Evolutionary Computing (2023)

Assignment (4):

Feature selection with an adaptive multi-objective EA solution

Shiraz University

Due date: 05/Jan (2024)

1 Introduction

Feature selection is an important preprocessing step in machine learning and data mining applications. The goal of feature selection is to identify and remove irrelevant or redundant features from a dataset, resulting in a more compact and informative subset of features. This can improve the performance of machine learning models in several ways - by reducing overfitting, improving interpretability, and decreasing training time.

The paper by Xue et al. presents a multi-objective binary genetic algorithm with adaptive operator selection (MOBGA-AOS) for feature selection. Genetic algorithms are well-suited for feature selection problems due to their population-based search and ability to find good approximations of the Pareto-optimal set. The key contribution of this work is the integration of an adaptive mechanism for selecting crossover operators during the search process.

2 Problem Statement

Feature selection involves multiple competing objectives - minimizing the number of selected features while maximizing classification accuracy. Multi-objective evolutionary algorithms like NSGA-II have been applied, but performance depends heavily on the search operators used. Different crossover operators have different search properties and there is no single best operator for all problems. This problem is about how to develop an adaptive EA that can automatically select good crossover operators during evolution to effectively solve multi-objective feature selection across diverse datasets.

MOBGA-AOS is a multi-objective optimization algorithm for feature selection in classification problems. It is based on NSGA-II but integrates adaptive operator selection to enhance the search. It uses a binary representation and pool of crossover operators. The key mechanism is adaptively selecting operators during evolution based on their past performance at generating non-dominated solutions. Operators that produce better offspring get higher selection probability. This allows dynamic selection of the most effective operators.

In summary, MOBGA-AOS follows the main steps of NSGA-II but integrates adaptive operator selection based on rewards for producing non-dominated offspring. The key aspect is updating the operator selection probabilities every LP generation based on their accumulated rewards.

2.1 Representation

Each individual is represented as a binary array with length equal to the number of features in the dataset. 1 indicates the feature is selected, 0 not selected.

As an example: “0101100”, means that the number of the original features is seven, and the second feature, the fourth feature and the fifth feature is selected.

2.2 Fitness Evaluation

Objective 1: Classification error achieved by evaluating solutions using k-NN (k = 3) with n fold cross validation (n = 3). Must be Minimized.

Objective 2: Number of selected features. Must be Minimized.

2.3 Operators

2.3.1 Crossovers

Uses a pool of 5 crossover operators: single-point, two-point, uniform, shuffle, reduced surrogate. Select crossover operator via roulette wheel based on OSP.

- **Single-point crossover**: Randomly select a crossover point. Swap genes beyond that point between two parents to create two offspring.
- **Two-point crossover**: Select two random crossover points. Swap genes between the points between parents to generate offspring.
- **Uniform crossover**: For each gene, randomly select from either parent with equal probability to create offspring.
- **Shuffle crossover**: Randomly shuffle genes of parents, then apply single-point crossover on shuffled parents.
- **Reduced surrogate crossover**: Only allow crossover points where parents have different genes. Randomly select one such point and apply single-point crossover.

2.3.2 Mutation

Uniform mutation with rate of P_m.

2.3.3 Environmental Selection

Use NSGA-II's original fast nondominated sorting and crowding distance to select survivors.

2.4 Credit assignment & Updating of OSP

To find out about the process of assigning credits and updating OSP (Operator Selection Probability vector), read section 3.5 and section 3.6 in the original paper.

Find more details about MOBGA-AOS in the original paper. (Download link)

3 Tasks

1. Implement MOBGA-AOS and apply it to the 6 provided datasets individually. To validate your results, you can compare your final obtained pareto front with those presented in the original paper (Fig 1). They should be at least relatively close.

2. Plot the obtained pareto front of your implementation.

3. Suggest a solution to converge to optimal solutions more quickly for datasets with a larger number of features (more than 250 features). Justify your solution.

4. Plot the hypervolume value for each generation.

5. Run your algorithm multiple times (minimum of 5 run) and report the average and standard deviation of the final values for IGD[1] and HV (hypervolume) for each run. Report these values in form of a table in your documentation file.

Datasets download link

Please complete the assigned tasks using each dataset provided individually.

You can use pre-defined libraries to compute IGD and HV values.

Notes:

- The only programming language allowed for this assignment is Python.
- Any sign of cheating would result in a zero grade for this assignment.
- You should upload your submissions at:
  https://quera.org/course/add_to_course/course/14736/
  All of the files should be in a ZIP file named in this format: "Lastname-SudentNumber.zip"
  Ex: "Zamani-4023040.zip"
- Your reports should be in a PDF file including: key points of your implementation, reports of your final results and answers of assignment questions (if given).

---

[1] Inverted Generational Distance