Université Libre de Bruxelles

Heuristic optimisation INFO-H413

Combinatorial optimization project
Assortment Planning Problem (AP)

Students :
Achten Alexandre 494484
Boistel Julien 440915

Professors :
Chicoisne Renaud
Aguayo Cristian

June 2024
Academic year 2023-2024

0 Introduction

This report focuses on the Assortment Planning Problem (AP). It can be formulated intuitively as follows: "A retailer wants to determine products to offer to its customers in order to maximise its expected revenue". More precisely, consider a set of products $\mathcal{I} = \{1, ..., n\}$ that the retailer can offer to his customers, and selling product i will give him a net revenue of $r_i > 0$. We assume that the products are sorted in decreasing order of revenue, i.e.

$$r_1 > r_2 > ... > r_n > 0 \tag{0.1}$$

and that each product i is bought according to a certain probability $P_i(S)$, which depends on:

1. The mean utilities $(\mu_j)_{j\in\mathcal{I}}$, for the customers when they buy product $j \in \mathcal{I}$,
2. The set of alternatives S made available to the customers.

These probabilities come from a discrete choice model called multinomial logit, and can be written as follows:

$$P_i(S) = \frac{e^{\mu_i}}{e^{\mu_0} + \sum_{j\in S} e^{\mu_j}} = \frac{e^{\mu_i}}{1 + \sum_{j\in S} e^{\mu_j}}, \quad \forall i \in \mathcal{I} \cup \{0\}$$

Where $\mu_0 = 0$ represents the utility - for the customers - of buying nothing. It is convenient to assume that selling nothing does come with a revenue $r_0 \geq 0$ (usually zero). The problem can be posed as the following combinatorial optimization problem :

$$\text{(AP)} \quad \max_{S\subseteq\mathcal{I}} \left\{ r_0 \cdot P_0(S) + \sum_{i\in S} r_i \cdot P_i(S) \right\} \tag{0.2}$$

In this report, different formulations of the problem are established. The properties are demonstrated and the different algorithms for solving the problem are explored and compared.

1 From Combinatorial Optimization to Linear Programming

Definition of variables
$\mathcal{I} = \{1, ..., n\}$: the set of all products
$i \in \mathcal{I}$: a product
r_i : revenue of product i
$S \subseteq \mathcal{I}$: set of choice for the customer
$\mu_j, j \in \mathcal{I}$: mean utilities when customer buy product $j \in \mathcal{I}$

The problem can be written in his (AP) form that's shown at Equation 1.1 and 0.2.

$$\text{(AP)} \quad \max_{S\subseteq I} \left\{ r_0 \cdot P_0(S) + \sum_{i\in S} r_i \cdot P_i(S) \right\} \tag{1.1}$$

with

$$P_i(S) = \frac{e^{\mu_i}}{e^{\mu_0} + \sum_{j\in S} e^{\mu_j}} = \frac{e^{\mu_i}}{1 + \sum_{j\in S} e^{\mu_j}}, \quad \forall i \in \mathcal{I} \cup \{0\}$$

It is possible to formulate the (AP) problem in a (AP-IP) problem by defining x_i that is the decision variable to decide if the product $i \in S$.

$$x_i = \begin{cases} 1 & \text{if product i} \in S \\ 0 & \text{else} \end{cases} \tag{1.2}$$

$$\Leftrightarrow \quad \max_{S\subseteq I} \left\{ r_0 \cdot \frac{1}{1+\sum_{j\in S} e^{\mu_j}} + \sum_{i\in S} r_i \cdot \frac{e^{\mu_i}}{1+\sum_{j\in S} e^{\mu_j}} \right\} \tag{1.3}$$

$$\Leftrightarrow \quad \max_{S\subseteq I} \left\{ \frac{r_0 + \sum_{i\in S} r_i e^{\mu_i}}{1+\sum_{j\in S} e^{\mu_j}} \right\}$$

$$\text{(AP-IP)} \quad \max_{x\in\{0,1\}^n} \left\{ \frac{r_0 + \sum_{i=1}^n x_i r_i e^{\mu_i}}{1+\sum_{i=1}^n x_i e^{\mu_i}} \right\} \tag{1.4}$$

The (AP-IP) formulation in Equation 1.4 is non-linear due to the structure of it's objective function. The numerator and denominator are both linear combinations and the ratio between two linear expressions is a non-linear expression.This non-linearity arises because changes in any x_i affect both the numerator and the denominator.

Now let's consider the new following change of variables:

$$\begin{cases} y_0 = \frac{1}{1+\sum_{j=1}^n x_j e^{\mu_j}} \\ y_i = \frac{x_i e^{\mu_i}}{1+\sum_{j=1}^n x_j e^{\mu_j}}, \quad \forall i \in I \end{cases} \tag{1.5}$$

By injecting them into Equation 1.3, the (AP-IPL) formulation can be found:

$$\begin{aligned} \text{(AP-IPL)} \quad \max_{y,y_0\geq 0} \quad & r_0 y_0 + \sum_{i=1}^n r_i y_i \\ \text{s.t.} \quad & y_0 + \sum_{i=1}^n y_i = 1, \\ & y_i \in \{0, y_0 e^{\mu_i}\}, \quad \forall i \in \mathcal{I} \end{aligned} \tag{1.6}$$

If we consider the continuous relaxation of Equation 1.6, the (AP-L) formulation can be obtained and is presented at Equation 1.7, with $y_0, y \in \mathbb{R}$.

$$\begin{aligned} \text{(AP-L)} \quad \max_{y,y_0\geq 0} \quad & r_0 y_0 + \sum_{i=1}^n r_i y_i \\ \text{s.t.} \quad & y_0 + \sum_{i=1}^n y_i = 1, \\ & y_i \leq y_0 e^{\mu_i}, \quad \forall i \in \mathcal{I} \end{aligned} \tag{1.7}$$

The dual linear programming of Equation 1.7 can be found by identifying the different terms present in the definition of a primal and dual programming presented here below.

Primal		**Dual**
Max $C^t x$		min $b^t \pi$
s.t. $Ax = b$,	$\Leftrightarrow$	s.t. $A^t \leq C$,
$x \geq 0$		$\pi \geq 0$

The following variables can be identified :

$$\begin{gathered} A = y_0 + \sum_{i=1}^{n} y_i \\ b = 1 \\ C = r_0 + \sum_{i=1}^{n} \end{gathered}$$

Thus the dual program (AP-LD) can be written in the form presented at Equation 1.8.

$$\begin{aligned} \text{(AP-LD)} \quad & \min_{\pi \geq 0, \pi_0 \in \mathbb{R}} \quad \pi_0 \\ & \text{s.t.} \quad \pi_0 - \sum_{i=1}^{n} \pi_i e^{\mu_i} \geq r_0, \\ & \qquad \pi_0 + \pi_i \geq r_i, \quad \forall i \in \mathcal{I} \end{aligned} \tag{1.8}$$

The relationship between the primal variables and the dual constraints is that it's the transposed matrix. The same reasoning applies to the dual variables and the primal constraints. This is shown in the definition of a primal and dual programme.

2 Ideal Formulation and a Greedy Algorithm

2.1 Primal

(a) For any $k \in \mathcal{I}$, the following solution is considered :

$$\begin{aligned} y_0^k &= \frac{1}{1 + \sum_{j=1}^{k} e^{\mu_j}} \\ y_i^k &= \begin{cases} \frac{e^{\mu_i}}{1+\sum_{j=1}^{k} e^{\mu_j}}, & \text{if } i \leq k \\ 0, & \text{otherwise} \end{cases} \qquad \forall i \in \mathcal{I} \end{aligned} \tag{2.1}$$

We need to check that (y_i^k, y_0^k) satisfies the constraints of (AP-L) by injecting them into the constraints.

$\begin{cases} y_0^k \geq 0 \\ y_i^k \geq 0 \end{cases}$ by definition.

1. The first constraint becomes :
   $y_0 + \sum_{i=1}^{n} y_i = 1$
   $\Leftrightarrow \frac{1}{1+\sum_{j=1}^{k} e^{\mu_j}} + \sum_{i=1}^{k} \frac{e^{\mu_i}}{1+\sum_{j=1}^{k} e^{\mu_j}} = \frac{1+\sum_{j=1}^{k} e^{\mu_j}}{1+\sum_{j=1}^{k} e^{\mu_j}} = 1$
   which is always true.

2. For the second constraint, two cases can be considered :
   $y_i \leq y_0 e^{\mu_i}$
   - For $i \leq k$, the inequality is saturated:
     $\Leftrightarrow \quad y_i^k = \frac{e^{\mu_i}}{1+\sum_{j=1}^k e^{\mu_j}} \leq \frac{1}{1+\sum_{j=1}^k e^{\mu_j}} \cdot e^{\mu_i} = y_0^k e^{\mu_i}$
   - For $i > k$:
     $\Leftrightarrow \quad y_i^k = 0 \leq \frac{e^{\mu_i}}{1+\sum_{j=1}^k e^{\mu_j}} = y_0^k e^{\mu_i}$

By injecting the values of (y_i^k, y_0^k) into the objective value of Equation 1.7, a new objective value is obtained :

$$\max_{k \in \{0,n\}} \quad \frac{r_0 + \sum_{i=1}^k r_i e^{\mu_i}}{1 + \sum_{i=1}^k e^{\mu_i}} \tag{2.2}$$

(b) Using the variable change in Equation 1.5, we can prove that the associated x^k is an integer.

$$\begin{aligned} \frac{e^{\mu_i}}{1+\sum_{i=1}^k e^{\mu_i}} x_i &= \begin{cases} \frac{e^{\mu_i}}{1+\sum_{i=1}^k e^{\mu_i}}, & \text{if } i \leq k \\ 0, & \text{otherwise} \end{cases} \\ &\Leftrightarrow \begin{cases} x_i = 1, & \text{if } i \leq k \\ x_i = 0, & \text{otherwise} \end{cases} \end{aligned} \tag{2.3}$$

We deduce that $x_i \in \{0, 1\} \subset \mathbb{Z}$.

(c) Let's suppose that the following inequation is true:

$$\begin{aligned} \frac{r_0 + \sum_{i=1}^k e^{\mu_i}}{1 + \sum_{i=1}^k e^{\mu_i}} &> \frac{r_0 + \sum_{i=1}^{k-1} e^{\mu_i}}{1 + \sum_{i=1}^{k-1} e^{\mu_i}} \\ \Leftrightarrow \quad \frac{r_0 + \sum_{i=1}^{k-1} e^{\mu_i} + r_k e^{\mu_k}}{1 + \sum_{i=1}^{k-1} e^{\mu_i} + e^{\mu_k}} &> \frac{r_0 + \sum_{i=1}^{k-1} e^{\mu_i}}{1 + \sum_{i=1}^{k-1} e^{\mu_i}} \end{aligned} \tag{2.4}$$

Let's pose :

$$\begin{cases} A = r_0 + \sum_{i=1}^{k-1} r_i e^{\mu_i} \\ B = 1 + \sum_{i=1}^{k-1} e^{\mu_i} \end{cases}$$

By injecting A and B in 2.4, we obtain:

$$\frac{A + r_k e^{\mu_k}}{B + e^{\mu_k}} > \frac{A}{B}$$

$$\Leftrightarrow \quad AB + r_k B e^{\mu_k} > AB + A e^{\mu_k}$$

$$\Leftrightarrow \quad r_k > \frac{A}{B}$$

$$\Leftrightarrow \quad r_k > \frac{r_0 + \sum_{i=1}^{k-1} r_i e^{\mu_i}}{1 + \sum_{i=1}^{k-1} e^{\mu_i}}$$

This is the expected expression.

(d) We want to prove that there is no value of k such that the Equation 2.5 is validated.

$$\begin{cases} r_k \leq \frac{r_0+\sum_{i=1}^{k-1} r_i e^{\mu_i}}{1+\sum_{i=1}^{k-1} e^{\mu_i}} \\ r_{k+1} > \frac{r_0+\sum_{i=1}^{k} r_i e^{\mu_i}}{1+\sum_{i=1}^{k} e^{\mu_i}} \end{cases} \tag{2.5}$$

Let's take the second equation and work on it:

$$r_{k+1} > \frac{r_0 + \sum_{i=1}^{k-1} r_i e^{\mu_i} + r_k e^{\mu_k}}{1 + e^{\mu_k} + \sum_{i=1}^{k-1} e^{\mu_i}}$$

$$\Leftrightarrow \quad r_{k+1} > \frac{\frac{r_0+\sum_{i=1}^{k-1} r_i e^{\mu_i}}{1+\sum_{i=1}^{k-1} e^{\mu_i}}(1 + \sum_{i=1}^{k-1} e^{\mu_i}) + r_k e^{\mu_k}}{1 + e^{\mu_k} + \sum_{i=1}^{k-1} e^{\mu_i}} \tag{2.6}$$

It is known that $r_k \leq \frac{r_0+\sum_{i=1}^{k-1} r_i e^{\mu_i}}{1+\sum_{j=1}^{k-1} e^{\mu_i}}$ (2.5) so let's inject this expression in 2.6:

$$\Leftrightarrow \quad r_{k+1} > \frac{\frac{r_0+\sum_{i=1}^{k-1} r_i e^{\mu_i}}{1+\sum_{j=1}^{k-1} e^{\mu_i}}(1 + \sum_{j=1}^{k-1} e^{\mu_i}) + \frac{r_0+\sum_{i=1}^{k-1} r_i e^{\mu_i}}{1+\sum_{j=1}^{k-1} e^{\mu_i}} e^{\mu_k}}{1 + e^{\mu_k} + \sum_{j=1}^{k-1} e^{\mu_i}}$$

$$\Leftrightarrow \quad r_{k+1} > \frac{r_0 + \sum_{i=1}^{k-1} r_i e^{\mu_i}}{1 + \sum_{j=1}^{k-1} e^{\mu_i}} \cdot \frac{1 + e^{\mu_k} + \sum_{j=1}^{k-1} e^{\mu_i}}{1 + e^{\mu_k} + \sum_{j=1}^{k-1} e^{\mu_i}}$$

$$\Leftrightarrow \quad r_{k+1} > \frac{r_0 + \sum_{i=1}^{k-1} r_i e^{\mu_i}}{1 + \sum_{j=1}^{k-1} e^{\mu_i}}$$

However, the Equation 2.5 tells us that r_k is smaller than this expression, thus:

$$\Leftrightarrow \quad r_{k+1} > r_k$$

This contradict the hypothesis 0.1. So there is no value of k validating the Equation 2.5.

(e) Let's prove that there is only one k^* such that:

$$r_k > \frac{r_0 + \sum_{i=1}^{k-1} r_i e^{\mu_i}}{1 + \sum_{j=1}^{k-1} e^{\mu_i}} \quad \text{and} \quad \frac{r_0 + \sum_{i=1}^{k} r_i e^{\mu_i}}{1 + \sum_{j=1}^{k} e^{\mu_i}} \geqslant r_{k+1} \tag{2.7}$$

Let's define the threshold function:

$$T(k) = \frac{r_0 + \sum_{i=1}^{k} r_i e^{\mu_i}}{1 + \sum_{j=1}^{k} e^{\mu_i}} \tag{2.8}$$

We have to find $r_k* > T(k^* - 1)$ *and* $T(k^*) \geq r_{k^*+1}$ As k increases, the threshold $T(k)$ will be a weighted average of the revenue r_j for $j \leq k$. There will be a turning point $k = k*$ where:

$$r_k^* > T(k^* - 1) \quad \textit{and} \quad T(k^*) \geq r_{k^*+1}$$

Before k^*, $r_k > T(k-1)$ indicates that each additional product still maintains a higher revenue than the threshold up to $k-1$.
After k^*, $r_k \leq T(k)$ indicates that adding any more product would result in a threshold greater than the next revenue r_{k+1}.
Uniqueness: We know than $T(k)$ is strictly increasing, therefore there can only be one point $k*$ where

$$r_k^* > T(k^*-1) \quad and \quad T(k^*) \geq r_{k^*+1}$$

(f) We will prove that we cannot have:

$$r_{k^*} < \frac{r_0 + \sum_{i=1}^{k^*} r_i e^{\mu_i}}{1 + \sum_{j=1}^{k^*} e^{\mu_i}} \tag{2.9}$$

Let's suppose this previous equation is true, we can deduce from the point (c) the following:

$$\frac{r_0 + \sum_{i=1}^{k^*-1} r_i e^{\mu_i}}{1 + \sum_{j=1}^{k^*-1} e^{\mu_i}} > \frac{r_0 + \sum_{i=1}^{k^*} r_i e^{\mu_i}}{1 + \sum_{j=1}^{k^*} e^{\mu_i}}$$

$$\Leftrightarrow \quad r_{k^*} < \frac{r_0 + \sum_{i=1}^{k^*-1} r_i e^{\mu_i}}{1 + \sum_{j=1}^{k^*-1} e^{\mu_i}}$$

However we showed in point (e) that:

$$\Leftrightarrow \quad r_{k^*} > \frac{r_0 + \sum_{i=1}^{k^*-1} r_i e^{\mu_i}}{1 + \sum_{j=1}^{k^*-1} e^{\mu_i}}$$

The Equation 2.9 is then impossible. We can continue the reasoning by stating the following:

$$r_{k^*} \not< \frac{r_0 + \sum_{i=1}^{k^*} r_i e^{\mu_i}}{1 + \sum_{j=1}^{k^*} e^{\mu_i}}$$

$$\Leftrightarrow \quad r_{k^*} \geq \frac{r_0 + \sum_{i=1}^{k^*} r_i e^{\mu_i}}{1 + \sum_{j=1}^{k^*} e^{\mu_i}}$$

$$\Rightarrow \quad r_1 > \ldots > r_{k^*} \geq \frac{r_0 + \sum_{i=1}^{k^*} r_i e^{\mu_i}}{1 + \sum_{j=1}^{k^*} e^{\mu_i}} \geq r_{k^*+1} > \ldots > r_n$$

2.2 Dual

(a) Considering for any $k \in \mathcal{I}$, (π^k, π_0^k) is considered as a solution where :

$$\pi_0^k = \frac{r_0 + \sum_{j=1}^{k} e^{\mu_j} r_j}{1 + \sum_{j=1}^{k} e^{\mu_j}}$$
$$\pi_i^k = \begin{cases} r_i - \frac{r_0 + \sum_{j=1}^{k} e^{\mu_j} r_j}{1 + \sum_{j=1}^{k} e^{\mu_j}}, & \text{if } i \leq k \\ 0, & \text{otherwise} \end{cases} \qquad \forall i \in \mathcal{I} \tag{2.10}$$

By injecting the solution (π^k, π_0^k) in the objective function of the (AP-LD) presented at the Equation 1.8, a new objective value is obtained.

$$\min_{k\in\{0,n\}} \quad \frac{r_0 + \sum_{i=1}^{k} r_i e^{\mu_i}}{1 + \sum_{j=1}^{k} e^{\mu_i}} \tag{2.11}$$

To check if the solution is feasible, it is injected in the (AP-LD) constraints (Equation 1.8).
For the first constraint, two cases are considered :
$\pi_0 - \sum_{i=1}^{k} \pi_i e^{\mu_i} \geq r_0$

- For $i \leq k$:
  $\pi_0^k - \sum_{i=1}^{k}(r_i - \pi_0^k)e^{\mu_i} \geq r_0$
  $\Leftrightarrow \pi_0^k - \sum_{i=1}^{k} r_i e^{\mu_i} + \pi_0^k \sum_{i=1}^{k} e^{\mu_i} \geq r_0$
  $\Leftrightarrow \pi_0^k(1 + \sum_{i=1}^{k} e^{\mu_i}) - \sum_{i=1}^{k} r_i e^{\mu_i} \geq r_0$
  $\Leftrightarrow (\frac{r_0+\sum_{j=1}^{k} e^{\mu_j} r_j}{1+\sum_{j=1}^{k} e^{\mu_j}})(1 + \sum_{i=1}^{k} e^{\mu_i}) - \sum_{i=1}^{k} r_i e^{\mu_i} \geq r_0$
  $\Leftrightarrow r_0 + \sum_{j=1}^{k} e^{\mu_j} r_j - \sum_{i=1}^{k} r_i e^{\mu_i} \geq r_0$
  $\Leftrightarrow r_0 \geq r_0$
  Which is always true.

- For $i > k$:
  $\pi_0^k \geq r_0$
  $\Leftrightarrow \frac{r_0+\sum_{j=1}^{k} e^{\mu_j} r_j}{1+\sum_{j=1}^{k} e^{\mu_j}} \geq r_0$
  $\Leftrightarrow r_0 + \sum_{j=1}^{k} e^{\mu_j} r_j \geq r_0(1 + \sum_{j=1}^{k} e^{\mu_j})$
  $\Leftrightarrow r_0 + \sum_{j=1}^{k} e^{\mu_j} r_j \geq r_0 + r_0 \sum_{j=1}^{k} e^{\mu_j}$
  $\Leftrightarrow \sum_{j=1}^{k} e^{\mu_j} r_j \geq r_0 \sum_{j=1}^{k} e^{\mu_j}$
  $\Leftrightarrow \sum_{j=1}^{k} e^{\mu_j}(r_j - r_0) \geq 0$
  Since $r_j > r_0, \forall j \in \mathcal{I}$, this inequality is always satisfied.

For the second constraint, two cases are also considered :
$\pi_0 + \pi_i \geq r_i, \quad \forall i \in \mathcal{I}$

- For $i \leq k$:
  $\Leftrightarrow \pi_0^k + \pi_i^k \geq r_i$
  $\Leftrightarrow \pi_0^k + (r_i - \pi_0^k) \geq r_0$
  $\Leftrightarrow r_i \geq r_i$
  Which is always true.

- For $i > k$:
  $\Leftrightarrow \pi_0^k \geq r_i$
  $\Leftrightarrow \frac{r_0+\sum_{j=1}^{k} e^{\mu_j} r_j}{1+\sum_{j=1}^{k} e^{\mu_j}} \geq r_i$
  $\Leftrightarrow r_0 + \sum_{j=1}^{k} e^{\mu_j} r_j - r_i \sum_{j=1}^{k} e^{\mu_j} \geq r_i$
  $\Leftrightarrow r_0 + \sum_{j=1}^{k} e^{\mu_j}(r_j - r_i) \geq r_i$
  Since $r_j > r_i, \forall j \leq k$, each term $(r_j - r_i)$ is positive. So the sum in the left part of the equation is positive or zero. So the left part is always greater or equal than r_i.

This implies that there exists a k such that $\pi_0^k \geq r_i, \forall i > k$, ensuring that (π^k, π_0^k) is feasible for AP-LD for such k.

(b) The primal and dual have the same objective value for k :

$$\frac{r_0 + \sum_{i=1}^{k} r_i e^{\mu_i}}{1 + \sum_{j=1}^{k} e^{\mu_j}}$$

Since it has been proved that (y^k, y_0^k) and (π^k, π_0^k) are feasible, it can be concluded that $S = \{1, ..., k\}$ is optimal for (AP). Since (AP) and (AP-L) have the same objective value, (AP-L) is an ideal formulation for (AP).

(c) To resolve the AP, the greedy algorithm can be used. This algorithm takes profit from the property at the Equation 0.1. The algorithm is presented beneath.

Algorithm 1 Greedy Algorithm

$S = \{\}$
$bestValue = 0$
for i in $\mathcal{I}$ **do**
  $S = S \cup i$
  $value = r_0 P_0(S) + \sum_{i \in S} r_i P_i(S)$
  if value $<$ bestValue **then**
    break
  else
    bestValue $=$ value
  end if
end for

Its complexity is $\mathcal{O}(n^2)$. The worst case happens when all the products i are added into S and the sum must be realised on the whole subset S containing all products i.

3 A more practical model

We now suppose that the retailer can only offer up to p products to its customers.

1. Starting from the AP-L, we can express a new model to take in account the new parameter p. This new model called the Mixed Integer Linear Program is formulated as follow.

$$\begin{aligned}
(\text{APC-MILP}) \quad & \max_{y_j \geq 0, z_i \in \{0,1\}} && r_0 y_0 + \sum_{i=1}^{n} r_i y_i \\
& \text{s.t.} && y_0 + \sum_{i=1}^{n} y_i = 1 \\
& && y_i \leq y_0 e^{\mu_i}, \quad \forall i \in I \\
& && y \leq z, \\
& && \sum_{i=1}^{n} z_i \leq p
\end{aligned} \tag{3.1}$$

The new variable z is a binary vector where $z_i = 1$ if a product is presented to the customer, $z_i = 0$ otherwise. The fourth constraint ensures that no more than p products are presented to the customer. The third constraint ensures that if a product is not presented to the client, its probability of being bought is null.

2. The number of constraints in the APC-MILP grows linearly with the size of the instance. The greedy algorithm is not suitable for such a problem with many constraints

3. Let's now present a adaptation of the AP-IP model to accomodate the new p parameter:

$$\begin{aligned}
(\text{APC-IP}) \quad & \max_{x \in \{0,1\}^n} \left\{ \frac{r_0 + \sum_{i=1}^{n} x_i r_i e^{\mu_i}}{1 + \sum_{i=1}^{n} x_i e^{\mu_i}} \right\} \\
& \text{s.t.} \quad \sum_{i=1}^{n} x_i \leq p
\end{aligned} \tag{3.2}$$

   (a) **Pricing:** Knowing that the constraint of APC-IP can be rewriten as follow:

$$\sum_{i=1}^{n} x_i \leq p \Leftrightarrow \frac{\sum_{i=1}^{n} x_i}{1 + \sum_{i=1}^{n} x_i e^{\mu_i}} \leq \frac{p}{1 + \sum_{i=1}^{n} x_i e^{\mu_i}}$$

   Let's induce a penalisation $\lambda \geq 0$ for the constraint of APC-IP. The Lagrangian function is:

$$\mathcal{L}(x, \lambda) = \frac{r_0 + \sum_{i=1}^{n} x_i r_i e^{\mu_i}}{1 + \sum_{i=1}^{n} x_i e^{\mu_i}} - \lambda \left(\frac{\sum_{i=1}^{n} x_i - p}{1 + \sum_{i=1}^{n} x_i e^{\mu_i}} \right) \tag{3.3}$$

   The new objective function is then:

$$\omega(\lambda) = \max_{x \in \{0,1\}^n} \frac{r_0 + \sum_{i=1}^{n} x_i \left(r_i e^{\mu_i} - \lambda \right)}{1 + \sum_{i=1}^{n} x_i e^{\mu_i}} \tag{3.4}$$

and the new revenues are:

$$\begin{cases} r_0(\lambda) = r_0 + \lambda \cdot p \\ r_i(\lambda) = \frac{r_i e^{\mu_i} - \lambda}{e^{\mu_i}} \end{cases} \tag{3.5}$$

By injecting these new values in the the Equation 3.4 we can recast the problem to give it a similar form as the (AP-L):

$$\begin{aligned} \text{(AP-L)}(\lambda) \quad \omega(\lambda) := \max_{y, y_0 \geq 0} \quad & r_0(\lambda) y_0 + \sum_{i=1}^{n} r_i(\lambda) y_i \\ \text{s.t.} \quad & y_0 + \sum_{i=1}^{n} y_i = 1, \\ & y_i \leq y_0 e^{\mu_i}, \quad \forall i \in \mathcal{I} \end{aligned} \tag{3.6}$$

(b) We will now implement a binary search algorithm to find the optimal value of λ

Algorithm 2 Binary Search for λ^*

$Best\_val = \inf$
$\lambda_{low} = 0$
$\lambda_{up} = \frac{r_1 - r_0}{p}$
for i in $maxIterations$ **do**
  $\lambda_{mid} = \frac{\lambda_{up} - \lambda_{low}}{2}$
  $objVal = \omega(\lambda_{mid})$
  if $objVal < Best\_val$ **and** is_feasable($\omega(\lambda_{mid})$) **then**
    $Best\_val = objVal$
  end if
  if $objVal < \omega(\lambda_{mid} + \varepsilon)$ **then**
    $\lambda_{up} = \lambda_{mid}$
  else
    $\lambda_{low} = \lambda_{mid}$
  end if
  if $\lambda_{mid} < \varepsilon$ **then**
    break
  end if
end for
$\lambda^* = \lambda_{mid}$

It is known that the maximum value for lambda is $\frac{r_1 - r_0}{p}$ because beyond this, the penality becomes large enough that is reduces the attractiveness of selecting any product beyond what is already contributing to the revenue maximization.

(c) **Heuristic:** It can be seen on line 7 of Algorithm 2 that when λ provide a feasible solution we compare the objective value with the best one so far and keep the best one.

(d) The lagrangian algorithm was implemented and a plot of the primal and dual bounds, on $n = 5000$, instances are presented at the Figure 1.

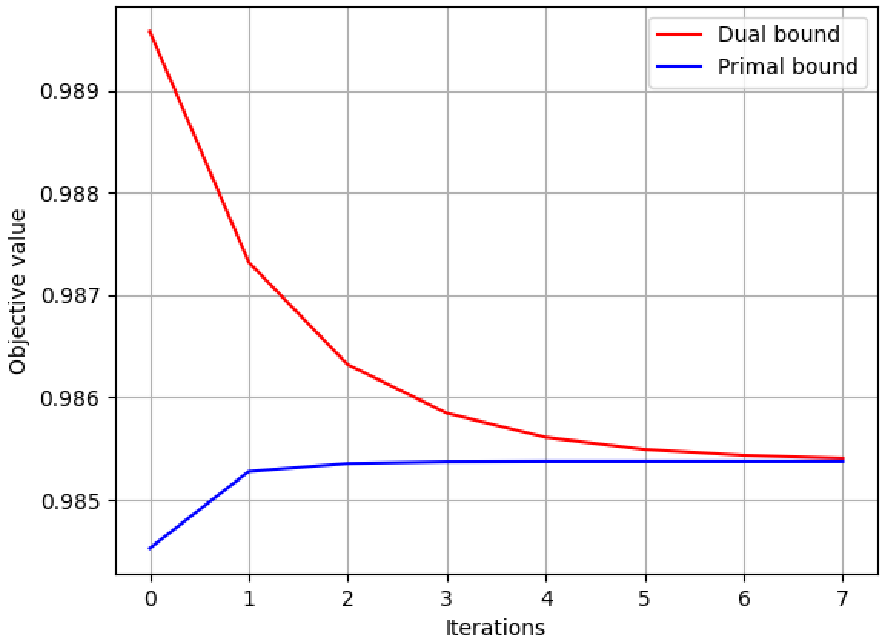

Figure 1: Primal and dual bounds in function of the number of iterations on a instance of $n = 5000$

If it's considered $r_1 > r_0$, it is possible to prove that the complexity of the lagrangian dual at a ε-precision requires solving a number of assortment planning problems in $\mathcal{O}(\log_2(\frac{r_1 - r_0}{p\varepsilon}))$.
Since the binary search over λ operates by halving the interval size at each step, to reach ε-precision, the number of steps is given by :
$\lceil \log_2(\frac{\lambda_{max}}{\varepsilon}) \rceil$
$\Leftrightarrow \mathcal{O}(\log_2(\frac{r_1 - r_0}{p\varepsilon}))$ since $\lambda_{max} \leq \frac{r_1 - r_0}{p}$.

4 Implementation and evaluation of models and algorithms

The tables provided below offer an quantitative evaluation of our models on different instance sizes. The metrics used are the minimum, average and maximum for the values of the objective function and the computation times with the different algorithms. Let's discuss the tables in details.
The AP is solved with a greedy algorithm that was implemented, the AP-L and APC-MILP are solved with the Gurobi solver and the AP-L(λ) is solved with a lagrangian relaxation that was also implemented.

4.1 Instances of size $n = 10$

The values are exactly the same of the resolution of AP, AP-L, APC-MILP and AP-L(λ). This indicates that at small scale the models performs similarly on the objective function. Concerning the execution times, the greedy algorithm gives better results. The APC-MILP is the slowest because it contains more constraints than the AP-L and the AP-L(λ) is the slowest.

Modelisation	$Value_{min}$	$Value_{avg}$	$Value_{max}$	t_{min} (s)	t_{avg} (s)	t_{max} (s)
AP	0.271592	0.631503	0.821510	0.000033	0.000118	0.000381
AP-L	0.271592	0.631503	0.821510	0.00100	0.00207	0.00400
APC-MILP	0.271592	0.631503	0.821510	0.0001	0.0021	0.0090
AP-L(λ)	0.271594	0.631504	0.821510	0.106271	0.236986	0.596347

Table 1: Modelisations on instances of $n = 10$

4.2 Instances of size $n = 5000$

The four models achieve once again similar results. The advantage in runtime of the greedy algorithm disappears, its performance is now comparable with the AP-L resolution with the Gurobi solver.

Modelisation	$Value_{min}$	$Value_{avg}$	$Value_{max}$	t_{min} (s)	t_{avg} (s)	t_{max} (s)
AP	0.982437	0.984725	0.987747	0.077502	0.150556	0.356821
AP-L	0.982437	0.984725	0.987747	0.06700	0.15615	0.38500
APC-MILP	0.982437	0.984725	0.987747	0.05500	0.06006	0.08500
AP-L(λ)	0.982469	0.984754	0.987773	6.495072	9.750247	22.302687

Table 2: Modelisations on instances of $n = 5000$

4.3 Instances of size $n = 10^6$

Here the greedy algorithm still achieve to reach the same objective value as the Gurobi solver but its runtime is worst. By limiting the execution time of the greedy algorithm we would have a worst solution. The AP-L(λ) takes a lot of time too.

Modelisation	$Value$	t (s)
AP	0.998917	468.268324
AP-L	0.998917	28.055
APC-MILP	0.998917	71.459
AP-L(λ)	0.999234	242.223968

Table 3: Modelisations on instance of $n = 10^6$

4.4 Discussion of the influence of p for the APC-MILP

The Table 4 presents the performance of the APC-MILP model across different instance sizes and capacity constraints p. The variability in the objective values with different p values suggests that as p approaches N, the model can more optimally choose from almost all items, which results in higher objective values. It can be seen that the value $p = 1$ slows the resolution for bigger instances. This can be explained by the fact that it makes one of the constraint way more constraining.

Instance size	p	$Value_{min}$	$Value_{avg}$	$Value_{max}$	t_{min} (s)	t_{avg} (s)	t_{max} (s)
10	1	0.255786	0.532608	0.708540	0.00200	0.00451	0.01400
	$N/5$	0.271592	0.608969	0.768771	0.00100	0.00373	0.01200
	$N/2$	0.271592	0.631392	0.818615	0.00100	0.00187	0.00500
	N	0.271592	0.631503	0.821510	0.0001	0.0013	0.0030
5000	1	0.715093	0.724467	0.730177	13.47100	32.63881	47.99500
	$N/5$	0.982437	0.984725	0.987747	0.08200	0.09984	0.16600
	$N/2$	0.982437	0.984725	0.987747	0.08300	0.10016	0.16600
	N	0.982437	0.984725	0.987747	0.0550	0.0699	0.1610
10^6	1	/	0.730096	/	/	169.194	/
	$N/5$	/	0.998917	/	/	78.863	/
	$N/2$	/	0.998917	/	/	79.258	/
	N	/	0.998917	/	/	74.254	/

Table 4: Results for different values of p on the APC-MILP

5 Conclusion

In this report, we have addressed the Assortment Planning Problem (AP), a key issue in retail management where the goal is to determine the optimal set of products to offer customers in order to maximize expected revenue.

We first discussed the basic formulation of AP and introduced the greedy algorithm as an initial solution method. Our subsequent analysis focused on more sophisticated models, including AP-L (a linearized version of AP), APC-MILP (Mixed Integer Linear Programming model considering capacity constraints), and APC-IP (Integer Programming model using Lagrangian relaxation).

Through their evaluation, it was observed that while all models achieve comparable objective values, their performance varies significantly with respect to computation times. For smaller instances ($n = 10$), the greedy algorithm outperformed other methods in speed. However, as the problem size increased ($n = 5000$ and $n = 10^6$), the efficiency of the greedy algorithm diminished, making the APC-MILP and APC-IP models more viable despite their higher complexity.

The influence of the capacity constraint parameter p on the APC-MILP model was particularly notable. As p increased, allowing the model to select from a larger pool of items, the objective values improved. However, setting $p = 1$ resulted in significantly longer resolution times for larger instances.