Algorithmic Methods for Mathematical Models

– COURSE PROJECT –

You work as a salesperson in a jewelry firm. The most important trade fair in the sector is taking place soon in a far away country, and you have to prepare for the event. There are n products that you would be interested in showing in the fair. However, due to several restrictions it is not possible to take them all. Since your firm has paid the cheapest airfare for the flight, first of all there is a strict limit on the dimensions of the luggage. Fortunately you have a suitcase that fits exactly. The suitcase has a rectangular shape with height x and width y, measured in millimetres. Products come in square boxes, which, because of the fragility of the merchandise, cannot be put in the suitcase one on top of the other. Also, to avoid that products move inside uncontrollably, there is a system of inner rails that require that boxes are placed with their sides parallel to the sides of the suitcase. Apart from the geometrical constraints, the total weight of the suitcase cannot exceed c grams. Finally, you think it would be convenient to take with you the most expensive and luxurious products, as you expect these could attract potential customers. How should you choose the products so as to maximize the accumulated price, while respecting the travelling constraints?

For the i-th product ($1 \leq i \leq n$), you have the following data:

- p_i: its price, in euros.
- w_i: its weight, in grams.
- s_i: the side of its (square) box, in millimetres.

All numbers in the input data are integer.

For example, let us consider the following data: $n = 5$, $x = 5$, $y = 7$, $c = 5$, $p = (4, 3, 1, 3, 2)$, $w = (3, 2, 1, 2, 1)$ and $s = (4, 4, 2, 2, 2)$. Let us also label the 1st product with letter A, the 2nd product with letter B, etc. In this case, an optimal choice are products B, D and E, with accumulated price $3 + 3 + 2 = 8$ and total weight $2 + 2 + 1 = 5$. A way of placing these products in the suitcase can be as follows:

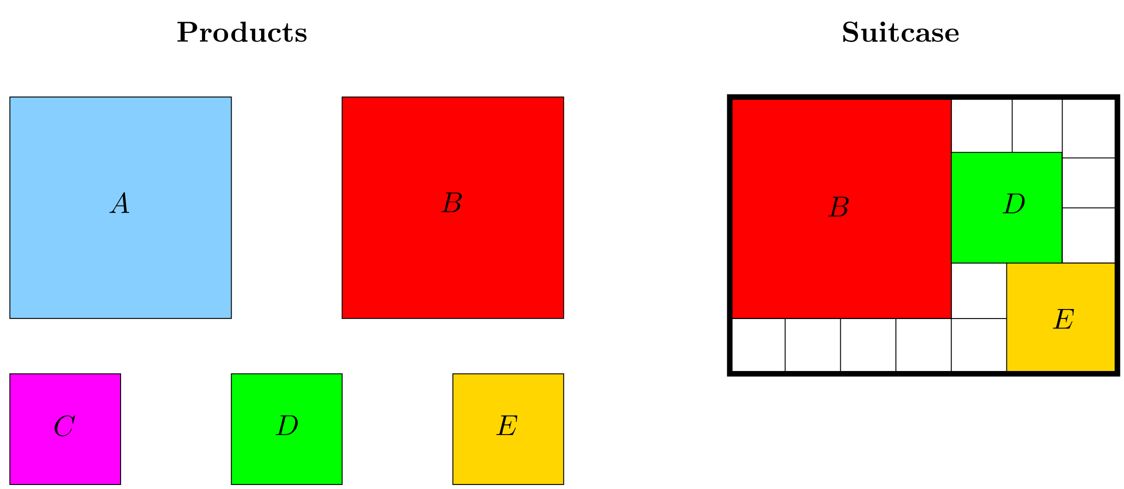

Figure 1: Example of how the products can be placed in the suitcase.

1. **Work to be done:**

   (a) State the problem formally. Specify the inputs and the outputs, as well as any auxiliary sets of indices that you may need, and the objective function.

   (b) Build an integer linear programming model for the problem and implement it in OPL.

   (c) Because of the complexity of the problem, heuristic algorithms can also be applied. Here we will consider the following:

      i. a greedy constructive algorithm,

      ii. a greedy constructive + a local search procedure,

      iii. GRASP as a meta-heuristic algorithm. You can reuse the local search procedure that you developed in the previous step.

      Design the three algorithms and implement them in the programming language you prefer.

   (d) Tuning of parameters and instance generation:

      Given an instance of input to the problem, the value of n is the *size* of the instance.

      i. Tune the α of the GRASP constructive phase with instances of large enough size.

      ii. Generate problem instances with increasingly larger size. Solving each instance with CPLEX should take from 1 to 30 min.

   (e) Compare the performance of CPLEX with the heuristic algorithms, both in terms of computation time and of quality of the solutions as a function of the size of the instances.

   (f) Prepare a report and a presentation of your work on the project.

2. **Report:**

   Prepare a report (8-10 pages) in PDF format including:

   - The formal problem statement.
   - The integer linear programming model, with a definition and a short description of the variables, the objective function and the constraints. Do not include OPL code in the document, but rather their mathematical formulation.
   - For the meta-heuristics, the pseudo-code of your constructive, local search, and GRASP algorithms, including equations for describing the greedy cost function(s) and the RCL.
   - Tables or graphs with the tuning of parameters, and with the comparative results.

   You should also give all your instances and sources (OPL code, programs of the meta-heuristics, instance generator) and instructions for using them, so that results can be reproduced.

3. **Presentation:**

   You are expected to make a presentation of your work at the end of the course (at most 10 minutes long; overtime presentations will be terminated). All members of the group must participate in the presentation and know all the work in the project. The slots of 28/05/24 and 31/05/24 will be devoted to these presentations. The schedule will be announced in its due time. The slides of the presentation in PDF format should be delivered together with the report by **26/05/24**.

   The slides can contain figures, plots, equations, algorithms, etc. with a very short text that helps to understand them. It is expected that you give a full explanation of those contents during your presentation. On the other hand, the report should contain that explanation in a well-organized manner as a text.