,

Solving real-world multi-objective engineering optimization problems with an Election-Based Hyper-Heuristic

Vinicius Renan de Carvalho and Jaime Simão Sichman

OptMAS 2018, Stockholm, Sweden

July 14th, 2018

Intelligent Techniques Laboratory
Computer Engineering Department
University of São Paulo (USP)

UNIVERSIDADE DE SÃO PAULO

LTI
Laboratório de
Técnicas Inteligentes

Evolutionary algorithms

- Evolutionary algorithms are algorithms which employ Darwin's theory of the survival of the fittest as their inspiration.
- They keep a population of solutions and generate new solutions using crossover and mutation operators;
- They needs a fitness function specification which tells how good is a solution;
- They are used to solve problems when there is not any problem-specific algorithm that gives a satisfactory solution in reasonable time.

Evolutionary algorithms - objective

Algorithm 1: Generic Evolutionary Algorithm

```
begin
    Initialize the population with random solutions;
    Evaluate solutions according to the objective function;
    while a termination condition is not satisfied do
        Select parents;
        Recombine pairs of parents;
        Mutate the resulting offspring;
        Evaluate new solutions according to the objective function;
        Select solutions to compose the next generation;
    end
end
```

Evolutionary algorithms - objective

Evolutionary algorithms can be classified according to their number of objectives (number of fitness functions) as mono-objective and multi-objective algorithms.

- Mono-objective evolutionary algorithms:
  - Genetic Algorithm (GA) [4]
- Multi-objective evolutionary algorithms (MOEA):
  - Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [3]
  - Strength Pareto Evolutionary Algorithm 2 (SPEA2) [11]
  - Indicator-Based Evolutionary Algorithm (IBEA) [10]
  - Generalized Differential Evolution (GDE3) [7]

Quality Indicators in Multi-objective Optimization

There are several quality indicators to tell us how good algorithms outcomes are:

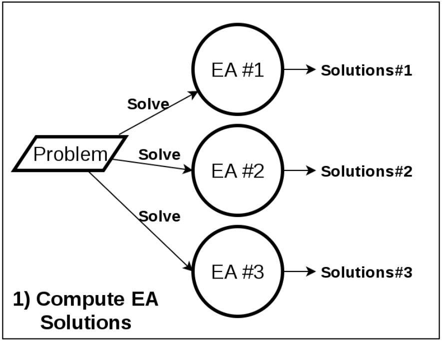
- Hypervolume;
- Ratio of non-dominated solutions;
- Hyper-area Ratio;
- Pareto Dominance Indicator;
- Uniform distribution of non-dominated population;
- Algorithm Effort;
- Epsilon;
- General Distance;
- Inverted General Distance;

Evolutionary Algorithms - How to choose one?

Choosing an evolutionary algorithm is not a trivial task. Different evolutionary algorithms produce different results when applied to different problems. Thus to choose an Evolutionary algorithm we have to:

- Use literature recommendations OR
- Perform a tuning and choose the best algorithm considering a quality indicator.

Evolutionary Algorithms - How to choose one?



3) Compare them to find the best one

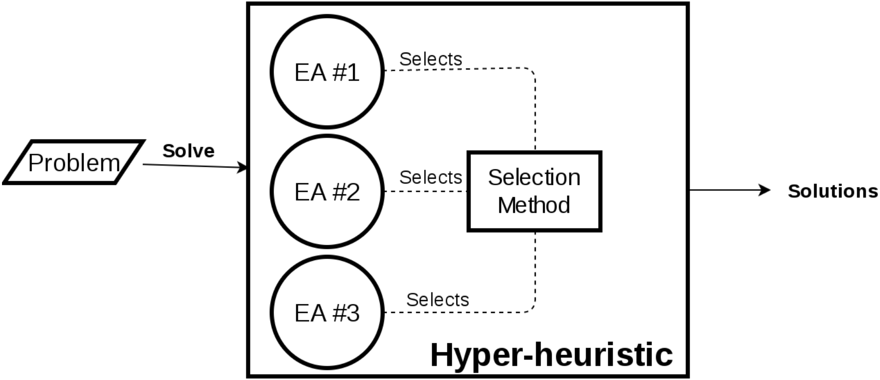
EA Results	Quality Indicator
Solutions#1	0.66
Solutions#2	0.33
Solutions#3	0.55

Evolutionary Algorithms - How to choose one?

Choosing an evolutionary algorithm is not a trivial task. Different evolutionary algorithms produce different results when applied to different problems. Thus to choose an Evolutionary algorithm we have to:

- Use literature recommendations OR
- Perform a tuning and choose the best algorithm considering a quality indicator OR
- Use a hyper-heuristic

Evolutionary algorithms - How to choose one?



Hyper-heuristic - Selection Method

Usually Hyper-heuristics employ a selection method. It can be:

- Roulette;
- A choice function;
- Multi-Armed Bandit approaches;
- Social Choice Based Approaches

MOABHH

We propose the Multi-Objective Agent-Based Hyper-Heuristic (MOABHH) [2] which has the following characteristics:

- Evolutionary algorithms (EA) as agents (*EA Agent*);
- Quality Indicators as agents (*Indicator Voters*);
- Share among *EA Agents* the number of solutions to generate;
- Allocate a bigger participation in generating new solutions to the top *EA Agents*;
- The top *EA Agents* are defined according to an election outcome, where *Indicator Voters* votes;
- We used Copeland voting method.

MOABHH - Population Sharing

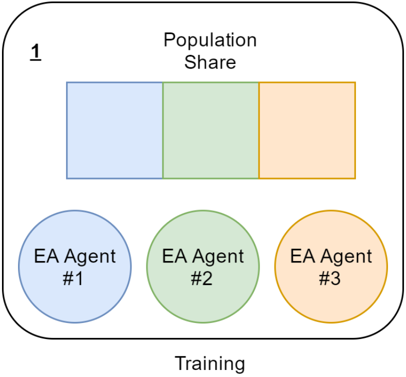


Figure 1: Population Sharing

MOABHH - Voting

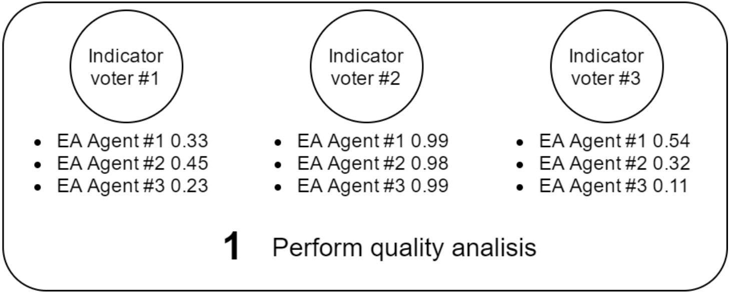


Figure 2: Voting method. First, all Indicator voter agents rank EA Agents based on their results.

MOABHH - Voting

2

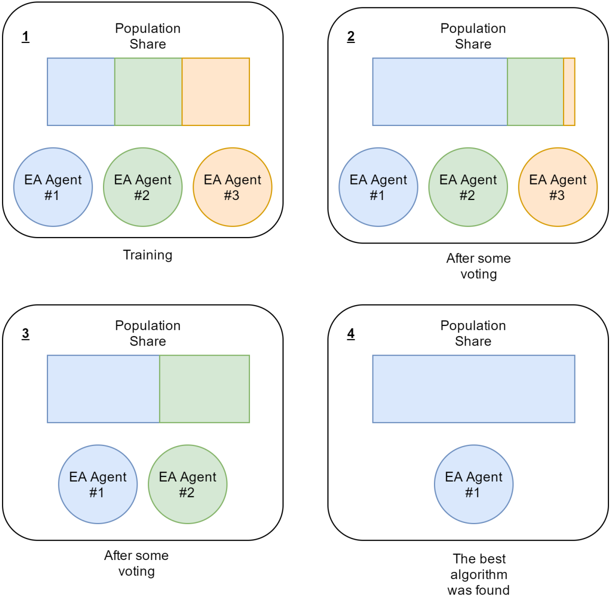
- One-on-one contest
- EA Agent#1 x EA Agent#2 on Voter #1
- EA Agent#1 x EA Agent#2 on Voter #2
- EA Agent#1 x EA Agent#2 on Voter #3
- EA Agent#1 x EA Agent#3 on Voter #1
- EA Agent#1 x EA Agent#3 on Voter #2
- EA Agent#1 x EA Agent#3 on Voter #3
- EA Agent#3 x EA Agent#2 on Voter #1
- EA Agent#3 x EA Agent#2 on Voter #2
- EA Agent#3 x EA Agent#2 on Voter #3

3

Candidate	Wins	Losses	Net	Final Rank
EA Agent#1	4	-1	3	1
EA Agent#2	3	-3	0	2
EA Agent#3	1	-4	-3	3

Figure 3: Voting method. In step 2 the Copeland voting is performed. In step 3 the Copeland ranking is generated.

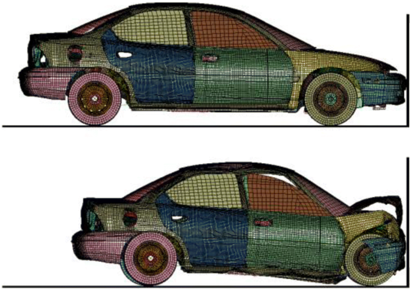
MOABHH - Population Sharing



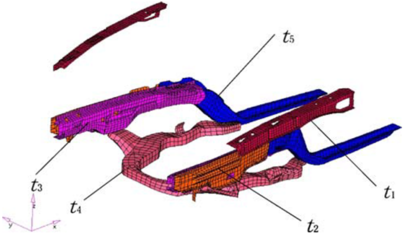
Motivations for a real-world application

- We just had performed our studies considering benchmarks:
  - WFG
  - DTLZ
  - ZDT
  - CEC09
- Pareto Front in known in advance for benchmarks. Thus we used IGD and GD as Indicator Voters;
- In real-world problems the Pareto Front is not know in advance;
- Other Indicator Voters have to be used;
- Real-world applications better propagate A.I. knowledge to other areas;

Applications: Crashworthiness



(a) A frontal crash



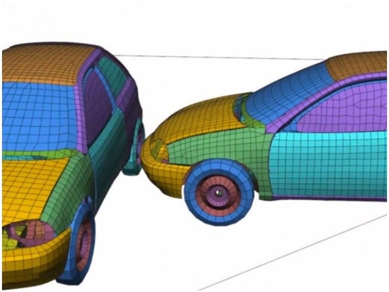
(b) Fender thickness decision variables

Figure 4: Liao et al. 2008

Problem Description:

- 3 objectives: (i) the mass, (ii) an integration of collision acceleration in the full frontal crash, (iii) the toe-board intrusion.
- 5 decision variables

Applications: Car Side Impact



Problem Description:

- 3 objectives: (i) the weight, (ii) the pubic force experienced by a passenger, (iii) the average velocity of the V-Pillar responsible for withstanding the impact load.
- 7 decision variables describing the thickness of B-Pillars, floor, cross members, door beam, roof rail, etc;
- 8 constraints.

Applications: Machining

Figure 5: A390 aluminum, widely used in automotive industry for cylinder liners and pistons etc., Source: http://www.alsi-alloys.com

Problem Description:

- 4 objectives: (i) min. the surface roughness, (ii) max. the surface integrity, (iii) max. the tool life, (iv) maximizing the metal removal rate.
- 3 decision variables Speed, feed and depth of cut;
- 3 constraints.

Applications: Water

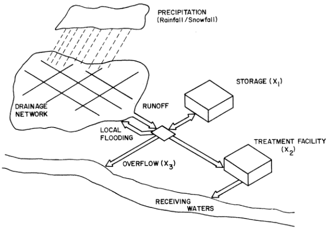


Figure 6: Musselman and Talavage, 1980

Problem Description:

- 5 objectives: (i) the drainage network cost, (ii) the storage facility cost, (iii) the treatment facility cost, (iv) the expected flood damage cost, and (v) the expected economic loss due to flood.
- 3 decision variables: storage capacity, the maximum treatment rate and the maximum allowable overflow rate;
- 7 constraints.

Experiments - Configuration

- 4 algorithms (candidates):
  - IBEA;
  - SPEA2;
  - NSGA-II;
  - GDE3.
- 6 Quality indicators (voters):
  - Hypervolume;
  - Ratio of non-dominated solutions;
  - Hyper-area Ratio;
  - Pareto Dominance Indicator;
  - Uniform distribution of non-dominated population;
  - Algorithm Effort.
- 40 independent runs.
- Kruskal-Wallis test with 1% of significance level.

Results

Table 1: Hypervolume, IGD and Epsilon Result Table

	Problem	GDE3	IBEA	NSGAII	SPEA2	MOABHH
Hyp.	Car Side Impact	4.4342E-01	**4.7710E-01**	3.7671E-01	4.4507E-01	**4.7161E-01**
	CrashWorthiness	**7.3603E-01**	7.0594E-01	6.6108E-01	7.2210E-01	**7.3985E-01**
	Water	5.6227E-01	5.0439E-01	4.3440E-01	4.9700E-01	**5.8632E-01**
	Machining	1.8393E-01	**2.7348E-01**	1.7288E-01	1.7705E-01	**2.7118E-01**
IGD	Car Side Impact	7.8878E-04	8.1957E-04	1.2878E-03	7.5318E-04	**6.6803E-04**
	Crash Worthiness	**6.9652E-04**	2.6247E-03	1.2639E-03	7.5822E-04	**4.2570E-04**
	Water	1.4869E-03	3.5247E-03	2.1495E-03	1.9045E-03	**8.9055E-04**
	Machining	1.6902E-03	**5.1369E-04**	1.6953E-03	1.7521E-03	**5.0530E-04**
Ep.	Car Side Impact	**1.6403E-01**	9.6482E-02	1.9015E-01	1.8226E-01	1.3509E-01
	Crash Worthiness	**5.3299E-02**	1.4667E-01	1.1723E-01	6.4985E-02	**4.3900E-02**
	Water	**1.4684E-01**	2.5247E-01	2.5750E-01	2.1015E-01	**1.1912E-01**
	Machining	4.8167E-01	**1.6378E-01**	4.9150E-01	5.0770E-01	**1.9654E-01**

Conclusions

- MOABHH was very competitive against the MOEAs;
- Most of the cases, it has found better Hypervolume, IGD and Epsilon averages;
- Sometimes with statistical difference;
- We believe that this makes this approach interesting for engineers to solve their real-world problems.

Future work

- Use different meta-heuristic, such as MOEA/D-DRA [9], MOEA/D-DD [8] and MOMBI-II [5];
- Use different voting methods, such as Kemeny [6] and Borda [1];
- Solve up to ten objectives problems.

Acknowledgments

CNPq

Thank you!

MOABHH - Pseudocode

Algorithm 2: MOABHH Pseudocode.

```
Input: Problem, MOABHH params
begin
    Initialize agents and artifacts;
    Generate a random population of solutions;
    while Training do
        Uniformly share the population among EA Agents;
        EA Agents execute for one generation;
        Update the main population;
    end
    while Executing do
        Evaluate EA Agents qualities;
        Perform the voting method;
        Share population among EA Agents according to voting results;
        EA Agents execute for γ generations;
        Update the main population;
    end
    return Main population
end
```

Where $\gamma = 12$

HH Assign

$$fx(pos, n) = \left\{ \begin{array}{l} 2^n \text{ if } pos = 1 \\ 0 \text{ if } pos = n \\ 2^{n-pos} \text{ otherwise} \end{array} \right\} \tag{1}$$

$$\forall pos \in rank \frac{fx(pos, n)}{\sum_{i=1}^{n} fx(i, n)} * \beta \tag{2}$$

Bibliography I

J. C. de Borda.
Mémoire sur les élections au scrutin.
Histoire de l'Académie Royale des Sciences, 1784.

V. R. de Carvalho and J. S. Sichman.
Applying Copeland Voting to Design an Agent-Based.
In *Proc. of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 972–980, 2017.

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan.
A fast and elitist multiobjective genetic algorithm: NSGA-II.
Evolutionary Computation, IEEE Transactions on, 6(2):182–197, Apr 2002.

Bibliography II

D. E. Goldberg and J. H. Holland.
Genetic algorithms and machine learning.
Machine learning, 3(2):95–99, 1988.

R. Hernández Gómez and C. A. Coello Coello.
Improved metaheuristic based on the r2 indicator for many-objective optimization.
In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO '15, pages 679–686, New York, NY, USA, 2015. ACM.

J. G. Kemeny.
Mathematics without numbers.
Daedalus, 88(4):577–591, 1959.

S. Kukkonen and J. Lampinen.
Gde3: The third evolution step of generalized differential evolution.
In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 1, pages 443–450. IEEE, 2005.

K. Li, K. Deb, Q. Zhang, and S. Kwong.
An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition.
IEEE Transactions on Evolutionary Computation, 19(5):694–716, oct 2015.

Bibliography IV

Q. Zhang, W. Liu, and H. Li.
The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances.
Technical Report CES-491, School of CS & EE, University of Essex, Feb 2009.

E. Zitzler and S. Künzli.
Indicator-based selection in multiobjective search.
In *PPSN*, volume 3242 of *Lecture Notes in Computer Science*, pages 832–842. Springer, 2004.

Bibliography V

E. Zitzler, M. Laumanns, and L. Thiele.
SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization.
In *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 95–100. International Center for Numerical Methods in Engineering, 2001.