Metaheuristics for Multi-objective Optimization: A Unified View

Prof. El-Ghazali Talbi

http://www.lifl.fr/~talbi
El-ghazali.talbi@univ-lille1.fr
Outline

• Multi-objective optimization: definitions, problems, etc

• A unified view of multi-objective metaheuristics

• Landscapes and performance analysis

• Software framework for multi-objective optimization: ParadisEO-MOEO
Multiobjective Optimization Problem (MOP)

\[
(MOP) = \left\{ \begin{array}{l}
\text{min } f(x) = (f_1(x), f_2(x), \ldots, f_n(x)) \\
\text{s.t. } x \in X
\end{array} \right.
\]

- \( n \geq 2 \) objective functions \((f_1, f_2, \ldots, f_n)\)
- \( x \in X \) is a decision vector
- \( X \) is the feasible set in the decision space
- \( Z \) is the feasible set in the objective space
**Pareto dominance** [Pareto 1896]

An objective vector $z \in Z$ dominates an objective vector $z' \in Z$ iff

- $\forall i \in \{1, \ldots, n\}, z_i \leq z'_i$
- $\exists j \in \{1, \ldots, n\}, z_j < z'_j$

**Non-dominated solution**
(eligible, efficient, non inferior, Pareto optimal)
Multi-objective Optimization Problem (MOP)

$X: \text{decision space}$

$Z: \text{objective space}$

efficient solution

dominated vector

non-dominated vector

efficient set

Pareto front

$x_1$ $x_2$

$f_1$ $f_2$
Multi-objective optimization problems

• **Academic problems**
  - Continuous optimization: ZDT, CTP, DTLZ,
  - Combinatorial optimization problems
    – Polynomially problems (assignment, spanning tree, shortest path)
    – NP-hard problems (TSP, QAP, knapsack, routing, scheduling)

• **Real-life applications**
  - Engineering design
  - Environment and energetics
  - Telecommunications
  - Control
  - Bioinformatics and computational biology
  - Transportation and logistics
Resolution Approaches

Multiobjective optimization as a part of the decision making process:

*A priori*
- Decision Maker (DM) *before* the resolution process

*A posteriori*
- Decision Maker (DM) *after* the resolution process

Interactive
- Decision Maker (DM) *during* the resolution process

\[\text{a priori knowledge} \rightarrow \text{DM} \rightarrow \text{preferences} \rightarrow \text{solver} \leftrightarrow \text{acquired knowledge} \rightarrow \text{results}\]
Resolution Methodologies

- **Exact Methods**
  - Problems of small size or specific structure

- **Metaheuristics**
  - Find a good approximation of the efficient set (or Pareto front)
  - Metaheuristics able to find multiple non-dominated solutions in a single run
What is a Good Approximation?

Approximating an efficient set is itself a bi-objective problem

- Min the distance to the Pareto front
  - well-converged efficient set approximation

- Max the diversity in the objective space (and/or decision space)
  - well-diversified efficient set approximation

![Diagram](image)
Genealogy of Metaheuristics
The number of multi-objective metaheuristics is growing exponentially!

- Very active research in the last two decades
- For each metaheuristic (e.g. EA, PSO, LS, TS, SA, ACO):
  - Hundreds of different designs
  - Hundreds of different implementations

- Give you the Catalog of the proposed algorithms: *I don’t like it*
  - May be bigger than a dictionary

- May have:
  - MO Evolutionary Algorithm 1 # MO Evolutionary Algorithm 2
  - MO Evolutionary Algorithm = MO Scatter Search 1 = MO PSO 1
  - MO Local Search 1 # MO Local Search 2
  - MO Iterated Local Search = MO GRASP
Just some algorithms: Compare with all those algorithms!
Motivations

⇒ A unified view

• Design and Implementation
  - Problem-dependent
  - Multi-objective-specific
  - Metaheuristic-specific

• Fine-grained decomposition of search mechanisms

• Common terminology and classification
  - Comparison of approaches (experimental analysis)
  - New approaches

---

metaheuristics for multiobjective optimization

Population based

Single solution based

ParadisEO-MOEO

Combinatorial and continuous MOP
A unified design view
Development process of a multi-objective metaheuristic

- Design concepts for metaheuristics
  - Representation
  - Constraint handling
  - Operators, and so on

- Design concepts for multiobjective metaheuristics
  - Fitness assignment
  - Diversity preserving
  - Elitism

- Implementation of a multiobjective metaheuristic
  - From scratch or no reuse
  - Code reuse
  - Design and code reuse (e.g., software framework ParadisEO–MOEO)
Design issues of multi-objective metaheuristics

• **Fitness assignment**
  • Guide the search towards Pareto optimal solutions for a better convergence.

• **Diversity preserving**
  • Generate a diverse set of Pareto solutions in the objective space and/or the decision space.

• **Elitism:**
  • Preservation and use of elite solutions.
  • Allows a robust, fast and a monotonically improving performance of a metaheuristic
Fitness Assignment

- **Scalar** approaches
  - Transformation to mono-objective problem(s)

- **Criterion-based** approaches
  - Each objective is treated separately

- **Dominance-based** approaches
  - The concept of dominance is used

- **Indicator-based** approaches
  - Use performance indicators to drive the search
Scalar approaches

• Aggregation methods
• Weighted metrics
• Goal programming
• \( \varepsilon \)-constraint approach
• Achievement functions

\[
f(x) = \sum_{i=1}^{n} \lambda_i f_i(x), \quad x \in S
\]

\[
(MOP(\lambda, z)) \begin{cases} 
\min (\sum_{j=1}^{n} \lambda_j |f_j(x) - z_j|^p)^{\frac{1}{p}} \\
\text{s.c.} \ x \in S
\end{cases}
\]

\[
(MCOP(\bar{z})) \begin{cases} 
\min (\sum_{j=1}^{n} \lambda_j \delta_j) \\
\text{s.c.} \ f_j(x) - \delta_j \leq \bar{z}_j, \ j \ in [1, n] \\
\delta_j \geq 0, \ j \ in [1, n] \\
x \in S
\end{cases}
\]

\[
(MOP(\lambda, z)) \begin{cases} 
\min \max_{j \in [1, n]} [w_j (f_j(x) - \bar{z}_j)] + \rho \sum_{j=1}^{n} (f_j(x) - \bar{z}_j) \\
\text{s.c.} \ x \in S
\end{cases}
\] (1.16)

\[
\begin{cases} 
\min \alpha \\
\text{s.c.} \ x \in S \\
f_i(x) \leq z \ast_i + \alpha \lambda_i, \ i = 1, \ldots, n \\
\sum_{i=1}^{n} \lambda_i = 1
\end{cases}
\]
Aggregation Metaheuristics

• **Weights**: Static, Multiple, Dynamic, Adaptive

• **Genetic algorithms** [Hajela et Lin 92]
  • Individual representation: solution + λ
  • Goal: generating various Pareto solutions

• **Simulated annealing** [Serafini 92]
  • Acceptance probability

• **Tabu search** [Dahl et al. 95]

• **Hybrid metaheuristics** [Talbi 98]
  • Greedy algorithm + Simulated annealing [Tuyttens 98]
  • Genetic algorithm (Local search) [Ishibuchi et Murata 98]
    – Selection with different weights
    – Local search on the produced individual (same weights)
Criterion-based Approaches: Sequential

- **Sequential approach**: Objectives are handled in sequential
- **Lexicographic selection** (priority order)
  - Tabu search, Genetic algorithms [Fourman 85]
  - Evolutionary strategies [Kursawe 91], …
Criterion-based Approaches: Parallel

- **Parallel approach**: Objectives are handled in parallel
- **Parallel selection (VEGA)** [Schaffer 85]

![Diagram showing population, sub-populations, and selection/reproduction/crossover/mutation](image)

- **Multi-sexual reproduction** [Lis & Eiben 96]
  - One class per objective
  - Reproduction (crossover) over several individuals
- **Ant colonies (pheromone/objective)**
  ➔ Tends to ignore compromised solutions
Dominance-based Approaches

• **Dominance relation** used during the fitness assignment process:
  - Pareto dominance
  - Weak dominance
  - Strict dominance
  - $\varepsilon$-dominance [Helbig & Pateva 1994]
  - g-dominance [Molina et al. 2009]
  - Guided domination
  - Fuzzy dominance
  - ...
Fitness assignment: Pareto ranking

• Pareto-based fitness assignment strategies
  • Dominance rank (e.g. used in MOGA)
    – Number of solutions which dominates the solution
  • Dominance depth (e.g. used in NSGA and NSGA-II)
  • Dominance count (e.g. combined with dominance rank in SPEA and SPEA2)
    – Number of solutions dominated by the solution
Indicator-Based Fitness Assignment

[Zitzler & Künzli 04]

Solutions compared on the basis of a binary quality indicator \( I \)

Fitness \( (A) = \) usefulness of A according to the optimization goal \( (I) \)

\[
\arg \min_{A \in \Omega} I(A, R)
\]

where \( \Omega \) represents the space of all efficient set approximations.

Examples of binary quality indicators:

- Additive epsilon indicator (\( I_{\epsilon+} \))
- Hypervolume indicator (\( I_{HD} \))
Diversity

Multi-modal optimization: locating every optima of the problem

- Independent iterative executions
- Sequential niching
  - Iterative execution with a penalization of the optima already found
- Parallel niching (sharing, crowding)
  - Only one execution
Diversity: Statistical density estimation

- **Kernel methods** *(sharing)*
  - Neighborhood of a solution in term of a function taking a distance as argument

- **Nearest neighbour** techniques
  - Distance of a solution to its $k^{th}$ nearest neighbour

- **Histograms**
  - Space divided onto neighbourhoods by an hypergrid

→ decision / objective space
Elitism

• Archive
  • External set storing non dominated solutions
  • Update criteria: size, convergence, diversity

• The archive can be involved in the search process:
  • Elitist selection

(a) Passive elitism
   Archive as an independent memory

(b) Active elitism
   Archive participates in the search process
Elitism

• No archive
  • Current approximation contained in the main population

• Unbounded archive
  • All nondominated solutions

• Bounded archive
  • A reasonable number of nondominated solutions

• Fixed-size archive
  • cf. SPEA2 [Zitzler et al. 2001]
A Model for Evolutionary Algorithms

Main issues

• Problem-dependent components
  representation, initialization, evaluation, variation (recombination, mutation)

• Multi-objective specific components
  fitness assignment, diversity preservation, archiving

• Metaheuristic specific components
  selection, replacement, stopping condition
# EMO Algorithms as Instances of the Model

<table>
<thead>
<tr>
<th>Components</th>
<th>NSGA-II [Deb et al. 02]</th>
<th>SPEA2 [Zitzler et al. 01]</th>
<th>IBEA [Zitzler and Künzli 04]</th>
<th>SEEA [Lefooghe et al. 10]</th>
</tr>
</thead>
<tbody>
<tr>
<td>fitness assignment</td>
<td>dominance-depth</td>
<td>dom-count + dom-rank</td>
<td>quality indicator</td>
<td>none</td>
</tr>
<tr>
<td>diversity preservation</td>
<td>crowding distance</td>
<td>k&lt;sup&gt;th&lt;/sup&gt; nearest neighbor</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>archiving</td>
<td>none</td>
<td>fixed-size archive</td>
<td>none</td>
<td>unbounded</td>
</tr>
<tr>
<td>selection</td>
<td>binary tournament</td>
<td>elitist selection</td>
<td>binary tournament</td>
<td>elitist selection</td>
</tr>
<tr>
<td>replacement</td>
<td>elitist replacement</td>
<td>generational replacement</td>
<td>elitist replacement</td>
<td>generational replacement</td>
</tr>
<tr>
<td>stopping condition</td>
<td>number of generations</td>
<td>number of generations</td>
<td>number of generations</td>
<td>user-defined</td>
</tr>
</tbody>
</table>
A Model for Dominance-based Local Search (DMLS)

Main issues

• Problem dependent components
  representation, initialization, evaluation, neighborhood, incremental evaluation

• Multi-objective specific components
  dominance relation, archiving

• Metaheuristic specific components
  current set selection, neighborhood exploration, stopping condition
# DMLS Algorithms as Instances of the Model

<table>
<thead>
<tr>
<th>Components</th>
<th>PLS-1 [Paquete et al. 04]</th>
<th>PLS-2 [Talbi et al. 01]</th>
<th>PAES [Knowles &amp; Corne00]</th>
<th>moRBC [Aguire &amp; Anaka 05]</th>
</tr>
</thead>
<tbody>
<tr>
<td>dominance relation</td>
<td>Pareto</td>
<td>Pareto</td>
<td>Pareto</td>
<td>Pareto</td>
</tr>
<tr>
<td>archiving</td>
<td>unbounded</td>
<td>unbounded</td>
<td>bounded hypergrid</td>
<td>bounded crowding</td>
</tr>
<tr>
<td>current set selection</td>
<td>partial 1 random sol.</td>
<td>exhaustive all solutions</td>
<td>partial μ solutions</td>
<td>partial 1 solution</td>
</tr>
<tr>
<td>neighborhood exploration</td>
<td>exhaustive all neighbors</td>
<td>exhaustive all neighbors</td>
<td>partial λ random neighbors</td>
<td>partial 1 dominating neighbor</td>
</tr>
<tr>
<td>stopping condition</td>
<td>natural all sol. visited</td>
<td>natural all sol. visited</td>
<td>user-defined</td>
<td>natural all sol. visited</td>
</tr>
</tbody>
</table>
Landscapes and Performance Analysis
Performance indicators

- Unary / Binary indicators

- Known Pareto optimal set / Unknown

- Cardinality, Distance, Volume

- Parameter less / additional parameters: reference point, ideal point, Nadir point, reference set, …
Performance indicators: Properties

• Monotonicity

• Objective scale independence

• Computational complexity

• Classification:
  • Convergence
  • Diversity (dispersion, extension)
  • Hybrid
PO known

- **Absolute efficiency (convergence)**
  - Proportion of Pareto solutions within $PO^*$
  \[ AE = \frac{|PO^* \cap PO|}{|PO|} \]

- **Distance ($PO^*$, PO)**
  - Worst distance
  \[ WD = \max \left( d(PO^*, y), y \in PO \right) \]
  
  - Mean distance
  \[ MD = \frac{\sum_{y \in PO} d(PO^*, y)}{|PO|} \]

- **Uniformity**
  \[ DIV = \frac{WD}{MD} \]
  
  \[ d(PO^*, y) = \min \left( d(x, y), x \in PO^* \right) \]
  \[ d(x, y) = \sum_{i=1}^{n} \lambda_i \left| f_i(x) - f_j(y) \right| \]
PO unknown

- **Relative efficiency**: number of solutions from A dominated by B

\[
\begin{align*}
A \neq B \\
ND(A \cup B) &= A \\
A \cap ND(A \cup B) &= \emptyset \\
ND(A \cup B) &= B \\
B - ND(A \cup B) \neq \emptyset \\
A \neq B \\
B \neq A
\end{align*}
\]
PO unknown: Convergence

Contribution: Evaluating the quality of the solutions from a set towards another one

\[ \text{Cont}(PO_1/PO_2) = \frac{|C/2 + W_1 + N_1|}{|C| + |W_1| + |N_1| + |W_2| + |N_2|} \]

Example:
- If \( PO_1 = PO_2 \) then \( \text{Cont}(PO_1/PO_2) = 0.5 \)
- If \( PO_1 > PO_2 \) then \( \text{Cont}(PO_1/PO_2) = 1 \)

\( C = 4 \)
\( W_1 = 4 - N_1 = 1 \)
\( W_2 = 0 - N_2 = 1 \)

- \( \text{Cont}(O,X) = 0.7 \)
- \( \text{Cont}(X,O) = 0.3 \)
PO unknown: Diversity

- **Entropy**: builds a niche around every solution of \( ND(PO_1 \cup PO_2) = PO^* \)
  - \( E(PO_1, PO_2) \): diversity of the solutions of \( PO_1 \) in comparison of those in the niches of \( PO^* \)

\[
E(PO_1, PO_2) = \frac{-1}{\ln(\gamma)} \sum_{i=1}^{PO^*} \left( \frac{1}{Ni} \frac{n_i}{||PO_1||} \ln \frac{n_i}{||PO_1||} \right)
\]
PO unknown: Hybrid

- S-metric / Hypervolume
  \cite{Zitzler99}

Size of the objective space enclosed by PO* and a reference point $Z^\text{ref}$
Other indicators

• **Generational distance** (convergence)

\[ I_{GD}^t(A, R) = \frac{\left(\sum_{u \in A} \min_{v \in R} \| F(u) - F(v) \|^2 \right)^{1/2}}{|R|} \]

• **Extent** (diversity)

\[ I_{ex}(A) = \left(\sum_{i=1}^{n} \max_{u, u' \in A} \| f_i(u) - f_i(u') \| \right)^{1/2} \]

• **Spread** (diversity)

\[ I_S = \frac{\sum_{u \in A} |\{u' \in A : \| F(u) - F(u') \| > \sigma \}|}{|A| - 1} \]

• **E-indicator** (convergence)

\[ I_{\varepsilon^+}(A, B) = \min_{\varepsilon \in \mathbb{R}} \{ \forall z \in B, \exists z' \in A : z'_i - \varepsilon \leq z_i, \forall 1 \leq i \leq n \} \]
Performance indicators

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Goal</th>
<th>Monotone</th>
<th>Complexity</th>
<th>Parameter</th>
<th>[] , Min-Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contribution</td>
<td>Conv.</td>
<td>Mon.</td>
<td>$O(</td>
<td>A</td>
<td>\cdot</td>
</tr>
<tr>
<td>Gen. Dist.</td>
<td>Conv.</td>
<td>No</td>
<td>$O(</td>
<td>A</td>
<td>\cdot</td>
</tr>
<tr>
<td>$\epsilon$ indicator</td>
<td>Conv.</td>
<td>Mon.</td>
<td>$O(n \cdot</td>
<td>A</td>
<td>\cdot</td>
</tr>
<tr>
<td>$I_p$</td>
<td>Conv.</td>
<td>Mon.</td>
<td>$O(</td>
<td>A</td>
<td>\cdot</td>
</tr>
<tr>
<td>$I_{ad}$</td>
<td>Conv.</td>
<td>No</td>
<td>$O(</td>
<td>A</td>
<td>\cdot</td>
</tr>
<tr>
<td>Spread</td>
<td>Div.</td>
<td>No</td>
<td>$O(</td>
<td>A</td>
<td>^2)$</td>
</tr>
<tr>
<td>Extent</td>
<td>Div.</td>
<td>No</td>
<td>$O(</td>
<td>A</td>
<td>^2)$</td>
</tr>
<tr>
<td>B. Entropy</td>
<td>Div.</td>
<td>No</td>
<td>$O((</td>
<td>R</td>
<td>+</td>
</tr>
<tr>
<td>U. Entropy</td>
<td>Div.</td>
<td>No</td>
<td>$O((</td>
<td>A</td>
<td>\cdot \mu^n)$</td>
</tr>
<tr>
<td>Hypervolume</td>
<td>Hybrid</td>
<td>Strict</td>
<td>$O(</td>
<td>A</td>
<td>^n)$</td>
</tr>
<tr>
<td>R-metrics</td>
<td>Hybrid</td>
<td>Mon.</td>
<td>$O(n \cdot</td>
<td>A</td>
<td>\cdot</td>
</tr>
</tbody>
</table>
Landscapes

How to describe a Pareto front?

• Convexity / Concave Pareto fronts
• Multi-modality and deceptive attractors
• Isolated optimum (Flat space)
• Continuous / Discontinuous
• Uniform distribution
Benchmarks: ZDT

- Convexity versus non-convexity of the Pareto optimal front (ZDT1 versus ZDT2).

- Discontinuities and gaps in the Pareto-optimal front (ZDT1 or ZDT2 versus ZDT3).

- Multiple locally Pareto optimal fronts towards the globally Pareto optimal front (ZDT1 versus ZDT4).

- Isolation and deception of the globally Pareto optimal front (ZDT1 versus ZDT5).

- Non-uniform density of solutions across the Pareto optimal front (ZDT2 versus ZDT6).
Supported / Non supported
Landscapes

**Aggregation**: supported solutions only

**Convexity**: Proportion of Pareto solutions belonging to the convex hull

Complexity: $O(n \log(n))$

- **Non-dominated solutions**
- **Unsupported solutions**
- **Convex hull**
- **Dominated solutions**
Multi-objectivization

A way to improve solving single-objective optimization problems

- **Objective function decomposition**
  - Several sub-objectives (separate conflicting goals)
  - Reduce the number of local optima

- **Helper objectives**
  - Adding new objectives correlated with the main objective
  - Break plateaus of the landscape $\rightarrow$ smooth landscape
Development process of a multi-objective metaheuristic

- Design concepts for metaheuristics
  - Representation
  - Constraint handling
  - Operators, and so on

- Design concepts for multiobjective metaheuristics
  - Fitness assignment
  - Diversity preserving
  - Elitism

- Implementation of a multiobjective metaheuristic
  - From scratch or no reuse
  - Code reuse
  - Design and code reuse (e.g., software framework ParadisEO–MOEO)
Framework for multi-objective metaheuristics: ParadisEO

parallel and distributed metaheuristics

single solution-based Metaheuristics
(LS, SA, TS, TA, VNS, ILS)

population-based metaheuristics
(GA, GP, ES, EDA, PSO, ...)

multiobjective metaheuristics

http://paradiseo.gforge.inria.fr
ParadisEO

• Design and code reuse
  • Conceptual separation between the solution methods and the problem to be solved

• Flexibility and adaptability
  • Adding or updating other optimization methods, search mechanisms, operators, representation…

• Utility
  • Broad range of methods, components, parallel and distributed models, hybridization mechanisms…

• Transparent and easy access to performance and robustness
  • Parallel and hybrid implementation transparent to the hardware platform

• Portability
  • Operating systems: Windows, Linux, MacOS
  • Material architectures: sequential, parallel, distributed

• Usability and efficiency
Software Frameworks/Librariries for multi-objective metaheuristics

<table>
<thead>
<tr>
<th>Framework/Library</th>
<th>Meta</th>
<th>Type</th>
<th>Metrics</th>
<th>Hybrid</th>
<th>Parallel</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>ParadisEO</td>
<td>S-meta, P-meta</td>
<td>White</td>
<td>Many</td>
<td>Yes</td>
<td>Yes</td>
<td>C++</td>
</tr>
<tr>
<td>TEA</td>
<td>EA</td>
<td>White</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>C++</td>
</tr>
<tr>
<td>PISA</td>
<td>EA</td>
<td>Black</td>
<td>Many</td>
<td>No</td>
<td>No</td>
<td>Any</td>
</tr>
<tr>
<td>O. BEAGLE</td>
<td>EA</td>
<td>White</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>C++</td>
</tr>
<tr>
<td>MOMHLib++</td>
<td>EA,LS,SA</td>
<td>White</td>
<td>Many</td>
<td>Yes</td>
<td>No</td>
<td>C++</td>
</tr>
<tr>
<td>MOEA–Matlab</td>
<td>EA</td>
<td>Black</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Matlab</td>
</tr>
<tr>
<td>MALLBA</td>
<td>S-meta, P-meta</td>
<td>White</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>C++</td>
</tr>
</tbody>
</table>

S-meta: S-metaheuristics; P-meta: P-metaheuristics; White: white box software; Black: black box software; LS: local search; SA: simulated annealing; EA: evolutionary algorithms.
Multi-objective Metaheuristics

Multi-objective problem

Problem-dependent components
- Representation
- Evaluation
- Initialization
- Neighborhood
- Incremental evaluation
- Recombination
- Mutation

(shared by all metaheuristics)

Multiobjective-specific components
- Fitness assignment
- Diversity preservation
- Archiving

(shared by all multi-objective metaheuristics)
Implementation of an evolutionary algorithm

- Implement a representation
- Implement a population initialization strategy
- Implement a way of evaluating a solution
- Implement suitable variation operators
- Instantiate a fitness assignment strategy
- Instantiate a diversity preservation strategy
- Instantiate a selection strategy
- Instantiate a replacement strategy
- Instantiate an archive management strategy
- Instantiate a continuation strategy

- Problem-specific components
- Generic components
  - Multi-objective
  - Metaheuristic
Implementation

• Implement a representation
• Implement a population initialization strategy
• Implement a way of evaluating a solution
• Implement suitable variation operators
• Instantiate a fitness assignment strategy
• Instantiate a diversity preservation strategy
• Instantiate a selection strategy
• Instantiate a replacement strategy
• Instantiate an archive management strategy
• Instantiate a continuation strategy
Representation

- **Evolving object**
- **Multi-objective evolving object**
- **Vector-based representation**
  - Vector of bits
  - Vector of integers
  - Vector of real values
  - Real-coded obj. values

Diagram:
- **EO** (evolving object)
- **MOEO** (Multi-objective evolving object)
- **ObjectiveVector**
- **moeoVector**
- **moeoBitVector**
- **moeoIntVector**
- **moeoRealVector**
- **moeoRealObjectiveVector**
- **std::vector**

Trait:
- **ObjectiveVectorTraits**
- **ObjectiveVectorType**
Implementation

• Implement a representation
• Implement a population \textit{initialization} strategy
• Implement a way of evaluating a solution
• Implement suitable variation operators
• Instantiate a fitness assignment strategy
• Instantiate a diversity preservation strategy
• Instantiate a selection strategy
• Instantiate a replacement strategy
• Instantiate an archive management strategy
• Instantiate a continuation strategy
Implementation

- Implement a representation
- Implement a population initialization strategy
- **Implement a way of evaluating a solution**
- Implement suitable variation operators
- Instantiate a fitness assignment strategy
- Instantiate a diversity preservation strategy
- Instantiate a selection strategy
- Instantiate a replacement strategy
- Instantiate an archive management strategy
- Instantiate a continuation strategy
Implementation

• Implement a representation
• Implement a population initialization strategy
• Implement a way of evaluating a solution
• Implement suitable variation operators
• Instantiate a fitness assignment strategy
• Instantiate a diversity preservation strategy
• Instantiate a selection strategy
• Instantiate a replacement strategy
• Instantiate an archive management strategy
• Instantiate a continuation strategy
Variation operators

- variation operators must be embedded to an eoTransform object

- mutation
- binary recombination
- quadratic recombination
- other operators
Implementation

- Implement a representation
- Implement a population initialization strategy
- Implement a way of evaluating a solution
- Implement suitable variation operators
- Instantiate a **fitness** assignment strategy
- Instantiate a diversity preservation strategy
- Instantiate a selection strategy
- Instantiate a replacement strategy
- Instantiate an archive management strategy
- Instantiate a continuation strategy
Fitness Assignment

dummy

scalar approaches

moeoDummyFitnessAssignment

moeoScalarFitnessAssignment

moeoAggregationFitnessAssignment

moeoAchievementFitnessAssignment

indicator-based approaches

moeoIndicatorBasedFitnessAssignment

moeoBinaryIndicatorBasedFitnessAssignment

moeoExpBinaryIndicatorBasedFitnessAssignment

criterion-based approaches

moeoCriterionBasedFitnessAssignment

dominance-based approaches

moeoDominanceBasedFitnessAssignment

moeoDominanceRankFitnessAssignment

moeoDominanceCountFitnessAssignment

moeoDominanceDepthFitnessAssignment

moeoDominanceCountRankingFitnessAssignment

used in IBEA

used in MOGA

NSGA

NSGA-II

used in SPEA2
Implementation

• Implement a representation
• Implement a population initialization strategy
• Implement a way of evaluating a solution
• Implement suitable variation operators
• Instantiate a fitness assignment strategy
• Instantiate a diversity preservation strategy
• Instantiate a selection strategy
• Instantiate a replacement strategy
• Instantiate an archive management strategy
• Instantiate a continuation strategy
Diversity Assignment

dummy

- used in MOGA & NSGA
- used in SPEA2
- used in NSGA-II
Implementation

• Implement a representation
• Implement a population initialization strategy
• Implement a way of evaluating a solution
• Implement suitable variation operators
• Instantiate a fitness assignment strategy
• Instantiate a diversity preservation strategy
• Instantiate a selection strategy
• Instantiate a replacement strategy
• Instantiate an archive management strategy
• Instantiate a continuation strategy
Selection

deterministic tournament

stochastic tournament

random

elitist
Implementation

- Implement a representation
- Implement a population initialization strategy
- Implement a way of evaluating a solution
- Implement suitable variation operators
- Instantiate a fitness assignment strategy
- Instantiate a diversity preservation strategy
- Instantiate a selection strategy
- **Instantiate a replacement strategy**
- Instantiate an archive management strategy
- Instantiate a continuation strategy
Replacement

- one-shot elitist
- iterative elitist
- generational
Implementation

• Implement a representation
• Implement a population initialization strategy
• Implement a way of evaluating a solution
• Implement suitable variation operators
• Instantiate a fitness assignment strategy
• Instantiate a diversity preservation strategy
• Instantiate a selection strategy
• Instantiate a replacement strategy
• Instantiate an archive management strategy
• Instantiate a continuation strategy
Archive
Dominance Relation

- Pareto dominance
- Weak dominance
- Strict dominance
- \(\varepsilon\)-dominance
- g-dominance
Implementation

• Implement a representation
• Implement a population initialization strategy
• Implement a way of evaluating a solution
• Implement suitable variation operators
• Instantiate a fitness assignment strategy
• Instantiate a diversity preservation strategy
• Instantiate a selection strategy
• Instantiate a replacement strategy
• Instantiate an archive management strategy

• Instantiate a continuation strategy
Performance Metrics

⇒ Online computation

- moeoMetric
- moeoUnaryMetric
- moeoBinaryMetric
- moeoSolutionUnaryMetric
- moeoVectorUnaryMetric
- moeoSolutionBinaryMetric
- moeoVectorBinaryMetric
- moeoEntropyMetric
- moeoContributionMetric
- moeoVectorVsVectorBinaryMetric
- moeoVectorVsVectorAdditiveEpsilonBinaryMetric
- moeoVectorVsVectorMultiplicativeEpsilonBinaryMetric
- moeoSolutionMetric

- Hypervolume
- Hypervolume difference
- Entropy
- Contribution
- Additive & Multiplicative epsilon
General-Purpose EMO Algorithm
State-of-the-art EMO Algorithms

• To instantiate a state-of-the-art multi-objective metaheuristic for a novel continuous MOP

→ The evaluation is the only component to be implemented
ParadisEO users: some statistics  (Up to Jan 2010)
http://paradiseo.gforge.inria.fr

- More than 10 236 downloads: 91% Academics, 9% Industrials, # App
  238 users in the user-list
- France: ECL, EMN, UNSA, LITA, EMSE, ENSMA, ENSIMAG, ENSTA …
- International: Cornell, Sandia, Lincoln, Waterloo, NanChang, Rome, Magdeburg, Cardiff, Melbourne,
  Auckland, Shanghai, Hochiminh, Limerick, Cardiff, Cambridge, Tsinghua, Columbia, Ulster,
  Fortaleza, Koc, Clemson, NASA, Minho …
- Industry: Eurodecision, Thales, Into-Technology-USA, Software Dev. Service-New York, PiloSoft-USA,
  Mobinets, …

34 452 visits
20 257 visitors
144 countries
3012 cities

Tutorials, Contributions
Conclusion

• Uniform view of hybrid multi-objective metaheuristics

  - **Low-level**: Functional composition of a single method.
  - **High-level**: Different methods are self-contained.

  - **Relay**: Pipeline fashion.
  - **Teamwork**: Parallel cooperating agents.
Conclusion

- Unified view of parallel multi-objective metaheuristics

- **Algorithm-Level**: Cooperative self-contained metaheuristics: Problem independent
- **Iteration-Level**: Parallelization of a single step of the metaheuristic: Problem independent
- **Solution-Level**: Parallelization of the processing of a single solution: Problem dependent
Exercises: what has to be done (design & implementation ?

- From the *mono-objective* resolution to the *multi-objective* resolution

- From the application of NSGA-II to IBEA evolutionary algorithms

- From the application of NSGA-II evolutionary algorithm to *particle swarm optimization* MOPSO and multi-objective *scatter search*

- Design of *interactive* multi-objective metaheuristics

- Handling *many-objective* MOPs

- Design of multi-objective metaheuristics for MOP *with uncertainties*