**Intelligent Systems**

**Exercise 9. Metaheuristics**

# Exercise description

The objective of this exercise is to apply the concepts and methods for Solving Problems with Metaheuristics. You must use a pseudo random number generator to produce each required number during the execution of the methods.

**Team members**

Write the student id, name, and campus of each member in a different line.

1:

2:

3:

**Problems to Solve**

**Problem 1**

The **Multiple Knapsack Problem (MKP)** is a real world subset problem and it has multiple applications in theory, as well as in practice. It also arises as a subproblem within several algorithms for more complex problems and these algorithms will further benefit from any improvement in the field of MKP. The following major applications can be mentioned as possible formulations of MKP: problems in cargo loading, cutting stock, bin-packing, budget control and financial management.

MKP can be thought as a resource allocation problem, where there are m resources (knapsacks) and n objects and every object j has a profit pj. Each resource has its own budget ci (knapsack capacity) and consumption wj of resource i by object j. The aim is maximizing the sum of the profits, while working within a limited budget.

Consider the instance of MKP defined by

n = 6;

m = 2;

(Pj) = (110, 150, 70, 80, 30, 5);

(Wj) = (40, 60, 30, 40, 20, 5);

(Ci) = (65, 85).

Examples:

(S1 : {5, 3}, S2: {1, 6}), profit = 95

(S1 : {2}, S2: {3, 4}), profit = 130

**Problem 2**

This is a **Simple Arithmetic Problem (SAP)** where the algorithm finds three numbers that add up to a target value.

Consider the instance of the SAP defined by

* Find three operands that add up to 20.
* The operands can have a value between -19 and 19.

Examples:

15 + -3 + 2 = 14

1 + 6 + 11 = 18

1. **Metaheuristic:** Ant System (AS)
   1. Run only two (2) complete cycles of searching for an Ant System trying to solve the MKP instance. A cycle ends when each ant cannot add more items to its knapsacks and a new level of pheromone of each arc of the graph is calculated.
   2. Simulate an AS with m = 5 ants (not necessarily every ant should start in a different node) and show the selected value for each parameter of the method.
   3. You mush show each required pseudorandom number, each performed calculation, and the effect of each step of the method.
2. **Metaheuristic:** Bees Algorithm (AB)
   1. Run two (2) complete cycles of a Bees Algorithm trying to solve the SAP instance. A cycle ends when the remaining bees for random search (scout bees) are assigned.
   2. Simulate a BA with n = 7 bees and show the selected value for each remaining parameter of the method.
   3. You must show each required pseudorandom number, each performed calculation, and the effect of each step of the method.
3. **Metaheuristic:** Particle Swarm Optimization (PSO)
   1. Run two (2) complete cycles of a Particle Swarm Optimization system trying to solve the SAP instance. A cycle ends when each particle for random search has its new velocity and position computed.
   2. Simulate a PSO with n = 7 particles with circular neighborhoods of size 3 and a maximum velocity (VMAX) of 5.
   3. You must show each required pseudorandom number, each performed calculation, and the effect of each step of the method.