MOEA/D

MOEA/D - Restart Strategy

Yuri Lavinas
Master Student - University of Tsukuba

System and Information Engineering

06/27/2018

What is MOP?

Multiobjective Optimization Problem have m multiple objective functions that must be optimized simultaneously.
Maximize[1] $F(x) = (f_1(x), f_2(x), ..., f_m(x))$,
subject to x in Ω.

- $F(x)$ objective functions;
- f_i is the i-th objective to be maximized;
- x is the decision vector;
- Ω is the decision space.

[1] All definitions are for maximization. Following inequalities should be reversed if the goal is to minimize.

Why is MOP interesting?

1. Many real-world scientific and engineering are MOP.
   - Water quality control, Groundwater pollution re-mediation, Design of marine vehicles, ... Coello et al. [2007].
   - Petrol extraction.
2. Hard problems: to balance the interests of the multi-objective as a whole is hard.

What is Pareto Set?

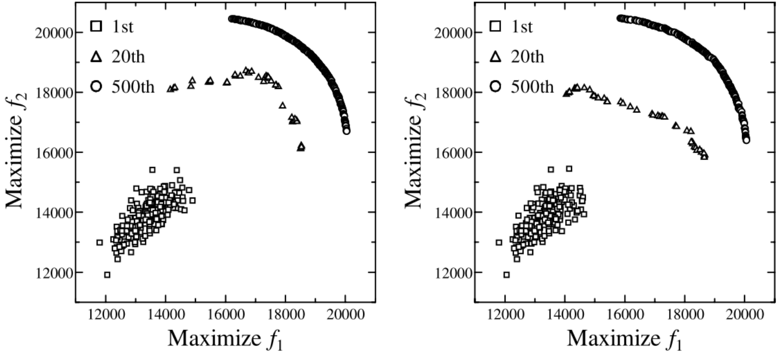
Objectives may be conflicting - The goal is to find good trade-off.

- Set of solutions.

Set of *optimum solutions* - Pareto set.

- Non-dominated solutions: no single solution provides a better trade-off in all objectives.

What is Pareto Set?



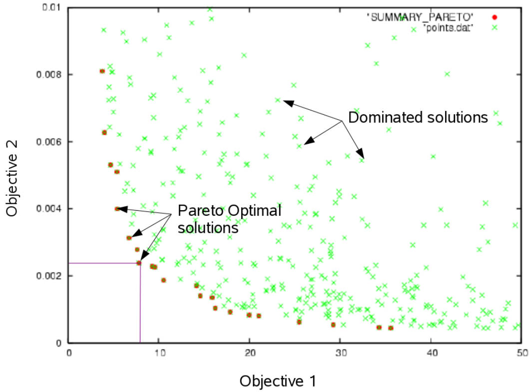
(a) The weighted sum. (b) The weighted Tchebycheff.

From Ishibuchi et al. [2009].

Non-dominated solutions

1. Let $u = (u_1, ..., u_m)$ and $v = (v_1, ..., v_m)$ vectors in Ω (the decision space).
   - $\forall i : u$ dominates v if $f_i(u) \leq f_i(v)$ and $\exists j : f_j(u) < f_j(v)$.
   - u dominates v, v is dominated by u, u is better that v.
2. A point x^* in Ω is called *Pareto Optimal* if no other point dominates x^*.

Pareto Front



From: http://www.cenaero.be/Page.asp?docid=27103&langue=EN

Pareto Front

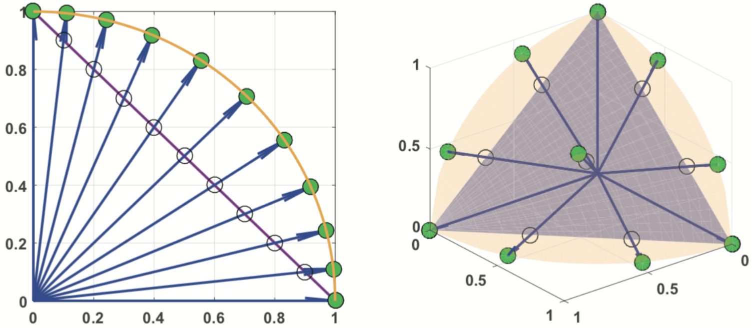
1. The set of all Pareto Optimal is called the *Pareto Set*.
   - $P^* = \{x \in \Omega : \nexists y \in \Omega$ and $F(y) \leq F(x)\}$
2. **Pareto Front** is the image of the Pareto Set in the objective space.
   - $\text{PF} = \{F(x) = (f_i(x), ..., f_m(x)) : x \in P^*\}$

Decompostion

MOEA/D represents a class of population-based meta-heuristics for solving Multi Objective Problems (MOPs).

- It is based on decomposition - one kind of scalarizing function
- One multi-objective problem becomes various single-objective sub-problems.
- A decomposition strategy generates weight vectors that defines the sub-problems.

Decomposition - 2 and 3 objective functions

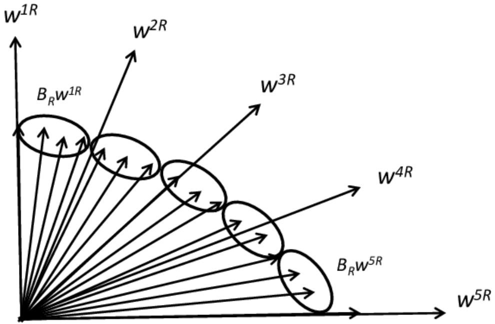


From: Chugh [2017].

Why use decomposition?

- It may be good at generating an even distribution of solutions in MOPs;
- It reduces the computation complexity when compared to other algorithms (NSGA-II) (at each generation), Zhang et al. [2009];
- Fitness assignment and diversity maintenance become easier to handle.

Decomposition + Aggregation Function



$f_3(x) = F * w_3$

In general, $f_i(x) = F * w_i$ Figure from: Chugh [2017].

Components of the MOEA/D

- Decomposition strategy: decomposes w/ weight vectors;
- Aggregation function: weight vector => single-objective sub-problems;
- Neighbourhood assignment strategy: Relationship between sub-problems;
- Variation Stack: New candidates solutions;
- Update Strategy: Maintain/discard candidate solutions;
- Constraint handling method: Constraint violation;
- Termination Criteria: when to stop the search.

Modifications Already Integrated

1. Cellular GA - proposed in the context of MOEA/D by Ishibuchi et al. [2009].
2. Latin Hypercube Sample - an alternative approach in initializing populations.
3. On-line Resource Allocation - proposed in the context of MOEA/D by Zhou and Zhang [2016].
4. Bet-and-Run: A kind of restart strategy - in the context of single-objective problems (SOP) by Friedrich et al. [2017]

Cellular GA and MOEA/D Ishibuchi et al. [2009]

- Why? MOEA/D can be viewed as a Cellular GA (cGA).
  - A cell can be seen as a specific "Neighbourhood assignment strategy", where each cell has its own weight vector.
- cGA is well explored in the context of SOP.

MOEA/D as cGA

- The main characteristic feature of MOEA/D as a cGA is the use of **local replacement** in addition to local selection.
  - Generated offspring for a cell is compared with not only the current solution of the cell but also its neighbours for possible replacement.
- Local replacement neighbourhood has greater effect on the performance than local selection neighbourhood.
  - Increasing its size tends to be better.

What is Latin Hypercube Sample

- Latin Hypercube Sample (LHS) was developed to generate a distribution of collections of parameter values from a multidimensional distribution, for more information see Stein [1987].

How it affects MOEA/D

- As defined in McKay et al. [1979], it could be a good method to use for selecting values of input variables.
- Therefore we expect that by using it, the initial population (ours input variable) would be better distributed along the search space.

What is Online Resource Allocation

- On-line Resource Allocation (ONRA) is an adaptation strategy that aim to adjust the behaviour of an algorithm in an on-line manner to suit the problem in question.

How it affects MOEA/D - Zhou and Zhang [2016].

- Some sub-problems can be more difficult to approximate that others. To better explore them, different computational resources are allocated to different sub-problems.
- The resources re-allocated is *the number of functions evaluations*.
  - From an equal amount to every sub-problem to an amount related to the difficulty of the sub-problem.

What is Restart Strategy

- Restart Strategy is a strategy used to avoid heavy-tailed running time distributions Gomes et al. [2000].
- If a execution of an algorithm does not conclude within a pre-determined limit or if the solution quality is unsatisfactory, we restart the algorithm Lissovoi et al. [2017].

Bet-and-Run framework

- It is defined in Fischetti and Monaci [2014]. as a number of short runs with randomized initial conditions, bet on the most promising run, and bring it to completion.
- To the best of our knowledge, only applied with EA in the context of SOP.

How it affects MOEA/D - Lissovoi et al. [2017].

- Initialisation can have a small beneficial effect even on very easy functions.
- Countermeasure when problems with promising and deceptive regions are encountered.
- Additional speed-up heuristic.

Unary Indicators

- Measure Pareto Sets independently.
- Power is restricted.
  - Cannot tell in general if a set is better than another.
- Focus on problem dependent and specifics.
  - Assumptions and knowledge should be specified.

1. Hyper-volume.
2. Error ratio.
3. Distance from reference set.

Binary Indicators

- Theoretically have no limitations.
- Analysis and presentation of results more difficult.

1. R1, R2, R3 indicators.
2. ε-Indicator.
3. Binary Hyper-volume.

Considerations

- Is complete - If, and only if $HV(A) > HV(B) \implies A$ is not worse than B.
- Is weakly compatible - $HV(A) > HV(B) \implies \not B$ dominates A.
- Assumptions - All points of a Pareto Set under consideration dominate the reference point.
  - Ishibuchi et al. [2018] proposed a method to specify the reference point from a viewpoint of fair performance comparison.

Considerations

- A large population size is **always** more beneficial than a small one.
- Measures both the convergence toward the Pareto Front and the diversity of non-dominated solutions.
- A monotonic increase of the hyper-volume over time cannot always be ensured.
    - For MOEA/D that is always true.

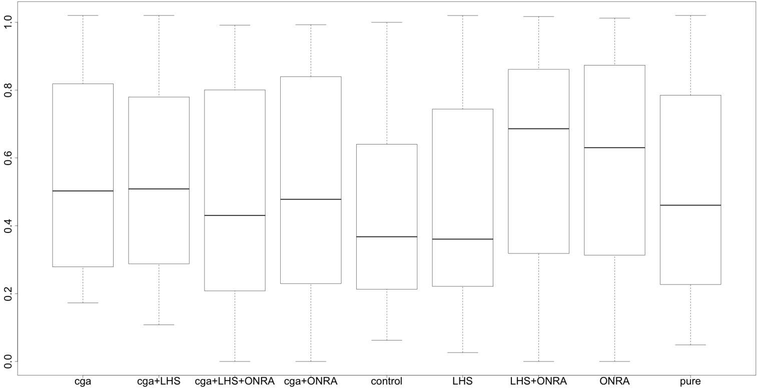
Considerations

- It compares 2 Pareto Sets.
  - It indicates which set is better and how much better
- If A is better than B $\implies I_\varepsilon(B, A) > 0$.
- If $I_\varepsilon(A, B) \leq 0$ and $I_\varepsilon(B, A) > 0 \implies A$ is better than B.

Experimental Design

1. Simple experiments - Check my understand and get insights.
2. DTLZ1, DTLZ2, DTLZ6 and DTLZ7 MOP benchmark functions - Available from the MOEADr package.
3. Every variation will be discussed based on the pilot data showed in the next slide by a box-plot figure.
4. Number of evaluations: 5 * 10 ˆ 4.
5. Based on the common variation: MOEA/D (variations 1 and 2 from MOEADr) and MOEA/D-DE.

Boxplot - HV


0.0
0.2
0.4
0.6
0.8
1.0
cga
cga+LHS
cga+LHS+ONRA
cga+ONRA
control
LHS
LHS+ONRA
ONRA
pure

cGA

1. MOEA/D as cGA has a high sensitivity on the parameters of local replacement and local selection.
2. Decreasing the size of competition neighbourhood increases. the non-dominated solutions, but degrades the search ability of the MOEA/D. **As already observed in Ishibuchi et al. [2009]** .

LHS

1. In some cases improves the results by a little.
2. It is cheap in terms of computational cost - it is only used once at each execution.

- This improve seems not to be significant.

ONRA

1. Computational costly -> more interactions than without it and we need to calculate the resources allocation every interaction.
2. It was beneficial in a few cases, while in others the overall quality decreased.
   - Considering the all functions and algorithms together it seems it leads to better results.

Bet-and-Run Strategy

1. Overall, this strategy combined with the MOEA/D lead to better final quality results.
2. Its performances become better when combined with other variations, specially with cGA+LHS.

Future works

1. cGA - On the fly parameter adaptation.
2. ONRA
    - Review my implementation and try the other methods proposed in Zhou and Zhang [2016]. Only the one considered to be the "best" was implemented.
3. Bet-and-Run strategy

- Use this strategy to add some adaptive technique.
- Use more instances based on the best one, instead of only one.
- Dynamic bet-and-run.
- Hierarchical bet-and-run - here it has only 2 phases.

Tinkle Chugh. Handling expensive multiobjective optimization problems with evolutionary algorithms. *Jyväskylä studies in computing 263.*, 2017.

Carlos A Coello Coello, Gary B Lamont, David A Van Veldhuizen, et al. *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer, 2007.

Matteo Fischetti and Michele Monaci. Exploiting erraticism in search. *Operations Research*, 62(1):114–122, 2014.

Tobias Friedrich, Timo Kötzing, and Markus Wagner. A generic bet-and-run strategy for speeding up stochastic local search. In *AAAI*, pages 801–807, 2017.

Carla P Gomes, Bart Selman, Nuno Crato, and Henry Kautz. Heavy-tailed phenomena in satisfiability and constraint satisfaction problems. *Journal of automated reasoning*, 24(1-2): 67–100, 2000.

Hisao Ishibuchi, Yuji Sakane, Noritaka Tsukamoto, and Yusuke Nojima. Adaptation of scalarizing functions in moea/d: An adaptive scalarizing function-based multiobjective evolutionary algorithm. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 438–452. Springer, 2009.

Hisao Ishibuchi, Ryo Imada, Yu Setoguchi, and Yusuke Nojima. How to specify a reference point in hypervolume calculation for fair performance comparison. *Evolutionary computation*, pages 1–29, 2018.

Andrei Lissovoi, Dirk Sudholt, Markus Wagner, and Christine Zarges. Theoretical results on bet-and-run as an initialisation strategy. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 857–864. ACM, 2017.

Michael D McKay, Richard J Beckman, and William J Conover. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

Michael Stein. Large sample properties of simulations using latin hypercube sampling. *Technometrics*, 29(2):143–151, 1987.

Qingfu Zhang, Wudong Liu, and Hui Li. The performance of a new version of moea/d on cec09 unconstrained mop test instances. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pages 203–208. IEEE, 2009.

Aimin Zhou and Qingfu Zhang. Are all the subproblems equally important? resource allocation in decomposition-based multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 20(1):52–64, 2016.

Table of Contents