COMPUTATIONAL INTELLIGENCE FOR OPTIMIZATION

2024 PROJECT GUIDELINES

Contact:

Berfin Sakallioglu, NOVA IMS

bsakallioglu@novaims.unl.pt

TIMELINE

27 April - Project Guidelines
17 May - Completion of Group and Topic Choices
The week of 20th May - Project Support Classes
The week of 27th May - Project Support Classes
3 June at 09h00 - Project Deadline
5 & 6 (maybe 7) June - Project Defenses

PROJECT DESCRIPTION

You are asked to apply your knowledge about Genetic Algorithms (GAs) to solve an optimization problem. You can choose one of the problems suggested in this document or propose your own project. If you choose to continue with your own project, the idea **must** be approved.

- All the communication will be done via email to: bsakallioglu@novaims.unl.pt.
- The group members and group topics will be chosen by the students filling the following sheet. The oral defense schedule will be published there as well: 2024_CIFO_PROJECTS
- Project delivery will be done via Moodle by only **one** member of the group.

Until the **17th of May**, you are expected to complete the sheet given in the link by choosing your group and topic. As stated before, if you will be working on your original idea, you must send an email and receive a confirmation before the 17th of May.

This is a group project, as such you are encouraged to do the project in a group. The maximum number of members for a group is 4. Groups of 3-4 people are encouraged, but not mandatory. If you are having trouble finding a group, please send an email. If you can reasonably justify that you are unable to do the project in a group, send an email and you can only proceed after receiving a prior authorization.

For any names not included in the sheet (even if you are doing the project alone) by the delivery deadline, it will be assumed that those students have opted to not partake in the project and will receive an automatic grade of 0.

NOTE: You can pass the course without undertaking the project, as it only counts for 35% of your grade. However, this is not encouraged as you will forfeit 35% of your grade. Please, remark that the grade of the project is the same in the first and second examination epoch, so between the two epochs you can only modify the 65% given by the final test.

PROJECT DEADLINE

The project deadline is the **3rd of June at 09:00 AM (09h00) on Monday morning**, that is in 37 days from the publication of this handout. Given the possibility of passing this course without undertaking the project, there is no ability to submit projects late, as any student who has not delivered theirs by the deadline will be considered as having opted to pursue solely the exam. Thereby, forfeiting 35% of their grade.

PROJECT DEFENSE

There will be a mandatory oral defense on the **5th and 6th of June (7th might be included)**. The exact time schedule for each group will be published closer to the date via the same sheet where the group lists are written. You will not be preparing any presentation for the defense. All the group members **must** join the defense and questions will be asked to each student about their project steps. These might include questions about the code implementations, experiments, or the results.

In situations where there are significant inconsistencies between the code, the report and/or the defense and/or plagiarism doubts arise, the defense duration will be extended with the involvement of additional professors.

PROJECTS

The projects suggested below are all expected to be solved using the Charles Genetic Algorithm Library developed throughout the semester. If you prefer to create your own library from scratch, you are free to do so. If you decide to undertake a more complex custom problem, you are free to use external GA libraries - although the library developed in class should be solid enough for any GA use-case.

Everyone is expected to demonstrate a good understanding of optimization problems and GAs. As a first step, think about how to represent such a problem in your code, and how to design a fitness function and appropriate genetic operators.

These are demonstrated by representing your problem well pragmatically, both as the **representation** itself as well as the definition of your **fitness** function. Keep in mind that the simpler the project the more implementations you are expected to do.

These are some basic steps expected from each project:

- Implement several selection methods
- Implement several mutation operators
- Implement several crossover operators

Illustrate and compare the performance of different approaches to your problem in your report. The report should focus **exclusively** on **your** implementation and the **results** you obtained, hence, have no literature review nor "Introduction to CIFO/ Genetic Algorithms" part. It should only focus on your project.

Try to illustrate decisions taken both in written form as well as using plots and illustrations. Your decisions should be based on statistical validation (i.e. run each selected configuration X times and compare the performance of your different configurations statistically).

VEHICLE ROUTING PROBLEM (VRP)

VRP is a generic name given to a whole class of problems concerning the optimal design of routes to be used by a fleet of vehicles to serve a set of customers. The goal is to locate optimal routes of some vehicles that begin from a depot and serve each customer one time and then return to the depot.

You will use GAs to evolve solutions for a VRP that you chose. This problem can be similar to Traveling Salesperson Problem (TSP) but there are many variants of VRP in literature (e.g. Capacitated VRP, Time-Dependent VRP, multi-objective etc.).

You are free to choose or decide which version to use. Your first step should be defining your problem. Please start with a quick review about the different variants and state your VRP assumptions at the beginning of your project (what are the restrictions, the rules, constraints). You can make your own rules or use already-existing cases. This repository could provide you with some datasets. But feel free to search for alternative sources including the datasets that are being used by example GA implementations.

IMAGE RECREATION

Your task is to recreate a target image using GAs. There are many approaches for this task and considering that we should keep the focus of the project in GAs, you are free to simplify your version of the problem. There are some examples with pixels, polygons and even in 2D. You are free to choose your target image and it can be black and white. The goal is to generate an image that is as close as to the target image.

SUDOKU SOLVER

Use GAs to evolve solutions to a Sudoku problem. Think about how to represent such a problem in your code, and how to design a fitness function and appropriate genetic operators. You should be able to implement Sudoku relatively easily and without using external libraries. How your Sudoku implementation looks visually is wholly irrelevant, this will not be considered for your grading process. You just want to be able to find good solutions - ideally solving Sudoku problems completely. You can compare your algorithm's performance on different difficulty levels and comment on those.

TIMETABLE / SCHEDULING PROBLEM

Utilize GAs to evolve solutions for solving a Timetable/Scheduling Problem. There are many examples of use cases: University Course Timetabling, School Classes Scheduling, Staff Scheduling (nurses at a hospital as an example), Exam Timetabling. You are free to create your own data with custom restrictions and definitions or you can find an already existing one (feel free to simplify any real-world data). You can have hard constraints and soft constraints for this task (i.e. constraints that must be met and the ones that are less crucial).

MORE IDEAS TO SEARCH FOR

There are several known optimization problems such as: Job Shop Problem, Stigler's Diet Problem. You can take a look at them.

You can invent your own combinatorial optimization problem which involves finding the best arrangement or combination of elements from a finite set (one example could be creating a seating chart for an event).

If you are feeling comfortable working with Neural Networks, you can expand your project to merge NNs and GAs. GAs can be used to optimize the weights of a NN (it can be a single-layer NN).

These examples can be extended, and original ideas are always encouraged, just make sure to receive a confirmation first.

RECOMMENDATIONS

You are recommended to use git versioning control as you develop the project, to avoid a sudden loss of all code on the day of delivery of the project. If you are unfamiliar with git, see: https://git-scm.com/book/en/v2 and https://youtu.be/2sjqTHE0zok.

Keep code simple and commented. Do not copy code from Stack Overflow and/or ChatGPT without knowing what it does! Obviously, take inspiration whenever you are stuck. But try to

understand how the code works and adapt solutions to your code. If you have parts of code that stand out from the rest of your code, and that particular part of the code is not commented – this will be investigated during the project defense.

If you have questions or require clarifications, use the practical lectures, or send an email.

PROJECT DELIVERABLES

- Code Implementation (link - can be a github link)
- Report (**YourGroupName**.pdf)

The code of your project should be delivered through a git repository. You may use any git service you like (GitHub, GitLab, etc.) as long as it can be **cloned**. All submitted projects will be cloned at exactly 09h01, the minute following delivery deadline, so no commits or changes made to the original repo after the deadline will be considered for evaluation.

Upload your report to Moodle before the deadline with the **link to the git repository on the first/cover page**. Do not forget to include your report in git as well. Do not forget to **check** if the link on the report is really working. If your repository is **private** and it is **not** possible to clone it, it will not be possible to grade your project and you will receive 0 points.

The report should be delivered as a PDF with your group name. Your report should not exceed 5 pages of text (except the cover page if you have one). In this module we trust you (or perhaps evolution?) enough to allow you to choose font and font size yourself*. Just keep it easy to read and relatively professional.

You will want to show what implementations you have done, and how the various implementations differ in terms of performance. Justify why and how you took decisions during your project.

Your report **must** mention the **division of labor** in your group.

*Reports delivered in font Comic Sans MS will receive 0 points.

A strong project will have implemented several selection approaches, several mutation operators, and several crossover operators. The operators must be coherent with the type of problem being addressed. These implementations will then be benchmarked and statistically compared. The code should be well formatted and commented. The report should be clear and easy to understand with useful illustrations. The following questions will be answered in the report:

Why did you choose the representation you did?

How did you design the fitness function? Did you try using different fitness functions, to see the impact on your GA?

Which configurations worked best together? How many did you try, and how did you determine the "best" one?

Do different operators affect the convergence of your GA?

Have you implemented elitism? Does the inclusion or exclusion of elitism impact your GA?

Do you get good results for the project you chose? What could be improved?

EVALUATION CRITERIA

• 35% Code Functionality: Assessment of the functionality of the code, the correct implementation of the GA techniques and interaction of the various components. While you will work on your projects in a group, you should ensure code is well-structured and works well in conjunction.

• 10% Code Structure: Assessment of the code quality and structure, the code should be clear and well-commented to be inspectable and understandable by anyone.

• 25% Report Content: Assessment of the contents of the report. The report should illustrate your project and not contain vast literature reviews (this is a report not an essay). A good report will justify choices undertaken in the project as well as demonstrate findings and results.

• 5% Report Structure: Assessment of the report structure, the report should be well organized, well written and grammatically correct. Proof-read your report well. An allowance for syntactical English errors is given.

• 25% Project Defense: Assessment of the project defense. All group members must join the defense and present knowledge about every aspect of the project.

*Keep in mind that we have no tolerance for plagiarism. In NOVA IMS, plagiarism is punished with the failure of both Epoch 1 and Epoch 2 of the exam (failure of the whole course).

FAQ

Can we do the project in 2, 3 or 4 people?

Yes.

Can I do the project alone or with 5 people?

Only if you have sent an email with valid reasonings and received a confirmation.

Hi Berfin, it's the day before the deadline and I forgot to tell you I was going to do the project on my own/I was going to implement my original idea/topic. Is that fine?

No, sorry. :(

My *insert excuse here*, can I get an extension on the project deadline?

See the answer above.

My *insert excuse here*, I lost all my code. What should I do?

Use a versioning control system from the start. See git.

Will we upload the code on Moodle?

No. You will only upload your report which will have the functioning link (it is crucial, please check it very well) for your git repository on the first page. And remember, only one member of the group will upload the report on Moodle.

Can I submit my project multiple times before the deadline?

Yes, I will not start grading or reviewing projects before the deadline. So, until then you can resubmit as many times as you want.

Can I change the name of the library?

Yes, if you do not want to use the name Charles (inspired by Charles Darwin) you can rename it to whatever you want.

Can the report be 6 pages?

If your visualizations are causing you to exceed pages, that is acceptable. You can also have an appendix for some extra visualizations. Just don’t exaggerate the page number, I trust you.