Two Stage Stochastic Optimization for Fixing Energy Reserves

Michel De Lara, Jean-Philippe Chancelier

March 18, 2019

1 Problem statement

We formalize the problem of fixing energy reserves in a day-ahead market as a two stage stochastic optimization problem. A decision has to be made at night of day J — which quantity of the cheapest energy production units (reserve) to be mobilized — to meet a demand that will materialize at morning of day $J+1$. Excess reserves are penalized; demand unsatisfied by reserves have to be covered by costly extra units. Hence, there is a trade-off to be assessed by optimization.

1.1 Stages

There are two stages, represented by the letter t (for time):

- $t = 0$ corresponds to night of day J;
- $t = 1$ corresponds to morning of day $J + 1$.

1.2 Probabilistic model

We suppose that the demand, materialized on the morning of day $J + 1$, can take a finite number S of possible values D_s, where s denotes a *scenario* in the finite set $\mathbb{S}$ (S=card($\mathbb{S}$)).

We denote π_s the probability of scenario s, with

$$\forall s \in \mathbb{S} \; , \;\; \pi_s > 0 \; , \;\; \sum_{s \in \mathbb{S}} \pi_s = 1 \; . \tag{1}$$

Notice that we do not consider scenarios with zero probability.

1.3 Decision variables

The decision variables are the scalar Q_0 and the finite sequence $(Q_{1,s})_{s\in\mathbb{S}}$ of scalars, as follows:

- at stage $t = 0$, the energy reserve is Q_0;
- at stage $t = 1$, a scenario s materializes and the demand D_s is observed, so that one decides of the recourse quantity $Q_{1,s}$.

The decision variables can be considered as indexed by a *tree* with one root (corresponding to the index 0) and as many leafs as scenarios in $\mathbb{S}$ (each leaf corresponding to the index $1, s$): Q_0 is attached to the root of the tree, and each $Q_{1,s}$ is attached to a leaf corresponding to s.

1.4 Optimization problem formulation

The balance equation between supply and demand is

$$Q_0 + Q_{1,s} = D_s \ , \ \ \forall s \in \mathbb{S} \ . \tag{2}$$

The energies mobilized at stages $t = 0$ and $t = 1$ display different features:

- at stage $t = 0$, the energy production has maximal capacity $Q_0^\sharp$, and producing Q_0 costs $\mathcal{C}_0(Q_0)$;
- at stage $t = 1$, the energy production is supposed to be unbounded, and producing Q_1 costs $\mathcal{C}_1(Q_1)$.

We consider the stochastic optimization problem

$$\min_{Q_0,\, Q_{1,s},\, s\in\mathbb{S}} \sum_{s\in\mathbb{S}} \pi_s \left[\mathcal{C}_0(Q_0) + \mathcal{C}_1(Q_{1,s})\right] \tag{3a}$$

$$\text{s.t.} \quad 0 \leq Q_0 \leq Q_0^\sharp \tag{3b}$$

$$0 \leq Q_{1,s} \qquad \forall s \in \mathbb{S} \tag{3c}$$

$$D_s = Q_0 + Q_{1,s} \qquad \forall s \in \mathbb{S} \tag{3d}$$

Here, we look for energy reserve Q_0 and recourse energy $Q_{1,s}$ so that the balance equation (3d) is satisfied (at stage $t = 1$) at minimum *expected cost* in (3a). By weighing each scenario s with its probability π_s, the optimal solution $\big(Q_0^\star, (Q_{1,s}^\star)_{s\in\mathbb{S}}\big)$ performs a compromise between scenarios.

2 Formulation on a tree with linear costs

Here, we suppose that the costs are linear:

$$\mathcal{C}_0(Q_0) = l_0 Q_0 \ , \ \ \mathcal{C}_1(Q_1) = l_1 Q_1 \ . \tag{4}$$

Therefore, the stochastic optimization problem (3) now becomes

$$\min_{Q_0,\, Q_{1,s},\, s \in \mathbb{S}} \sum_{s \in \mathbb{S}} \pi_s (l_0 Q_0 + l_1 Q_{1,s}) \tag{5a}$$

$$\text{s.t.} \qquad 0 \leq Q_0 \leq Q_0^\sharp \tag{5b}$$

$$0 \leq Q_{1,s} \qquad \forall s \in \mathbb{S} \tag{5c}$$

$$D_s = Q_0 + Q_{1,s} \qquad \forall s \in \mathbb{S} \tag{5d}$$

This optimization problem (5) is linear. When the number S of scenarios is not too large, we can use linear solvers.

Question 1 *We consider the case when $\mathbb{S} = \{L, M, H\}$ has $S = 3$ scenarios (low, medium, high). We want to transform the linear optimization problem* (5) *under a form adapted to a linear solver.*

a) ***[1+1+1]*** *Expand the criterion* (5a)*. Expand the inequalities* (5b)–(5c) *into an array of five scalar equations (one inequality per equation), and the equalities* (5d) *into an array of three scalar equations (one equality per equation).*

b) ***[1+1+1+1+1]*** *Let x be the column vector $x = (Q_0, Q_{1L}, Q_{1M}, Q_{1H})$. Propose matrices A_e, A_i and column vectors b_e, b_i and c such that the linear optimization problem* (5) *can be written under the form*

$$\min_{x \in \mathbb{R}^4} c'x \tag{6a}$$

$$s.t. \qquad A_e x = b_e \tag{6b}$$

$$A_i x \leq b_i \tag{6c}$$

Question 2 *We are going to numerically solve the linear optimization problem* (5)*.*

a) ***[2]*** *Interpret the code below. What is the macro* `linpro` *doing? What is* `lopt`*? Copy the code into a file named* tp_q1.sce*.*

b) ***[1+2]*** *Solve a numerical version of problem* (5) *with $S = 3$ scenarios and the parameters in the code below, by executing the file* tp_q1.sce*. What is the optimal value $Q_0^\star$ of the reserve? What is the optimal value $Q_{1,L}^\star$? Can you explain why these values are optimal? (there is an economic explanation based on relative costs; there is a mathematical explanation based on the properties of the solutions of a linear program).*

c) ***[3]*** *Then, increase and decrease the value of the unitary cost l_1 (especially above and below l_0). Show the numerical results that you obtain. What happens to the optimal values $Q_0^\star$? Explain why (make the connection with the properties of the solutions of a linear program).*

```
// Formulation on a tree with linear costs.
// Numerical resolution by linear programming

// Constant initialization
S=3;// Number of random scenarios
q0m=30;// max capacity for q0

// Demand
if S==3 then
  D=[15;20;50];
  Pr=[0.2;0.6;0.2];// Probabilities of Demand
else
  D=grand(S,1,'uin',5,50);
  Pr=grand(S,1,'unf',0,1);
  Pr=Pr ./sum(Pr);// Probabilities of Demand
end

// Constants used in the cost function
l10=2;l11=5;

// a revoir pour passer a des contraintes egalité et utiliser lb et ub
c=[l10;l11 .*Pr];// cost coefficients

// inequality constraints (bounds on production)
Ai=[-eye(S+1,S+1);eye(1,S+1)];
bi=[zeros(S+1,1);q0m];

// equality constraints, i.e. production equals demand
Ae=[ones(S,1),eye(S,S)];
be=[D];

// solving by linear
// xopt should be [  15,  0, 5, 35 ] when S=3

// scicoslab version with linpro
A=[Ae;Ai];b=[be;bi];
[xopt,lopt,fopt]=linpro(c,A,b,[],[],size(be,'*'))
```

Question 3 *We are going study the impact of the number S of scenarios on the numerical resolution of the linear optimization problem* (5)*.*

a) *[2] Take $S = 100$. What is the optimal value $Q_0^\star$ of the reserve? Identify the scenario $\bar{s}$ with the lowest demand. What is the optimal value $Q_{1,\bar{s}}^\star$? Explain.*

b) ***[1]*** *For what value of n in $S = 10^n$ can you no longer solve numerically?*

3 Formulation on a tree with quadratic convex costs

Here, we suppose that the costs are quadratic and convex:

$$\mathcal{C}_0(Q_0) = \frac{1}{2}K_0 Q_0^2 + l_0 Q_0 \ , \ \ K_0 > 0 \ , \ \ \mathcal{C}_1(Q_1) = \frac{1}{2}K_1 Q_1^2 + l_1 Q_1 \ , \ \ K_1 > 0 \ . \tag{7}$$

The optimization problem (3) is quadratic convex:

$$\min_{Q_0, Q_{1,s}, s \in \mathbb{S}} \sum_{s \in \mathbb{S}} \pi_s \left[\frac{1}{2}K_0 Q_0^2 + l_0 Q_0 + \frac{1}{2}K_1 Q_{1,s}^2 + l_1 Q_{1,s} \right] \tag{8a}$$

$$\text{s.t.} \quad 0 \le Q_0 \le Q_0^\sharp \tag{8b}$$

$$0 \le Q_{1,s} \qquad \forall s \in \mathbb{S} \tag{8c}$$

$$D_s = Q_0 + Q_{1,s} \qquad \forall s \in \mathbb{S} \tag{8d}$$

When the number S of scenarios is not too large, we can use quadratic solvers.

Question 4 *We are going to numerically solve the quadratic convex optimization problem* (8)*.*

a) ***[1]*** *Interpret the code below. What is the macro* ***quapro*** *doing? Copy the code into a file named* tp_q2.sce*.*

b) ***[1]*** *Solve a numerical version with $S = 3$ scenarios and the parameters in the code below, by executing the files* tp_q1.sce *and* tp_q2.sce*. What is the optimal value $Q_0^\star$ of the reserve? What is the optimal value $Q_{1,L}^\star$?*

c) ***[1+1+2]*** *Check numerically that $(Q_0^\star, Q_{1,L}^\star, Q_{1,H}^\star, Q_{1,M}^\star)$ is an inner solution to the optimization problem* (8a)*, that is, check numerically that the inequalities* (8b) *and* (8c) *are strict. What is the difference with the optimal solution of Question 2? Discuss the difference (make the connection with the properties of the solutions of a linear program).*

d) ***[2+1]*** *Compute the derivatives of the cost functions in* (7)*. Check numerically (giving the details of computation) that the optimal solution $(Q_0^\star, Q_{1,L}^\star, Q_{1,M}^\star, Q_{1,H}^\star)$ satisfies the following relation between marginal costs:*

$$\mathcal{C}_0'(Q_0^\star) = \pi_L \mathcal{C}_1'(Q_{1,L}^\star) + \pi_M \mathcal{C}_1'(Q_{1,M}^\star) + \pi_H \mathcal{C}_1'(Q_{1,H}^\star) \ . \tag{9}$$

```
// Formulation on a tree with quadratic costs
// Numerical resolution by brute force quadratic programming

// just to get common data
exec('tp_q1.sce');

// the problem is now quadratic; we use a quadratic solver
A=[Ae;Ai];b=[be;bi];
KK0=10;KK1=1;
KK=diag([KK0,KK1*Pr' .*ones(1,S)]);

// scicoslab version with quapro
[xopt,lopt,fopt]=quapro(KK,c,A,b,[],[],size(be,'*'))
```

Question 5 *We are going to numerically solve the quadratic convex optimization problem* (8) *after changing the relative values of the (quadratic) parameters K_0 and K_1. For the parameters K_0, l_0, l_1, we take the same values as those in Question 2 (hence $l_1 > l_0$) and in Question 4.*

a) ***[1]*** *Take $K_1 > K_0$ with $K_1 \approx K_0$. What are the optimal value $Q_0^\star$ of the reserve and the optimal value $Q_{1,L}^\star$?*

b) ***[1]*** *Take $K_1 > K_0$ with $K_1 >> K_0$. What are the optimal value $Q_0^\star$ of the reserve and the optimal value $Q_{1,L}^\star$?*

c) ***[2]*** *Discuss.*

Question 6 *We are going study the impact of the number S of scenarios on the numerical resolution of the quadratic convex optimization problem* (8)*.*

a) ***[1]*** *Take $S = 100$. Solve a numerical version of problem* (8) *with the parameters K_0, l_0, K_1, l_1 as in the code of Question 4. What is the optimal value $Q_0^\star$ of the reserve? Identify the scenario $\bar{s}$ with the lowest demand. What is the optimal value $Q_{1,\bar{s}}^\star$?*

b) ***[1]*** *For what value of n in $S = 10^n$ can you no longer solve numerically?*

Question 7 **This theoretical question may be ignored (by those who want to focus on numerical results)**

For this question, we temporarily ignore the inequalities (8b) *and* (8c) *in* (8)*. Therefore, we consider the optimization problem* (8a) *with equality constraints* (8d)*, that is:*

$$\min_{Q_0,\, Q_{1,s},\, s\in\mathbb{S}} \sum_{s\in\mathbb{S}} \pi_s \left[\frac{1}{2}K_0 Q_0^2 + l_0 Q_0 + \frac{1}{2}K_1 Q_{1,s}^2 + l_1 Q_{1,s}\right] \tag{10a}$$

$$s.t. \qquad D_s = Q_0 + Q_{1,s}\ , \quad \forall s \in \mathbb{S}\ . \tag{10b}$$

a) ***[2]*** *Compute the Hessian matrix of the criterion*

$$J_0 : \big(Q_0, (Q_{1,s})_{s\in\mathbb{S}}\big) \in \mathbb{R}\times\mathbb{R}^S \mapsto \sum_{s\in\mathbb{S}} \pi_s \left[\frac{1}{2}K_0 Q_0^2 + l_0 Q_0 + \frac{1}{2}K_1 Q_{1,s}^2 + l_1 Q_{1,s}\right] . \tag{11}$$

What are the dimensions of the Hessian matrix?

b) ***[3]*** *Why does optimization problem* (10) *have a solution? (Beware of the domain)*

c) ***[2]*** *Why is the solution unique?*

d) ***[1+2]*** *Why are the equality constraints* (10b) *qualified? Why does an optimal solution* $\big(Q_0^\star, (Q_{1,s}^\star)_{s\in\mathbb{S}}\big)$ *of* (10) *satisfy the Karush-Kuhn-Tucker (KKT) conditions (first-order optimality conditions)?*

e) ***[2]*** *Why is a solution of the KKT conditions an optimal solution of* (10)*?*

f) ***[1]*** *Write the Lagrangian* $\mathcal{L}_0\left(Q_0, (Q_{1,s})_{s\in\mathbb{S}}, (\mu_s)_{s\in\mathbb{S}}\right)$ *associated with problem* (10).

g) ***[2]*** *Deduce the KKT conditions. Show that there exist* $(\mu_s^\star, s\in\mathbb{S})$ *such that*

$$\mathcal{C}_0'(Q_0^\star) - \sum_{s\in\mathbb{S}} \mu_s^\star = 0 \quad \text{and} \quad \pi_s \mathcal{C}_1'(Q_{1,s}^\star) - \mu_s^\star = 0 \ , \ \ \forall s\in\mathbb{S} \ . \tag{12}$$

h) ***[2]*** *Deduce that — when* $\big(Q_0^\star, (Q_{1,s}^\star)_{s\in\mathbb{S}}\big)$ *is an inner optimal solution to problem* (8a) *— we have the following relation between marginal costs:*

$$\mathcal{C}_0'(Q_0^\star) = \sum_{s\in\mathbb{S}} \pi_s \mathcal{C}_1'(Q_{1,s}^\star) \ . \tag{13}$$

Give an economic interpretation of this equality.

4 Formulation on a fan with quadratic convex costs

When the number S of scenarios is too large, Problem (5) — be it linear or convex — cannot be solved by direct methods.

4.1 Dualization of non-anticipativity constraints

To bypass this problem, we use a "trick" consisting in introducing new decision variables $(Q_{0,s})_{s\in\mathbb{S}}$, instead of the single decision variable Q_0, and we write

$$Q_{0,s} = \sum_{s'\in\mathbb{S}} \pi_{s'} Q_{0,s'} \ , \ \ \forall s\in\mathbb{S} \ . \tag{14}$$

These equalities are called the *non-anticipativity* constraints. Indeed, the equations (14) express that

$$Q_{0,s} = Q_{0,s'} \ , \ \ \forall (s,s') \in \mathbb{S}^2 \ , \tag{15}$$

that is, the decision at stage $t = 0$ does not depend on the scenario s, hence cannot anticipate the future. Later, we will treat the constraints (14) by duality.

Therefore, the stochastic optimization problem (3) now becomes

$$\min_{Q_{0,s},\, Q_{1,s},\, \forall s \in \mathbb{S}} \sum_{s\in\mathbb{S}} \pi_s \left[\mathcal{C}_0(Q_{0,s}) + \mathcal{C}_1(Q_{1,s}) \right] \tag{16a}$$

$$\begin{aligned}
\text{s.t} \qquad & 0 \leq Q_{0,s} \leq Q_0^\sharp && \forall s \in \mathbb{S} && \text{(16b)} \\
& 0 \leq Q_{1,s} && \forall s \in \mathbb{S} && \text{(16c)} \\
& D_{1,s} = Q_{0,s} + Q_{1,s} && \forall s \in \mathbb{S} && \text{(16d)} \\
& Q_{0,s} = \sum_{s'\in\mathbb{S}} \pi_{s'} Q_{0,s'} && \forall s \in \mathbb{S} &&
\end{aligned}$$

By the assumption (1) that there are no scenarios with zero probability ($\pi_s > 0$), we replace each equality in the last equation by the equivalent one

$$\pi_s Q_{0,s} = \pi_s \sum_{s'\in\mathbb{S}} \pi_{s'} Q_{0,s'} \ , \ \ \forall s \in \mathbb{S} \ . \tag{16e}$$

We attach, to each equality above a *multiplier* $\lambda_{0,s}$. We put

$$Q = \big((Q_{0,s})_{s\in\mathbb{S}}, (Q_{1,s})_{s\in\mathbb{S}} \big) \ , \ \ \lambda = (\lambda_{0,s})_{s\in\mathbb{S}} \ . \tag{17}$$

The corresponding *Lagrangian* is

$$\mathcal{L}(Q,\lambda) = \sum_{s\in\mathbb{S}} \pi_s \left[\mathcal{C}_0(Q_{0,s}) + \mathcal{C}_1(Q_{1,s}) + \lambda_{0,s}(Q_{0,s} - \sum_{s'\in\mathbb{S}} \pi_{s'} Q_{0,s'}) \right] \tag{18a}$$

$$= \sum_{s\in\mathbb{S}} \pi_s \left[\mathcal{C}_0(Q_{0,s}) + (\lambda_{0,s} - \sum_{s'\in\mathbb{S}} \pi_{s'} \lambda_{0,s'}) Q_{0,s} + \mathcal{C}_1(Q_{1,s}) \right] \ . \tag{18b}$$

Question 8 **This theoretical question may be ignored (by those who want to focus on numerical results)**

When the costs are quadratic and convex as in (7)*, show that*

a) ***[3]*** *the criterion*

$$J : Q \in \mathbb{R}^S \times \mathbb{R}^S \mapsto \sum_{s\in\mathbb{S}} \pi_s \left[\frac{1}{2} K_0 Q_{0,s}^2 + l_0 Q_{0,s} + \frac{1}{2} K_1 Q_{1,s}^2 + l_1 Q_{1,s} \right] \tag{19}$$

in (16a) *is a-strongly convex in* $((Q_{0,s})_{s\in\mathbb{S}}, (Q_{1,s})_{s\in\mathbb{S}})$*, and provide a possible* $a > 0$*,* *b)* ***[2]*** *the domain defined by the constraints in the optimization problem* (16) *is closed,*

c) ***[2]*** *the domain defined by the constraints in the optimization problem* (16) *is convex,*

d) ***[2+1]*** *the optimization problem* (16) *has a solution, and it is unique, denoted by* $Q^\star$*,*

e) ***[1]*** *there exists a multiplier* $\lambda^\star$ *such that* $(Q^\star, \lambda^\star)$ *is a saddle point of the Lagrangian* $\mathcal{L}$ *in* (18).

4.2 Uzawa algorithm

We consider the following optimization problem

$$\min_{u \in \mathbb{U}^{\mathrm{ad}}} \quad J(u) \tag{20a}$$

$$s.t. \quad \Theta(u) = 0 \ , \tag{20b}$$

under the assumptions that

- the set $\mathbb{U}^{\mathrm{ad}}$ is a closed convex subset of a Euclidian space $\mathbb{R}^N$,
- the criterion $J : \mathbb{R}^N \to \mathbb{R}$ is an a-strongly convex ($a > 0$) and differentiable function,
- the constraint mapping $\theta : \mathbb{R}^N \to \mathbb{R}^M$ is affine, κ-Lipschitz ($\kappa > 0$),
- the Lagrangian $\mathcal{L}(u, \lambda) = J(u) + \langle \lambda, \Theta(u) \rangle$ admits a saddle-point over $\mathbb{U}^{\mathrm{ad}} \times \mathbb{R}^M$.

Then the following algorithm — called *dual gradient* algorithm, or *Uzawa algorithm* — converges toward the optimal solution of Problem (20), when $0 < \rho < 2a/\kappa^2$.

Data: Initial multiplier $\lambda^{(0)}$, step ρ
Result: optimal decision and multiplier;
repeat
  $u^{(k)} = \arg\min_{u \in \mathbb{U}^{\mathrm{ad}}} \mathcal{L}(u, \lambda^{(k)})$;
  $\lambda^{(k+1)} = \lambda^{(k)} + \rho \Theta(u^{(k)})$;
until $\Theta(u^{(k)}) = 0$;

Algorithm .1: Dual Gradient Algorithm

4.3 Numerical resolution by Uzawa algorithm (quadratic convex case)

When the costs are quadratic and convex as in (7), the optimization problem (16) becomes:

$$\min_{Q_{0,s},\, Q_{1,s},\, \forall s \in \mathbb{S}} \sum_{s \in \mathbb{S}} \pi_s \left[\frac{1}{2} K_0 Q_{0,s}^2 + l_0 Q_{0,s} + \frac{1}{2} K_1 Q_{1,s}^2 + l_1 Q_{1,s} \right] \tag{21a}$$

$$\text{s.t} \quad 0 \leq Q_{0,s} \leq Q_0^\sharp \qquad \forall s \in \mathbb{S} \tag{21b}$$

$$0 \leq Q_{1,s} \qquad \forall s \in \mathbb{S} \tag{21c}$$

$$D_{1,s} = Q_{0,s} + Q_{1,s} \qquad \forall s \in \mathbb{S} \tag{21d}$$

$$Q_{0,s} = \sum_{s' \in \mathbb{S}} \pi_{s'} Q_{0,s'} \qquad \forall s \in \mathbb{S} \tag{21e}$$

Question 9 **This theoretical question may be ignored (by those who want to focus on numerical results)**
When the costs are quadratic and convex as in (7)*, identify in the optimization problem* (21) *the corresponding elements in the Uzawa algorithm .1:*

a) ***[1]*** *decision variable u,*

b) ***[1]*** *decision set $\mathbb{R}^N$ (for what N?),*

c) ***[2]*** *affine constraints mapping $\Theta : \mathbb{R}^N \to \mathbb{R}^M$, corresponding to the constraints* (21e) *(why is it κ-Lipschitz, and for which κ?).*

d) ***[3]*** *Explain why the Uzawa algorithm converges towards the optimal solution of Problem* (21)*.*

Question 10 *We are going to numerically solve the quadratic convex optimization problem* (21) *by Uzawa algorithm.*

a) ***[3]*** *Detail what the code below is doing ; explain how the code implements the Uzawa algorithm. Why do we use the macro* ***quapro****? What is the dual function? Copy the code into a file named* tp_q3.sce.

b) ***[2]*** *Solve a numerical version with $S = 3$ scenarios. What is the solution $\big(Q_0^\star, (Q_{1,s}^\star)_{s \in \mathbb{S}}\big)$ given by the algorithm? Do you have that $D_{1,s} = Q_0^\star + Q_{1,s}^\star$, for all $s \in \mathbb{S}$?*

c) ***[1]*** *Then, try with $S = 100$. What is the value $Q_0^\star$ of the reserve given by the algorithm? Identify the scenario $\bar{s}$ with the lowest demand. What is the value $Q_{1,\bar{s}}^\star$ given by the algorithm?*

d) ***[1]*** *For what value of n in $S = 10^n$ can you no longer solve numerically?*

```
// formulation on a fan
// Constant initialization

// Constant initialization
S=3;// Number of random scenarios
q0m=30;// max capacity for q0

// Demand
if S==3 then
  D=[15;20;50];
  Pr=[0.2;0.6;0.2];// Probabilities of Demand
else
  D=grand(S,1,'uin',5,50);
  Pr=grand(S,1,'unf',0,1);
  Pr=Pr ./sum(Pr);// Probabilities of Demand
end

// Constants used in the cost function
ll0=2;ll1=5;
KK0=10;KK1=1;

// Uzawa iterations when the dualized constraints are the S constraints
//  Q0(ss) = sum(Pr.*Q0);

Q0=zeros(S,1);
Q1=zeros(S,1);
f0=zeros(S,1);

rho=5;

lambda=zeros(S,1);
// iterations of the Uzawa algorithm
for it=0:30 do
  // decomposed minimizations (loop over scenarios ss)
  for ss=1:S do
    // inequality constraints (bounds)
    // bounds on (Q0s,Q1s)
    ll=[ll0*Pr(ss);ll1*Pr(ss)];// cost coefficients
    Ai=[-eye(2,2);eye(1,2)];
    bi=[zeros(2,1);q0m];
    // equality constraints i.e production equals demand
    Ae=[1,1];
```

```
    be=[D(ss)];
    A=[Ae;Ai];b=[be;bi];
    KK=Pr(ss)*diag([KK0,KK1]);
    //
    cc=ll+Pr(ss)*[lambda(ss)-sum(Pr .*lambda);0];
    [xopt,lopt,fopt]=quapro(KK,cc,A,b,[],[],size(be,'*'));
    Q0(ss)=xopt(1);
    Q1(ss)=xopt(2);
    f0(ss)=fopt;
    printf("Dual function %f\n",sum(f0));
  end
  Q0bar=sum(Pr .*Q0);
  lambda=lambda+rho*(Pr .*(Q0-Q0bar));
end
```

5 Formulation on a fan with linear costs

Here, we suppose that the costs are linear, as in (4).

5.1 Difficulties in applying Uzawa algorithm with linear costs

The optimization problem (16) becomes:

$$
\begin{aligned}
&\min_{Q_{0,s},\, Q_{1,s},\, \forall s\in\mathbb{S}} \sum_{s\in\mathbb{S}} \pi_s \left[l_0 Q_{0,s} + l_1 Q_{1,s} \right] && && (22a)\\
&\text{s.t} \quad 0 \le Q_{0,s} \le Q_0^\sharp && \forall s \in \mathbb{S} && (22b)\\
&\qquad\; 0 \le Q_{1,s} && \forall s \in \mathbb{S} && (22c)\\
&\qquad\; D_{1,s} = Q_{0,s} + Q_{1,s} && \forall s \in \mathbb{S} && (22d)\\
&\qquad\; Q_{0,s} = \sum_{s'\in\mathbb{S}} \pi_{s'} Q_{0,s'} && \forall s \in \mathbb{S} && (22e)
\end{aligned}
$$

Question 11 *We are going to numerically solve the linear optimization problem* (22)*.*

a) *[2]* *Detail what the code below is doing. Why do we use the macro* ***linpro****? Copy the code into a file named* tp_q4.sce*.*

b) *[3]* *Solve a numerical version with* $S = 3$ *scenarios. What do you observe regarding convergence of the Uzawa algorithm? Can you explain why?*

```
// formulation on a fan with linear cost
// Uzawa does not work
```

```
S=3;// Number of random scenarios
q0m=30;// max capacity for q0
D=[15;20;50];
Pr=[0.2;0.6;0.2];// Probabilities of Demand
rho=0.1;
// Constants used in the cost function
ll0=2;ll1=5;

// Uzawa iterations when the dualized constraints are the S constraints
//  Q0(ss) = sum(Pr.*Q0);

Q0=zeros(S,1);
Q1=zeros(S,1);
f0=zeros(S,1);

lambda=zeros(S,1);
for it=0:30 do
  // decomposed minimizations
  for ss=1:S do
    // inequality constraints (bounds)
    // bounds on (Q0s,Q1s)
    ll=[ll0*Pr(ss);ll1*Pr(ss)];// cost coefficients
    Ai=[-eye(2,2);eye(1,2)];
    bi=[zeros(2,1);q0m];
    // equality constraints i.e production equals demand
    Ae=[1,1];
    be=[D(ss)];
    A=[Ae;Ai];b=[be;bi];
    //
    cc=ll+Pr(ss)*[lambda(ss)-sum(Pr .*lambda);0];
    [xopt,lopt,fopt]=linpro(cc,A,b,[],[],size(be,'*'));
    Q0(ss)=xopt(1);
    Q1(ss)=xopt(2);
    f0(ss)=fopt;
    printf("Dual function %f\n",sum(f0));
  end
  Q0bar=sum(Pr .*Q0);
  lambda=lambda+rho*(Pr .*(Q0-Q0bar));
end
```

5.2 Dualization of non-anticipativity constraints

To bypass the difficulty in applying Uzawa algorithm with linear costs, we use a "trick" consisting in introducing new decision variables $\overline{Q_0}$ and $\left(Q_{0,s}\right)_{s\in\mathbb{S}}$, instead of the single decision variable Q_0, and we write

$$Q_{0,s} = \overline{Q_0} \ , \ \ \forall s \in \mathbb{S} \ . \tag{23}$$

These equalities are another form of the *non-anticipativity* constraints. Indeed, the equations (23) express that the decision at stage $t = 0$ cannot anticipate the future, hence cannot depend on the scenario s. We will treat these constraints by duality.

Therefore, the stochastic optimization problem (3) now becomes

$$\begin{aligned}
&\min_{\overline{Q_0},\, Q_{0,s},\, Q_{1,s},\, \forall s\in\mathbb{S}} \sum_{s\in\mathbb{S}} \pi_s \left[l_0 Q_{0,s} + l_1 Q_{1,s} \right] && && \text{(24a)}\\
&\text{s.t} \qquad 0 \leq Q_{0,s} \leq Q_0^{\sharp} && \forall s \in \mathbb{S} && \text{(24b)}\\
&\qquad\quad\ \ 0 \leq Q_{1,s} && \forall s \in \mathbb{S} && \text{(24c)}\\
&\qquad\quad\ \ D_{1,s} = Q_{0,s} + Q_{1,s} && \forall s \in \mathbb{S} && \text{(24d)}\\
&\qquad\quad\ \ Q_{0,s} = \overline{Q_0} && \forall s \in \mathbb{S} && \text{(24e)}
\end{aligned}$$

By the assumption (1) that there are no scenarios with zero probability ($\pi_s > 0$), we replace each equality in (24e) by the equivalent one

$$\pi_s Q_{0,s} = \pi_s \overline{Q_0} \ , \ \ \forall s \in \mathbb{S} \ . \tag{25}$$

We attach, to each equality above a *multiplier* $\lambda_{0,s}$. We put

$$Q_0 = \left(Q_{0,s}\right)_{s\in\mathbb{S}} \ , \ \ Q_1 = \left(Q_{1,s}\right)_{s\in\mathbb{S}} \ , \ \ \lambda = \left(\lambda_{0,s}\right)_{s\in\mathbb{S}} \ . \tag{26}$$

5.3 Augmented Lagrangian and obstacles to decomposition

5.4 Progressive Hedging algorithm (quadratic solver)

Question 12 *We are going to numerically solve the linear optimization problem* (24) *by the Progressive Hedging algorithm.*

a) *[3]* *Detail what the code below is doing. Why do we use the macro* **`quapro`**? *Explain the two roles of the new parameter* **`rr`**. *Copy the code into a file named* tp_q5.sce.

b) *[2]* *Solve a numerical version with $S = 3$ scenarios. What do you observe regarding convergence of the Uzawa algorithm? Can you explain why?*

```
// formulation on a fan
// with linear cost and augmented lagrangian
```

```
S=3;// Number of random scenarios
q0m=30;// max capacity for q0
D=[15;20;50];
Pr=[0.2;0.6;0.2];// Probabilities of Demand
// Constants used in the cost function
ll0=2;ll1=5;
// Constant used both as a quadratic term and as a gradient step
rr=0.1;

// Uzawa iterations when the dualized constraints are the S constraints
//   Q0s = sum(Pr.*Q0)

Q0=zeros(S,1);
Q1=zeros(S,1);
f0=zeros(S,1);

lambda=zeros(S,1);
Q0bar=0;
for it=0:300 do
  // decomposed minimizations
  // we alternate minimization
  for ss=1:S do
    // inequality constraints (bounds)
    // bounds on (Q0s,Q1s)
    ll=[ll0*Pr(ss);ll1*Pr(ss)];// cost coefficients
    Ai=[-eye(2,2);eye(1,2)];
    bi=[zeros(2,1);q0m];
    // equality constraints i.e production equals demand
    Ae=[1,1];
    be=[D(ss)];
    A=[Ae;Ai];b=[be;bi];
    KK=rr*Pr(ss)*diag([1,0]);
    cc=ll+[Pr(ss)*lambda(ss);0];
    cc=cc+[-rr*Q0bar*Pr(ss)];
    [xopt,lopt,fopt]=quapro(KK,cc,A,b,[],[],size(be,'*'));
    Q0(ss)=xopt(1);
    Q1(ss)=xopt(2);
    f0(ss)=fopt;
  end
  // updates of Q0bar
  Q0bar=sum(Pr .*Q0);
  lambda=lambda+rr*(Pr .*(Q0-Q0bar));
```

```
end

// solution (Q0bar,Q1)
Q1=max(0,(D-Q0bar));
// to be compared with tp_q1
```

5.5 Progressive Hedging algorithm (linear solver)

A Additional code for "Formulation on a tree with linear costs"

```
// Formulation on a tree with linear costs.
// Numerical resolution by linear programming
// using nsp linprog (glpk).

// Constant initialization
S=3;// Number of random scenarios
q0m=30;// max capacity for q0

// Demand
if S==3 then
  D=[15;20;50];
  Pr=[0.2;0.6;0.2];// Probabilities of Demand
else
  D=grand(S,1,'uin',5,50);
  Pr=grand(S,1,'unf',0,1);
  Pr=Pr ./sum(Pr);// Probabilities of Demand
end

// Constants used in the cost function
ll0=2;ll1=5;

// a revoir pour passer a des contraintes egalité et utiliser lb et ub
c=[ll0;ll1 .*Pr];// cost coefficients

// inequality constraints (bounds on production)
Ai=[-eye(S+1,S+1);eye(1,S+1)];
bi=[zeros(S+1,1);q0m];

// equality constraints, i.e. production equals demand
Ae=[ones(S,1),eye(S,S)];
be=[D];
```

```
// solving by linear
// xopt should be [  15,  0, 5, 35 ] when S=3

if exists('%nsp') then
  // in nsp linprog is glpk
  [xopt,fopt,flag]=linprog(c,Ai,bi,Ae,be);
  // we can also use quapro if available with a 0 quadratic term
  if exists('quapro','callable') then
    A=[Ae;Ai];b=[be;bi];
    [xopt_q,lagr_q,fopt_q]=quapro(zeros(S+1,S+1),c,A,b,[],[],size(be,'*'))
  end
else
  // scicoslab version with linpro
  A=[Ae;Ai];b=[be;bi];
  [xopt,lopt,fopt]=linpro(c,A,b,[],[],size(be,'*'))
end
```

B Additional code for "Formulation on a tree with quadratic convex costs"

```
// Formulation on a tree with quadratic costs
// Numerical resolution by brute force quadratic programming
// using nsp quapro or optim

if exists('%nsp') then
  load_toolbox('quapro');
end

// just to get common data
exec('tp_q1.sce');

// the problem is now quadratic; we use a quadratic solver if available
A=[Ae;Ai];b=[be;bi];
KK0=10;KK1=1;
KK=diag([KK0,KK1*Pr' .*ones(1,S)]);

if exists('%nsp') then
  // we can also use quapro if available with a 0 quadratic term
  if exists('quapro','callable') then
    [xopt_q,lagr_q,fopt_q]=quapro(KK,c,A,b,[],[],size(be,'*'))
```

```
  end
else
  // scicoslab version with quapro
  [xopt,lopt,fopt]=quapro(KK,c,A,b,[],[],size(be,'*'))
end

// BEWARE: using a global optim if quapro is not available
// we eliminate the equality constraint to use optim on q0

function [f,g,ind]=costf(q0,ind)
  q1=max(0,(D-q0));
  // expression of the cost
  f=sum(Pr .*(l10*q0+l11*q1))+(1/2)*[q0;q1]'*KK*[q0;q1]
  // expression of the cost derivative
  g=sum(Pr .*(l10-l11*(D-q0 >= 0)))+KK(1,1)*q0+-sum((D-q0 >= 0) .*(KK(2:$,2:$)*q1))
endfunction

if exists('%nsp') then
  Q0=optim(costf,0,xinf = 0,xsup = q0m)
else
  [fo,Q0]=optim(costf,'b',0,q0m,0)
end

// solution (Q0,Q1)
Q1=max(0,(D-Q0))
```