Projects - Metaheuristic Algorithms

2019-2020

There are three main categories of projects:

I. Research papers review
Aim: study a couple (2-5) of recently published papers (in well recognized journals in the field) related to a specific topic and prepare a review of the main results reported in the paper.
Content: the review should contain: (i) a description of the problem addressed in the paper(s); (ii) a clear presentation of the main contribution(s) of the paper(s); (iii) a discussion on the similarities/differences between the approaches presented in different papers or a critical review of the results; (iv) practical illustration of the results (if it is the case) - it might involve implementing some algorithms or using some already existing implementations; (v) a list of limits of the proposed solutions/methods (if it is the case) + list of open problems (providing ideas of how they might be addressed represents a plus).
Proposed topics (based on papers selected from IEEE Transactions on Evolutionary Computation - https://cis.ieee.org/publications/t-evolutionary-computation):

1. Ant Colony Optimization (ACO)
2. Covariance Matrix Adaptation - Evolution Strategy (CMA-ES)
3. Cooperative Coevolution
4. Estimation of Distribution Algorithms (EDA)
5. Evolutionary Neural Networks
6. Genetic Programming (GP)
7. Memetic Algorithms (MA)
8. Multimodal Optimization
9. Multiobjective Optimization
10. Parameter Tuning in Metaheuristic Algorithms
11. Particle Swarm Optimization (PSO)
12. Applications of Metaheuristic algorithms

Remark: the paper(s) available in the corresponding folder represent only the starting point - usually a couple of other papers mentioned in the list of references included in the paper should be also read.

II. Application oriented projects

1. **Nurse rostering.** This is a practical problem asking to find a schedule of nurses in a hospital. It is in fact a constraint satisfaction problem which can be solved by using different meta-heuristics (evolutionary algorithms (EA), estimation of distribution algorithms (EDA), ant colony optimization (ACO), variable neighborhood search (VNS), tabu search (TS), hyperheuristics etc).

   Requirement: Select a problem variant, identify and describe an adequate method (based on EA, EDA, ACO, VNS or TS), implement and test it.

   Variants: 5 (EA, EDA, ACO, VNS, TS)

   Biblio: applications/nurse_rostering

2. **Timetabling.** It is also a constraint satisfaction problem characterized by the existence of hard constraints (they should be satisfied) and soft constraints (they can be broken but the score of the timetabling becomes smaller). There are approaches based on simulated annealing (SA), evolutionary algorithms (EA), ant colony optimization (ACO), tabu search (TS), variable neighbourhood search (VNS), particle swarm optimization (PSO)

   Requirement: Select a problem variant, identify and describe an adequate method (based on SA, EA, ACO, TS, VNS or PSO), implement a variant and test it.

   Variants: 6 (SA, EA, ACO, TS, VNS or PSO)

   Biblio: applications/timetabling

   (e.g. SUMO - http://sumo.dlr.de/wiki/Main_Page).

3. **Regular Expressions Inference.** The regular expressions are useful to extract information from text. Automatic synthesis of regular expressions starting from examples can be done by genetic programming.

   Requirement: Analyze the implementation(s) available at https://github.com/MaLeLabTs/RegexGenerat and
   http://regex.inginf.units.it, adapt and test them for generating regular expressions which allow to extract text fragments (a.g. numerical values in a text).

   Biblio: applications/RegularExpressionsInference

4. **Community detection in networks.** The aim is to identify the groups of highly interconneted nodes in a network. The problem is similar to that of graph partitioning such that the nodes in a partition are highly connected while the nodes in different partitions are sparsely connected. It can be solved by using clustering techniques but also evolutionary algorithms.

   Requirement: Study aof the community detection problem, implementation of an EA and its testing for a test problem (e.g. Zachary's club problem).

   Biblio: applications/communities_detection

5. **Software testing.** The evolutionary algorithms and other nature inspire metaheuristics (e.g. ACO) can be used to generate test cases for software analysis.

   Requirement: Study of the applicability of evolutionary algorithm in software testing. Choose an evolutionary algorithm to generate test cases for a software product and implement it.

   Biblio. applications/software_testing

6. **Automated Program Repairing.** Automatic generation of patches used to remove bugs from software or to make an application more efficient can be done using genetic programming. The patches are usually small pieces of code which are generated starting from some templates and are evaluated using the behavior of the program for some testcases.

   Requirement: Comparative study of different approaches in automated patches generation. A Genetic Programming based example should be implemented.

   Biblio. applications/AutomatedProgramRepair

7. **Multilevel thresholding.** Image segmentation means identifying regions in the image in such a way that the pixels belonging to the same region are similar while those belonging to different regions are dissimilar. For gray level images, one of the simplest segmentation technique relies on threshold(s). The threshold(s) estimation can be formulated as an optimization problem, e.g. find the threshold values which maximize the inter-regions variance.

   Requirement: Implementing a metaheuristic algorithm (starting from one of the variants described in the papers provided in the bibliography) to estimate multiple thresholds + testing it on some test images (http://sipi.usc.edu/database/, http://www.imageprocessingplace.com/root_files_V3/image_databases.htm)

   Biblio: applications/ImageThresholding

8. **Image registration.** The aim of image registration is to align images of the same scene/object taken from different viewpoints or using different devices. The task to be solved is to transform a source image such that it matches a target image. If some corresponding points in the two images are known then by using them a geometric transformation can be identified. The evolutionary algorithms can be used to estimate the parameters of transformation models.

   Requirement: Choice and implementation of a metaheuristic (e.g. differential evolution, evolutionary algorithm or particle swarm optimization) for the estimation of the parameters of a transformation ensuring the registratiom of two medical or satellite images.

   Biblio: applications/ImageRegistration

9. **Data analysis and prediction.** Lately, in the context of major conferences in evolutionary computing (e.g. GECCO, CEC) have been organized competitions on industrial challenges, most of them addressing prediction problems.

   Requirement: Select an approach and illustrate its performance for other examples than those reported. Description of problems and solutions available at:

   - GECCO 2013 (http://www.spotseven.de/gecco-challenge/gecco-challenge-2013/)
   - GECCO 2014 (http://www.spotseven.de/gecco-challenge/gecco-challenge-2014/)
   - GECCO 2015 (http://www.spotseven.de/gecco-challenge/gecco-challenge-2015/)
   - GECCO 2016 - 2019 http://www.human-competitive.org/awards

III. Comparative studies of tools which can be used for the implementation of various metaheuristics. Examples of tools which can be compared:
Metaheuristics:

1. **ECJ.** http://cs.gmu.edu/ eclab/projects/ecj/
2. **JavaEVA.** http://www.ra.cs.uni-tuebingen.de/software/EvA2/

3. **JCLEC.** http://jclec.sourceforge.net/
4. **JMetal.**http://jmetal.github.io/jMetal/
5. **EO.** http://eodev.sourceforge.net/
6. **DEAP - Distributed Evolutionary Algorithms in Python** http://code.google.com/p/deap/
7. **pySTEP - Python Strongly Typed gEnetic Programming** http://pystep.sourceforge.net/
8. **jmp75** - a simplified metaheuristics framework for .NET. https://github.com/jmp75/metaheuristics
9. **OPT4J** - open source Java-based framework for evolutionary computation. http://opt4j.sourceforge.net/
10. **HeuristicLab** - a framework for heuristic and evolutionary algorithms. http://dev.heuristiclab.com/

Requirement. At least two tools should be selected, described and tested for a simple problem (similar with those used in the lab examples). The project will contain a comparative study of the selected tools. Teams of 2-3 students can be involved in such a project.

Structure of the projects of type II and III. Each project will consist of:

1. A report (6-12 pages) structured as follows:
   - abstract (5 lines): will describe the aim of the report and the main result/remark/idea
   - introduction: will contain a state of the art concerning the studied method/application; it is based on the study of the papers specified as references
   - detailed presentation of the models/algorithms/problems related to the subject
   - detailed presentation of the application + results
   - conclusions and open problems
   - references
2. A functional application illustrating the behaviour of the method(s) described in the report or using the software tools selected for comparison.

Remarks.

- The papers suggested for references can be dowloaded from http://staff.fmi.uvt.ro/~daniela.zaharie/ma2019/projects

  These papers should be used to start the study - other references could be used as well. The report should be based on at least two references from the provided list.

- The report should not contain paragraphs directly copied from other papers. The existence of such paragraphs is considered to be plagiarism and such a report cannot be accepted. In order to avoid plagiarism you should synthesize the main ideas using you own words.