Nature-inspired Algorithms

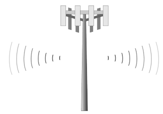
Lecture 7

Algorithm Engineering Group
Hasso Plattner Institute, University of Potsdam

3 July 2017

What happens when a problem has more than one objective?

Many problems have several **conflicting** objectives!



data rate, latency, energy efficiency

cost, environmental impact

value, risk

- often simplify by treating them as one (aggregation)
- remaining objectives treated as constraints (ϵ-constraint method)

Multiobjective optimization

Let $\mathcal{X}$ denote the domain. We move from optimizing

$$f\colon \mathcal{X} \to \mathbb{R}$$

to

$$f\colon \mathcal{X} \to \mathbb{R}^k := x \mapsto (f_1(x), f_2(x), \dots f_k(x))$$

Now, notion of **optimum** changes!
Trying to find **compromises** or **trade-offs** rather than a single solution

Optimum (single-objective)

$$\{x \in \mathcal{X} : f(x) \geq f(y) \forall y \in \mathcal{X}\}$$

Optimum (multi-objective)

What if the best points for f_1 are the worst points for f_2?

Need a new definition of optimum

Dominance

Let $x, y \in \mathcal{X}$. We say x *dominates* y (written as $x \preceq y$) if and only if

1. $f_i(x) \leq f_i(y) \, \forall i \in \{1, \dots, k\}$
2. $\exists j \in \{1, \dots, k\} \; f_j(x) < f_j(y)$

So $x \preceq y$ when it is not worse in any objective, and is better in at least one

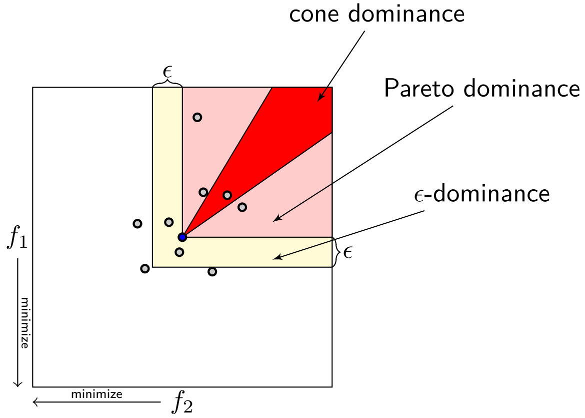
Pareto optimality

A solution $x \in \mathcal{X}$ is *Pareto optimal* if there does not exist another $y \in \mathcal{X}$ such that $y \preceq x$.

There is no solution that would decrease one criterion without *simultaneously increasing* another.

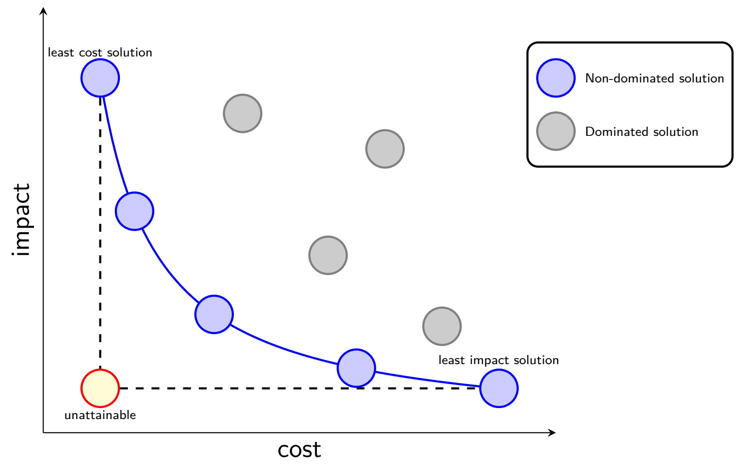
- *Pareto set*: $\{x \in \mathcal{X} : x \text{ is Pareto optimal}\}$
- *Pareto front*: its image in objective space

Dominance



Pareto optimality

Image of the objective function is a set of points in $\mathbb{R}^k$



Evolution

> Evolutionary algorithms seem particularly suitable to solve multiobjective optimization problems, because they deal simultaneously with a set of possible solutions (the so-called population). This allows us to find several members of the Pareto optimal set in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of the traditional mathematical programming techniques. Additionally, evolutionary algorithms are less susceptible to the shape or continuity of the Pareto front (e.g., they can easily deal with discontinuous or concave Pareto fronts), whereas these two issues are a real concern for mathematical programming techniques.
>
> – Carlos A. Coello Coello, 2001

Evolution

Need a fitness function for selection. How to deal with $f\colon \mathcal{X} \to \mathbb{R}^k$? How to deal with comparing points that do not dominate one another?

Aggregation

$$f'(x) = \sum_{i=1}^{k} w_i f_i(x)$$

where $w_i \geq 0$ and $\sum_i w_i = 1$.

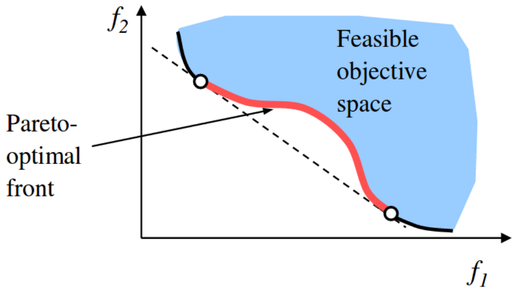
Each w_i is a weighting coefficient that represents the relative importance of each objective function.

Now, just use f' in a single-objective algorithms, e.g., $(\mu{+}\lambda)$ EA.

Evolution

Severe problems with aggregation

1. How to correctly set the weights?
2. Nonconvex Pareto fronts:



https://engineering.purdue.edu/~sudhoff/ee630/Lecture09.pdf

Lexicographic ordering

If the objectives can be *ranked* in order of importance, we can impose a total order on the solutions. WLOG, f_1 is most important, f_2 second, etc.

$$x \prec y \text{ if } f_i(x) < f_i(y) \text{ where } i = \min\{j \in \{1, \ldots, k\} : f_j(x) \neq f_j(y)\}$$

- Efficient and easy to implement.
- Requires a pre-defined ordering of objectives and its performance will be affected by it.
- Not appropriate when k is very large

Again can use single-objective EAs that use Lexicographic order for selection.

EMO algorithms

Keep an *archive* separate from the population to store good individuals

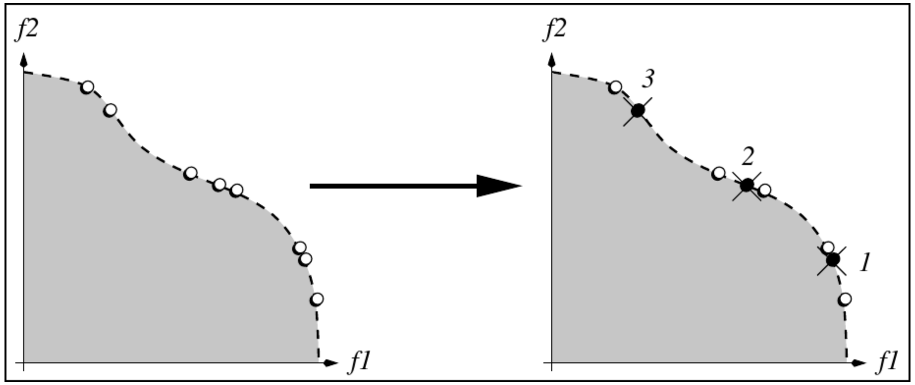
SPEA2

Input: Offspring population size M
Input: Archive size N
Output: Nondominated set A^*
$P_0 \leftarrow$ initial population, $A_0 \leftarrow \emptyset$, $t \leftarrow 0$;
while *stopping criterion not met* **do**
  Compute fitness of individuals in P_t and A_t;
  Copy all nondominated individuals in P_t and A_t to A_{t+1};
  if $|A_{t+1}| > N$ **then** use *truncation operation*;
  if $|A_{t+1}| < N$ **then** fill with best dominated individuals from $A_t \cup P_t$;
  Binary tournament selection with replacement on A_{t+1} to fill mating pool;
  Recombination and mutation to mating pool and set P_{t+1} to the resulting population;
  $t \leftarrow t + 1$;
end

EMO algorithms

SPEA2 truncation operation

Remove the individual with minimum distance to another individual, break ties with second-smallest distance, etc.



Zitzler, Laumanns and Bleuler, 2004

EMO algorithms

Evolve a set of non-dominated solutions

SEMO

Choose x uniformly at random from $\{0,1\}^n$;
$P \leftarrow \{x\}$;
while *true* **do**
  Select one element $x \in P$ uniformly;
  Create offspring x' by mutation;
  $P \leftarrow P \setminus \{x'' \in P : x' \preceq x''\}$;
  if $\nexists x'' \in P$ *such that* $(x'' \preceq x' \vee f(x'') = f(x'))$ **then**
    $P \leftarrow P \cup \{x'\}$
  end
end

Goals for EMO algorithm

1. Minimize the *distance* of the Pareto front produced by our algorithm with respect to the true Pareto front (assuming we know its location).
2. Maximize the *spread* of solutions found, so that we can have a distribution of vectors as smooth and uniform as possible.
3. Maximize the *number of unique elements* of the Pareto optimal set found.

(Zitzler, 2000)

Evaluating the performance of an EMO algorithm

In the single-objective case, we can measure the time until an optimal solution (or ρ-approximation) is found. What about the multi-objective case?

Enumeration. Enumerate the entire search space and compare the true Pareto front obtained against those fronts produced by any EMO approach.

Spread. Use of a statistical metric such as the chi-square test to measure how well solutions are spread along the Pareto front.

Generational Distance. Estimates how far is our current Pareto front from the true Pareto front of a problem using the Euclidean distance (measured in objective space) between each vector and the nearest member of the true Pareto front.

Coverage. Measure the size of the objective value space area which is covered by a set of nondominated solutions.

Some further reading

Dimo Brockhoff and Tobias Wagner. 2016. *GECCO 2016 Tutorial on Evolutionary Multiobjective Optimization.* In Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion (GECCO '16 Companion), Tobias Friedrich (Ed.). ACM, New York, NY, USA, 201-227. DOI: `https://doi.org/10.1145/2908961.2926974`

Branke, Jürgen, Kalyanmoy Deb, and Kaisa Miettinen, eds. *Multiobjective optimization: Interactive and evolutionary approaches.* Vol. 5252. Springer Science & Business Media, 2008. **[Available electronically @ Universitätsbibliotek]**

Kalyanmoy Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Inc., 2001

Zitzler E., Laumanns M., Bleuler S. (2004) *A Tutorial on Evolutionary Multiobjective Optimization.* In: Gandibleux X., Sevaux M., Sörensen K., T'kindt V. (eds) Metaheuristics for Multiobjective Optimisation. Lecture Notes in Economics and Mathematical Systems, vol 535. Springer, Berlin, Heidelberg