Evolutionary Algorithms: Lecture 4

Jiří Kubalík Department of Cybernetics, CTU Prague



http://labe.felk.cvut.cz/~posik/xe33scp/

Multi-Objective Optimization

- :: Many real-world problems involve multiple objectives
 - Conflicting objectives
 - A solution that is extreme with respect to one objective requires a compromise in other objectives.
 - A sacrifice in one objective is related to the gain in other objective(s).
 - Motivation example: Buying a car
 - two extreme hypothetical cars 1 and 2,
 - cars with a trade-off between cost and comfort – A, B, and C.



©Kalyanmoy Deb: Multi-Objective Optimization using Evolutionary Algorithms.

Which solution out of all of the trade-off solutions is the best with respect to all objectives?
 Without any further information those trade-offs are indistinguishable.

 \Rightarrow a number of optimal solutions is sought in multiobjective optimization!

:: General form of multi-objective optimization problem

• x is a vector of n decision variables: $x = (x_1, x_2, ..., x_n)^T$;

- Decision space is constituted by variable bounds that restrict each variable x_i to take a value within a lower $x_i^{(L)}$ and an upper $x_i^{(U)}$ bound;
- Inequality and equality constraints
- A solution x that satisfies all constraints and variable bounds is a feasible solution, otherwise it si called an infeasible solution;
- Feasible space is a set of all feasible solutions;
- Objective functions f(x) = (f₁(x), f₂(x), ..., f_M(x))^T constitute a multi-dimensional objective space.

Decision and Objective Space



©Kalyanmoy Deb: Multi-Objective Optimization using Evolutionary Algorithms.

• For each solution x in the decision space, there exists a point in the objective space

$$f(x) = z = (z_1, z_2, ..., z_M)^T$$

Task is to design a beam, defined by two decision variables

- diameter d,
- length *l*.

that can carry an end load P and is optimal with respect to the following **objectives**

- f_1 minimization of weight,
- f₂ minimization of deflection.
 Obviously, conflicting objectives!

subject to the following **constraints**





- the developed maximum stress σ_{max} is less than the allowable strength S_y ,
- the end deflection δ is smaller than a specified limit δ_{max} .

Cantilever Design Problem: Decision and Objective Space



©Kalyanmoy Deb: Multi-Objective Optimization using Evolutionary Algorithms.

Dominance and Pareto-Optimal Solutions

:: Domination: A solution $x^{(1)}$ is said to dominate the other solution $x^{(2)}$, $x^{(1)} \leq x^{(2)}$, if $x^{(1)}$ is no worse than $x^{(2)}$ in all objectives and $x^{(1)}$ is strictly better than $x^{(2)}$ in at least one objective.



©Kalyanmoy Deb: Multi-Objective Optimization using Evolutionary Algorithms.

Solutions A, B, C, D are *non-dominated* solutions (Pareto-optimal solutions) Solution E is *dominated* by C a B (E is non-optimal). Dominance is

- is not reflexive because any solution does not dominate itself,
- is not symmetric because $p \preceq q$ does not imply $q \preceq p$,
- is not antisymmetric implies from previous line,
- is transitive because if $p \leq q$ and $q \leq r$, then $p \leq r$.



©Kalyanmoy Deb: Multi-Objective Optimization using Evolutionary Algorithms.

Principles of Multi-Objective Optimization

- :: Every finite set of solutions P can be divided into two non-overlapping sets
 - non-dominated set P_1 contains all solutions that do not dominate each other, and
 - dominated set P_2 at least one solution in P_1 dominates any solution in P_2 .
- :: Two goals in multi-objective optimization
- 1. Quality To find a set of solutions as close as possible to the Pareto-optimal front.
- 2. **Spread** To find a set of solutions as diverse as possible.



©Kalyanmoy Deb: Multi-Objective Optimization using Evolutionary Algorithms.

Non-Conflicting Objectives

:: There exist multiple Pareto-optimal solutions in a problem only if the objectives are conflicting to each other.

• If this does not hold then the cardinality of the Pareto-optimal set is one.

Example: Cantilever beam design problem

- f_1 minimizing the end deflection δ ,
- f_2 minimizing the maximum developed stress in the beam σ_{max} .



©Kalyanmoy Deb: Multi-Objective Optimization using Evolutionary Algorithms.

Differences with Single-Objective Optimization

:: Two (orthogonal) goals instead of one

- progressing towards the Pareto-optimal front,
- maintaining a diverse set of solutions in the non-dominated set.

:: Dealing with two search spaces

- objective vs. decision space,
- in which space the diversity must be achieved?

:: No artificial fix-ups

 weighted sum approach – multiple objectives are weighted and summed together to create a composite objective function.

Its performance depends on the chosen weights.

 ε-constraint method – chooses one of the objective functions and treats of the objectives as constraints by limiting each of them within certain predefined limits.

Also depends on the chosen constraint limits.

Difficulties with Classical Optimization Algorithms

- The convergence to an optimal solution depends on the chosen initial solution.
- Most algorithms tend to get stuck to a suboptimal solution.
- An algorithm efficient in solving one optimization problem may not be efficient in solving a different opt. problem.
- Algorithms are not efficient in handling problems having a discrete search space.
- Algorithms cannot be efficiently used on a parallel machine

Pareto Archived Evolution Strategy (PAES)

Knowles, J.D., Corne, D.W. (2000) Approximating the nondominated front using the Pareto archived evolution strategy. Evolutionary Computation, 8(2), pp. 149-172

Multiple Objective Genetic Algorithm (MOGA)

Carlos M. Fonseca, Peter J. Fleming: Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization, In Genetic Algorithms: Proceedings of the Fifth International Conference, 1993

Niched-Pareto Genetic Algorithm (NPGA)

Jeffrey Horn, Nicholas Nafpliotis, David E. Goldberg: A Niched Pareto Genetic Algorithm for Multiobjective Optimization, Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, 1994

SPEA2

Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm For Multiobjective Optimization, In: Evolutionary Methods for Design, Optimisation, and Control, Barcelona, Spain, pp. 19-26, 2002

NSGA

Srinivas, N., and Deb, K.: Multi-objective function optimization using non-dominated sorting genetic algorithms, Evolutionary Computation Journal 2(3), pp. 221-248, 1994

NSGA-II

Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T Meyarivan: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II, In Proceedings of the Parallel Problem Solving from Nature VI Conference, 2000

• • • •

Non-Dominated Sorting Genetic Algorithm (NSGA)

- :: Common features with the standard GA
 - variation operators crossover and mutation,
 - selection method Stochastic Reminder Roulette-Wheel,
 - standard generational evolutionary model.
- :: What distinguishes NSGA from the SGA
 - fitness assignment scheme which prefers non-dominated solutions, and
 - fitness sharing strategy which preserves diversity among solutions of each non-dominated front.

:: Algorithm NSGA

- 1. Initialize population of solutions
- 2. Repeat
 - Calculate objective values and assign fitness values
 - Generate new population

Until stopping condition is fulfilled

:: Diversity preservation method originally proposed for solving multi-modal optimization problems so that GA is able to sample each optimum with the same number of solutions.

:: Idea – diversity in the population is preserved by degrading the fitness of similar solutions

:: Algorithm for calculating the shared fitness function value of i-th individual in population of size ${\cal N}$

1. calculate *sharing function* value with all solutions in the population according to

$$Sh(d) = \begin{array}{l} 1 - (\frac{d}{\sigma_{share}})^{\alpha}, & \text{if } d \leq \sigma_{share} \\ \mathbf{0}, & \text{otherwise.} \end{array}$$

2. calculate niche count nc_i as follows

$$nc_i = \sum_{j=1}^N Sh(d_{ij})$$

3. calculate shared fitness as

$$f_i' = f_i/nc_i$$

:: Remark: If d = 0 then Sh(d) = 1 meaning that two solutions are identical. If $d \ge \sigma_{share}$ then Sh(d) = 0 meaning that two solutions do not have any sharing effect on each other.

Fitness Sharing: Example

- :: Bimodal function six solutions and corresponding shared fitness functions
 - $\sigma_{share} = 0.5, \ \alpha = 1.$

Sol.	String	Decoded	$\chi^{(i)}$	fi	nci	f'i
i		value				ť
1	110100	52	1.651	0.890	2.856	0.312
2	101100	44	1.397	0.948	3.160	0.300
3	011101	29	0.921	0.246	1.048	0.235
4	001011	11	0.349	0.890	1.000	0.890
5	110000	48	1.524	0.997	3.364	0.296
6	101110	46	1.460	0.992	3.364	0.295



©Kalyanmoy Deb: Multi-Objective Optimization using Evolutionary Algorithms.

- :: Let's take the first solution
 - $d_{11} = 0.0$, $d_{12} = 0.254$, $d_{13} = 0.731$, $d_{14} = 1.302$, $d_{15} = 0.127$, $d_{16} = 0.191$
 - $Sh(d_{11}) = 1$, $Sh(d_{12}) = 0.492$, $Sh(d_{13}) = 0$, $Sh(d_{14}) = 0$, $Sh(d_{15}) = 0.746$, $Sh(d_{16}) = 0.618$.
 - $nc_1 = 1 + 0.492 + 0 + 0 + 0.746 + 0.618 = 2.856$
 - $f'(1) = f(1)/nc_1 = 0.890/2.856 = 0.312$

NSGA: Fitness Assignment

- **::** Input: Set *P* of solutions with assigned objective values.
- **:: Output**: Set of solutions with assigned fitness values (the bigger the better).
- 1. Choose sharing parameter σ_{share} , small positive number ϵ , initialize $F_{max} = PopSize$ and front counter front = 1
- 2. Find set $P' \subset P$ of non-dominated solutions
- 3. For each $q \in P'$
 - assign fitness $f(q) = f_{max}$,
 - calculate sharing function with all solutions in P' niche count ncq among solutions of P' only, the normalized Euclidean distance d_{ij} is calculated
 - calculate shared fitness $f'(q) = f(q)/nc_q$.

4.
$$f_{max} = min(f'(q) : q \in P') - \epsilon$$

 $P = P \setminus P'$
 $front = front + 1$

5. If not all solutions are assessed go to step 2, otherwise stop.

$$d_{ij} = \sqrt{\sum_{k=1}^{n} (\frac{x_k^{(i)} - x_k^{(j)}}{x_k^{max} - x_k^{min}})^2}$$

- :: Example:
 - First, 10 solutions are classified into different non-dominated fronts.
 - Then, the fitness values are calculated according to the fitness sharing method.
 - The sharing function method is used front-wise.
 - Within a front, less dense solutions have better fitness values.



©Kalyanmoy Deb: Multi-Objective Optimization using Evolutionary Algorithms.

:: Computational complexity

- Governed by the non-dominated sorting procedure and the sharing function implementation.
 - non-dominated sorting complexity of $O(MN^3)$.
 - sharing function requires every solution in a front to be compared with every other solution in the same front, total of $\sum_{j=1}^{\rho} |P_j|^2$, where ρ is a number of fronts. Each distance computation requires evaluation of n differences between parameter values. In the worst case, when $\rho = 1$, the overall complexity is of $O(nN^2)$.

:: Advantages

- Assignment of fitness according to non-dominated sets makes the algorithm converge towards the Pareto-optimal region.
- Sharing allows phenotypically diverse solutions to emerge.

:: Disdvantages

- sensitive to the sharing method parameter σ_{share} .
 - some guidelines for setting the parameter based on the expected number of optima q.

$$\sigma_{share} = \frac{0.5}{\sqrt[n]{q}}$$

- or dynamic update procedure of σ_{share} .

NSGA-II

- :: Fast non-dominated sorting approach
 - Computational complexity of $O(MN^2)$.
- :: Diversity preservation
 - the sharing function method is replaced with a crowded comparison approach,
 - parameterless approach.
- :: Elitist evolutionary model

NSGA-II: Diversity preservation

:: Density estimation



©Kalyanmoy Deb: Multi-Objective Optimization using Evolutionary Algorithms.

:: Crowded comparison operator

Every solution in the population has two attributes

- 1. non-domination rank (i^{rand}) , and
- 2. crowding distance ($i^{distance}$).

A partial order \prec_n is defined as:

$$i \prec_n j \text{ if}(i^{rank} < j^{rank}) \text{ or } ((i^{rank} = j^{rank}) and (i^{distance} > j^{distance}))$$

NSGA-II: Evolutionary Model

- 1. Current population P_t is sorted based on the non-domination Each solution is assigned a fitness equal to its non-domination level (1 is the best).
- 2. The usual binary tournament selection, recombination, and mutation are used to create a child population Q_t of size N.
- 3. Combined population $R_t = P_t \cup Q_t$ is formed. Elitism is ensured.
- 4. Population P_{t+1} is formed according to the following schema



©Kalyanmoy Deb: Multi-Objective Optimization using Evolutionary Algorithms.



©Kalyanmoy Deb et al.: A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II.

NSGA-II: Simulation Results cont.



©Kalyanmoy Deb et al.: A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II.

:: Binary tournament selection with modified domination concept is used to choose the better solution out of the two solutions i and j, randomly picked up from the population.

:: In the presence of constraints each solution in the population can be either **feasible** or **infeasible**, so that there are the following three possible situations:

- both solutions are feasible,
- one is feasible and other is not,
- both are infeasible.

:: Constrained-domination: A solution i is said to constrained-dominate a solution j, if any of the following conditions is true

- 1. Solution i is feasible and solution j is not.
- 2. Solutions i and j are both infeasible, but solution i has a smaller overall constraint violation.
- 3. Solutions i and j are feasible, and solution i dominates solution j.

NSGA-II: Simulation Results cont.

Comparison of NSGA-II and Ray-Kang-Chye's Constraint handling approach

Ray, T., Tai, K. and Seow, K.C. [2001] "Multiobjective Design Optimization by an Evolutionary Algorithm", Engineering Optimization, Vol.33, No.4, pp.399-424



©Kalyanmoy Deb et al.: A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II.

NSGA-II: Simulation Results cont.

Comparison of NSGA-II and Ray-Kang-Chye's Constraint handling approach



©Kalyanmoy Deb et al.: A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II.