Accelerated Pattern Search with Variable Solution Size for Simultaneous Instance Selection and Generation

Hoang Lam Le[1], Ferrante Neri[2], Dario Landa-Silva[1], Isaac Triguero[1,3]

hoang.le,dario.landasilva,isaac.triguero@nottingham.ac.uk;f.neri@surrey.ac.uk
[1]COL Group, School of Computer Science, University of Nottingham, Nottingham NG8 1BB, United Kingdom
[2]NICE Group, Department of Computer Science, University of Surrey, Guildford GU2 7XH, United Kingdom
[3]DaSCI Andalusian Institute in Data Science and Computational Intelligence, University of Granada, Spain

ABSTRACT

The search for the optimum in a mixed continuous-combinatorial space is a challenging task since it requires operators that handle both natures of the search domain. Instance reduction (*IR*), an important pre-processing technique in data science, is often performed in separated stages, combining instance selection (*IS*) first, and subsequently instance generation (*IG*). This paper investigates a fast optimisation approach for *IR* considering the two stages at once. This approach, namely Accelerated Pattern Search with Variable Solution Size (APS-VSS), is characterised by a variable solution size, an accelerated objective function computation, and a single-point memetic structure designed for *IG*.

APS-VSS is composed of a global search crossover and three local searches (*LS*). The global operator prevents premature convergence to local optima, whilst the three *LS* algorithms optimise the reduced set (*RS*). Furthermore, by using the k-nearest neighbours algorithm as a base classifier, APS-VSS exploits the search logic of the *LS* to accelerate, by orders of magnitude, objective function computation. The experiments show that APS-VSS outperforms existing algorithms using the single-point structure, and is statistically as competitive as state-of-the-art *IR* techniques regarding accuracy and reduction rates, while reducing significantly the runtime.

KEYWORDS

Memetic Algorithm, Pattern Search, Instance Reduction, Classification, Data Science, Combinatorial/ Continuous Optimisation

ACM Reference Format:
Hoang Lam Le[1], Ferrante Neri[2], Dario Landa-Silva[1], Isaac Triguero[1,3]. 2022. Accelerated Pattern Search with Variable Solution Size for Simultaneous Instance Selection and Generation. In *Genetic and Evolutionary Computation Conference Companion (GECCO '22 Companion), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3520304.3529020

1 INTRODUCTION

With the explosion in the amount of data in the digital age, the size of available training data (*TR*) has become enormous in many areas, potentially having noise and imperfections. Pre-processing is essential to make downstream processes more accurate. Among other data pre-processing techniques, data reduction techniques such as feature selection, instance reduction, or discretisation result in a cleaner and smaller dataset that is free from noise, redundant or irrelevant samples. Most of the existing *IR* solutions were proposed to enhance the performance of the well-known Nearest Neighbour (NN) classifier [2], which will be also adopted in this work. However, any other classifier can use the resulting reduced dataset for learning [1].

The purpose of *IR* is the identification of a smaller set of the starting dataset which is as informative as the original source. Research about *IR* can be categorised into instance selection *IS* [4] and instance generation *IG* [9]. The former has frequently been modelled as a binary combinatorial optimisation problem as it deals with the decision whether or not to include a sample in the final subset, whilst the latter may be modelled as a continuous optimisation problem, considering generating new examples non-existing in the source but better to represent the training data.

Considering both reduction rate and accuracy, state-of-the-art performance in *IR* techniques is achieved by hybrid approaches in which *IS* and *IG* combine their effectiveness to produce a final reduced dataset [5]. In practice, all hybrid techniques proposed so far in the literature, make use of cascade approaches which perform *IS* and *IG* in separated and subsequent stages. In contrast, a method that determines the most appropriate number of instances while performing *IG* has not been explored yet.

Popular hybrid approaches combine a Steady State Memetic Algorithm (SSMA) [3] plus an *IG* method [5, 10]. After optimising the number of samples using SSMA, hybrid approaches feed the subset of well-distributed samples to an *IG* algorithm to refine their positions. Despite its effectiveness, this type of hybrid methods are typically very time-consuming due to the expensive cost of computing the objective function, not only caused by SSMA but also by the *IG* phase (if tackled by evolutionary methods). A recent simple, yet effective memetic approach [5] is able to yield major savings in computational overhead during the *IG* phase but it still relies on SSMA feeding in a subset of compact and well-rep resentative samples among classes. Thus, the computational cost of the overall *IR* process is still high.

APS-VSS is proposed to perform a fast reduction of the data by simultaneously doing *IS* and *IG*. Our proposal stems from the single-point memetic structure (SPMS), proposed in [5, 7]. SPMS algorithm is a domain-specific technique for *IG* belonging to the family of Memetic Computing, which features an accelerated objective function evaluation. Conversely, the proposed APS-VSS

GECCO '22 Companion, July 9–13, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9268-6/22/07.
https://doi.org/10.1145/3520304.3529020

integrates two domain-specific *LS* components to optimise the total number of instances appearing in the reduced set within the *IG* local search of [5].

2 ACCELERATED PATTERN SEARCH WITH VARIABLE SOLUTION SIZE

Section 2.1 highlights the novelty, Section 2.2 defines the problem, and Section 2.3 presents the implementation details.

2.1 Novelty of the proposed approach

Two main branches of *IR* may belong to different search space domains. While *IS* may be considered in the combinatorial search space, *IG* is in the continuous domain. Though there are *IR* techniques working in both search spaces but they are applied sequentially. The current state-of-the-art approaches are accurate but may be extremely computationally expensive [10]. For example, with the dataset "Magic" (19020 samples, 10 features), on average of 10 runs, SSMA-SFLSDE consumed 68.3 hours to complete [10], while SSMA-LSHADE took 92.6 hours [5]. Unlike previous studies which address *IR* in separated stages, APS-VSS performs the selection and generation within a single framework.

2.2 Problem Definition

Given an instance I having m features and belonging to a class w. In a supervised classification problem, a TR set contains l entries of I. An *IR* algorithm aims to provide a smaller set RS of p instances ($p < l$) by either selection or generation from the examples in TR. Let x be a candidate solution formed by flattening matrix RS into a one-dimensional array.

$$\mathbf{x} = (x_1, x_2, \ldots, x_n) \tag{1}$$

where x_i represents the design variable of the feature perturbation process and $n = m \cdot p$.

Let $\mathbf{e^i}$ be a vector in an n-dimensional space whose elements are all zeros except from the i^{th} element which is one. LSIR [7] is based on a greedy *LS* in the family of Pattern Search algorithms [6]. Every single element in the one-dimensional array x is perturbed one at a time in its feasible range of values. The perturbed x is then evaluated to detect if any improvement is found. The adjusted value can replace the current one if the trial solution is non-worsening.

2.3 Algorithmic Description

The proposed APS-VSS (Algorithm 1) partly makes use of the successful search logic of SPMS-ALS [5] but completely reinterprets the *IR* problem representation and hence the search space associated with it. APS-VSS employs two novel extra *LS* components, named LS^{eli} and LS^{asc}, respectively.

LS^{eli} (lines 3-8) discards any elements in *RS* whose absence does not affect the current solution quality. Although this simple idea of elimination relies heavily on the quality of initial sampling, the output is yet promising as the *RS* can be enhanced by the generation phase of the greedy LS (lines 9-26).

Different to LS^{eli}, LS^{asc} (lines 19-25) is called only occasionally as its main role is to confirm the importance of the adjusted sample. It is necessary because the feature perturbation process may adjust it into an already existing (redundant) sample in *RS*.

Algorithm 1 Accelerated Pattern Search with Variable Solution Size (symbol '=' indicates an assignment)

1: **INPUT** x
2: **while** local budget and precision conditions are not met **do**
3: **for** $h = 1 : p$ **do**
4: $\mathbf{x}_h$ = $\mathbf{x}$ after removing the elements $b_{h1}, b_{h2}, \ldots, b_{hm}$
5: **if** $f(\mathbf{x_h}) \geq f(\mathbf{x})$ **then**
6: $x = \mathbf{x_h}$
7: **end if**
8: **end for**
9: **for** $i = 1 : n$ $(n = m \cdot p)$ **do**
10: $\mathbf{x^t} = \mathbf{x} - \rho \cdot \mathbf{e}^i$
11: **if** $f(\mathbf{x^t}) \geq f(\mathbf{x})$ **then**
12: $\mathbf{x} = \mathbf{x^t}$
13: **else**
14: $\mathbf{x^t} = \mathbf{x} + \frac{\rho}{2} \cdot \mathbf{e}^i$
15: **if** $f(\mathbf{x^t}) \geq f(\mathbf{x})$ **then**
16: $\mathbf{x} = \mathbf{x^t}$
17: **end if**
18: **end if**
19: **if** $\mod(i, m) = 0$ **then**
20: j = i / m ▷ Get index of the generated example
21: $\mathbf{x}_t^{'}$ = $\mathbf{x}_t$ after removing the elements $b_{j1}, b_{j2}, \ldots, b_{jm}$
22: **if** $f\left(\mathbf{x}_t^{'}\right) \geq f(\mathbf{x}_t)$ **then**
23: $\mathbf{x} = \mathbf{x}_t^{'}$
24: **end if**
25: **end if**
26: **end for**
27: **if** x has **not** been updated **then**
28: halve the exploratory radius ρ
29: **if** $\rho < \epsilon$ **then**
30: Randomly generate a candidate solution $\mathbf{x^r}$
31: Apply Crossover between x and $\mathbf{x^r}$ to generate a new trial vector x^t
32: Reinitialise $\mathbf{x} = \mathbf{x^t}$
33: **end if**
34: **end if**
35: **end while**
36: **RETURN** x
37:

Finally, the gene-resampling mechanism (lines 27-34), taken from SPMS-ALS, is an important countermeasure to prevent the sample adjustment from overfitting. An excessive effort on sample adjustment is likely to yield an overfitted RS, which results in poor performance on unseen data [5].

3 EXPERIMENTS AND ANALYSIS

Section 3.1 introduces the experimental setup, Section 3.2 highlights the effectiveness of LS^{eli} and LS^{asc}, Sections 3.3 and 3.4 discuss the performance between APS-VSS against different models, and Section 3.5 presents advantages of APS-VSS. Details of all numerical results can be found at the repository[1].

3.1 Experimental Setup

We use 57 multi-class datasets with various sizes from the KEEL dataset repository, and group them into small and medium categories based on the number of samples. To validate the results, we use a 10-fold stratified cross-validation (10-fcv) procedure to partition each dataset. Thus, the reported training and test performance of each dataset is a 10-fold average.

Compared algorithms include baselines and state-of-the-art *IR* solutions. Baselines are Nearest Neighbour (1NN) employing the

[1] https://github.com/lehoanglam20000/APS-VSS

entire TR for training, LSIR [7] and SPMS-ALS [5]. State-of-the-art approaches are hybrid techniques including SSMA-LSHADE [5, 8], SSMA-SFLSDE [10], SSMA-SPMS-ALS [5].

The same parameters suggested by the authors in their original works have been used. Apart from the shared parameters obtained from SPMS-ALS including ρ, $N_{\max}$, ρ_{Red}, ρ_{Thr} and Gr [5], APS-VSS requires the tuning of the P_{init} parameter, which is the percentage of initialisation. As described in Section 2.3, LS^{eli} relies on the quality of initial sampling, P_{init} is tuned in a wide range, from 5% up to 70%. P_{init} = 5 meaning 5% of samples of a class are randomly allocated to the initial RS.

3.2 Search behaviour

As an example, Figure 1 characterises the search behaviour of APS-VSS and SPMS-ALS on the zoo dataset.

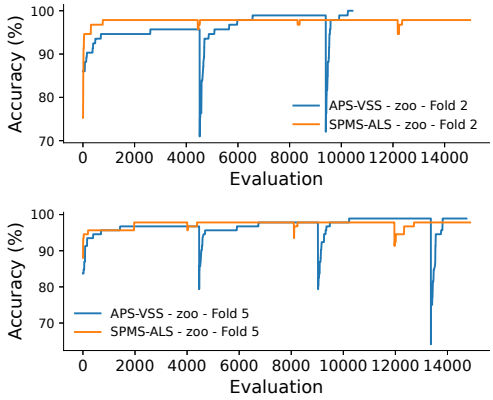

Figure 1: Search behaviour of APS-VSS and SPMS-ALS at dataset zoo, folds 2 and 5.

The search consists of several cycles, separated by a straight drop. The first cycle begins with random initialisation of RS, and end after about 4,500 evaluations. APS-VSS and SPMS-ALS do not share the same initial percentage of samples, as they start at different positions. Then, the global evolutionary operator introduces new materials to let the search start at a new region, initialising the start of the second cycle.

At Fold 2, through the feature perturbation, the accuracy value of SPMS-ALS progressively increases and remains unchanged at 97-98% until the end of the first search cycle, while APS-VSS only reaches 94-95%. Around evaluation 4500, the upward trends at both algorithms suddenly drop sharply due to the effect of the global gene-resampling crossover operator. It depends on the parameters used for the operator to observe how far the accuracy drops down. A new similar pattern of the search restarts from evaluation 4500 as the elements in the RS have been refreshed. APS-VSS gradually develops the accuracy whilst SMPS-ALS goes back to its previous peak, which can be attributed to the impact of the two new LS components in APS-VSS.

Different from Fold 2, the two algorithms used up the computational budget in Fold 5. SPMS-ALS shows a better performance at a few first cycles, but eventually APS-VSS obtains higher accuracy. APS-VSS is likely more consistent to make progression after each restart than SPMS-ALS.

3.3 Comparison with baseline models

Table 1 presents the average training and test, the number of Wins, Ties and Losses of APS-VSS with baseline models in both small and medium datasets. The $p-$value is calculated to confirm if APS-VSS statistically outperforms the algorithm in the row.

Table 1: Performance of APS-VSS and baselines

		TRAINING	TEST	APS-VSS vs			p-value
		Acc ± Std	Acc ± Std	Win	Tie	Lose	APS-VSS vs
SMALL	1NN	0.7367 ± 0.012	0.7388 ± 0.061	27	1	**12**	**0.031**
	LSIR	0.8693 ± 0.014	0.7415 ± 0.061	**28**	0	**12**	**0.001**
	SPMS-ALS	0.8733 ± 0.011	0.7549 ± 0.062	25	0	15	**0.044**
	APS-VSS	**0.8753** ± 0.015	**0.7656** ± 0.064	–	–	–	–
MEDIUM	1NN	0.8322 ± 0.006	0.8308 ± 0.017	**11**	11	**3**	0.08
	LSIR	0.9052 ± 0.004	0.8609 ± 0.014	8	0	9	0.644
	SPMS-ALS	0.9199 ± 0.003	0.8668 ± 0.011	6	1	10	0.431
	APS-VSS	**0.9271** ± 0.005	**0.8682** ± 0.014	–	–	–	–

APS-VSS has obtained the highest average test accuracy in either small or medium datasets. p-values at the last column have confirmed the difference is significant in small datasets, while no significant difference found in medium ones.

APS-VSS has a substantially greater number of Wins than Losses with respect to the three baseline models in small datasets. Specifically, it is more than 2 times compared with LSIR and 1NN, and more than 1.5 times compared with SPMS-ALS. On the contrary, those figures are distributed mostly equal im medium datasets which does not help distinguish which one outperforms the other.

3.4 Comparison with state-of-the-art models

In Table 2 p-value at the last column refers to the Wilcoxon statistical test of the algorithm in the row to APS-VSS.

Table 2: Performance of APS-VSS and state-of-the-art models

		TRAINING	TEST	APS-VSS vs			p-value
		Acc ± Std	Acc ± Std	Win	Tie	Lose	vs APS-VSS
SMALL	SSMA-LSHADE	0.8687 ± 0.010	**0.7792** ± 0.057	11	0	29	0.008
	SSMA-SFLSDE	0.8684 ± 0.011	0.7767 ± 0.059	15	0	25	0.096
	SSMA-SPMS-ALS	0.8727 ± 0.013	0.7670 ± 0.057	**19**	0	**21**	0.979
	APS-VSS	**0.8815** ± 0.015	0.7623 ± 0.061	–	–	–	–
MEDIUM	SSMA-LSHADE	0.9069 ± 0.004	**0.8706** ± 0.012	5	0	12	0.145
	SSMA-SFLSDE	0.9059 ± 0.004	0.8675 ± 0.013	**6**	0	**11**	0.712
	SSMA-SPMS-ALS	0.9264 ± 0.003	0.8700 ± 0.011	5	0	12	0.145
	APS-VSS	**0.9271** ± 0.005	0.8682 ± 0.014	–	–	–	–

Though average training performance goes up in the order of SSMA-LSHADE, SSMA-SFLSDE, SSMA-APMS-ALS and APS-VSS, the average test performance does not follow the same pattern in both small and medium datasets.

In either small or medium datasets, APS-VSS has no significant difference with and SSMA-SFLSDE, SSMA-APMS-ALS. This information shows the robustness of APS-VSS with respect to several state-of-the-art methods in IR techniques. SSMA-LSHADE outperforms statistically APS-VSS in small datasets but the significant difference is rejected in medium datasets. Note that SSMA-LSHADE is a very complicated meta-heuristic technique, demanding an excessive amount of runtime to complete a run.

3.5 Advantages of APS-VSS

Reduction Rate: Figure 2 plots a bar chart starting with the original size in *TR*, followed by the reduction rate of SPMS-ALS, Hybrid approaches, and APS-VSS. At each algorithm, three pieces of information are written above each bar. At topmost, it is the average size of the reduced set, then the percentage of the average size in relation to the original size, and at last the percentage of reduction. On average, APS-VSS has reduced 97.06% and 98.75% on small and medium datasets, respectively, while the hybrid approaches have saved 96.97% and 98.82% the original data size on small and medium datasets, respectively.

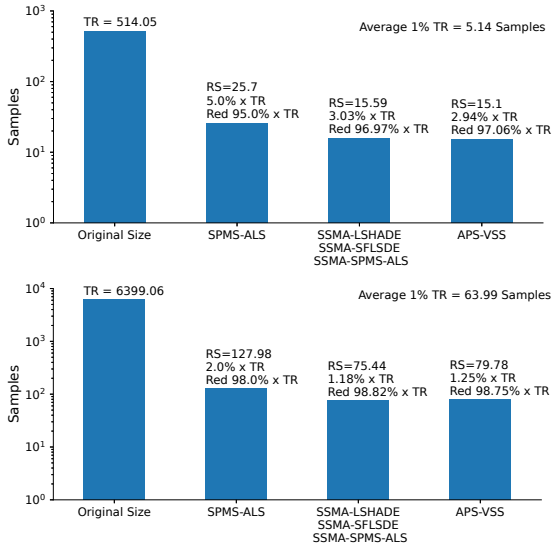

Figure 2: Reduction rate of APS-VSS against other examined algorithms over 57 datasets.

Runtime: Figure 3 shows how much faster the proposed APS-VSS is against all the comparison algorithms in small and medium datasets, respectively. Given the runtime of APS-VSS as T(s), we also display, on top of each bar, the proportion of runtime that other algorithms take in relation to T(s) in both types of datasets.

On small datasets, APS-VSS spends around 9 seconds while LSIR and SPMS-ALS are 1(s) or 2(s) lower. Other hybrid approaches substantially consume more runtime than APS-VSS. Particularly, SSMA-LSHADE uses more than 10 times, SSMA-SFLSDE takes approximate 6 times, and SSMA-SPMS-ALS is 1.38 times the runtime used in APS-VSS. The benefit of the runtime is more obviously observed in medium datasets, where any compared algorithm requires more runtime than our solution. Specifically, APS-VSS reduces 3.58 times consumed in SSMA-SPMS-ALS and 43.79 times in SSMA-LSHADE. With respect to LSIR and SPMS-ALS, the range is from 1.07 times and 3 times, respectively.

4 CONCLUSION

This paper has proposed an approach for handling simultaneously *IS* and *IG* within one algorithmic framework. The proposed algorithm is a single point memetic structure that perturbs candidate solutions in the continuous space, i.e., performing the *IG* endowed

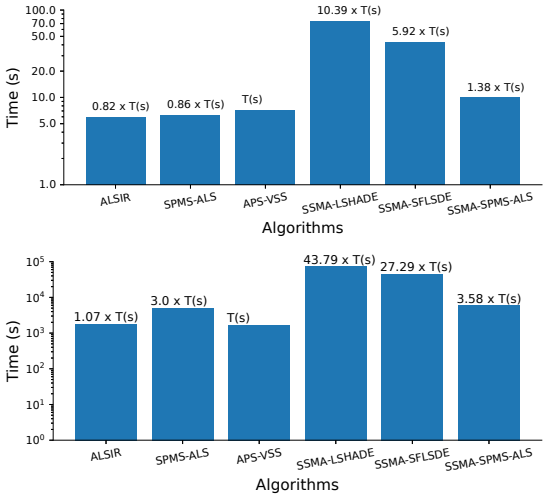

Figure 3: Average runtime (in seconds) of of APS-VSS against other examined algorithms over 57 datasets.

with two *LS* mechanisms that attempt to shorten the length of the solutions. This algorithmic approach is new in the domain-specific body of literature of *IR*.

The robustness of the proposed method with respect to performance, runtime and reduction rate is proved competitive to a few state-of-the-art hybrid methods existing in the literature, and statistically better than published algorithms using single-point search structure. The outputs of APS-VSS are valuable in the context of real-world problems when an application is required to be processed in a timely fashion. Also, these preliminary results may offer initial empirical evidence for investigating mixed continuous/combinatorial optimisation in data science.

REFERENCES

[1] J. R. Cano, F. Herrera, and M. Lozano. 2003. Using Evolutionary Algorithms as Instance Selection for Data reduction in KDD: an experimental study. *IEEE Transactions on Evolutionary Computation* 7, 6 (Dec 2003), 561–575.
[2] T. M. Cover and P. E. Hart. 1967. Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory* 13, 1 (1967), 21–27.
[3] S. García, J. R. Cano, and F. Herrera. 2008. A Memetic Algorithm for Evolutionary Prototype Selection: A Scaling up Approach. *Pattern Recognition* 41, 8 (2008), 2693–2709.
[4] S. García, J. Derrac, J.R. Cano, and F. Herrera. 2012. Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 3 (2012), 417–435.
[5] H. L. Le, F. Neri, and I. Triguero. 2021. SPMS-ALS: A Single-Point Memetic structure with accelerated local search for instance reduction. *Swarm and Evolutionary Computation* (2021), 100991.
[6] F. Neri and S. Rostami. 2021. Generalised Pattern Search Based on Covariance Matrix Diagonalisation. *SN Comput. Sci.* 2, 3 (2021), 171.
[7] F. Neri and I. Triguero. 2020. A Local Search with a Surrogate Assisted Option for Instance Reduction. In *Applications of Evolutionary Computation (Lecture Notes in Computer Science, Vol. 12104)*, Pedro A. Castillo, Juan Luis Jiménez Laredo, and Francisco Fernández de Vega (Eds.). Springer, 578–594.
[8] R. Tanabe and A. S. Fukunaga. 2014. Improving the Search Performance of SHADE Using Linear Population Size Reduction. In *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1658–1665.
[9] I. Triguero, J. Derrac, S. García, and F. Herrera. 2012. A Taxonomy and Experimental Study on Prototype Generation for Nearest Neighbor Classification. *IEEE Transactions on Systems, Man, and Cybernetics–Part C* 42, 1 (2012), 86–100.
[10] I. Triguero, S. García, and F. Herrera. 2011. Differential Evolution for Optimizing the Positioning of Prototypes in Nearest Neighbor Classification. *Pattern Recognition* 44, 4 (2011), 901–916.