Multiobjective Optimization Algorithms

Sébastien Verel

LISIC
Université du Littoral Côte d'Opale
Equipe OSMOSE

verel@univ-littoral.fr
http://www.lisic.univ-littoral.fr/~verel

Master informatique WeDSci, ULCO,

2020, version 0

Single Objective Optimization

Inputs

- Search space: Set of all feasible solutions,

$$\mathcal{X}$$

- Objective function: Quality criterium

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

Goal

Find the best solution according to the criterium

$$x^{\star} = \mathrm{argmax}\ f$$

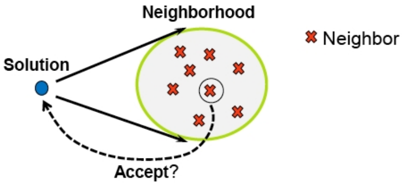
But, sometime, the set of all best solutions, good approximation of the best solution, good 'robust' solution...

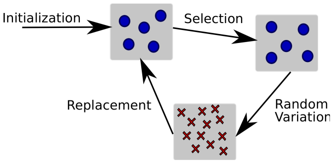
Search algorithms

Principle

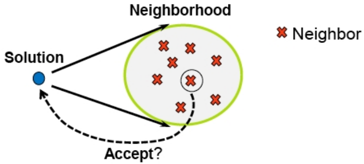
Enumeration of the search space

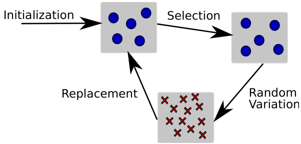
- A lot of ways to enumerate the search space
- Using random sampling: Monte Carlo technics
- Local search technics:





Search algorithms





- **Single solution-based**: Hill-climbing technics, Simulated-annealing, tabu search, Iterative Local Search, etc.
- **Population solution-based**: Genetic algorithm, Genetic programming, ant colony algorithm, etc.

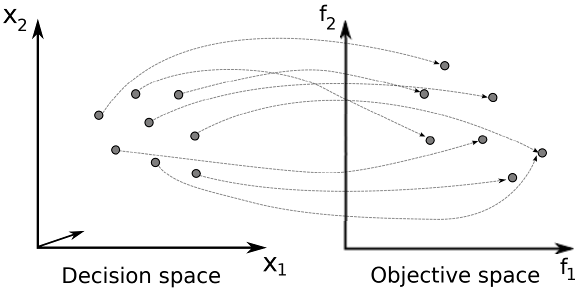
Design components are well-known

- Probability to decrease,
- Memory of path, of sub-space
- Diversity of population, etc.

Multiobjective optimization

Multiobjective optimization problem

- $\mathcal{X}$: set of feasible solutions in the decision space
- $M \geqslant 2$ objective functions $f = (f_1, f_2, \ldots, f_M)$ (to maximize)
- $\mathcal{Z} = f(\mathcal{X}) \subseteq \mathbb{R}^M$: set of feasible outcome vectors in the objective space

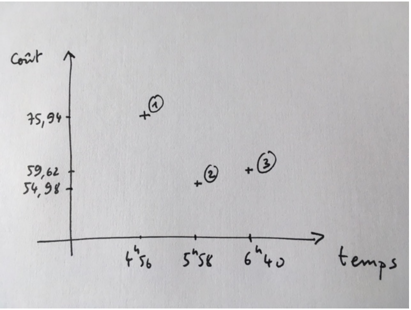


Pareto dominance definition

Pareto dominance relation (maximization)

A solution $x \in \mathcal{X}$ dominates a solution $x' \in \mathcal{X}$ ($x' \prec x$) iff

- $\forall i \in \{1, 2, \ldots, M\}$, $f_i(x') \leqslant f_i(x)$
- $\exists j \in \{1, 2, \ldots, M\}$ such that $f_j(x') < f_j(x)$

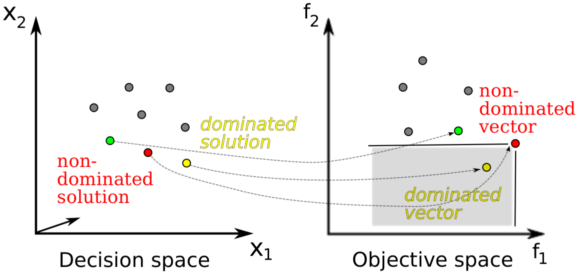


Pareto Optimale solution

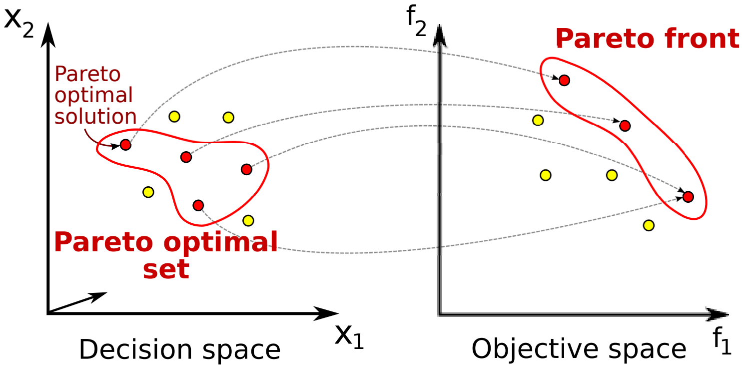
Definition: non-dominated solution

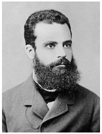
A solution $x \in \mathcal{X}$ is non-dominated (or Pareto optimal, efficient) iff

$$\forall x' \in \mathcal{X} \setminus \{x\},\ x' \prec x$$



Pareto set, Pareto front



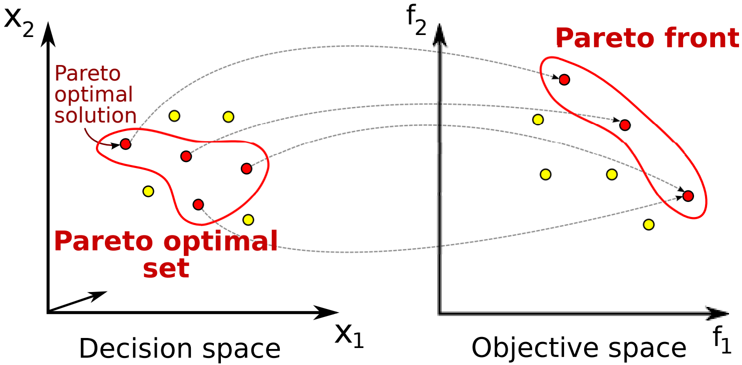


Vilfredo Pareto (1848 - 1923)
source: wikipedia

Multiobjective optimization goal

Goal

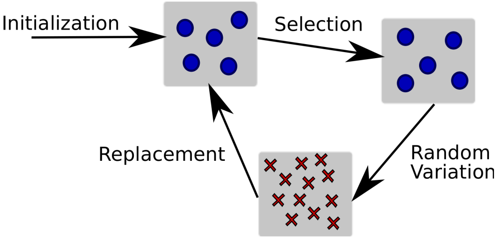
Find the Pareto Optimal Set,
or a good approximation of the Pareto Optimal Set
And not a single solution for a single agregated objective



MO algorithms

Population-based algorithm

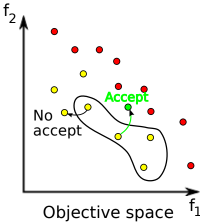
A MO algorithm is an Evolutionary Algorithm :
the goal is to find a set of solutions

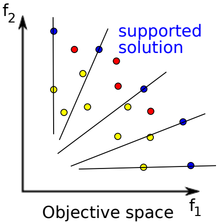


Main types of MO local search algorithms

Three main classes:

- Pareto-based approaches: directly or indirectly focus the search on the Pareto dominance relation.
  Pareto Local Search (PLS), Global SEMO, NSGA-II, etc.
- Indicator approaches: Progressively improvement the indicator function: IBEA, SMS-MOEA, etc.
- Scalar approaches: multiple scalarized aggregations of the objective functions: MOEA/D, etc.





Pareto-based approaches

Simple dominance-based EMO with unbounded archive

local search: Pareto Local Search (PLS)

Pick a random solution $x_0 \in X$
$A \leftarrow \{x_0\}$
repeat
  Select a non-visited $x \in A$
  Create $N(x)$ by flipping each bit of x in turns
  Flag x as visited
  $A \leftarrow$ non-dom. from $A \cup N(x)$
until all-visited $\vee$ maxeval

[Paquete *et al.* 2004][7]

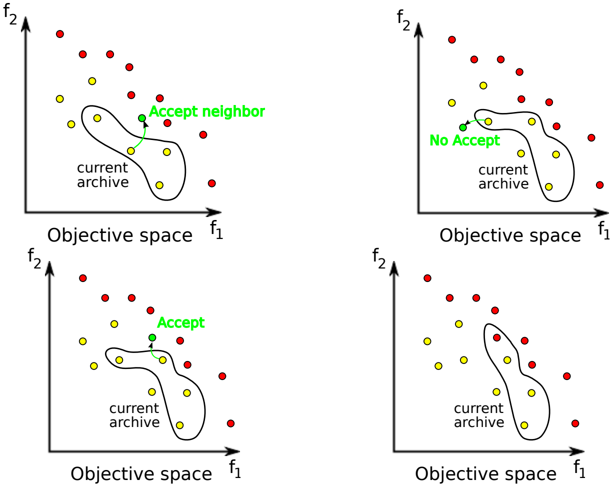
global search: Global-Simple EMO (G-SEMO)

Pick a random solution $x_0 \in X$
$A \leftarrow \{x_0\}$
repeat
  Select $x \in A$ at random
  Create x' by flipping each bit of x with a rate $1/N$
  $A \leftarrow$ non-dom. from $A \cup \{x'\}$
until maxeval

[Laumanns *et al.* 2004][5]

A Pareto-based approach: Pareto Local Search

- Archive solutions using **Dominance relation**
- Iteratively improve this archive by exploring the neighborhood



SMS-MOEA: $\mathcal{S}$ metric selection- EMOA

[Beume *et al.* 2007][1]

$P \leftarrow$ initialization()
repeat
  $q \leftarrow$ Generate(P)
  $P \leftarrow$ Reduce($P \cup \{q\}$)
until maxeval

Reduction

- Remove the worst solution according to non-dominated sorting, and $\mathcal{S}$ metric

Algorithm 2. Reduce(Q)

1: $\{\mathcal{R}_1, \ldots, \mathcal{R}_v\} \leftarrow$ fast-nondominated-sort(Q)   /* all v fronts of Q */
2: $r \leftarrow \text{argmin}_{s \in \mathcal{R}_v}[\Delta_{\mathcal{S}}(s, \mathcal{R}_v)]$ /* $s \in \mathcal{R}_v$ with lowest $\Delta_{\mathcal{S}}(s, \mathcal{R}_v)$ */
3: **return** $(Q \backslash \{r\})$ /* eliminate detected element */

IBEA: Indicator-Based Evolutionary algorithm
[Zitzler *et al.* 2004][9]

$P \leftarrow$ initialization()
repeat
    $P^{'} \leftarrow$ selection(P)
    $Q \leftarrow$ random_variation($P^{'}$)
    Evaluation of Q
    $P \leftarrow$ replacement(P, Q)
until maxeval

Fitness assignment

- Pairwise comparison of solutions in a population w.r.t. indicator I
- Fitness value: "loss in quality" in the population P if x was removed

$$f(x) = \sum_{x^{'} \in P \setminus \{x\}} (-e^{-I(x^{'},x)/\kappa})$$

- Often the ϵ-indicator is used

Decomposition based approaches: MOEA/D

Principe

Divide the multi-objective problem
into several single-objective sub-problems

- cf. slides présentation à Nagano: `2014-06-19-nagano-moeadxy.pdf`
- et slides suivants (new)

Original MOEA/D [8] (minimization)

```
/* μ sub-problems defined by μ directions */
(λ^1, ..., λ^μ) ← initialization_direction()
Initialize ∀i = 1..μ B(i) the neighboring sub-problems of sub-problem i
/* one solution for each sub-problem */
(x^1, ..., x^μ) ← initialization_solution()
repeat
   for i = 1..μ do
      Select x and x' randomly in {x_j : j ∈ B(i)}
      y ← mutation_crossover(x, x')
      for j ∈ B(i) do
         if g(y|λ_j, z_j*) < g(x_j|λ_j, z_j*) then
            x_j ← y
         end if
      end for
   end for
until max_eval
```

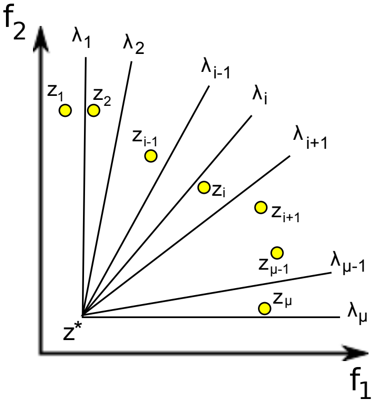
$B(i)$ is the set of the T closest neighboring sub-problems of sub-problem i
$g(\ |\lambda_i, z_i^\star)$: scalar function of sub-pb. i with λ_i direction, and $z_i^\star$ reference point

MOEA/D steady-state variant

Another MOEA/D (minimization)

/* μ *sub-problems defined by* μ *directions* */
$(\lambda^1, \dots, \lambda^\mu) \leftarrow$ initialization_direction()
Initialize $\forall i = 1..\mu$ $B(i)$ the neighboring sub-problems of sub-problem i
/* *one solution for each sub-problem* */
$(x^1, \dots, x^\mu) \leftarrow$ initialization_solution()
repeat
    Select i at random $\in 1..\mu$
    Select x randomly in $\{x_j \ : \ j \in B(i)\}$
    $y \leftarrow$ mutation_crossover(x_i, x)
    for $j \in B(i)$ **do**
        if $g(y|\lambda_j, z_j^\star) < g(x_j|\lambda_j, z_j^\star)$ **then**
            $x_j \leftarrow y$
        end if
    end for
until max_eval

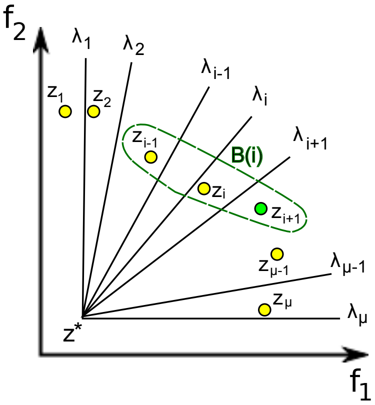
Representation of steady-state MOEA/D



Population at iteration t

- Minimization problem
- One solution x_i for each sub pb. i
- Representation of solutions in objective space: $z_i = g(x_i|\lambda_i, z_i^\star)$
- Same reference point for all sub-pb. $z^\star = z_1^\star = \ldots = z_\mu^\star$
- Scalar function g: Weighted Tchebycheff
- Neighborhood size $\sharp B(i) = T = 3$

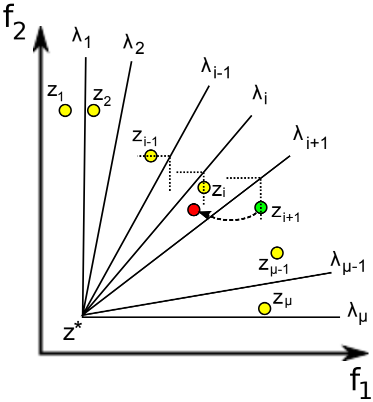
Representation of steady-state MOEA/D



From the neigh. $B(i)$ of sub-pb. i, x_{i+1} is selected

- Minimization problem
- One solution x_i for each sub pb. i
- Representation of solutions in objective space: $z_i = g(x_i|\lambda_i, z_i^\star)$
- Same reference point for all sub-pb. $z^\star = z_1^\star = \ldots = z_\mu^\star$
- Scalar function g: Weighted Tchebycheff
- Neighborhood size $\sharp B(i) = T = 3$

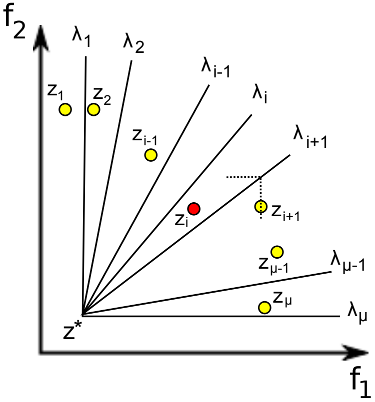
Representation of steady-state MOEA/D



The mutated solution y is created

- Minimization problem
- One solution x_i for each sub pb. i
- Representation of solutions in objective space: $z_i = g(x_i|\lambda_i, z_i^\star)$
- Same reference point for all sub-pb. $z^\star = z_1^\star = \ldots = z_\mu^\star$
- Scalar function g: Weighted Tchebycheff
- Neighborhood size $\sharp B(i) = T = 3$

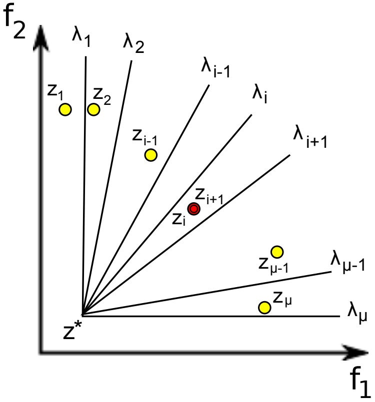
Representation of steady-state MOEA/D



According to scalar fonction,
y is worst than x_{i-1},
y is better than x_i and replaces it.

- Minimization problem
- One solution x_i for each sub pb. i
- Representation of solutions in objective space: $z_i = g(x_i | \lambda_i, z_i^\star)$
- Same reference point for all sub-pb. $z^\star = z_1^\star = \ldots = z_\mu^\star$
- Scalar function g:
  Weighted Tchebycheff
- Neighborhood size $\sharp B(i) = T = 3$

Representation of steady-state MOEA/D



According to scalar fonction,
y is also better than x_{i+1}
and replaces it for the next iteration.

- Minimization problem
- One solution x_i for each sub pb. i
- Representation of solutions in objective space: $z_i = g(x_i | \lambda_i, z_i^\star)$
- Same reference point for all sub-pb. $z^\star = z_1^\star = \ldots = z_\mu^\star$
- Scalar function g:
  Weighted Tchebycheff
- Neighborhood size $\sharp B(i) = T = 3$

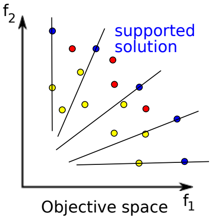
Decomposition based approaches: MOEA/D

Main issues

1. Impact of the scalar function:
   cf. Slide suivant et `ppsn2014poster-impactScalarFunction.pdf` [Derbel *et. al.*, 2014] [2]
2. Direction of search:
   cf. slides 30-34 et [Derbel *et. al.*, 2014] [3]
3. Cooperation between sub-problems:
   cf. slides Nagano: `2014-06-19-nagano-moeadxy.pdf` [Gauvain *et al.*, 2014] [6]
4. Parallelization:
   cf. algorithm of "A fine-grained message passing MOEA/D" [Derbel *et al.*, 2015] [4]

Scalar approaches: scalarizing function

- multiple scalarized aggregations of the objective functions



Different aggregations

- Weighted sum:

$$g(x|\lambda) = \sum_{i=1..m} \lambda_i f_i(x)$$

- Weighted Tchebycheff:

$$g(x|\lambda, z) = \max_{i=1..m} \{\lambda_i |z_i - f_i(x)|\}$$

- ... cf. poster PPSN 2014

Nicola Beume, Boris Naujoks, and Michael Emmerich.
Sms-emoa: Multiobjective selection based on dominated hypervolume.
European Journal of Operational Research, 181(3):1653–1669, 2007.

Bilel Derbel, Dimo Brockhoff, Arnaud Liefooghe, and Sébastien Verel.
On the impact of multiobjective scalarizing functions.
In *Parallel Problem Solving from Nature–PPSN XIII*, pages 548–558. Springer, 2014.

Bilel Derbel, Jérémie Humeau, Arnaud Liefooghe, and Sébastien Verel.
Distributed localized bi-objective search.
European Journal of Operational Research, 239(3):731–743, 2014.

Bilel Derbel, Arnaud Liefooghe, Gauvain Marquet, and El-Ghazali Talbi.
A fine-grained message passing moea/d.
In *Evolutionary Computation (CEC), 2015 IEEE Congress on*, pages 1837–1844. IEEE, 2015.

Marco Laumanns, Lothar Thiele, and Eckart Zitzler.
Running time analysis of evolutionary algorithms on a simplified multiobjective knapsack problem.
Nat Comput, 3(1):37–51, 2004.

Gauvain Marquet, Bilel Derbel, Arnaud Liefooghe, and El-Ghazali Talbi.
Shake them all!
In *Parallel Problem Solving from Nature–PPSN XIII*, pages 641–651. Springer, 2014.

L. Paquete, M. Chiarandini, and T. Stützle.

Pareto local optimum sets in the biobjective traveling salesman problem: An experimental study.
In *Metaheuristics for Multiobjective Optimisation*, volume 535 of *Lecture Notes in Economics and Mathematical Systems*, chapter 7, pages 177–199. Springer, 2004.

Qingfu Zhang and Hui Li.
Moea/d: A multiobjective evolutionary algorithm based on decomposition.
Evolutionary Computation, IEEE Transactions on, 11(6):712–731, 2007.

Eckart Zitzler and Simon Künzli.
Indicator-based selection in multiobjective search.
In *Parallel Problem Solving from Nature-PPSN VIII*, pages 832–842. Springer, 2004.