As part of the optimization project, which aims to develop an educational platform to solve the Bin-Packing problem, the last part of the latter aims to propose an efficient hybridization schema by combining the approximate methods.

Hybridization is the action of combining or cooperating two or more optimization methods, the aim of this cooperation is to provide results superior to the two combined methods.

In order to offer our hybridization schemes, we need an evolutionary method as well as heuristic

The genetic algorithm(GA) is inspired by the natural selection process which belongs to the largest class of evolutionary algorithms. The basic idea is to try to imitate a simple image of natural selection, in order to find a good algorithm. GAs are very efficient in their principle of research and improvement. In addition, they are not limited by hypotheses on the field of exploration of the research space and the problem. In addition, like any other evolutionary method, GAs are less likely to be trapped in local minima because they manipulate in parallel a set of possible solutions to a given problem. They use a set of genetic operators (crossing, mutation and replacement). These algorithms use a vocabulary similar to that of genetics and biology.

Heuristic algorithms are very widely used to tackle practical problems in operations research, because so many are NP-hard [1] and exhaustive search is often computationally intractable.

[1] Michael R. Garey and David S. Johnson. Computers and Intractability: a Guide to the Theory of NP-Completeness. Freeman, 1979.

Literature Review:

The development of heuristics has led to various algorithms tailored to specific problems, yet no single heuristic provides high-quality results across all problem instances. Certain problems exhibit features that make particular heuristics effective, but these features might not be present in other problems, reducing the heuristic’s performance. Hyper-heuristics have emerged as a research focus, aiming to create more general algorithms capable of efficiently solving various problems. Ochoa et al. (2012) introduced HyFlex, a software framework designed to develop cross-domain search methodologies. HyFlex offers a common interface for handling different combinatorial problems and provides problem-specific algorithm components, acting as a benchmark for developing and comparing the generality of selection hyper-heuristics. This framework has been used to test algorithms across domains such as maximum satisfiability, bin packing, flow shop scheduling, personnel scheduling, traveling salesman, and vehicle routing.

HyFlex has inspired numerous studies, such as Burke et al. (2010b), which compared various hyper-heuristics combining heuristic selection and acceptance approaches. Burke et al. (2011) provided further extensions, and Burke et al. (2012) proposed a general packing methodology for 1D, 2D, and 3D bin packing and knapsack packing. They developed a genetic programming system to generate high-quality heuristics for different problems, albeit with a significant computational cost per instance. The CHeSC 2011 competition, using HyFlex, was won by Misir et al. (2011) with an algorithm that intelligently selected and paired heuristics and adapted their parameters online. Misir et al. (2013) later extended this work by focusing on single heuristic sets and experimental limits. Kalender et al. (2013) introduced a selection hyper-heuristic combining simulated annealing-based move acceptance with a learning heuristic selection algorithm.

HyFlex and similar systems typically employ a selection hyper-heuristic approach that perturbs complete candidate solutions to improve quality, involving limited search. In contrast, Sim and Hart (2013) discussed a selection hyper-heuristic approach constructing solutions incrementally, which requires significant search effort to create the solution-builder but results in a lower cost for generating solutions to unseen problems compared to HyFlex methods. This incremental approach has also been used for solving constraint satisfaction problems (Terashima-Marín et al., 2008).

The literature highlights various heuristic-selection mechanisms. For instance, Cowling et al. (2000) used a choice function based on single and paired heuristics performance, and Burke et al. (2006) employed case-based reasoning for timetabling problems. Bai et al. (2012) proposed a learning approach that updates heuristic selection weights based on performance after each learning period. Walker et al. (2012) utilized HyFlex for vehicle routing problems using adaptive multiple neighborhood iterated local search algorithms.

Much of the research on combinatorial optimization problems focuses on developing methods to produce the best solutions for benchmark problem instances rather than generalized solutions. Hyper-heuristics aim to provide general solutions by exploring a heuristic space comprising low-level heuristics, which can be either constructive (building solutions) or perturbative (improving initial solutions). For example, in the one-dimensional bin-packing problem, construction heuristics include first-fit, best-fit, next-fit, and worst-fit. Hyper-heuristics may use methodologies like variable neighborhood search, tabu search, and genetic programming to select or generate low-level heuristics for different problem domains.Nelishia Pillay,

Despite its importance, the two-dimensional irregular bin-packing problem (2DIBPP) has received less attention in the literature compared to other bin-packing problems. Hyper-heuristic methods, particularly those based on genetic algorithms, have been applied to both regular and irregular 2D bin-packing problems. Terashima-Marín et al. (2010) developed heuristics for selecting and placing pieces in bins, and Lopez-Camacho et al. (2013, 2014) extended one-dimensional heuristics to two-dimensional cases, achieving good results on convex polygon instances. Other studies have explored heuristic approaches allowing piece rotation, such as Martinez-Sykora et al. (2017), Abeysooriya et al. (2018), Liu et al. (2020), and Zhang et al. (2022).

Evolutionary algorithms (EAs) are widely used in operations research due to their effectiveness in searching large spaces, although they often lack performance guarantees and may appear as black-box algorithms. To address acceptability issues, EAs can be used to generate solution processes applicable to many problem instances, built from simple heuristics. For instance, in large exam timetabling problems, a system might switch heuristics based on problem characteristics. An example is using a modern learning classifier system, XCS, to learn rules associating problem states with specific heuristics, successfully solving a large set of one-dimensional bin-packing problems.

In conclusion, hyper-heuristics represent a significant step towards creating generalized solution processes for a variety of combinatorial optimization problems. They offer a way to construct and select heuristics dynamically, adapting to different problem instances, and thus provide a versatile approach to solving complex problems efficiently.