Evolutionary multi-objective algorithm design issues

Karthik Sindhya, PhD
Postdoctoral Researcher
Industrial Optimization Group
Department of Mathematical Information Technology
Karthik.sindhya@jyu.fi
http://users.jyu.fi/~kasindhya/
Objectives

The objectives of this lecture are to:

• Address the design issues of evolutionary multi-objective optimization algorithms
  – Fitness assignment
  – Diversity preservation
  – Elitism

• Explore ways to handle Constraints
References


Algorithm design issues

• The approximation of the Pareto front is itself multi-objective.
  – Convergence: Compute solutions as close as possible to Pareto front quickly.
  – Diversity: Maximize the diversity of the Pareto solutions.

• It is impossible to describe
  – What a good approximation can be for a Pareto optimal front.
  – Proximity to the Pareto optimal front.
• Unlike single objective, multiple objectives exists.
  – Fitness assignment and selection go hand in hand.
• Fitness assignment can be classified in to following categories:
  – Scalarization based
    • E.g. Weighted sum, MOEA/D
  – Objective based
    • VEGA
  – Dominance based
    • NSGA-II
Scalarization based (Aggregation based):

- Aggregate the objective functions to form a single objective.
- Vary the parameters in the single objective function to generate multiple Pareto optimal solutions.

Fitness assignment

$$f_1(x), f_2(x), ..., f_k(x)$$

$$w_1f_1(x) + w_2f_2(x), ..., w_kf_k$$

Or, $$\max(w_i(f_i - z_i))$$
• Advantages – Weighted sum
  – Easy to understand and implement.
  – Fitness assignment is computationally efficient.
  – If time available is short can be used to quickly provide a Pareto optimal solution.

• Disadvantages - Weighted sum
  – Non-convex Pareto optimal fronts cannot be handled.
• Objective based
  – Switch between objectives in the selection phase.
    • Every time an individual is chosen for reproduction, a different objective decides.
  – E.g. Vector evaluated genetic algorithm (VEGA) proposed by David Schaffer.
    • First implementation of an evolutionary multi-objective optimization algorithm.
    • Subpopulations are created and each subpopulation is evaluated with a different objective.
Advantages

– Simple idea and easy to implement.
– Simple single objective genetic algorithm can be easily extended to handle multi-objective optimization problems.
– Has tendency to produce solutions near the individual best for every objective.
  • An advantage when this property is desirable.

Disadvantages

– Each solution is evaluated only with respect to one objective.
  • In multi-objective optimization algorithm all solutions are important.
– Individuals may be stuck at local optima of individual objectives.
Fitness assignment

• Dominance based
  – Pareto dominance based fitness ranking proposed by Goldberg in 1989.

• Different ways
  – **Dominance rank**: Number of individuals by which an individual is dominated.
    • E.g. MOGA, SPEA2
  – **Dominance depth**: The fitness is based on the front an individual belongs.
    • NSGA-II
  – **Dominance count**: Number of individuals dominated by an individual.
    • SPEA2
Fitness assignment

Dominance count

Dominance rank

Dominance depth
Diversity preservation

• Chance of an individual being selected
  – **Increases**: Low number of solutions in its neighborhood.
  – **Decreases**: High number of solutions in its neighborhood.

• There are at least three types:
  – Kernel methods
  – Nearest neighbor
  – Histogram
- Kernel methods:
  - Sum of $f$ values, where $f$ is a function of distance.
  - E.g. NSGA

- Nearest neighbor
  - The perimeter of the cuboid formed by the nearest neighbors as the vertices.
  - E.g. NSGA-II
Diversity preservation

• Histogram
  – Number of elements in a hyperbox.
  – E.g. PAES
Elitism

• Elitism is needed to preserve the promising solutions

No archive strategy

Old population - Offspring - New population

Old population - Offspring - Archive

New Archive
• **Penalty function approach**
  
  – For every solution, calculate the overall constraint violation, OCV (sum of Constraint violations).
  
  – \( F_m(x_i) = f_m(x_i) + OCV \)

  • Solution - \((x_i), f_m(x_i)\) - \(m^{th}\) objective value for \(x_i\), \(F_m(x_i)\) — Overall \(m^{th}\) objective value for \(x_i\).

  • OCV is added to each of the objective function values.

• **Use constraints as additional objectives**
  
  – Usually used when feasible search space is very narrow.
Deb’s constraint domination strategy

- A solution \( x_i \) \textit{constraint dominates} a solution \( x_j \), if any is true:
  - \( x_i \) is feasible and \( x_j \) is not.
  - \( x_i \) and \( x_j \) are both infeasible, but \( x_i \) has a smaller constraint violation.
  - \( x_i \) and \( x_j \) are feasible and \( x_i \) dominates \( x_j \).

- Advantages:
  - Penalty less approach.
  - Easy to implement and clearly distinguishes good from bad solutions.
  - Can handle, if population has only infeasible solutions.

- Disadvantages:
  - Problem to maintain diversity of solutions.
  - Slightly infeasible and near optimal solutions are preferred over feasible solutions far from optima.