A Multi-objective Genetic Programming Biomarker Detection Approach in Mass Spectrometry Data

Abstract Mass spectrometry is currently the most commonly used technology in biochemical research for proteomic analysis. The main goal of proteomic profiling using mass spectrometry is the classification of samples from different clinical states. This requires the identification of proteins or peptides (biomarkers) that are expressed differentially between different clinical states. However, due to the high dimensionality of the data and the small number of samples, classification of mass spectrometry data is a challenging task. Therefore, an effective feature manipulation algorithm either through feature selection or construction is needed to enhance the classification performance and at the same time minimise the number of features. Most of the feature manipulation methods for mass spectrometry data treat this problem as a single objective task which focuses on improving the classification performance. This paper presents two new methods for biomarker detection through multi-objective feature selection and feature construction. The results show that the proposed multi-objective feature selection method can obtain better subsets of features than the single-objective algorithm and two traditional multi-objective approaches for feature selection. Moreover, the multi-objective feature construction algorithm further improves the perfomance over the multi-objective feature selection algorithm. The paper is the first multi-objective genetic programming approach for biomarker detection in mass spectrometry data.

1 Introduction

Nowadays, much attention is given to the high-throughput mass spectrometry (MS) technology in proteomics. MS enables the detection and the discrimination of patterns between diseased and healthy samples of complex mixtures of proteins [1]. MS datasets typically consist of tens or thousands of mass to charge (m/z) ratios. Each m/z value corresponds to a mass of a certain peptide and reflects the abundance of this peptide through an intensity value [2]. From machine learning perspective, each of the abundances of the peptides is a feature for classification. This causes the critical issue of *curse of dimensionality*, which leads to the degradation of classification performance due to the large number of features and the small number of examples.

Feature manipulation can help solving the biomarker detection problem [1]. It provides means to transform the representation of the input to a classification algorithm to improve its performance [3]. Feature manipulation consists of *feature selection* and *feature construction*.

While feature selectin aims at selecting a subset of relevant original features, feature construction aims at generating new high-level features. Generally, feature selection and construction methods can be divided into filter, wrapper

or embedded approaches [3]. In filter approaches, features are evaluated with some relevance measure such as, *t*-statistics [4] and mutual information [5]. Although the filter approach is efficient in terms of computational cost, most of the features selected by the filter approach are still correlated [2]. Therefore, features are mostly redundant and include some sort of data noise, which leads to the reduction of their effectiveness in terms of classification accuracy. In wrapper approaches, an inductive algorithm (mostly a classifier) is wrapped as an evaluation criterion to the selected features. Although a wrapper approach is more effective than a filter approach, its computational cost is a major obstacle to the use of this approach. Moreover, in high feature-to-sample ratio data, the wrapper approach may face the problem of overfitting [2]. The embedded approach also uses an inductive algorithm but the main difference from the wrapper approach is that the inductive algorithm is used for both feature selection and classification. Therefore, embedded approaches can overcome the disadvantages of wrapper approaches.

When two or more conflicting objectives occur and an optimal decision is needed to be taken, this results in a multi-objective problem. Multi-objective optimization is evaluated in terms of the trade-off between the conflicting objectives, which have to be minimised or maximised.

Genetic programming (GP) is an algorithm which, inspired by natural evolution, searches for good solutions in a population of programs. GP proved to be an effective technique for feature selection, feature construction and classification especially for high dimensional data [6].

Many feature selection techniques have been proposed to detect the potential biomarkers in MS data [7, 8]. Despite the great promise of the previously proposed methods, none of these methods considered the number of features as an important objective to optimise. Although some studies considered the relative importance of the number of features to classification accuracy [9, 10], the major limitation of these approaches is the prior specification of the relative importance of each objective. More related work can be seen from [11],, which are not detailed here due to the page limit.

Multi-objective optimisation offers solutions to the optimisation of different conflicting objectives.

Biomarker detection must consider the trade-off between the classification performance and the number of features. The number of features should be as small as possible to be able to pass them to experimental validation. Therefore, for evaluation of biomarker detection, two objectives should be considered, which are maximizing the classification performance and at the same time minimising the number of features. This paper represents the first attempt to use GP as a multi-objective approach to biomarker detection.

1.1 Goals

The overall goal of this paper is to develop GP-based multi-objective feature selection and construction approaches to classification of MS data. In feature selection, the proposed GP method uses the ideas of NSGAII [12] and SPEA2

[13] to evolve models that keep the balance between the conflicting objectives. We notate these methods as $NS\text{-}GPMOFS$ and $SP\text{-}GPMOFS$. The main goal here is to evolve a Pareto front of non-dominated solutions, which include a small number of selected original features and achieve a better classification accuracy than using the whole set of features.

In feature construction, a single evolved tree is used to construct multiple features to replace the original features by combining them using the GP functions. Multi-objective optimisation is used to reduce the number of constructed features while keeping the high classification accuracy. We notate these methods as $NS\text{-}GPMOFC$ and $SP\text{-}GPMOFC$.

In both approaches, an embedded approach is used to take advantages of the low computational cost and better classification accuracy.

Precisely, we will investigate the followings:

- whether using GP as a multi-objective approach to feature selection can produce better solutions than using the single objective GP algorithm,
- whether using multi-objective GP feature selection methods ($NS\text{-}GPMOFS$ and $SP\text{-}GPMOFS$) can select feature subsets that improve the classification performance and reduce the number of features than using the traditional multi-objective algorithms ($NSGAII$ and $SPEA2$), and
- whether the GP-based methods ($NS\text{-}GPMOFC$ and $SP\text{-}GPMOFC$) can further improve the performance feature subset evolved by the multi-objective GP feature selection methods ($NS\text{-}GPMOFS$ and $SP\text{-}GPMOFS$).

1.2 Organisation

The rest of the paper is organised as follows. Section 2 describes the GP-based multi-objective feature selection and the GP-based multi-objective feature construction approaches. Section 3 describes the experimental design that includes the settings and the MS datasets used. Section 4 presents the experimental results and discussions. Section 5 concludes the paper.

2 The GP Multi-Objective Approaches

This section describes the two multi-objective GP approaches.

2.1 The GP Multi-objective Feature Selection Approach

In this section, we propose a new approach to feature selection for MS data with the aim of biomarker detection using multi-objective GP, with two main objectives to explore the Pareto front of feature subsets. The objectives here are maximising the classification accuracy and minimising the number of features used in each individual of the population. As mentioned earlier, an embedded approach is taken in the proposed algorithm. GP is employed here as a classifier as well, and the number of correctly classified instances in the training set is stored in an external archive. The classification accuracy is used to assess the

first objective. The second objective here is to minimise the cardinality of the selected features (number of features selected automatically in the GP tree). When a new solution is evolved, it is compared to the other solutions stored in the archive. If the evolved solution is not worse in both objectives and it is better than a solution in the list in at least one of the objectives, it will dominate that solution. Pareto optimal contains the set of non-dominated solutions where a specific solution can not improve any of the objectives without degrading at least one of the other conflicting objectives [14]. The non-dominated solution forms the Pareto front in which no solution can be judged better than the others.

2.1.1 Pareto Fitness Assignment in $NS\text{-}GPMOFS$ and $SP\text{-}GPMOFS$

In evolutionary multi-objective optimisation, solutions are usually ranked according to their performance on the different objectives to measure the Pareto dominance. The Pareto dominance is measured through the dominance rank or dominance count [13] (or both) of a certain solution. Dominance rank of a solution is the number of solutions that dominates this solution, while the dominance count is the number of solutions that a given solution dominates. A solution with a smaller number of solutions that dominate it (lower rank) and a higher count is a better solution.

We investigate two mechanisms to measure the Pareto fitness. The first uses the dominance rank of a solution S_i for evaluating the fitness which is similar to the idea of NSGAII [12], i.e., the number of other solutions in the population that dominate S_i, and we call this method as $NS\text{-}GPMOFS$. Similar to SPEA2 [13], the second mechanism uses both dominance rank and dominance count in the Pareto refined fitness, and this method is named as $SP\text{-}GPMOFS$.

2.1.2 Crowding Distance Measure

In addition to the previously mentioned Pareto dominance measures used in the fitness, a crowding distance measure is used to generate more diversity among the population [15]. The crowding distance used is the Manhattan distance between the solutions. This distance measure is used only when two or more solutions have the same Pareto dominance measures, which means that if solutions have an equal rank, then the solution with smaller crowding distance is selected. The crowding distance is the average distance between the two solutions with each of the objectives, where a lower distance indicates a better result.

2.1.3 $NS\text{-}GPMOFS$ and $SP\text{-}GPMOFS$ Algorithms

Algorithm 1 shows the pseudocode of $GPMOFS$ algorithms. The input is D, the dataset, and the output is the Pareto front archive of solutions (PF). At each generation, the parent and offspring populations are merged. The fittest individuals (according to the two objectives) in this merged population acts as the new population ($CHILD$) in the next generation. The population is reduced to size N (original size of the population) using dominance rank and crowding distance for $NS\text{-}GPMOFS$. While for $SP\text{-}GPMOFS$ dominance rank, dominance count and

the crowding distance are measured. The size of $CHILD$ is the same as the size of the original population and it is produced using the traditional genetic operators (crossover and mutation operators). In case of $SP\text{-}GPMOFS$, the size of PF (Pareto front solutions) is kept fixed while in $NS\text{-}GPMOFS$ it does not have a specific size. Another difference between using $NS\text{-}GPMOFS$ and $SP\text{-}GPMOFS$ is the use of elitism in $SP\text{-}GPMOFS$, which is not used in $NS\text{-}GPMOFS$. The non-dominated solutions in $CHILD$ are identified and copied to PF. These steps are repeated until the maximum number of generations is reached. At the end of the evolutionary search, the solutions of PF are used to project the datasets and passed for evaluation. The evaluation is done through both classification accuracy and the number of features used in each solution in the archive.

Algorithm 1 Pseudo-Code of $NS\text{-}GPMOFS$ and $SP\text{-}GPMOFS$

Require: D, a dataset that contains a vector of instances with m original features.
Ensure: PF, a Pareto front (PF) of a set of solutions (low-level features).
begin
Divide D into training and test sets.
Initialise the population (P)
while Maximum generation is not reached **do**
Evaluate the two objectives of each individual { // Acc, $|F|$}
Select the individuals using the selection method
Generate new population ($CHILD$) using the genetic operators
if $NS\text{-}GPMOFS$ is used **then**
Non-dominated sorting of the individuals based on ranking and the crowding distance
else if $SP\text{-}GPMOFS$ **then**
evaluate the individuals based on ranking, count, and the crowding distance
end if
Copy both $CHILD$ and P to $Archive$
Identify the individuals who have non-dominated solutions in $Archive$ and add to Pareto front (PF)
Select a population of size N based upon ranking and crowding distance
Generate new population ($CHILD$) using the genetic operators
end while
Use the solutions in PF to project the test set
Calculate the test set classification accuracy of the different solutions
Calculate the number of selected features in each solution in PF
return a vector S that contain the number of features and classification accuracy of each solution in PF

2.2 The GP Multi-objective Feature Construction Approach

$GPMOFC$ constructs new high-level features from the original features (resulted features from the tree branches). In addition to the features constructed from the branches (all the internal function nodes), the final feature constructed from the root node of the tree is also used.

2.2.1 $SP\text{-}GPMOFC$ and $NS\text{-}GPMOFC$ Algorithm Algorithm 2 describes the pseudocode of $SP\text{-}GPMOFC$ and $NS\text{-}GPMOFC$. The two algorithms are similar to the feature selection algorithms ($SP\text{-}GPMOFS$ and $NS\text{-}GPMOFS$) except for the feature sets. The difference between the two algorithms for feature selection and construction is that instead of using the original features se-

lected in $SP\text{-}GPMOFS$ and $NS\text{-}GPMOFS$, the high-level features are constructed to optimise the second objectives in $SP\text{-}GPMOFC$ and $NS\text{-}GPMOFC$.

Algorithm 2 Algorithm of $NS\text{-}GPMOFC$ and $SP\text{-}GPMOFC$

Require: D, a dataset that contains a vector of instances with m original features.
Ensure: PF, A Pareto front (PF) (solutions with high-level features).
begin
Divide D into 50% for training and 50% testing.
Randomly Initialise the population (P)
Save the high-level features resulting from the branches and the root of the individual tree
while Maximum generation is not reached **do do**
Evaluate the number of constructed features and Acc of each individual
Select the individuals using the selection method
Generate new population ($CHILD$) using the genetic operators
if $NS\text{-}GPMOFC$ **then**
Non-dominated sorting of the individuals based on ranking and the crowding distance
else if $SP\text{-}GPMOFC$ **then**
Non-dominated sorting of the individuals based on ranking, count and the crowding distance
end if
Copy both $CHILD$ and P to $Archive$
Identify the individuals who have non-dominated solutions in $Archive$ and add to Pareto front (PF)
Select a population of size N based upon ranking and crowding distance
Generate new population ($CHILD$) using the genetic operators
end while
Use the solutions in PF to project test set
Calculate the test set classification accuracy of the different solutions
Calculate the number of high-level features in each solution in PF
return a vector S that contain the number of high-level features and classification accuracy of each solution in PF
end

2.3 Overview of the Two Systems

As shown in Figure 1, after preprocessing of the MS spectra datasets, the system for $GPMOFS$ or $GPMOFC$ starts by dividing the dataset into training and test sets. Each program in the population uses a subset of features in its tree terminal nodes and generates the objective value. The objective value (classification accuracy) is measured by GP individual classifier's accuracy that is passed as a objective value to measure the dominance. Dominance rank, dominance count and crowding distance are used to measure the dominance of the solutions. After the objective calculation, the objective value of each solution is compared to the Pareto front archive. If the solution in the archive is dominated by the new solution, the new solution will replace it in the archive. Each solution in the Pareto front has a subset of features that were selected in the terminal nodes. The Pareto front solutions are used to project the datasets, therefore if the size of the archive is n, there will be n projected datasets. To test the subsets of features, the test set is evaluated using GP classifier. As explained earlier, the main difference between $GPMOFS$ and $GPMOFC$ is the use of low-level and high-level features.

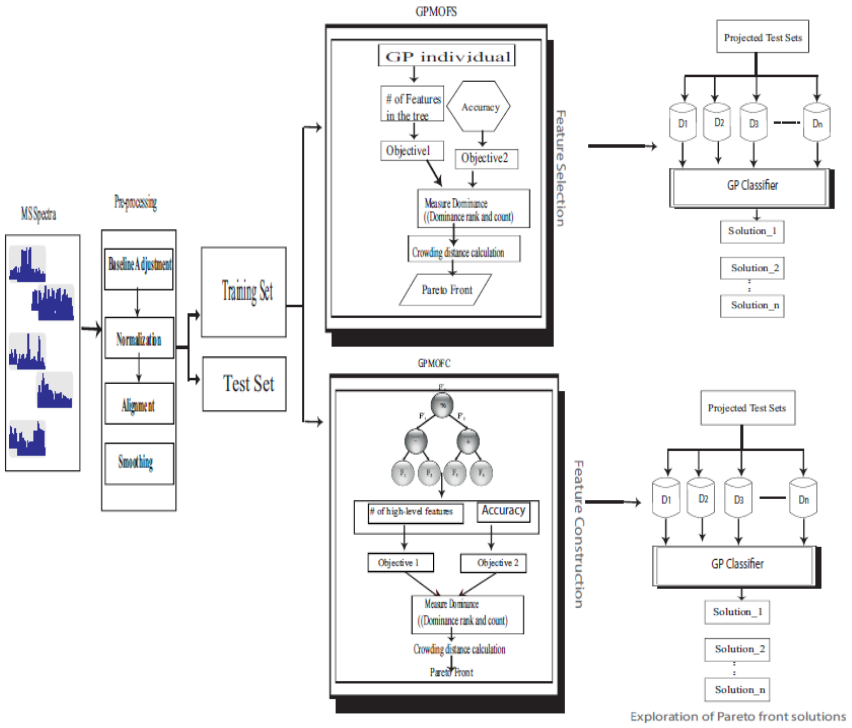

Figure 1: General overview of the multi-objective approaches

2.3.1 Objective Functions For both $GPMOFS$ and $GPMOFC$, the first objective is to maximise the classification accuracy (Acc). The second objective used is to minimise the number of features selected or constructed by each GP tree in the terminal nodes $|F|$.

Acc is defined as:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP, TN, FP and FN are the true positives, true negatives, false positives, and false negatives, respectively. For each instance of the training set, if the output of the program is less than or equal to zero then the instance is classified as class 1, otherwise it is classified as class 2.

3 Experiment Design

This section explains the MS datasets used to test $GPMOFS$ and $GPMOFC$, GP operators and parameters, benchmark algorithms used for comparison, and the evaluation criteria.

3.1 MS Datasets

To test the effectiveness of the proposed GP multi-objective approaches, eight different MS datasets are used.

- OVA1 and OVA2 [16]: OVA1 is composed of 216 spectra where 121 spectra are cancerous samples and 95 spectra are healthy ones, while OVA2 consists of 253 spectra with 162 spectrum are cancerous samples and 91 are healthy samples. The number of features is 15000 and 15154 in OVA1 and OVA 2, respectively.
- PAN [17]: The dataset has 181 spectral examples, where 80 are in the affected class and 101 are in the healthy class. The number of features in each spectrum is 6771.
- TOX [18]: The dataset consists of 62 spectra (28 in the positive and 34 in the negative class) and each spectrum has 45200 features.
- HCC [19]: HCC has 150 spectra (78 affected and 72 non-affected) with 36802 features in each spectrum.
- DGB [20]. This dataset contains three groups of samples (78 healthy, 25 hepatocellular carcinoma and 25 chronic liver diseased). The total number of features is 16075.
- Pros dataset [21]: This dataset is composed of four classes which are: Healthy (63 samples), Benign stage$_1$ (190 samples), Prostate Cancer stage$_2$ (26 samples) and Prostate Cancer stage$_3$ (43 samples). The number of features in Pros is 15000. For DGB and Pros datasets, we used only two classes of instances.
- Appleminus: This dataset is composed of 365 features with ten instances of each class. Three classes contain five predefined biomarkers, and the last class is not spiked-in. Only one of the spiked-in classes and the non-spiked class are used in our algorithms.

Several preprocessing steps were applied to each of the datasets. The preprocessing of MS data is important to convert the data to a homogeneous matrix which can be used for feature selection and classification of the data [20]. The preprocessing steps used in our experiments include baseline adjustment, spectrum normalisation, alignment and filtering with different parameters for each dataset. The baseline removal is used to remove the low-range noise. The baseline is estimated by passing a window on the spectra and the minimum m/z values are calculated. A piecewise linear interpolation method is used for the regression of the baseline. To make the intensity values range the same, normalisation is performed. The normalisation of the spectra is done by calculating the area under the curve [18] and rescaling the spectra to have a maximum intensity value of 300. This is done by using the $msnorm$ function in the Matlab toolbox [22]. After normalisation is performed, alignment of the peaks is performed to match the similar peaks across all the spectra. Finally, smoothing of the spectra is done to remove the low signal fluctuation. Smoothing is done via a Savitzky-Golay filter. Pros and TOX datasets were already baseline adjusted. Therefore, both of the datasets were only filtered and normalised. Table 1 shows the running parameters of the preprocessing steps used with each of the datasets. The parameters are selected based on the original papers of the datasets [17–20]. The spike-in Appleminus dataset is available in NetCDF format, and it is preprocessed using XCMS [23] with the settings described in [24].

Table 1: Preprocessing parameters

	OVA1 & OVA2	TOX	PAN	HCC	DGB	Pros
Window size for baseline removal	500	-	200	50	200	-
Smoothing frame size	5	6	3	6	6	3
Maximum intensity after normalisation	300					

3.2 Performance Evaluation

GP as a classifier is used to test the selected features in each solution in the archive on the test sets.The performance is evaluated according to both the classification accuracy of the test set and the number of features.

3.3 Terminal Set, Function Set, Genetic Operators and Parameters

In the experiments, we adopt standard tree-based GP, which produces a single floating point number [25] for each instance in the dataset. Each of the output values is then used to determine the classification accuracy of the genetic program. The initial population is generated using the ramped-half-and-half method [26]. The function set consists of the four standard elementary mathematical operators $\{+, -, \%, \times\}$ and also a square root $\sqrt{}$ operator. The $\%$ and $\sqrt{}$ are "protected" where $\%$ returns zero for division by zero and $\sqrt{}$ returns zero for negative numbers. The terminal set has only variable terminals that are the feature values. The population size is set to 1024. Crossover and mutation probabilities are 0.8, and 0.2, respectively, and tournament selection is used with the size of 7. The GP, NSGAII and SPEA2 implementations used in the experiments are based on the Evolutionary Computing Java-based (ECJ) package [27]. Other parameters for NSGAII and SPEA2 are set as the default values in the ECJ library. The evolution terminates at a maximum number of 20 generations.

For each dataset, the experiment is repeated for 30 independent runs with 30 different random seeds. Each run outputs a set of non-dominated solutions in the Pareto front. The 30 sets of non-dominated solutions from the 30 runs are combined to one set by removing the dominated solutions from the different sets.

3.4 Benchmark Algorithms

$GPMOFS$ and $GPMOFC$ are compared to the following benchmark algorithms:

1. Standard (Single-Objective) GP is the standard GP classification framework using the overall classification accuracy as a single objective to maximise. The features selected in the terminal nodes of the tree are treated as the selected features.
2. NSGAII: Multi-objective optimisation using NSGAII and Fisher criterion based class separability for feature selection [28]. The evaluation is done

through both the higher Fisher criterion and the smaller number of features. The first objective which is maximising the Fisher criterion or the class separability, that is defined as,

$$\text{Fitness function = Fisher criterion} = \sum_{n=1}^{N} |\frac{\mu_i - \mu_j}{\sigma_i^2 - \sigma_j^2}|$$

where μ_i and μ_j are the means, σ_i^2 and σ_j^2 are the variances of the samples which belong to class i and class j, respectively. N is the number of samples in the training set. The second objective is minimising the number of features.

3. SPEA2: Multi-objective optimisation using SPEA2 where the first objective is to maximise Fisher criterion and the second objective is to minimise the number of features.

Similar to $GPMOFS$ and $GPMOFC$, the population size is set to 1024 and the number of generations is 20. For both NSGAII and SPEA2, each individual is encoded as a binary vector. The length of the vector is equal to the total number of features in the dataset. Hence, if the bit is 1, this means that the feature is selected and if the bit is 0 the feature is not selected.

4 Results and Discussions

Figure 2 shows the results of $GPMOFS$ compared to using the single objective GP method, and the SPEA2 and NSGAII, while Figure 3 shows the results of $GPMOFC$ compared to $GPMOFS$. The multi-objective methods have different numbers of non-dominated solutions. The results are the non-dominated solutions obtained from the 30 independent runs. The x-axis refers to the number of features selected by each method, whereas the y-axis indicates the classification accuracy. Each figure is divided into a number of sub-figures where each sub-figure represents the results of each dataset.

4.1 Performance of $GPMOFS$

It can be noticed from Figure 2 that using SP-$GPMOFS$ has the potential to evolve solutions, which have better classification performance and a smaller number of features than using NS-$GPMOFS$ in seven out of the eight datasets. The proposed method also outperformed the single objective GP approach and the two benchmark multi-objective methods SPEA2 and NSGAII, on all the eight datasets. This supports our hypothesis that using multi-objective GP can improve the feature selection performance from both the classification accuracy and the number of features points of view.

In some cases, NS-$GPMOFS$ and SP-$GPMOFS$ have common solutions such as in the TOX, and HCC datasets during the left region of the front. Only in the TOX dataset, NS-$GPMOFS$ evolves solutions at the right region of the frontier which have better accuracy, but the number of features in these solutions are larger. In the Appleminus dataset, NS-$GPMOFS$ is the best followed by SP-$GPMOFS$. The single-objective GP method for the Appleminus dataset

has evolved solutions with a large number of features and lower accuracy compared to the multi-objective approaches.

The multi-objective approaches SPEA2 and NSGAII for feature selection are both used with Fisher criterion for comparison to the proposed method. Comparing $NS\text{-}GPMOFS$ and $SP\text{-}GPMOFS$ with SPEA2 and NSGAII, it is clear that GP has improved the performance of both NSGAII and SPEA2 for feature selection. This can be explained by the GP capability to select the subsets of features that are more relevant to classification. Using multi-objective optimisation along with GP improves both objectives of reducing the number of features and having a better performance. This suggests that GP improves the capability of the multi-objective approaches through its ability to select the better subsets of features.

4.2 Comparison of $GPMOFS$ and $GPMOFC$

Considering the experimental results of $GPMOFC$ that are shown in Figure 3, it can be noticed that the multi-objective feature construction is better than the multi-objective feature selection in most cases. For OVA1, $SP\text{-}GPMOFC$ is the best with a smaller number of features. For PAN and HCC, feature construction approaches evolve better solutions than the feature selection algorithms. In dataset OVA2, $SP\text{-}GPMOFC$ is equivalent to $SP\text{-}GPMOFS$ and it outperforms $NS\text{-}GPMOFS$.

The results suggest that multi-objective feature construction tends to achieve the balance between reducing the dimensionality and improving the performance better than multi-objective feature selection. This supports our first hypothesis that feature construction can further improve the multi-objective feature manipulation performance through the construction of high-level features that identify the interactions and relations between the original low-level features.

The exceptions to the observaton mentioned above that multi-objective feature construction can achieve better results than the multi-objective feature selection on these datasets are the TOX and Pros datasets. For Appleminus dataset, $NS\text{-}GPMOFS$ has the best set of solutions and the two feature construction methods come next. For these two datasets, $GPMOFS$ is better than $GPMOFC$. $GPMOFC$ tries to reduce the number of constructed features and decreases the dimensionality better than $GPMOFS$ in these two datasets, but this came on the account of the classification performance. However, the gap between the selection and construction is very small. Both selection and construction can achieve 100% accuracy with a number of features of 10-12 for feature selection and 12-16 for feature construction, from over 15,000 features in Pros and 45,000 in TOX.

4.3 Comparison of $GPMOFC$ to single objective GP, SPEA2 and NSGAII approaches

Comparing Figure 2 and Figure 3, $GPMOFC$ is outperforming both SPEA2 and NSGAII in all the cases. If the results of $GPMOFC$ and the single objective

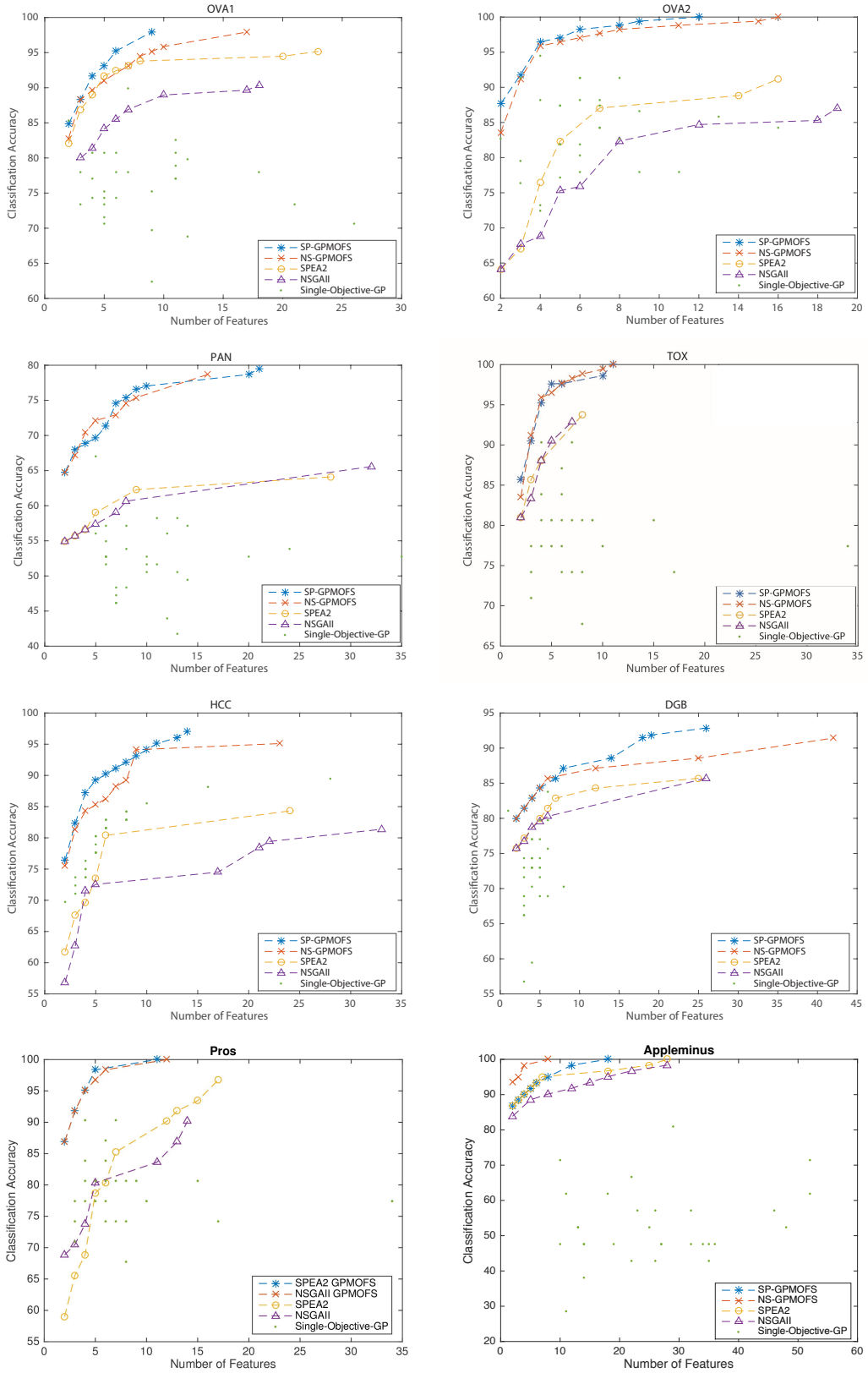

Figure 2: Experimental Results for $GPMOFS$

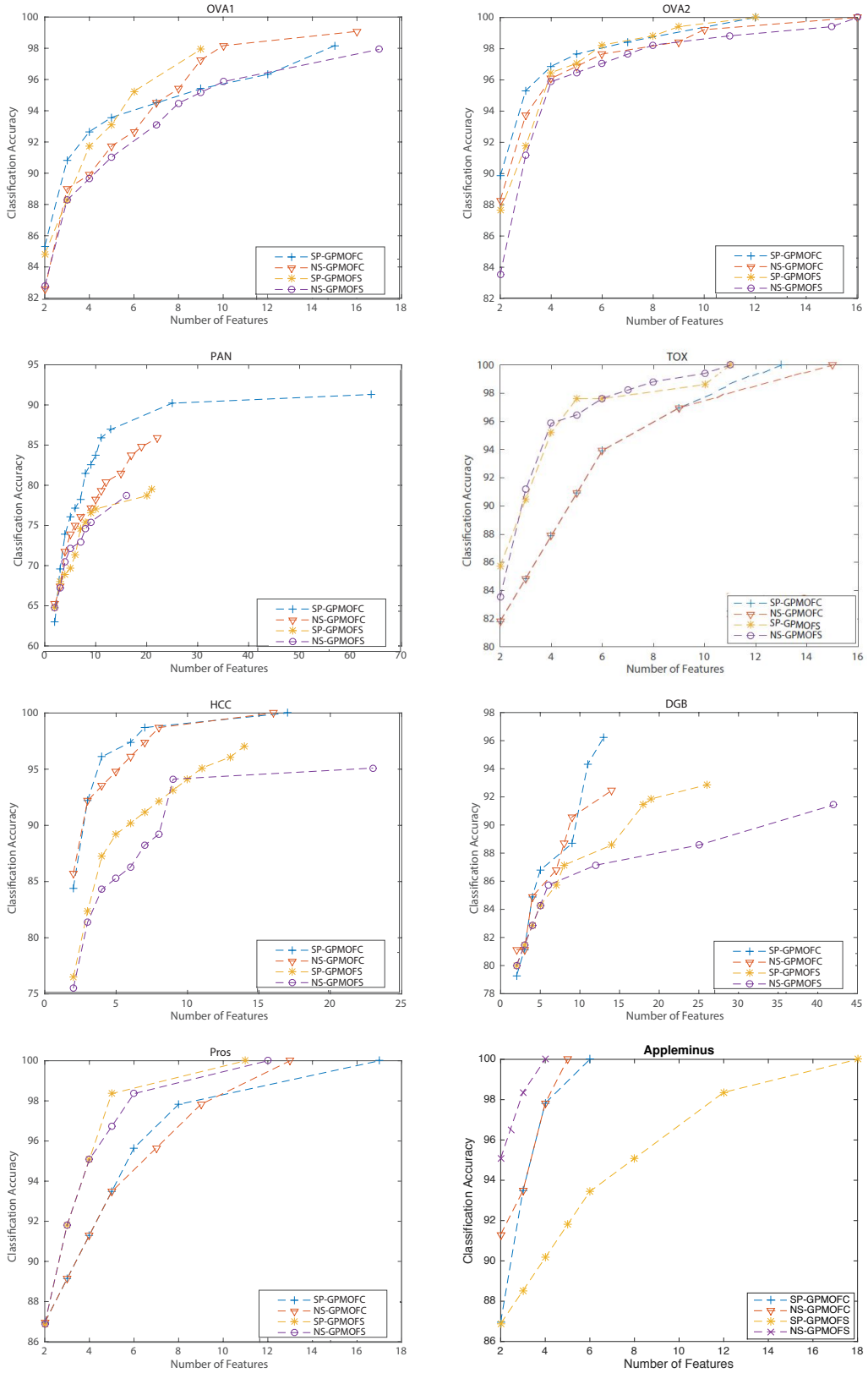

Figure 3: Experimental Results for $GPMOFC$

GP are compared, it is also clear the multi-objective construction is better on all the tasks.

This indicates the increased effectiveness of using the high-level features over the selected original features, and gives more credibility to GP as a feature construction approach.

4.4 Biomarker Detection

We tested the features selected from the Appleminus dataset to check the number of detected predefined biomarkers by each method. Table 2 shows the selection status of the biomarker by each of the multi-objective feature selection methods. It can be noticed from Table 2 that $SP\text{-}GPMOFS$ has outperformed the other three methods and managed to detect the five predefined biomarkers. SPEA2 detected four predefined biomarkers while both $NS\text{-}GPMOFS$ and $NSGAII$ detected three out of the five predefined biomarkers. This suggests that $SP\text{-}GPMOFS$ has better performance in terms of biomarker detection as well as higher accuracy solutions with a smaller number of features

Table 2: Identified spike-in biomarkers by $SP\text{-}GPMOFS$, $NS\text{-}GPMOFS$, $SPEA2$ and $NSGAII$

m/z value	$SP\text{-}GPMOFS$	$NS\text{-}GPMOFS$	$SPEA2$	$NSGAII$
(5 Biomarkers) 463.0	✓	✓	✓	✗
447.09	✓	✓	✓	✓
273.03	✓	✓	✓	✓
435.13	✓	✗	✗	✗
227.07	✓	✗	✓	✓

5 Conclusions

This paper proposes the first multi-objective biomarker detection approach for MS data. Moreover, the paper also presents the first multi-objective feature construction algorithm that is applied to MS data.

In Section 2 of the paper, $GPMOFS$, the proposed GP multi-objective feature selection method manages the trade-off between the classification accuracy and the cardinality of features. According to the results, $GPMOFS$ evolves non-dominated solutions, which has the potential to solve the problem of high dimensionality and a small number of examples in MS data. The method outperforms the single-objective feature selection GP method in terms of both objectives. The method uses the embedded capability of GP to select features with the dominance rank, dominance count and crowding distance to evaluate the solutions. The proposed method also outperforms both SPEA2 and NSGAII multi-objective feature selection approaches using Fisher criterion.

The second part of the paper presents $GPMOFC$, the first multi-objective feature construction method on MS data. For the construction of multiple high-level features, the features generated from the branches of the evolved GP tree

in addition to the root features are used. This generates a number of new high-level features, which has the potential to improve the classification performance. To reduce the dimensionality by generating a smaller number of features, $GPMOFC$ uses ideas from SPEA2 and NSGAII to keep the trade-off between the number of features and the classification performance. The results show that $GPMOFC$ outperformed $GPMOFS$ in almost all the cases, and, it was also better than SPEA2 and NSGAII approaches and the single objective GP feature selection method.

References

1. Morris, J.S., Coombes, K.R., Koomen, J., Baggerly, K.A., Kobayashi, R.: Feature extraction and quantification for mass spectrometry in biomedical applications using the mean spectrum. Bioinformatics **21**(9) (2005) 1764–1775
2. Yang, P., Zhang, Z.: A Clustering Based Hybrid System for Mass Spectrometry Data Analysis. In Chetty, M., Ngom, A., Ahmad, S., eds.: Pattern Recognition in Bioinformatics. Volume 5265 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2010) 98–109
3. Liu, H., Motoda, H.: Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publishers, Norwell, MA, USA (1998)
4. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., Lander, E.S.: Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Science **286**(5439) (1999) 531–537
5. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. Pattern Analysis and Machine Intelligence, IEEE Transactions on **27**(8) (2005) 1226–1238
6. Neshatian, K., Zhang, M.: Unsupervised Elimination of Redundant Features Using Genetic Programming. In: Australasian Conference on Artificial Intelligence. (2009) 432–442
7. Gertheiss, J., Tutz, G.: Supervised feature selection in mass spectrometry-based proteomic profiling by blockwise boosting. Bioinformatics **25**(8) (2009) 1076–1077
8. Somnath, D.: Classification of Breast Cancer versus Normal Samples from Mass Spectrometry Profiles Using Linear Discriminant Analysis of Important Features Selected by Random Forest. Statistical Applications in Genetics and Molecular Biology **7**(2) (2008) 1–14
9. Muni, D., Pal, N., Das, J.: Genetic programming for simultaneous feature selection and classifier design. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **36**(1) (2006) 106–117
10. Ahmed, S., Zhang, M., Peng, L.: Improving feature ranking for biomarker discovery in proteomics mass spectrometry data using genetic programming. Connection Science (doi:10.1080/09540091.2014.906388) (2014) 1–29
11. Kourid, A., Batouche, M.: Biomarker Discovery Based on Large-Scale Feature Selection and MapReduce. In: Computer Science and Its Applications, (2015). (Volume 456.) 81–92
12. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast Elitist Multi-Objective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6** (2000) 182–197
13. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In: Evolutionary Methods for Design, Optimisation, and Control, CIMNE, Barcelona, Spain (2002) 95–100

14. Ngatchou, P., Zarei, A., El-Sharkawi, M.: Pareto Multi Objective Optimization. In: Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems. (2005) 84–91
15. Bhowan, U., Johnston, M., Zhang, M., Yao, X.: Evolving Diverse Ensembles Using Genetic Programming for Classification With Unbalanced Data. IEEE Trans. Evolutionary Computation **17**(3) (2013) 368–386
16. Petricoin, Ardekani, A.M., Hitt, B.A., Levine, P.J., Fusaro, V.A., Steinberg, S.M., Mills, G.B., Simone, C., Fishman, D.A., Kohn, E.C., Liotta, L.A.: Use of proteomic patterns in serum to identify ovarian cancer. The Lancet **359** (2002) 572–577
17. Hingorani, S.R., III, E.F.P., Maitra, A., Rajapakse, V., King, C., Jacobetz, M.A., Ross, S., Conrads, T.P., Veenstra, T.D., Hitt, B.A., Kawaguchi, Y., Johann, D., Liotta, L.A., Crawford, H.C., Putt, M.E., Jacks, T., Wright, C.V., Hruban, R.H., Lowy, A.M., Tuveson, D.A.: Preinvasive and invasive ductal pancreatic cancer and its early detection in the mouse. Cancer Cell **4**(6) (2003) 437–450
18. Petricoin, E.F., Rajapaske, V., Herman, E.H., Arekani, A.M., Ross, S., Johann, D., Knapton, A., Zhang, J., Hitt, B.A., Conrads, T.P., Veenstra, T.D., Liotta, L.A., Sistare, F.D.: Toxicoproteomics: Serum Proteomic Pattern Diagnostics for Early Detection of Drug Induced Cardiac Toxicities and Cardioprotection. Toxicologic Pathology (2004) 122–130
19. Ressom, H., Varghese, R.S., Orvisky, E., Drake, S., Hortin, G., Abdel-Hamid, M., Loffredo, C.A., Goldman, R.: Ant Colony Optimization for Biomarker Identification from MALDI-TOF Mass Spectra. In: Proceedings ofthe 28th IEEE Annual International Conference in Engineering in Medicine and Biology Society. (2006) 4560–4563
20. Armañanzas, R., Saeys, Y., Inza, I., García-Torres, M., Bielza, C., van de Peer, Y., Larranaga, P.: Peakbin Selection in Mass Spectrometry Data Using a Consensus Approach with Estimation of Distribution Algorithms. IEEE/ACM Transactions on Computational Biology and Bioinformatics **8**(3) (2011) 760–774
21. Petricoin, E.F., Ornstein, D.K., Paweletz, C.P., Ardekani, A., Hackett, P.S., Hitt, B.A., Velassco, A., Trucco, C., Wiegand, L., Wood, K., Simone, C.B., Levine, P.J., Linehan, W.M., Emmert-Buck, M.R., Steinberg, S.M., Kohn, E.C., Liotta, L.A.: Serum Proteomic Patterns for Detection of Prostate Cancer. Journal of National Cancer Institute **94**(20) (2002) 1576–1578
22. MATLAB: version 7.10.0 (R2010a). The MathWorks Inc., Natick, Massachusetts (2010)
23. Smith, C., Want, E., O'Maille, G., Abagyan, R., Siuzdak, G.: XCMS: Processing mass spectrometry data for metabolite profiling using Nonlinear peak alignment, matching, and identification. Analytical Chemistry (2006) 779–787
24. Datta, S.: Feature Selection and Machine Learning with Mass Spectrometry Data. In Matthiesen, R., ed.: Mass Spectrometry Data Analysis in Proteomics. Volume 1007 of Methods in Molecular Biology. Humana Press (2013) 237–262
25. Koza, J.: Genetic Programming III: Darwinian Invention and Problem Solving : John R.Koza...[et Al.]. A Bradford book. Elsevier Science & Tech (1999)
26. Neshatian, K., Zhang, M., Johnston, M.: Feature Construction and Dimension Reduction Using Genetic Programming. In: Australian Conference on Artificial Intelligence. (2007) 160–170
27. Luke, S.: Essentials of Metaheuristics. second edn. Lulu (2013) http://cs.gmu.edu/∼sean/book/metaheuristics/.
28. Soyel, H., Tekguc, U., Demirel, H.: Application of NSGA-II to feature selection for facial expression recognition. Computers & Electrical Engineering **37**(6) (2011) 1232–1240