

Programmation linéaire : l'usine

Le problème

Une usine agro-alimentaire met en boîte des petits pois carottes. Elle dispose chaque heure de 300 kg de petits pois et de 140 kg de carottes. La chaîne de production est conçue pour produire deux types de boîtes :

Boîte A : 3 kg de petits pois et 0,8 kg de carottes.

Une boîte génère alors un bénéfice de 2,30 €.

Boîte B : 2 kg de petits pois et 1,4 kg de carottes.

Une boîte apporte alors un bénéfice de 2 €.

Combien l'entreprise doit-elle fabriquer, pendant chaque heure, de boîtes de chaque type pour espérer obtenir un bénéfice maximum ?

Complément culturel

Ce type de problème fait appel à ce que l'on appelle la *programmation linéaire*, qui, comme son nom l'indique, est avant tout de la programmation !

Notons x et y le nombre de boîtes de chaque type respectivement fabriquées.

Les contraintes se traduisent par : $3x + 2y \leq 300$ et $0,8x + 1,4y \leq 140$.

x et y sont des quantités positives. Ces inégalités entraînent en particulier $x \leq 100$ et $y \leq 100$.

Les programmes



Le programme est – en partie – identique à celui développé avec les deux autres logiciels (voir ci-contre).

le programme

```
>>> def prog():
    x=0
    bmax=0
    xmax=0
    ymax=0
    while x<=100:
        y=0
        while (y<=-4/7*x+100) and (y<=-1.5*x+150):
            b=2.3*x+2*y
            if b>bmax:
                bmax=b
                xmax=x
                ymax=y
            y=y+1
        x=x+1
    return (bmax, xmax, ymax)

>>> prog()
(262.19999999999999, 54, 69)
```

le résultat

SCRATCH

Scratch



On effectue un balayage de $[0 ; 100] \times [0 ; 100]$ par le couple $(x ; y)$ tout en lui imposant les contraintes $y \leq -1,5x + 150$ et $y \leq -4 / 7 x + 100$.

On évalue pour chaque couple le bénéfice obtenu, et s'il est maximum par rapport à ceux trouvés précédemment, on le prend comme bénéfice maximum.

```

quand pressé
à x attribuer 0
à Bmax attribuer 0
à xBmax attribuer 0
à yBmax attribuer 0
répéter jusqu'à x > 100
à y attribuer 0
répéter jusqu'à y > -4 / 7 * x + 100 ou y > -1.5 * x + 150
à B attribuer 2.3 * x + 2 * y
si B > Bmax
à Bmax attribuer B
à xBmax attribuer x
à yBmax attribuer y
changer y par 1
changer x par 1
dire Bénéfice maximum : pendant 2 secondes
dire Bmax pendant 2 secondes
  
```



AlgoBox

```

VARIABLES
- x EST_DU_TYPE NOMBRE
- Bmax EST_DU_TYPE NOMBRE
- xBmax EST_DU_TYPE NOMBRE
- yBmax EST_DU_TYPE NOMBRE
- y EST_DU_TYPE NOMBRE
- B EST_DU_TYPE NOMBRE

DEBUT_ALGORITHME
x PREND_LA_VALEUR 0
Bmax PREND_LA_VALEUR 0
xBmax PREND_LA_VALEUR 0
yBmax PREND_LA_VALEUR 0
TANT_QUE (x <= 100) FAIRE
  DEBUT_TANT_QUE
  y PREND_LA_VALEUR 0
  TANT_QUE ((y <= -4/7*x+100) ET (y <= -1.5*x+150))
  DEBUT_TANT_QUE
  B PREND_LA_VALEUR 2.3*x+2*y
  SI (B > Bmax) ALORS
    DEBUT_SI
    Bmax PREND_LA_VALEUR B
    xBmax PREND_LA_VALEUR x
    yBmax PREND_LA_VALEUR y
    FIN_SI
  y PREND_LA_VALEUR y+1
  FIN_TANT_QUE
  x PREND_LA_VALEUR x+1
  FIN_TANT_QUE
AFFICHER "Le bénéfice maximum (en €) est : "
AFFICHER Bmax
AFFICHER " Il est obtenu pour x = "
AFFICHER xBmax
AFFICHER " et pour y = "
AFFICHER yBmax
FIN_ALGORITHME
  
```

Prolongements

La même entreprise doit utiliser au moins 90 % des matières premières en petits pois et elle s'engage à fournir au total au moins 80 boîtes. Sachant que la production d'une boîte de type A coûte 0,15 € et la production d'une boîte de type B coûte 0,10 €, combien l'entreprise doit-elle fabriquer de boîtes de chaque sorte pour minimiser ses coûts de production ?

Développer un programme avec Scratch, AlgoBox ou Python afin d'obtenir la réponse.

Remarque : on limitera volontairement le balayage à x et y variant de 0 à 100, avec bien entendu les contraintes imposées par l'énoncé.