PROJECT 1 - 2023/2024

Optimization of a Purchase Order in a Marketplace

1 Introduction

This case study focuses on a prescriptive algorithm intended to help buyers in a marketplace to order all the items they want to buy at a minimal total cost. A marketplace is an online platform where anyone can sell their products or services. The marketplace in this case study is a marketplace for buying and selling LEGO® parts, minifigures and sets, both new and used.

The aim of the project is to develop a decision support system based on a Metaheuristic, a Mixed-Integer Programming Model or a Matheuristics[1], to solve this purchasing order decision problem.

2 Description of the Problem

An online marketplace for buying and selling LEGO® bricks wants to develop a purchase decision support system for its customers. The Marketplace has several sellers who offer a large number of references (SKUs) of LEGO® bricks.

Each SKU (i) is characterised by its type of part, item number (N_i), name, colour (L_i), volume (V_i) and weight (P_i). Each seller (j) is characterised by their nationality (N_j), point of sale location (S_j) and the SKUs (i) they have in inventory in quantity (I_{ji}) at a unit price (C_{ji}).

The purchasing process for the customers consists of having a list of SKUs (i) with the respective quantities to be purchased (Q_i) and choosing what to order from each seller.

Some sellers require a minimum purchase value of parts per order (min_j) and a minimum average value per SKU ($racio_j$).

The shipping cost is added to the parts cost for each order placed with a vendor. The shipping costs are based on the total weight of the parts in an order, according to the weight ranges shown in Table 1. Some vendors offer free shipping on orders over a certain amount ($free_j$). A fixed cost (T) resulting from bank commissions is added in each international order.

[1] Heuristic algorithms made by the interoperation of metaheuristics and mathematic programming (MP) techniques.

As the number of SKUs is over 100,000, the number of vendors is over 15,000 and each vendor does not have available all the SKUs of the catalogue, it is very difficult to determine the order list with the minimum total cost.

Table 1 – Shipping weight ranges

Weight Range [g]
0 - 50
50 - 100
100 - 250
250 - 500
500 - 1000
1000 - 2000

The objective is to develop a decision support system that allows a customer, for a given list of parts, to determine which vendors, SKUs and quantities to purchase from each vendor in order to ensure that all the parts on the list are purchased at the lowest total cost.

3 Data

Several data files are available to develop the solution. For each file type, Table 2 shows the respective data fields.

Table 2 – Data fields in files

File	Fields
categories	Category ID; Category Name
colors	Color ID; Color Name; RGB; Type; Parts; In Sets; Wanted; For Sale; Year From; Year To
itemtypes	Item Type ID; Item Type Name
euroexchangerate	currency; rate
parts	Category ID; Category Name; Number; Name; Weight (in Grams); Dimensions
vendors	store; country; minmoeda; minvalor; free; racio
itemcostvendor	Item No; Color ID; Description; Status; Stock; store; unitcurrency; unitprice
ShippingRates	Ranges x Countries
instances	Type; Item No; Item Name; Qty; Color ID; Extra?; Alternate?; Match ID; Counterpart?

A file example with the **output file format** it also available and is the format that must be used by the groups.

The problem data files are available in the Moodle page.

4 Instances Benchmark results

Four instances are provided for validation and evaluation of the solution proposed. Table 3 presents the main features of the instances provided, as well as the benchmark results.

Table 3 – Instances main features and best-known results

Instance			Best Known Results			
			#Suppliers		Total Parts	
Name	# SKUs	Total Qty	International	National	Cost [€]	Time [s]
S-40469-1	70	135	0	2	10.97	6
S-40517-1	61	125	1	1	12.74	4
S-9500-1	165	740	3	0	175.91	12
S-10325-1	385	1500	2	0	164.94	21

5 Rules and Deliverables

5.1 Project

The assignment is mandatory and must be solved and reported on by a working group, consisting of a maximum of 2 students.

The proposed solution should be coded using Python as programming language.

The problem is to be solved using one of the following approaches:

1. Metaheuristics
    a. GRASP
    b. Tabu Search
    c. Simulating Annealing
    d. Genetic Algorithm.
2. MIP model
3. Matheuristics

Each approach has different **milestones**:

1. Metaheuristics
    - First phase: implement and validate a constructive heuristic. The deliveries must include the pseudo-code, the code, and the validation process output.
    - Second phase: local search. The deliveries must include the pseudo-code, the code, and the solutions for all given instances using the xxx.xls format.
2. MIP model
    - First phase: Development of the MIP model. The deliveries must include the MIP model formulation.
    - Second phase: Implementation. The deliveries must include the pseudo-code, the code, and the solutions for all given instances using the xxx.xls format.

3. Matheuristics
    - First phase: Development of the MIP model. The deliveries must include the framework and a MIP model formulation.
    - Second phase: Implementation. The deliveries must include the pseudo-code, the code, and the solutions for all given instances using the xxx.xls format.

5.2 deliverables

The **deliverables** of the project are:

- The report of the solution proposal, that should include:
    a) the description and explanation of the proposed solution
    b) a performance analysis of the proposed solution
        i. two comparisons are required against the benchmark results of Table 3 – Instances main features and best-known results: one without time limit and another with a time limit of 10 min.
- The solution implementation (algorithms/applications)
- The presentation

5.3 Grading and key dates

The grade given to each student has two main components; the first results from work submitted by the group and the second from the presentation and discussion of the work. The result of the second component is individual. The submission dates of the deliverables and the weight in the evaluation of the first component of the work are indicated in Table 4.

Table 4 - Submission dates and grade weights

Deliverable	Due Date	Weight (%)
Solution and Report		40
First Phase	January 24, 2024	
Second Fase	February 5, 2024	
Presentation and Discussion	February 7-9, 2024	60

No submissions are accepted beyond the dates indicated in the table. The presentation and discussion of the work are scheduled for the week of 5-9, February, at a time to be agreed with each group.

5.4 Presentation format

The work developed by each group will be presented and discussed after its conclusion.

The presentation has a maximum duration of 10 minutes and has as main objective the demonstration of the functionalities of the developed solution. The presentation is followed by an up to 20 min discussion.

At the beginning of the presentation a new instance will be provided to the group to be run during the presentation. The time limit of the solution must be set for 10 minutes.

5.5 Files for submission

The Report of the solution proposal and the project Solution and Presentation files must be submitted by one of the members of the group via Moodle, on the UC page. If more than one file is to be submitted in each deliverable, a single zip-file containing all files must be created for submission. The name of the submitted files must follow the format indicated in Table 5.

Table 5 - File name format

Deliverable	File Format
Report	Group xx – Report_x_Phase
Solution	Group xx – Solution_x_Phase

A template for the report of the solution proposal is available in UC page in Moodle.