Design Choices

Introduction

A monotone submodular optimisation function is a function that exhibits diminishing returns and non-decreasing fitness values. These qualities make greedy methods highly effective for single-objective cases. However, when attempting to solve these submodular optimisation functions with a multi-objective population-based solution, the goal shifts to approximating a diverse Pareto front of optimal solutions that provide a trade off for competing objectives.

Algorithm Overview

Each individual solution respected a uniform constraint, ensuring that all subsets selected were of fixed size, maintaining comparability across the population. The population-based evolutionary algorithm utilised to address the multi-objective nature of the optimisation function will combine several key concepts, the most basic of which being the following:

- Crossover and Mutation operations
- Randomly generated unique individuals

To maintain and ensure diversity amongst the population, the following key concepts will be employed:

- Island model population structure
- Pareto-based tournament selection
- Periodic migration between islands
- Crowding distance-based diversity control

Design Choices

1. **Diversity Maintenance**

A key challenge in multi-objective optimisation is maintaining diversity within the population such that it does not prematurely converge to a local optima. To address this concern, an island model was implemented. The specific island model utilised within this evolutionary algorithm was that of two separate subpopulations which were capable of independent evolution. Every set number of generations a random individual from both subpopulations would be exchanged to encourage exploration of new solutions and prevent premature convergence.

In addition to this, a crowding distance mechanism was applied to maintain diversity and prevent individuals with similar objective values from dominating the population.

2. **Selection Strategy**

To ensure the selection of individuals to undergo crossover operation was diverse but not always the global best solution, Pareto tournament selection was utilised. This form of tournament selection would favour an individual that dominated all other individuals within the tournament, and if no individual was found to show dominance random selection would occur.

3. **Variation Operators**

A crossover and mutation operator would be utilised to enable the evolutionary process of recombination between individuals. Each child would be produced through the crossover between two parent solutions, with these children subject to a random bit-flip mutation.

4. **Pareto Front Extraction**

After the final generation has concluded, the two island subpopulations are merged into one, with unique non-dominated solutions forming the Pareto front. This Pareto front is sorted and printed for visualisation within the terminal.

Design Process and Iteration

The final algorithm design was the result of a culmination of problems and solutions, birthing an effective evolutionary algorithm to solve multi-objective optimisation problems:

- **Baseline implementation:** A single-objective EA was implemented and utilised as the foundation for the multi-objective EA. This involved the use of crossover and mutation operators.
- **Island model:** Dividing the population into two separate subpopulations was an effective method to improve diversity and performance across problem instances.
- **Crowding distance:** The initial Pareto front that was produced would converge to a single value, making the Pareto front largely non-existent. Introducing crowding distance prevented clustering of similar solutions, allowing for the maintenance of diversity across the Pareto front.
- **Migration:** Migration between the two islands was found to provide the most benefit when scheduled for every 25 generations.

Conclusion

The final algorithm effectively balances convergence and diversity throughout execution. Across the tested population sizes, larger populations were able to produce a more diverse and complete Pareto front, while smaller populations converged faster and produced notably smaller Pareto fronts.