

My beautiful CI Project

Cristian Andres Camargo Giraldo CRISTIAN.ANDRES.CAMARGO@ESTUDIANATAT.UPC.EDU

Rodrigo Pablo Carranza Astrada RODRIGO.PABLO.CARRANZA@ESTUDIANATAT.UPC.EDU

Santiago del Rey Juárez SANTIAGO.DEL.REY@ESTUDIANATAT.UPC.EDU

Yazmina Zurita Martel YAZMINA.ZURITA@ESTUDIANATAT.UPC.EDU

Abstract

This is a great project and therefore it has a concise abstract.

Keywords: List of keywords

1. Problem statement and goals

One of the most popular problems studied in constrained combinatorial optimization is the knapsack problem. Given a finite set of objects with associated weights and values, the objective is to maximize the value of a collection formed by these items without exceeding a predefined weight limit.

The knapsack problem has diverse practical applications which makes it particularly interesting. For example, it has been applied to production and inventory management ?, financial models ? and queueing operations in computer systems ? or manufacturing ?.

There are several variations to the knapsack problem. We could consider there are multiple copies of each item or take into consideration their volume in addition to their weight. However, we will focus on the simplest case, the one dimensional 0-1 knapsack problem, where the only constraint is the weight and the number of copies of each item is limited to 1.

Although the premise might look simple on the surface, this is actually an NP-hard problem. Thus, there is no known algorithm that achieves optimal solutions in polynomial time for all cases of this problem. Still, a sufficiently good solution can be found quickly by resorting to heuristics methods. The one that we will explore is an effective and commonly used metaheuristic known as genetic algorithm.

Genetic algorithms (GAs) belong to the family of techniques known as evolutionary algorithms. They draw inspiration from natural selection and genetic processes to provide solutions to complex optimization problems and model evolutionary systems.

The main process can be described as an evolutionary cycle. They first initialise a population of chromosomes, which represent candidate solutions. Some of these individuals will be drawn from the population by means of a selection mechanism and compared according

to their fitness. In the reproduction phase, the genetic material of the best individuals will be combined to form offspring. Also, some values of the offspring's chromosomes will be mutated. Finally, a replacement strategy is set to generate the next generation population. This process is repeated for a number of generations or until some convergence criterion is met.

A solution to the knapsack problem can be represented as a sequence of binary values $\{p_1, p_2, \dots, p_m\}$, where m is the number of objects. A value $p_i = 1$ would mean that the solution contains the object p_i and $p_i = 0$ that it does not. We will apply GAs to eighteen different cases, each of them specified by a file containing the number of objects m and the capacity K of a knapsack in the first line and the value v_i and weight w_i of each object p_i in subsequent lines.

The goal of this project is to find the optimal solution to each of the eighteen knapsack problems proposed by applying GAs. To this end, we will have to choose and implement suitable mechanisms in each of the phases of the evolutionary cycle. The description of these methods, as well as the reasons for their election, are detailed in section [THE CI METHODS].

2. Previous work

This is a very important part, because it puts your work in context.

3. The CI methods

Do not repeat well-known theory or formulas. Just mention which methods you use and why you choose them, and provide relevant citations.

4. Results and Discussion

The main part of the document.

5. Strengths and weaknesses

Be critic with your work ...

6. Conclusions and future work

The conclusions are not a mere repetition of the abstract. Basically, you should describe “what you know now that you did *not* before doing the work”. In addition, mention what would be natural follow-up lines of work.

References

Appendix A. Proof of theoretical results

(if applicable)

Appendix B. Implementation details

(if applicable)