# Accelerating a multi-objective memetic algorithm for feature selection using hierarchical k-means indexes

Francia Jiménez, Claudio Sanhueza, Regina Berretta, and Pablo Moscato
School of Electrical Engineering and Computing
The University of Newcastle, NSW, Australia
{francia.jimenezfuentes,claudio.sanhuezalobos,regina.berretta,pablo.moscato}@newcastle.edu.au

## ABSTRACT

The $(\alpha, \beta)$-$k$ Feature Set Problem is a mathematical model proposed for multivariate feature selection. Unfortunately, addressing this problem requires a combinatorial search in a space that grows exponentially with the number of features. In this paper, we propose a novel index-based Memetic Algorithm for the Multi-objective $(\alpha, \beta)$-$k$ Feature Set Problem. The method is able to speed-up the search during the exploration of the neighborhood on the local search procedure. We evaluate our algorithm using six well-known microarray datasets. Our results show that exploiting the natural feature hierarchies of the data can have, in practice, a significant positive impact on both the solutions' quality and the algorithm's execution time.

## CCS CONCEPTS

• **Applied computing → Multi-criterion optimization and decision -making**;

## KEYWORDS

Feature Set problem, Multi-Objective Optimization, Memetic Algorithm.

**ACM Reference Format:**
Francia Jiménez, Claudio Sanhueza, Regina Berretta, and Pablo Moscato . 2018. Accelerating a multi-objective memetic algorithm for feature selection using hierarchical k-means indexes. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3205651.3205707

## 1 INTRODUCTION

The $(\alpha, \beta)$-$k$ Feature Set Problem is a combinatorial optimization model proposed for finding a subset of features that can explain a certain dichotomy [3]. It is a generalization of the well-known $k$-Feature Set Problem which was proved to be NP-complete by Davies and Russell in 1994 [4]. The idea is to consider the effect of a subset of features over each pair of samples [3]. The $\alpha$ value represents the support of features for discriminating samples in the feature set of size $k$. In other words, the $\alpha$ value is the minimum number of features in the feature set that have different values on all pair of samples with different classes. The $\beta$ value represents the support of features for describing samples within a class. The $\beta$ value is the minimum number of features in the feature set of size $k$ with the same value on all pair of samples with the same class.

This model has been widely used in bioinformatics to select a subset of features (e.g., biomarkers) to discriminate between samples in studies investigating prostate cancer [7], Parkinson's [8] and Alzheimer's diseases [12].

The most recent application of the $(\alpha, \beta)$-$k$ Feature Set Problem was presented in [9]. We proposed for the first time a multi-objective optimization algorithm to address the problem. We implemented a memetic algorithm to solve it, and we studied the impact of different local search algorithms and initialization heuristics. We proposed two novel clustering-based heuristics to improve the searching process of features. The clustering-based heuristic improved the performance of the algorithm; however, it increases the execution time of the algorithm in comparison with other alternatives. Thus, we propose a new algorithm to speed-up the optimization process.

## 2 SPEEDING UP THE LOCAL SEARCH EXPLORATION

Search trees are data structures that can be used to search and store data [2]. In the literature, we can find several data structures of this type such as binary search trees, red-black trees, interval trees, among others. Overall, the advantage of this type of data structure is to perform search operations more efficiently. In particular, we based our strategy on a hierarchical *k-means* clustering tree [11] to speed-up the local search exploration. In addition, we employ nearest-neighbor queries to find similar features quickly. Combined, they form an indexing scheme called hierarchical *k-means* index [10].

In Algorithm 1, we present the pseudocode of our *IndexLS heuristic* (*IndexLS*). The first stage aims to obtain the $\alpha$ *node* representation of the current individual, that we called remaining solution $rs$. For each pair of samples with different target value, the method computes how many features are needed to cover it $\alpha^*$ times (i.e., *remainCoverage*). Then, we normalize the remaining solution ($rs$) by the maximum observed value. Once we have the $rs$, we search in the tree of features (i.e., KnnSearch($rs$,*maxFeat*)) for those with similar $\alpha$ *node* expressions (i.e., *similarFeatures*). We remove from the *similarFeatures* set all the features that are in the current individual. Since the *similarFeatures* set is defined, we need to generate the neighbors. We add the most similar feature until all the pair of samples with different target value satisfy the $\alpha^*$.


*GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan*

ACM ISBN 978-1-4503-5764-7/18/07.
https://doi.org/10.1145/3205651.3205707

**Algorithm 1** *IndexLS*, an index-based local search algorithm.

**Input:** current individual ($S$), alpha parameter ($\alpha^*$), maximum number features ($maxFeat$)
**Output:** improved individual ($S$)

```
rs ← ∅ //remaining solution
for all α node ∈ α nodes do
    remainCoverage ← α* − AlphaCover(α node, S)
    if remainCoverage > 0 then
        rs ← rs ∪ (α node, remainCoverage)
    end if
end for
maximum ← MaximumCoverage(rs)
rs ← Normalize(rs, maximum)
similarFeatures ← KnnSearch(rs, maxFeat) − S
while α(S) < α* & maxFeat ≠ ∅ do
    feature ← get(similarFeatures, 0)
    while feature ∈ S & pos ≤ 3 do
        feature ← get(similarFeatures, pos)
        pos ← pos + 1
    end while
    S ← S ∪ feature
    maxFeat ← maxFeat − 1
    similarFeatures ← similarFeatures − feature
    if maxFeat > 0 & similarFeatures = ∅ then
        similarFeatures ← KnnSearch(rs, maxFeat)
    end if
end while
    return S
```

## 3 EXPERIMENTS AND RESULTS

To test our *IndexLS heuristic*, we use the same memetic algorithm for the multi-objective $(\alpha, \beta)$-$k$ Feature Set Problem presented on [9]. We represent the individual as a set of selected features (i.e., a bit array of size $n$ where the position $j$ ($1 \leq j \leq n$) has the value of 1 if the feature is selected in the solution ($S$) (i.e., $f_j = 1$) and 0 otherwise. ). We use the same genetic operators as [9] *RandomInit* initialization, DetBitFlip(3) mutation, *Intersect* recombination, and *Elitist* replacement strategy. We execute 30 trials for each dataset, and the algorithm runs for a maximum of 100 generations.

In our experiments, we use six real-world microarray datasets which were previously studied in [5, 9]. In Table 1, we present the average normalized hypervolume as quality indicator [1]. Higher hypervolume value means a better performance of the algorithm. In Table 1, we present the clustering-based results presented in [9]. We applied the Wilcoxon statistical test [6] to compare our results with the clustering-based experiments presented in [9].

**Table 1: Normalized hypervolume using *IndexLS*. For each dataset, we present the average normalized hypervolume using our *IndexLS*, and the clustering-based results from [9].**

| Dataset | Previous clustering results [9] | | IndexLS heuristic results | |
|---|---|---|---|---|
| | Hypervolume | Time | Hypervolume | Time |
| DownSyn | 0.5318 | 108 | 0.4722 | 1 |
| Smoking | 0.5942 | 50,155 | 0.5993 | 13,958 |
| Bruta | 0.5451 | 2,642 | 0.5222 | 239 |
| PdParkinson | 0.4871 | 5,006 | 0.5774 | 154 |
| Prostate | 0.5852 | 45,482 | 0.6164 | 6,999 |
| Parkinson | 0.5216 | 82,332 | 0.5424 | 99,951 |

For the datasets PdParkinson, Prostate, and Parkinson, our *IndexLS heuristic* is better than the clustering-based heuristics. In fact, the average hypervolume is significantly higher (Wilcoxon statistical test with a 5% of significance, $p = 9.3\times10^{-11}$, $p = 8.2\times10^{-7}$, and $p = 7.2\times10^{-5}$). Despite the fact that Smoking dataset does not show a significant difference ($p = 0.561$), we are still confident on the usefulness of the index-based heuristics, because it reduces the time significantly. Considering the execution time, we observe that our index-based heuristics is in average 28× faster than the clustering-based heuristics. For instance, with our index-based heuristic we processed PdParkinson dataset approximately 32× faster.

## 4 CONCLUSIONS AND FUTURE WORK

We proposed a novel index-based heuristic for local search stages in a memetic algorithm. We found that our index-based heuristic have good performance when evaluated with microarray datasets.

Our proposed heuristic for local search, *IndexLS heuristic*, has a remarkably positive impact on the performance of the memetic algorithm. The execution time are significantly lower than the previous multi-objective solution presented in [9].

Although our results show that our *IndexLS heuristic* improves the performance, in the future, we will study the impact of *IndexLS* parameters; the effect of the implemented index structure; and an effective way to introduce this information during the initialization procedure.

## REFERENCES


[1] Shi Cheng, Yuhui Shi, and Quande Qin. 2012. On the performance metrics of multiobjective optimization. In *Advances in Swarm Intelligence*. Springer, 504–512.
[2] Thomas H Cormen. 2009. *Introduction to algorithms*. MIT press.
[3] Carlos Cotta, Christian Sloper, and Pablo Moscato. 2004. Evolutionary Search of Thresholds for Robust Feature Set Selection: Application to the Analysis of Microarray Data. In *Applications of Evolutionary Computing*. Lecture Notes in Computer Science, Vol. 3005. Springer Berlin Heidelberg, 21–30.
[4] Scott Davies and Stuart Russell. 1994. NP-completeness of searches for smallest possible feature sets. In *Proceedings of the 1994 AAAI fall symposium on relevance*. AAAI Press, 37–39.
[5] Mateus Rocha de Paula, Regina Berretta, and Pablo Moscato. 2016. A fast memetic heuristic approach for the ($\alpha$,$\beta$)-k-feature set problem. *Journal of Heuristics* 22, 2 (2016), 199–220.
[6] Warren J Ewens and Gregory R Grant. 2006. *Statistical methods in bioinformatics: an introduction*. Springer Science & Business Media.
[7] Martin Gomez Ravetti, Regina Berretta, and Pablo Moscato. 2009. Novel Biomarkers for Prostate Cancer Revealed by ($\alpha$,$\beta$)-k-Feature Sets. In *Foundations of Computational Intelligence Volume 5: Function Approximation and Classification*. Springer Berlin Heidelberg, 149–175.
[8] Mou'ath Hourani, Regina Berretta, Alexandre Mendes, and Pablo Moscato. 2008. Genetic signatures for a rodent model of Parkinson's disease using combinatorial optimization methods. *Bioinformatics: Structure, Function and Applications* (2008), 379–392.
[9] Francia Jiménez, Claudio Sanhueza, Regina Berretta, and Pablo Moscato. 2017. A Multi-objective Approach for the ($\alpha$, $\beta$)-k-feature Set Problem Using Memetic Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '17)*. ACM, New York, NY, USA, 207–208. https://doi.org/10.1145/3067695.3076106
[10] Marius Muja and David Lowe. 2009. Flann-fast library for approximate nearest neighbors user manual. *Computer Science Department, University of British Columbia, Vancouver, BC, Canada* (2009).
[11] David Nister and Henrik Stewenius. 2006. Scalable recognition with a vocabulary tree. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, Vol. 2. Ieee, 2161–2168.
[12] Nisha Puthiyedth, Carlos Riveros, Regina Berretta, and Pablo Moscato. 2016. Identification of differentially expressed genes through integrated study of alzheimer's disease affected brain regions. *PloS one* 11, 4 (2016), e0152342.