UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

Department of Computer Science
University of Pretoria

Artificial Intelligence
COS 710

Assignment 3: Solving Multi-objective Optimisation with Evolutionary Computation-based Algorithms

Deadline: 4 May 2016, at 23h00

Do any one of the assignments listed below. Note that you have to submit a pdf document, containing a technical report wherein you describe what you have done, and present and discuss your results. **Your report must follow the IEEE conference style** (template) and the number of pages is limited to **12 pages**. You can find templates at: `http://www.ieee.org/conferences_events/conferences/publishing/templates.html`.

Remember that you cannot report results for only one run of an algorithm on a problem, but you have to report results as averages and standard deviations over a sufficient number of independent runs of the algorithms. I usually use 30 independent runs. When you want to determine if one approach is better than another, the outcome of such a comparison should be based on a formal statistical test, such as the Mann-Whitney U test. You also have to explain why you use specific parameter values and motivate your choice with appropriate references or results.

You can use any programming language to implement the algorithms. However, a Scala library (CIlib) is available that is maintained by the CIRG research group. You can download the source code at:
`https://github.com/cirg-up/cilib`.
In addition, jMetal is a Java library that already contains many multi-objective optimisation algorithms. You can access jMetal at:

http://jmetal.sourceforge.net.

1 Option 1: Constrained Multi-objective Optimisation

Multi-objective optimisation problems (MOOPs) are optimisation problems with more than one objective. Since there is no single optimum as is the case with a single-objective optimisation problem (SOOP), the goal of a multi-objective optimisation algorithm (MOA) is to find the optimal set of trade-off solutions referred to as the Pareto-optimal set (POS) in the decision variable space and the Pareto-optimal front (POF) in the objective space. The MOA has to find a set of solutions that are as close as possible to the POF and that contains a good spread of solutions on the POF.

This study investigates the performance of the following MOAs on a set of benchmark functions:

- A variant of the non-dominated sorting genetic algorithm II (NSGA-II) [1]. Here you can select a variant and have to motivate your choice. You can find some of the variants in the jMetal library or by reading the literature.
- An evolutionary strategy algorithm, referred to as Pareto archived evolution strategy (PAES) [2].
- A co-evolutionary algoithm, referred to as the multiobjective evolutionary algorithm based on decomposition (MOEA/D) [3].
- A cultural algorithm [4].

For each of the algorithms listed above, you have to motivate your choices for the parameter values with appropriate references.

Evaluate the performance of these MOAs on the MOO benchmark suite of the CEC2010 competition [5]. Remember to provide the reference for the selected functions and to discuss the functions' characteristics in your report. Aim to select functions with different characteristics.

Also discuss the performance measure(s) that you are using to analyse the performance of these algorithms. The following performance measures have to be used:

- Inverse generational distance (IGD) [6]
- Hypervolume [7, 8]

You can calculate the hypervolume according to [9] if you do not make use of the jMetal library. **Add your own measure to incorporate the constraint violation and discuss why you have selected that specific measure**.

To deal with constraints, incorporate the following two approaches in each algorithm (therefore you will have two versions of each algorithm):

Approach 1
Penalty function - refer to Section A.7 on page 563 of the textbook.

Approach 2
Modified version of Pareto-dominance, i.e. if you compare two solutions, $\mathbf{x}_1$ and $\mathbf{x}_2$:

- If the two solutions are non-dominated and both do not violate any constraints, both remain non-dominated.
- If the two solutions are non-dominated, $\mathbf{x}_2$ violates constraints and $\mathbf{x}_1$ does not violate any constraint, $\mathbf{x}_1$ dominates $\mathbf{x}_2$.
- If the two solutions are non-dominated and both violate constraints, the one that violates the least constraints dominates the other.
- If $\mathbf{x}_1$ dominates $\mathbf{x}_2$, but $\mathbf{x}_1$ violates constraints while $\mathbf{x}_2$ does not, $\mathbf{x}_2$ dominates $\mathbf{x}_1$.
- In all other cases, the normal Pareto-dominance relations are used.

2 Option 2: Scalability of Multi-objective Algorithms

MOOPs are optimisation problems with more than one objective, but normally less than four objectives. However, many-objective optimisation problems deal with problems that have more than four objectives. Similar to a MOA, the goal of a many-objective optimisation algorithm (MaOA) is to find the optimal set of trade-off solutions referred to as the POS in the decision variable space and the POF in the objective space. The MaOA has to find a set of solutions that are as close as possible to the POF and that contains a good spread of solutions on the POF. In addition, since there are more than three objectives, the number of dominated solutions increases as the number of objectives increases. Therefore, the MaOA has to use a different approach than a MOA when comparing the quality of two solutions.

This study investigates the performance of the following MaOAs on a set of benchmark functions:

- A variant of the NSGA-II [1]. Here you can select any variant of NSGA-II, but you have to motivate your choice. You can find some of the variants in the jMetal library or by reading the literature.
- An evolutionary strategy algorithm, referred to as PAES [2].
- A co-evolutionary algoithm, referred to as the MOEA/D [3].
- A cultural algorithm [4].

For each of the algorithms listed above, you have to motivate your choices for the parameter values with appropriate references. In addition, you have to adapt the way in which they guide their search as follows:

- Instead of using all of the objective functions when evaluating a solution, you should for each iteration randomly select 3 objectives and then evaluate all solutions using those three objectives for the parts of the algorithm that guides the search (e.g. selection for crossover and mutation).

- You will then apply the Pareto-dominance comparison taking into account only the selected three objectives.
- However, when doing either Pareto-ranking or adding solutions to the archive at the end of the run, the solutions are compared using all of the objective functions.

The following benchmark functions should be used in this study:

- DTLZ1, DTLZ2, DTLZ3 [10]
- WFG6, WFG7 [11]
- An adapted version of DTLZ2 [12] where each objective function i is scaled with 10^{i-1}

Each algorithm should be evaluated on each of the functions listed above for 4, 6, 8, and 10 objective functions. For more information, you can refer to [12].

Also discuss the performance measures that you are using to analyse the performance of these algorithms. The following performance measure have to be used:

- IGD [6]

Add also at least two other performance measures and motivate your choice. You can find other measures in either the jMetal library or in the literature.

3 Marking and General Guidelines

For this assignment you have to submit a research report where you discuss your findings appropriate to the option chosen above. This is not a course in technical and report writing; however, you should at least attempt to follow some accepted document writing techniques and to make your report as readable as possible. You are more likely to obtain a higher mark if your report generates a good impression with the marker and is void of general errors like spelling and grammar mistakes.

You are strongly advised to download some research papers from the CIRG website (`cirg.cs.up.ac.za`) to get a feel for how to write a report. The following is a general guideline of how to structure your report; however, strict adherence to this guideline is not a requirement for this assignment.

Abstract
The abstract is a summary of the study and the results that were obtained. After reading your abstract, the reader should know what your report is about and whether it is relevant to what s/he requires information on. The abstract should not be longer than 200 words.

1. Introduction
The introduction sets the stage for the remainder of your report. You usually have very general statements here. The introduction prepares the reader for what to expect from reading your report. In general, the introduction should either contain or be a summary of your ENTIRE report.

2. Background
A very high level discussion on the problem domain and the algorithms and/or approaches that you have used. Do not be too specific on the algorithms and approaches. This section is typically where the "base cases" of concepts that appear throughout the remainder of your report are discussed. It is also an ideal place to refer a reader to other sources containing relevant information on the topic but which is outside the scope of your assignment. It is the perfect place for pseudo code. Remember to discuss very generally. After reading this section the marker should be able to determine whether or not you know what you're talking about.

3. Implementation
In this section you discuss how you approached, implemented and solved your assignment choice. Mention, for example, the values set for the algorithm's control parameters, how many simulations you have run and what the characteristics for candidate solutions to your problems are. After reading this section (in addition to the background) the reader should be able to duplicate your experiments to obtain similar results to those obtained by you. This is also the section where your discussion specializes on the concepts mentioned in the background section. Be very specific in your discussions in this section.

4. Research Results
In this section you discuss how you approached, implemented and solved your assignment choice. Mention, for example, the values set for the algorithm's control parameters, how many simulations you have run and what the characteristics for candidate solutions to your problems are. After reading this section (in addition to the background) the reader should be able to duplicate your experiments to obtain similar results to those obtained by you. This is also the section where your discussion specializes on the concepts mentioned in the background section. Be very specific in your discussions in this section.

5. Conclusion(s)
Very general conclusions about the assignment that you have done. This section "answers" the questions and issues that you've raised and investigated. This section is, in general, a summary of what you have done, what the results where and finally what you concluded from these results. This is the final section in your document so be sure that all the issues raised up until now are answered here. This is also the perfect section to discuss what you have learnt in doing this assignment.

References

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, apr 2002.

[2] J. Knowles and D. Corne, "The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation," in *Proceedings of the Congress on Evolutionary Computation*, vol. 1, 1999, pp. 98–105.

[3] Q. Zhang, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[4] C. Coello Coello and R. Becerra, "Evolutionary multiobjective optimization using a cultural algorithm," in *Proceedings of the IEEE Swarm Intelligence Symposium*, Honolulu, Hawai, 1-5 April 2003, pp. 6–13.

[5] P. N. S. R. Mallipeddi, "Problem definitions and evaluation criteria for the cec 2010 competition on constrained real- parameter optimization," Nanyang Technological University, Singapore, Tech. Rep., 2010. [Online]. Available: http://www3.ntu.edu.sg/home/epnsugan/index_files/CEC10-Const/TR-April-2010.pdf

[6] D. van Veldhuizen, "Multiobjective evolutionary algorithms: classification, analyses, and new innovations," Ph.D. dissertation, Graduate School of Engineering Air University, 1999.

[7] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms âĂŤ a comparative case study," vol. 1498, pp. 292–301, 1998.

[8] ——, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, nov 1999.

[9] C. Fonseca, L. Paquete, and M. Lopez-Ibanez, "An improved dimension-sweep algorithm for the hypervolume indicator," in *Proceedings of Congress on Evolutionary Computation*, july 2006, pp. 1157–1163.

[10] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proceedings of Congress on Evolutionary Computation*, Honolulu, 12-17 May 2002, pp. 825–830.

[11] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, oct. 2006.

[12] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.