UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Engineering, Built Environment and IT
Department of Computer Science

COS314

Assignment 2

Due 28 April 2024

1. **Instructions**

    1. A zipped folder containing the data for this assignment is attached.
    2. Only Java or C++ may be used to complete this assignment.
    3. The programs must be executable (JAR) and be able to run without linking to libraries via the IDE (in the case of C++). Please note the programs will not be run in IDEs but as a piece of commercial software (marks may be lost if this is not complied to).
    4. Read-me instructions are to be included.
    5. **NB: The report will not be marked without submission of the code.**
    6. Submission is through ClickUP no email submissions will be allowed.
    7. For the purpose of evaluation all code must be seeded (Listing 1) and run by initially requesting the seed value.

2. **Background**
The Knapsack problem is a popular optimization challenge. It involves filling a knapsack with a collection of items, each with a weight and value, while keeping the total weight of the items in the knapsack within a given limit and maximizing the total value of the selected items. The problem can be defined as selecting a subset of $\mathbf{n}$ items with weight $\mathbf{w}_i$ and value $\mathbf{v}_i$ that fit into a knapsack of capacity $\mathbf{W}$. The Knapsack problem is categorized as NP-hard, which implies that there is no polynomial-time algorithm that can solve it optimally for all scenarios. However, there are various efficient algorithms and meta-heuristics that can provide acceptable approximate solutions. The Knapsack problem has numerous practical applications, such as in logistics, finance, scheduling, and resource allocation.

3. **Assignment Question- 30Marks**

The purpose of this assignment is to compare the effectiveness of applying a meta-heuristic and its hybrid to solve instances of this problem, namely, a Genetic Algorithm and a GA + Local Search. For the provided problem instances, a GA and a GA + Local Search algorithm are to be developed to solve the given instances. The results are to be presented in the format of the table provided below.

The report should include the following

1. GA configuration description (7 marks code + report)
2. GA + local search, configuration description. (8 marks code + report)
3. A description of the local search and justification of its selection. (2 marks)
4. Experimental setup.(including table of parameters for both programs) (3 marks)

5. A table (exemplified below) presenting the results. (4 marks correctnes of output + report)
6. Statistical analysis of differences in performance need to be presented. Specifically a one-tailed z-test (5% level) is used. The null hypothesis is that the means are equivalent.(2 marks)
7. A critical analysis of the results.(4 marks)

Please note with respect to 1 and 2 the initial configuration values are usually obtained from literature and then used as starting points to perform parameter tuning. These sources need to be referenced and a justification of the final values used should be presented. A zipped folder containing the problem instances and the known optimums(not to be used in your implementation) accompanies this file.

Table 1: Comparison of GA and GA+local on 10 knapsack problem instances

Problem Instance	**Algorithm**	**Seed Value**	**Best Solution**	**Known Optimum**	**Runtime (seconds)**
f1_ld_kp_10_269	GA−LS	xxx	xxx	xxx	xxx
	GA	xxx	xxx	xxx	xxx
f2_l_d_kp_20_878	GA−LS	xxx	xxx	xxx	xxx
	GA	xxx	xxx	xxx	xxx
Instance X	GA−LS	xxx	xxx	xxx	xxx
	GA	xxx	xxx	xxx	xxx
f10_l_d_kp_20_879	GA−LS	xxx	xxx	xxx	xxx
	GA	xxx	xxx	xxx	xxx

Listing 1: A seeding Java code example

```
1 import java.util.Random;
2
3 long seed = System.currentTimeMillis();
4 Random rand = new Random(seed);
5 System.out.println("Seed␣Value" + seed);
```