performance based only on instance data, not algorithm performance data.

In recent work, Liefooghe et al. [10] introduce several new cheap features using short random and adaptive walks. This work showed significant features that helped discriminate between the performance of three algorithms (IBEA [11], MOEA/D [12], NSGA-II [13]) on a range of problem instances using ρmnk landscapes. Liefooghe et al. used a CART decision tree to provide an easily interpretative model to distinguish which algorithm is likely to be best for a given instance. A more complex random forest model [14] was also explored.

In this paper, we build upon this work by carrying out the first Instance Space Analysis (ISA) [15] of combinatorial multi-objective problems using the same instances, features and algorithms as [10]. Our aim is twofold: to create a visual complement using the CART decision tree features, and explore how the strengths and weaknesses of algorithms depend on the chosen criteria for good performance (i.e. best performing algorithm, or close to best). ISA also provides insights into how features change across the space and their correlation with algorithm performance. As such, we are able to generate a mapping of each algorithm's "footprint," i.e., the

Funding was provided by the Australian Research Council through the Australian Laureate Fellowship FL140100012.

regions of the space where it has strengths or weaknesses. Support vector machines (SVMs) are then used to decide which algorithm is recommended across various regions of the instance space, based on a set of calculated features. Through this analysis, we also assess the diversity of the ρmnk problem instances, and identify sparse areas in the instance space. Our analysis is comprised of three stages:

1) visually supporting conclusions of [10] – performing the first instance space analysis on the CART subset of features to see if similar rules can be visualized,
2) extending explanations – analysis using feature selection methods to consider additional features that may provide further insights, and
3) relaxing performance – exploring the impact when the definition of "good" is relaxed from best hyper-volume to be within 1% of best hyper-volume.

The remainder of this paper is organized as follows. In Section II, we introduce the general multi-objective combinatorial optimization problem, and the benchmark instances used for evaluation. We then describe the features that are utilized in this paper. In Section III we describe the instance space methodology for generating a $2D$ visualization of algorithm performance across instances. Results are presented in Section IV where we validate the relationship between features and algorithm performance in the previous paper [10]. Alternative instance spaces generated by different features are also presented, and we investigate the impacts of relaxed conditions of "good" performance to obtain stronger insights into algorithm strengths and weaknesses. Section V concludes the paper with a discussion of future research directions.

II. Background

A. Definition

Without loss of generality, consider a multi-objective minimization problem with m objectives and n decision variables:

$$\begin{aligned} \min\ \mathbf{y} &= \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), ..., f_m(\mathbf{x})) \\ \text{where } \mathbf{x} &= (x_1, ..., x_n) \in X \\ \mathbf{y} &= (y_1, ..., y_m) \in Y \end{aligned}$$

where $\mathbf{x}$ is known as the decision vector, X is the search space, $\mathbf{y}$ the objective vector and Y the objective or fitness space, and $\mathbf{f}(\mathbf{x})$ the fitness function. In multi-objective combinatorial optimization, X is a discrete set, i.e. $X \coloneqq \{0,1\}^n$. Given two solutions $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$, $\mathbf{x}^{(1)}$ dominates $\mathbf{x}^{(2)}$ (denoted as $\mathbf{x}^{(1)} \prec \mathbf{x}^{(2)}$) if it is better in at least one objective, and at least as good in all others. Solution $\mathbf{x}^{(1)}$ is said to be non-dominated with respect to $\mathbf{x}^{(2)}$, and $\mathbf{x}^{(2)}$ is dominated with respect to $\mathbf{x}^{(1)}$. This relation then holds for the objective vectors, with $\mathbf{y}^{(1)}$ is non-dominated with respect to $\mathbf{y}^{(2)}$. A solution $\mathbf{x}^\star$ is Pareto optimal (PO) if there is no solution $x \in X$ that dominates it. Similarly, $y^\star \in Y$ is non-dominated if there does not exist any $y \in Y$ such that $y^\star$ is dominated by y. The set of PO solutions forms the Pareto set, and their corresponding mapping into the objective space is known as the Pareto front (PF). Generating a good approximation of the PF is the main goal in solving MOPs.

B. Benchmark Instances

The selection of a representative subset of MOP instances is required in order to study the problem domain. For the purposes of the study, ρmnk-landscapes [8] are used to construct the test instances for combinatorial MOPs. This is due in part to their ability to control properties such as multimodality and objective correlation. A ρmnk-landscape can be formally defined as:

$$\begin{aligned} &\max\ f_i(x) = \frac{1}{n}\sum_{j=1}^{n} f_{ij}(x_j, x_{j_1}, \dots x_{j_k}) \quad i \in \{1, \dots, m\} \\ &\text{s.t. } x_j \in \{0,1\} \qquad\qquad\qquad\qquad\qquad\quad j \in \{1, \dots, n\} \end{aligned}$$

where n is the number of bits in the decision space, m the number of objectives and k the number of epistatic interactions influencing the contribution of each x_j. The fitness $f_i(x)$ of a solution $x \in X$ corresponds to the average value of the contributions in its n fitness components. Problem instances can be tuned from smooth to rugged by increasing k from 0 to $(n-1)$. The parameter $\rho > \frac{-1}{m-1}$ controls the degree of correlation between the objectives.

C. Features

The purpose of features is to quantify structural characteristics about a problem instance. They are used to summarize instances, and to identify the cause of difficulty. Features are used for gaining insights into the relationship between algorithms and instances, and consequently enabling suitable algorithm selection using performance prediction.

Multi-objective landscapes require the inclusion of multiple PO solutions, due to the trade-offs resulting from conflicting objectives. Similarly to the single-objective case, characteristics such as multimodality and ruggedness are empirically linked to instance difficulty and algorithm performance [16], [17]. Therefore, it is desirable to identify features that correlate well with these characteristics. In particular, useful features should be captured using the fewest function evaluations.

Liefooghe et al. [10] introduced local features which compute landscape information by using random or adaptive walks to sample the neighbourhood of solutions. The neighbourhood relation is defined as $N \colon X \mapsto 2^X$, and is here based on the 1-bit-flip operator. In a random walk, a random neighbour is selected at each step. During an adaptive walk, an improving (i.e., dominating) neighbour is selected at each step using a Pareto hill climber (PHC) [8].

During a walk of size ℓ, the neighbourhood N of each sample is explored. The proportion of dominated (#inf), dominating (#sup), incomparable (#inc), non-dominated (#lnd), and supported (#lsupp) solutions in the neighbourhood are measured. Features relating to the hyper-volume are also calculated. These include the hyper-volume covered by each

TABLE I: Features collected via sampling methods, used in this paper.

Description	**Random walk**		**Adaptive walk**
estimated correlation between objectives	f_cor_rws		
length of walk			length_aws
	First autocorrelation	**Average**	**Average**
proportion of dominated solutions	#inf_r1_rws	#inf_avg_rws	#inf_avg_aws
proportion of dominating solutions	#sup_r1_rws	#sup_avg_rws	#sup_avg_aws
proportion of incomparable solutions	#inc_r1_rws	#inc_avg_rws	#inc_avg_aws
proportion of non-dominated solutions	#lnd_r1_rws	#lnd_avg_rws	#lnd_avg_aws
proportion of supported solutions	#lsupp_r1_rws	#lsupp_avg_rws	#lsupp_avg_aws
hyper-volume covered by a single solution	hv_r1_rws	hv_avg_rws	hv_avg_aws
difference of hyper-volume covered by a single solution and its neighbourhood	hvd_r1_rws	hvd_avg_rws	hvd_avg_aws
hyper-volume covered by the entire neighbourhood	nhv_r1_rws	nhv_avg_rws	nhv_avg_aws

solution (hv), the difference between the hyper-volume covered by the neighbour and current solution (hvd), and the hyper-volume covered by the whole neighbourhood (nhv).

For samples collected during random and adaptive walks, features are measured and averaged. Additionally, the first autocorrelation coefficient of random walks is known to characterize the ruggedness of the single-objective landscapes [16]. Therefore, to accommodate this in the multi-objective case, the first autocorrelation coefficient of the above features during random walks is also measured. Lastly, the correlation between objectives is measured during the random walk. The full list of 26 features obtained during random and adaptive walks can be found in Table I.

D. Prior Work

Several new local features defined above were introduced in the work of Liefooghe et al. [10] with the intention of characterizing the structure of combinatorial MOPs. Tree-based predictive models were used to highlight the differences in important features between local and global search heuristics on small instances.

Large scale experiments were also performed for feature-based performance prediction. We utilize the same experimental setup in this paper. A total of 1,000 ρmnk-landscapes are created using latin hypercube sampling. The parameters of these instances are generated within the domains of $n \in \{64, \ldots, 256\}$, $k \in \{0, \ldots, 8, \}$, $m \in \{2, \ldots, 5, \}$ and $\rho \in [-1, 1]$ with $\rho > \frac{-1}{m-1}$. The local features in Section II-C are measured, as well as the values of m and n, as they are known parameters in black-box settings.

Three state-of-the-art EMO algorithms were chosen as representatives of different evolutionary approaches. The indicator based method is represented by IBEA [11], scalarization by MOEA/D [12], and dominance by NSGA-II [13]. These are run using default parameters in jMetal 4.5 [18]. All algorithms use a population size of 100, 2-point crossover with a rate of 0.9, and a bit-flip mutation rate of $1/n$. The budget is set at 10^6 evaluations. Performance is measured by the hyper-volume metric, with "good" initially defined as the best hyper-volume. Later we relax this goodness definition to be within 1% of best hyper-volume.

A CART decision tree was used in [10] to distinguish between the preferred algorithm for an instance, based on measured features. The features selected in the decision tree included lnd_avg_aws, hv_r1_rws, hvd_avg_rws, lnd_avg_aws and hv_avg_aws. A randomized trees model was used to generate the final output, with an error rate of 0.014 for the best statistical rank.

III. Methodology

Instance Space Analysis (ISA) is a methodology that allows the visualization of a group of problem instances in a $2D$ space. Within the *instance space* it is possible to observe trends in algorithm performance; hence, we are able to identify and objectively measure the regions of strength (and conversely, weaknesses) for an algorithm, which are known as *footprints*. Moreover, ISA gives us information about the distribution of the instances, allowing us to identify sparse regions where new instances may be needed. First described by Smith-Miles et al. [15], ISA can be automatically performed through publicly available webtools [19] or using the MATLAB toolkit [20]. In general, ISA involves the following steps:

1) collecting the meta-data, which corresponds to a table where each row represents a test instance and each column is a feature or the performance of an algorithm;
2) selecting a subset of features which best distinguish between different algorithm performances, and variations in problem instances;
3) projecting onto a $2D$ space for visualization;
4) measuring the algorithm footprints; and
5) generating new test instances in sparse regions.

In this study, we will not be generating new test instances, and instead focus on the analysis of the algorithm footprints. Details on all the steps of the ISA methodology can be found in previous work [21], [22]. However, we briefly summarise them here for reference. Once the meta-data has been collected, a binary performance measure is calculated that defines when an algorithm is "good." As mentioned above, "good" is here initially defined as largest hyper-volume or, later, as being within 1%. Then, the features are pre-processed by bounding the outliers within the mean plus or minus five times the interquartile range. Next, both the features and performance

measures are normalized and standardized using Box-Cox and Z transformations.

We filter out those features that are not highly predictive of algorithm performance, by calculating the absolute value of the Pearson correlation between them. Then, we cluster these meta-features using as similarity metric $1-|\rho_{x,y}|$, where $\rho_{x,y}$ is the correlation between two features. Silhouette analysis is used to determine the number of clusters. Retaining one meta-feature per cluster to reduce redundancy, we consider all valid combinations. Using PCA with two components as dimensionality reduction, and Random Forests (RF) to predict the binary measure, we determine which projection has the highest out-of-the-bag average accuracy. We use PCA and RFs as they are cheaper alternatives than the algorithms used in the following stages. This approach ensures that our selected meta-features are predictive of algorithm performance across the portfolio.

Next, we use the Prediction Based Linear Dimensionality Reduction (PBLDR) method [22] to find the final $2D$ projection, which creates the most linear trends of algorithm performance and feature values across the instance space, to assist visualization of directions of hardness and feature correlations. BFGS is used to solve numerically the underlying optimization problem in PBLDR, which is known to have infinite solutions. As such, we calculate 30 different projections and select the one with the highest topological preservation, defined as the correlation between high- and low-dimensional distances.

The instance space analysis assists in guiding automatic algorithm selection by highlighting which algorithms are compatible within regions of the space. For new instances, features are measured and projected onto the instance space. The location of this instance and whether it lies within any algorithm's footprint can be used to guide selection. We repeat this analysis for each of the three stages mentioned in Section I: (A) supporting conclusions drawn using the CART subset, (B) extending explanations through the use of the feature selection method used above and (C) relaxing performance by defining "good" performance as within 1% of best hyper-volume.

IV. Results

A. Supporting Conclusions

The final projection matrix of the generated instance space using the CART features, shown in Fig. 2, is defined by the following linear projection to $2D$. Each instance is a point in the $2D$ space shown in Fig 2, defined by:

$$\mathbf{Z}_{CART} = \begin{bmatrix} -0.1628 & -0.6232 \\ 0.4698 & 0.1364 \\ -0.4443 & 0.079 \\ -0.3854 & -0.048 \\ -0.249 & 0.7237 \end{bmatrix}^{\top} \begin{bmatrix} \text{lnd\_avg\_rws} \\ \text{hv\_r1\_rws} \\ \text{hvd\_avg\_rws} \\ \text{lnd\_avg\_aws} \\ \text{hv\_avg\_aws} \end{bmatrix} \quad (1)$$

While MOEA/D is the algorithm that performs best on average, good performance is almost evenly split between NSGA-II, as seen in Fig. 1b and 1c. The footprints for each algorithm are mostly clustered. Fig. 2 shows hv_avg_aws has a good correlation with the splitting of spaces in which each algorithm is superior. NSGA-II performs well on instances with a larger value, while the opposite is true for MOEA/D. This is a similar recommendation given by the CART model. This feature is known to strongly negatively correlate with the number of objectives m [10], suggesting that MOEA/D is preferred for more objectives. Thus, our approach shows an alternative approach to the CART model and generates a visual representation of performance.

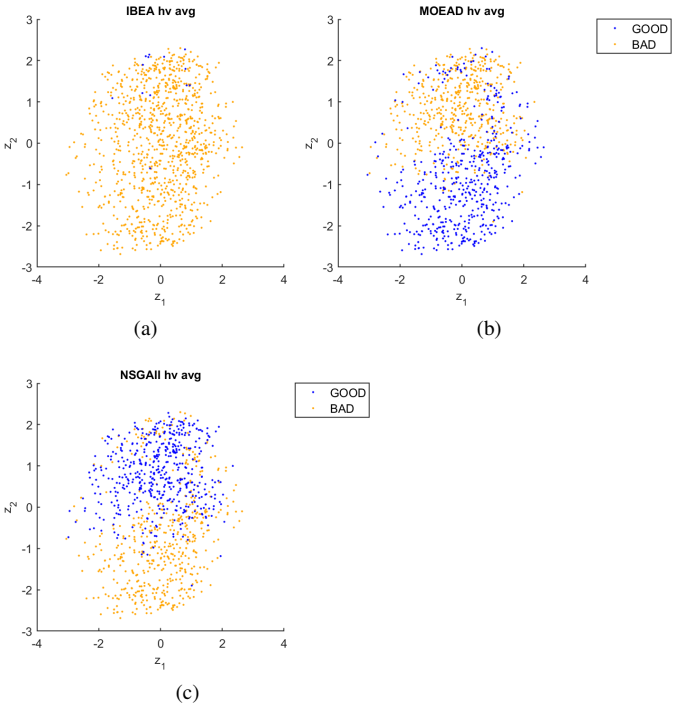


Fig. 1: Algorithm footprints in the projected instance space of (1) using the CART subset of features.

In all generated instance spaces, it is evident that IBEA only performs well on a subset of the instances that NSGA-II performs well on. Fig. 7 shows that it is never recommended by the SVM as the preferred algorithm. The SVM in Fig. 7a shows that MOEA/D and NSGA-II are each selected as the preferred algorithm in approximately half of the instances.

B. Extending Explanations

When using feature selection on the complete set, only two features were shared with those in the CART subset (hv_avg_aws, lnd_avg_rws) shown in Fig. 4. The final projection matrix of the generated instance space in Fig. 4 is defined by:

$$\mathbf{Z}_{featselect} = \begin{bmatrix} 0.2209 & -0.2881 \\ -0.3158 & 0.1127 \\ 0.2037 & 0.3388 \\ -0.3081 & -0.0352 \\ -0.1906 & 0.0698 \\ 0.2783 & 0.3268 \\ 0.4636 & -0.0265 \end{bmatrix}^{\top} \begin{bmatrix} \text{sup\_avg\_rws} \\ \text{hvd\_avg\_aws} \\ \text{inc\_avg\_rws} \\ \text{hv\_avg\_aws} \\ \text{nhv\_avg\_aws} \\ \text{lnd\_avg\_rws} \\ \text{m} \end{bmatrix} \quad (2)$$

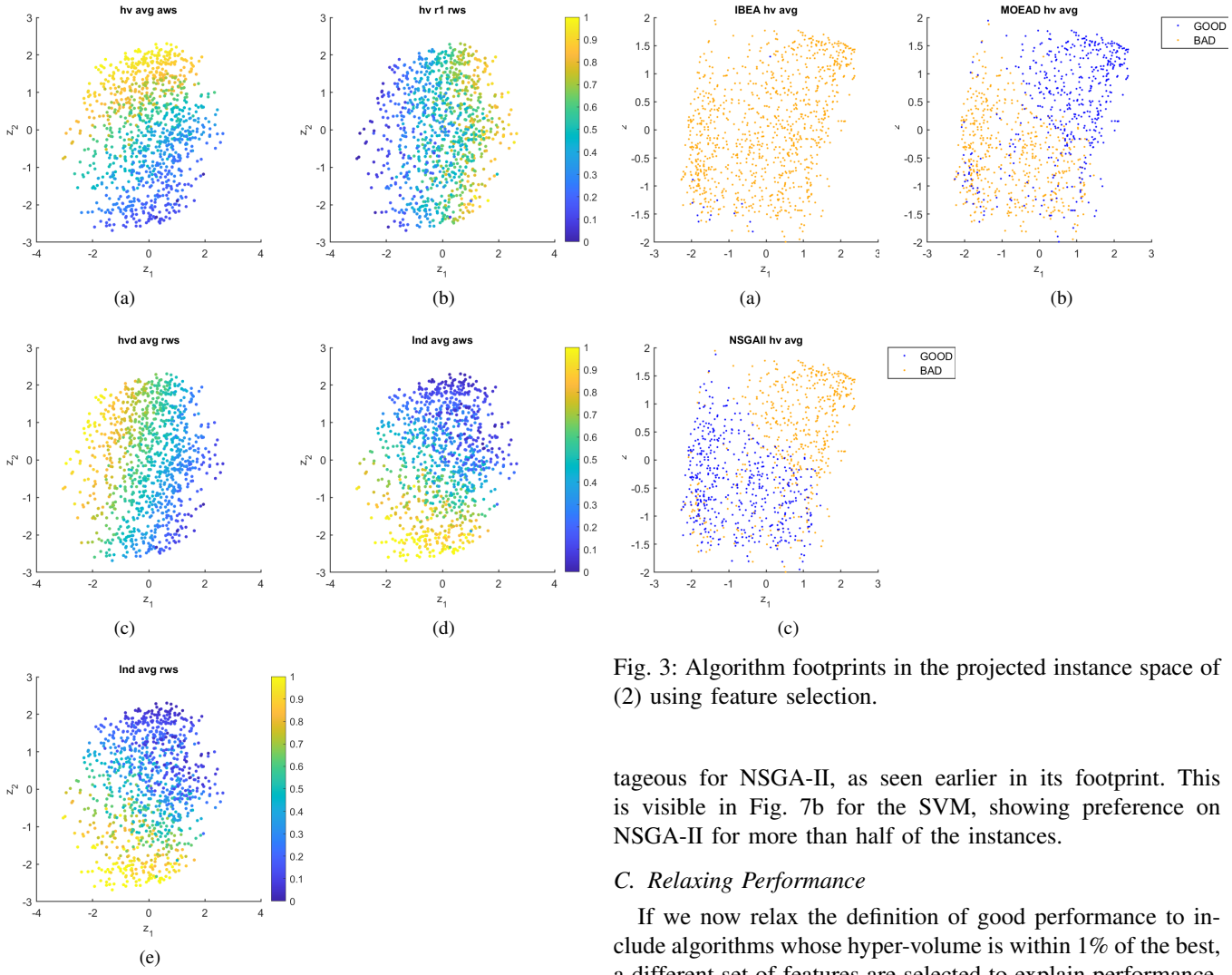


Fig. 2: Distribution of normalized CART subset of features in the projected instance space of (1). The color scale corresponds to normalized feature values.

We observe the negative correlation between m and hv_avg_aws, with MOEA/D's footprint occupying instances with larger m. The performance of MOEA/D on larger values of sup_avg_rws is also observed. This feature is correlated with the objective correlation ρ [10], suggesting that MOEA/D performs well when ρ is larger, i.e. when the degree of conflict between the objectives is small.

The subset of features selected shows a similar partitioning of the instance space between MOEA/D and NSGA-II's performance, as shown in Fig. 3b and 3c. However, the projection seems to favour NSGA-II slightly, showing a clearer partition of best performance. Here, the feature hv_avg_aws correlates with algorithm performance, as do inc_avg_rws and lnd_avg_rws. As a result, these features are able to generate a useful instance space for inferring algorithm performance.

The projection of the instance space here is more advantageous for NSGA-II, as seen earlier in its footprint. This is visible in Fig. 7b for the SVM, showing preference on NSGA-II for more than half of the instances.

Fig. 3: Algorithm footprints in the projected instance space of (2) using feature selection.

C. Relaxing Performance

If we now relax the definition of good performance to include algorithms whose hyper-volume is within 1% of the best, a different set of features are selected to explain performance. The final projection matrix of the generated instance space in Fig. 6, can be defined by:

$$\mathbf{Z}_{relaxedperf} = \begin{bmatrix} 0.5246 & -0.189 \\ -0.0935 & 0.1986 \\ 0.1557 & -0.5934 \\ -0.3895 & -0.1462 \\ 0.4879 & -0.0472 \\ 0.0813 & 0.3494 \\ 0.0447 & 0.2091 \end{bmatrix}^{\top} \begin{bmatrix} \text{lnd\_avg\_aws} \\ \text{hv\_avg\_aws} \\ \text{m} \\ \text{length\_aws} \\ \text{inc\_avg\_aws} \\ \text{hvd\_avg\_aws} \\ \text{nhv\_avg\_aws} \end{bmatrix} \quad (3)$$

The feature hv_avg_aws is once again selected as a good predictor for algorithm performance.

In relaxed conditions of "good", Fig. 5c shows that the space where NSGA-II displays good performance is more cleanly partitioned. However, MOEA/D performs so strongly that its footprint overlaps with that of NSGA-II's, without any clear separation. While the earlier analyses showed the two algorithms complementing each other, here the performance of MOEA/D is superior. Due to the shared good performance on much of the space, the SVM has difficulties in identifying clustered regions within the instance space where MOEA/D performs well. Therefore, the features currently used may

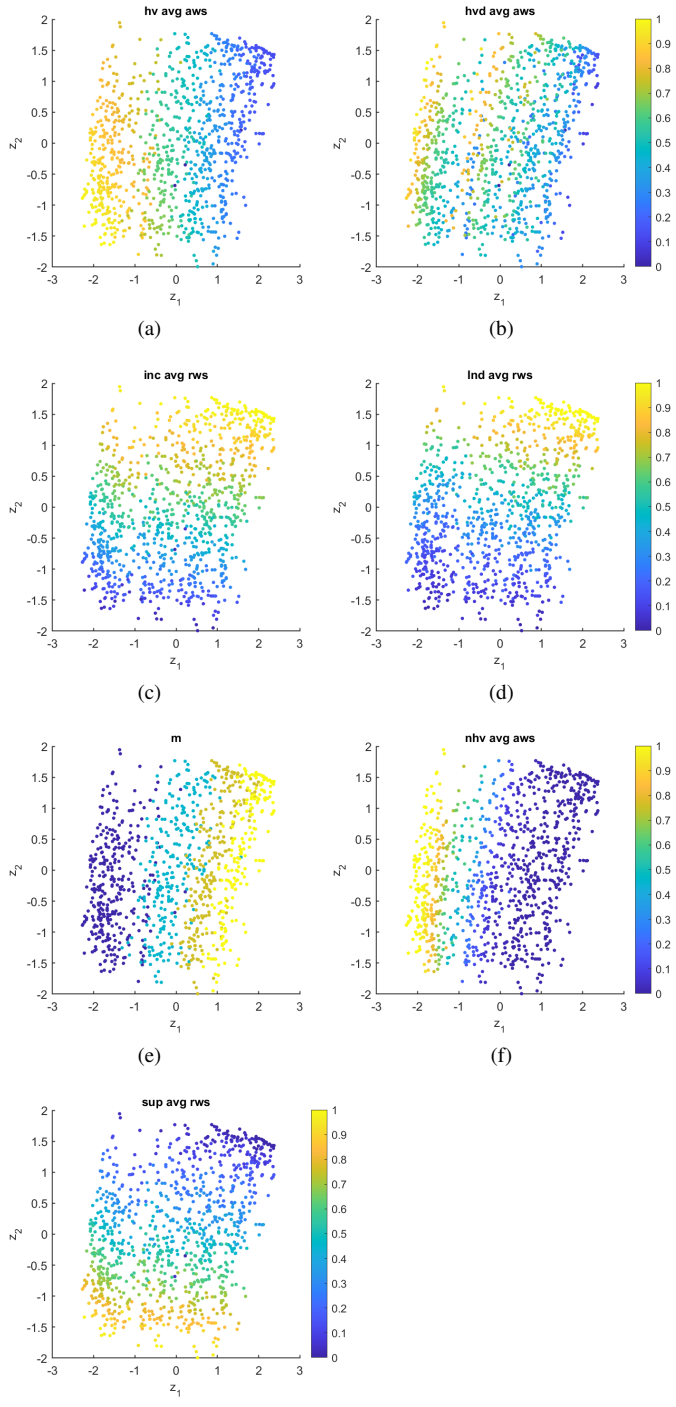


Fig. 4: Distribution of normalized subset of features in the projected instance space of (2).

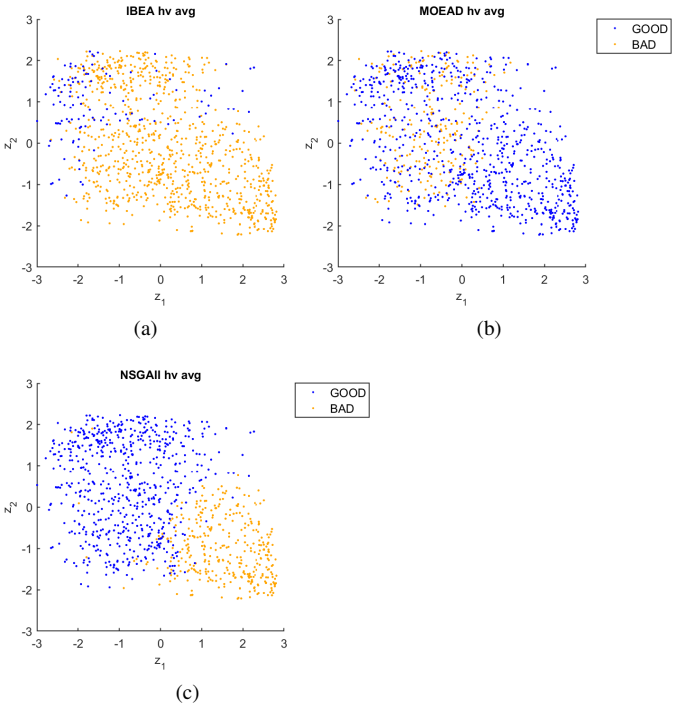


Fig. 5: Algorithm footprints in the projected instance space of (3) using feature selection when performance is relaxed.

not be sufficient as the definition of best becomes more lenient. The feature hv_avg_aws remains correlated with the performance of NSGA-II. Another feature that correlates is length_aws, an estimator of the number of local optima [10]. The larger the length, the larger the size of the basin of attraction, and thus the lower number of local optima. Since NSGA-II does not perform well for smaller values length_aws, we infer that it struggles when there is a larger number of local optima.

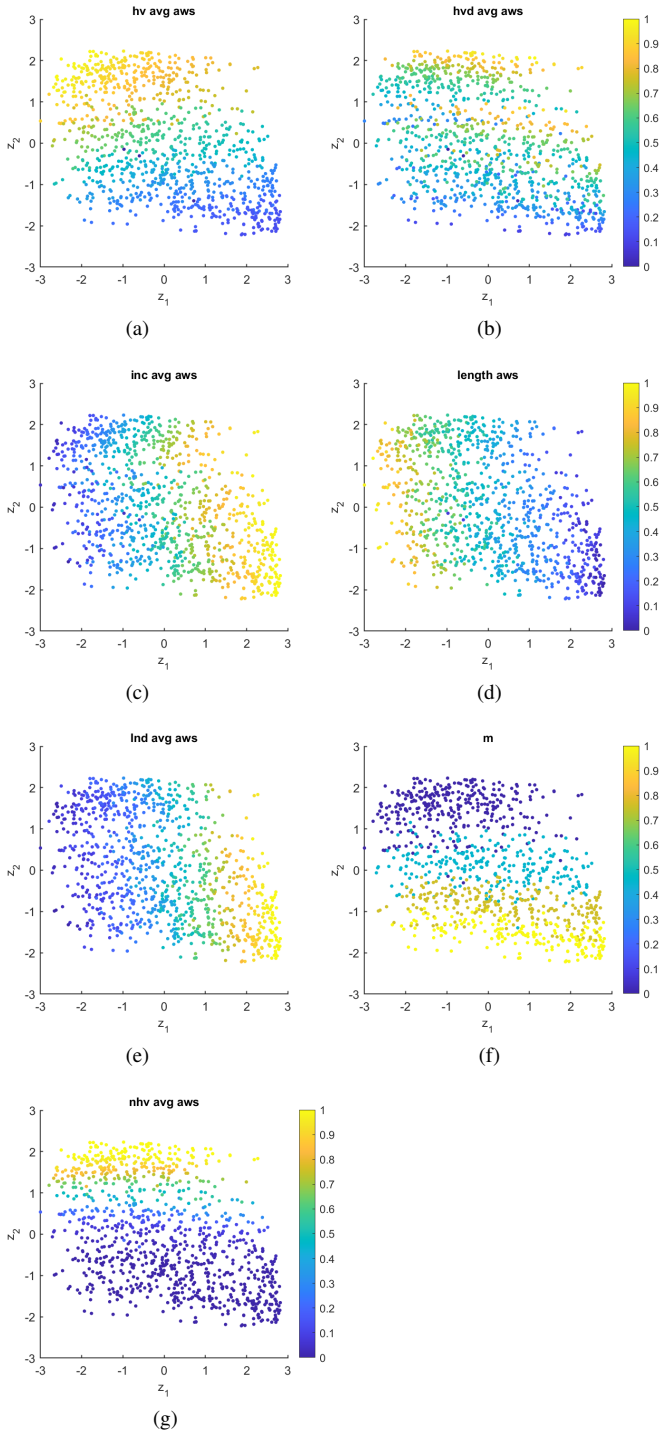
The boundaries of NSGA-II's footprint are once again clearer than MOEA/D's. The few instances MOEA/D performed poorly on were scattered within the domain where both MOEA/D and NSGA-II performed well, and thus the SVM shown in Fig. 7c always conservatively recommended NSGA-II over that entire domain.

D. Algorithm Selection Results

The test instances for all analyses cover a large portion of the instance spaces generated with minor empty pockets closer to the origin, as seen in Fig. 7. We conclude that the set of test instances studied are quite diverse and span the instance space sufficiently to infer performance behaviours across the instance space.

V. Conclusions and Future Work

In this paper, we carried out the first Instance Space Analysis (ISA) of combinatorial MOPs, using as a basis the work by Liefooghe et al. [10]. ISA complements that of the decision tree analysis, offering additional insights into algorithm strengths and weaknesses through visualizations. The development of such insights into the transparency of algorithm performance allows for better automated algorithm selection for new, untested instances. When an untested instance is encountered, features are evaluated mapped into their instance space co-ordinates. Given these co-ordinates, the



(a) (b) (c) (d) (e) (f) (g)

Fig. 6: Distribution of normalized subset of features in the projected instance space of (3) when performance is relaxed.

suitable algorithms can be inferred by selecting those which show dominant performance over the region. Moving forward, there is the opportunity to extend the approach to consider a broader range of algorithms and instance features to explore a more comprehensive understanding of algorithm behaviours.

Investigation of the algorithm footprints showed that MOEA/D displays strong performance under relaxed conditions of goodness, as seen in Fig. 5. However, its weaknesses are sparsely located. Therefore, despite MOEA/D being the higher performing algorithm, the SVM selected NSGA-II more often due to NSGA-II's consistent region of good performance. While these features may be sufficient in selecting the best algorithm by hyper-volume, when the condition of goodness is relaxed, they may not provide enough information to separate algorithm performance.

We therefore focus our attention toward developing features that hold more discrimination between algorithms in future, as well as expanding our algorithm portfolio. Furthermore, we are interested in developing features that can be calculated cheaply to minimize the total cost of function evaluations. The amalgamation of this work will provide further insights, assisting with better informed algorithm selection in future in the multi-objective space.

REFERENCES

[1] J. D. Schaffer, "Some experiments in machine learning using vector evaluated genetic algorithms (artificial intelligence, optimization, adaptation, pattern recognition)," Ph.D. dissertation, Nashville, TN, USA, 1984, aAI8522492.

[2] J. Rice, "The algorithm selection problem," ser. Advances in Computers, M. Rubinoff and M. C. Yovits, Eds. Elsevier, 1976, vol. 15, pp. 65 – 118. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0065245808605203

[3] K. M. Malan and A. P. Engelbrecht, "A survey of techniques for characterising fitness landscapes and some possible ways forward," *Information Sciences*, vol. 241, pp. 148 – 163, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025513003125

[4] O. Mersmann, M. Preuss, and H. Trautmann, "Benchmarking evolutionary algorithms: Towards exploratory landscape analysis," in *Parallel Problem Solving from Nature, PPSN XI*, R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 73–82.

[5] J. Knowles and D. Corne, "Instance generators and test suites for the multiobjective quadratic assignment problem," in *Evolutionary Multi-Criterion Optimization*, C. M. Fonseca, P. J. Fleming, E. Zitzler, L. Thiele, and K. Deb, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 295–310.

[6] L. Paquete and T. Stützle, "Clusters of non-dominated solutions in multiobjective combinatorial optimization: An experimental analysis," in *Multiobjective Programming and Goal Programming*, V. Barichard, M. Ehrgott, X. Gandibleux, and V. T'Kindt, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 69–77.

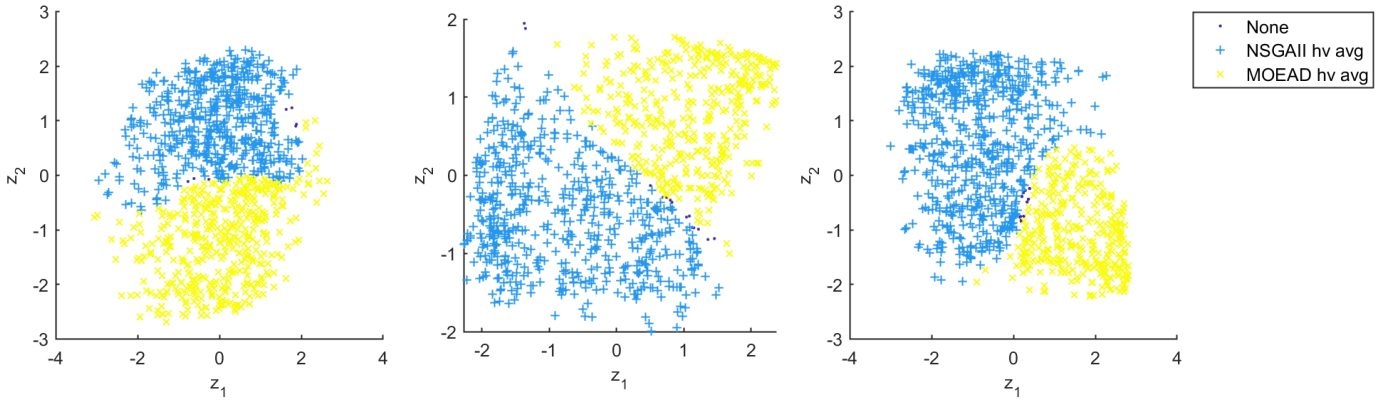
[7] H. E. Aguirre and K. Tanaka, "Working principles, behavior, and performance of moeas on mnk-landscapes," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1670 – 1690, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221706005455

[8] S. Verel, A. Liefooghe, L. Jourdan, and C. Dhaenens, "On the structure of multiobjective combinatorial search space: Mnk-landscapes with correlated objectives," *European Journal of Operational Research*, vol. 227, no. 2, pp. 331 – 342, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221712009630

[9] A. Liefooghe, S. Verel, H. Aguirre, and K. Tanaka, "What makes an instance difficult for black-box 0–1 evolutionary multiobjective optimizers?" in *Artificial Evolution*, P. Legrand, M.-M. Corsini, J.-K. Hao, N. Monmarché, E. Lutton, and M. Schoenauer, Eds. Cham: Springer International Publishing, 2014, pp. 3–15.

[10] A. Liefooghe, F. Daolio, S. Verel, B. Derbel, H. Aguirre, and K. Tanaka, "Landscape-aware performance prediction for evolutionary multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, 2019. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02294201

[11] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature - PPSN VIII*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 832–842.



(a) CART subset projected using (1) (b) Feature selection projected using (2) (c) Relaxed performance projected using (3)

Fig. 7: SVM algorithm recommendations for each of the three instance space projections defined by equations (1), (2), (3).

[12] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, Dec 2007.

[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr 2002.

[14] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, p. 3–42, Apr. 2006. [Online]. Available: https://doi.org/10.1007/s10994-006-6226-1

[15] K. Smith-Miles, D. Baatar, B. Wreford, and R. Lewis, "Towards objective measures of algorithm performance across instance space," *Computers & Operations Research*, vol. 45, pp. 12 – 24, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0305054813003389

[16] E. Weinberger, "Correlated and uncorrelated fitness landscapes and how to tell the difference," *Biological Cybernetics*, vol. 63, no. 5, pp. 325–336, Sep 1990. [Online]. Available: https://doi.org/10.1007/BF00202749

[17] P. Kerschke, H. Wang, N. Preuss, C. Grimme, A. Deutz, H. Trautmann, and M. Emmerich, "Towards analyzing multimodality of continuous multiobjective landscapes," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2016, pp. 962–972.

[18] J. J. Durillo and A. J. Nebro, "jmetal: A java framework for multi-objective optimization," *Advances in Engineering Software*, vol. 42, no. 10, pp. 760 – 771, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0965997811001219

[19] K. Smith-Miles, "Melbourne algorithm test instance library with data analytics (MATILDA)," Available at https://matilda.unimelb.edu.au, 2020.

[20] M. Muñoz, "Instance space analysis: A toolkit for the assessment of algorithmic power," Source code is available at https://github.com/andremun/InstanceSpace, 2020.

[21] S. Kandanaarachchi, M. A. Muñoz, and K. Smith-Miles, "Instance space analysis for unsupervised outlier detection," in *EDML@SDM*, 2019.

[22] M. A. Muñoz, L. Villanova, D. Baatar, and K. Smith-Miles, "Instance spaces for machine learning classification," *Mach. Learn.*, vol. 107, no. 1, p. 109–147, Jan. 2018. [Online]. Available: https://doi.org/10.1007/s10994-017-5629-5