ASI - Recherche Opérationelle - Examen final - 2nd partie - **1 h 30**

Exercice 1 — Il est libre Max (mais attention quand même) — 4 points

On considère le problème d'optimisation suivant :

$$\begin{cases} \max\limits_{x\in\mathbf{R}} & -x^2-1 \\ \text{avec} & (2-x)(x-4)\geq 0 \end{cases}$$

1. donner les conditions nécessaires pour qu'un point x puisse être une solution du problème,
2. écrire le lagrangien du problème
3. montrez que la formulation duale du problème est

$$\begin{cases} \max\limits_{\lambda\in\mathbf{R}} & \dfrac{-\lambda^2+9\lambda+1}{1+\lambda} \\ \text{avec} & \lambda\geq 0 \end{cases}$$

4. calculez la solution du problème dual (en faisant dans un premier temps abstraction de la contrainte) et en déduire la solution du problème primal
5. on considère maintenant le même problème mais perturbé par un réel ε.

$$\begin{cases} \max\limits_{x\in\mathbf{R}} & -x^2-1 \\ \text{avec} & (2-x)(x-4)\geq \varepsilon \end{cases}$$

si l'on note $J^*(\varepsilon)=-x^*(\varepsilon)^2-1$ la valeur du cout de la solution du problème perturbé comme une fonction de ε, calculez $\frac{dJ^*(\varepsilon)}{d\varepsilon}$ au point $\varepsilon=0$, que vous exprimerez comme une fonction de λ.

Exercice 2 — Heureux qui comme Unyque — 4 points

Soit n un entier positif et $\alpha_i, i=1,n$ un vecteur donné de n réels positifs. Soit C une quantité (supposée réelle et positive) à transmettre à travers les n canaux disponible. Le problème est donc de trouver comment distribuer l'information à envoyer sur les n canaux, c'est-à-dire trouver le vecteur des $x_i, i=1,n$ représentant la quantité d'information à envoyer par le canal i, tout en maximisant l'entropie.

$$\begin{cases} \max\limits_{\mathbf{x}\in\mathbf{R}^n} & \sum\limits_{i=1}^{n} \log\left(\dfrac{1}{\alpha_i+x_i}\right) \\ \text{avec} & \mathbf{x}\geq 0 \\ \text{et} & \sum\limits_{i=1}^{n} x_i = C \end{cases}$$

1. Donnez les conditions d'optimalité du problème.
2. Reformulez ces conditions en éliminant les multiplicateurs de Lagrange associés aux contraintes d'inégalitées
3. En déduire que $x_i=\max(0,1/\mu^*-\alpha_i)$ où μ^* est la valeur optimale du multiplicateur de Lagrange associé à la contrainte d'égalité.
4. Montrez que la solution du problème est alors donnée par une équation en μ. Proposez une méthode pour la résoudre.

Exercice 3 — El Clásico — 4 points

1. Donnez la forme standard du programme linéaire suivant :

$$\begin{cases} \max\limits_{\mathbf{x}\in\mathbf{R}^n,\mathbf{y}\in\mathbf{R}^\ell} & \sum\limits_{i=1}^{n} c_i|x_i| - \sum\limits_{j=1}^{\ell} d_j\, y_j \\ \text{avec} & A\mathbf{x} = \mathbf{b}_1 \\ \text{et} & B\mathbf{x} + C\mathbf{y} \leq \mathbf{b}_2 \\ & 0 \leq y_j \leq 1, \quad j = 1, \ell \end{cases}$$

2. Proposez un algorithme efficace permettant d'obtenir la solution du problème lorsque la matrice $B = 0$.
3. Donnez la fonction matlab `function` $[x, y]$ = `resoud`$(A, C, b1, b2, c, d)$ mettant en œuvre votre algorithme. Vous pourrez utiliser les fonctions que vous avez utilisé en TP (`linprog, quadprog, cvx...` voir annexe).

Annexes : quelques doc matlab

```
>> help linprog
 LINPROG Linear programming.
    X = LINPROG(f,A,b) attempts to solve the linear programming problem:

              min f'*x    subject to:   A*x <= b
               x

    X = LINPROG(f,A,b,Aeq,beq) solves the problem above while additionally
    satisfying the equality constraints Aeq*x = beq.

    X = LINPROG(f,A,b,Aeq,beq,LB,UB) defines a set of lower and upper
    bounds on the design variables, X, so that the solution is in
    the range LB <= X <= UB. Use empty matrices for LB and UB
    if no bounds exist. Set LB(i) = -Inf if X(i) is unbounded below;
    set UB(i) = Inf if X(i) is unbounded above.

>> help quadprog
 QUADPROG Quadratic programming.
    X = QUADPROG(H,f,A,b) attempts to solve the quadratic programming problem:

              min 0.5*x'*H*x + f'*x   subject to:  A*x <= b
               x

    X = QUADPROG(H,f,A,b,Aeq,beq) solves the problem above while
    additionally satisfying the equality constraints Aeq*x = beq.

    X = QUADPROG(H,f,A,b,Aeq,beq,LB,UB) defines a set of lower and upper
    bounds on the design variables, X, so that the solution is in the
    range LB <= X <= UB. Use empty matrices for LB and UB if no bounds exist. Set
    LB(i) = -Inf if X(i) is unbounded below; set UB(i) = Inf if X(i) is unbounded above.

>> help cvx
  CVX: A system for disciplined convex programming.
    CVX is a modeling framework for building, constructing, and solving
    disciplined convex programs.

    cvx_setup        - Sets up and tests the cvx distribution.
    commands         - Top-level commands to create and control CVX.
    functions        - Additional functions created specifically for CVX.
    keywords         - Keywords for declaring variables and objectives
    lib              - Code for internal use by CVX.

% exemple
 cvx_begin
   variable x(n)
   dual variables y z
   minimize( c' * x )
   subject to
      y : A * x == b;
      z : x >= 0;
cvx_end
```