Online Optimization of an Energy Storage System Profitability with Reserve Commitment

A Competitive Analysis Approach

Abdessamad El Kabid, Ahmed Mansour Bourassine, Lucas Levy

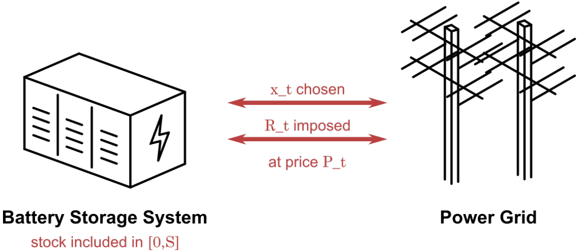
December 17th, 2024

EA Recherche MAP 511

Plan

1. Problem formulation and chosen approach
2. Contributions
3. Proofs

Industrial problem



The operating cost is $\sum_t P_t(x_t + R_t)$

Initial paper (Sossou Edou 2024), PGMO project.
Stochastic approach, tries to minimize $\mathbb{E}[\sum_t P_t x_t]$

Problem formulation

The problem in its general form is

$$
\begin{array}{ll}
\text{minimize} & \displaystyle\sum_{t=1}^{T} P_t(x_t + R_t) \\
\text{s.t.} & \\
& \begin{array}{ll}
s_t = s_{t-1} + x_t + R_t & \forall t \in [T] \\
s_t \in [0, S] & \forall t \in [T] \\
x_t + R_t \in [-C^{\max}, +C^{\max}] & \forall t \in [T] \\
x_t \text{ depends only on } R_1, \ldots, R_{t-1}, P_1, \ldots, P_t & \forall t \in [T]
\end{array}
\end{array}
$$

The R_t and P_t are unknown beforehand but are respectively in the intervals $[-R^{\max}, +R^{\max}]$ and $[p^{\min}, p^{\max}]$ with $p^{\min} \geqslant 0$

Competitive Analysis

Given an online decision algorithm ALG and an instance I of the problem, we denote

- $ALG(I)$ the resulting cost when the algorithm ALG is applied on I.
- $OPT(I)$ the optimal value of the cost for the offline problem on the instance I (the whole instance is known beforehand).

Competitive Analysis

Given an online decision algorithm ALG and an instance I of the problem, we denote $ALG(I)$ the resulting cost when the algorithm ALG is applied on I and $OPT(I)$ the optimal value of the cost for the offline problem on the instance I.

Definition

Let $c \in \mathbb{R}$ and ALG an online decision algorithm. ALG is c-competitive if for all instances I, $ALG(I) \leqslant c\, OPT(I)$

Definition Competitive ratio

The competitive ratio of ALG is

$$\inf\{c \in \mathbb{R};\ ALG \text{ is } c\text{-competitive}\}$$

Acceptable algorithm

We call an algorithm *acceptable* if for every instance I, the inequalities $s_t \in [0, S]$ and $|x_t + R_t| \leq C^{\max}$ are checked at every time t.

Proposition

An algorithm is acceptable if and only if for every time t we have

$$x_t \leqslant \min(C^{\max} - R^{\max}, S - s_{t-1} - R^{\max})$$

$$x_t \geqslant \max(-C^{\max} + R^{\max}, -s_{t-1} + R^{\max})$$

Remark: An unacceptable algorithm has a competitive ratio of $+\infty$.

Plan

1. Problem formulation and chosen approach
2. Contributions
   - Constant prices
   - Known prices, unknown demand
   - No demand, unknown prices
   - Worst-case cost
3. Proofs

Plan

Constant prices

With constant positive prices, the problem can be written as

$$
\begin{aligned}
\text{minimize} \quad & \sum_{t=1}^{T} (x_t + R_t) \\
\text{s.t.} \quad & \\
& s_t = s_{t-1} + x_t + R_t && \forall t \in [T] \\
& s_t \in [0, S] && \forall t \in [T] \\
& x_t + R_t \in [-C^{\max}, +C^{\max}] && \forall t \in [T] \\
& x_t \text{ depends only on } R_1, \dots, R_{t-1} && \forall t \in [T]
\end{aligned}
$$

Intuition: $\sum_t (x_t + R_t) = s_T - s_0$ so we just want to minimize s_T, i.e sell as much as possible.

Constant prices

Algorithm Minimizing x

---

for all $t \in [T]$ **do**

$x_t = \max(R^{\max} - C^{\max}, R^{\max} - s_{t-1})$

end for

Theorem 1, Optimal algorithm for constant prices

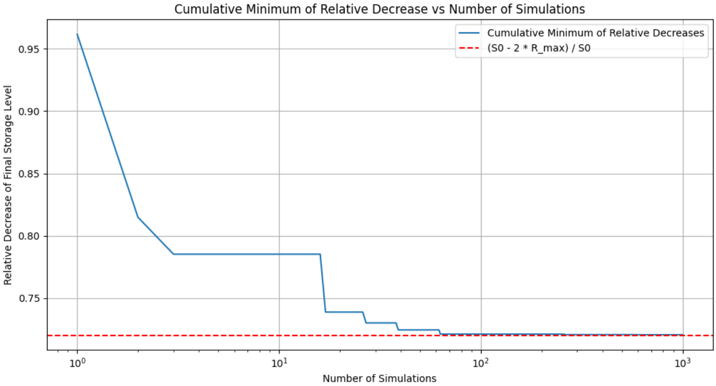
If $\min(C^{\max}, S) \geqslant 2R^{\max}$, then the previous algorithm is acceptable, and has a competitive ratio of

$$c = \frac{\max(2R^{\max} - s_0, -T(C^{\max} - 2R^{\max}))}{\max(-s_0, -TC^{\max})}$$

This is the best competitive ratio that can be achieved.

Constant Price : Numeric validation

- Simulate many instances
- Keep track of the minimum score (compared to optimal case)
- plot the results ; convergence to the true competitiveness ratio



Plan

Known prices Unknown Rt

With known prices beforehand, the problem can be written as

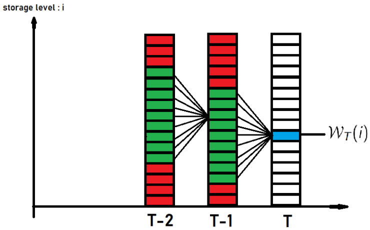
$$
\begin{aligned}
\text{minimize} \quad & \sum_{t=1}^{T} P_t(x_t + R_t) \\
\text{s.t.} \quad & \\
& s_t = s_{t-1} + x_t + R_t && \forall t \in [T] \\
& s_t \in [0, S] && \forall t \in [T] \\
& x_t + R_t \in [-C^{\max}, +C^{\max}] && \forall t \in [T] \\
& x_t \text{ depends only on } R_1, \ldots, R_{t-1} && \forall t \in [T] \\
& P_t \in \mathbb{R}^T
\end{aligned}
$$

Known prices Unknown Rt

we define

$$\mathcal{W}_T(i) = \min_{x \in E} \left(\mathcal{W}_{T-1}(x) + gain(x, i) \right)$$

- E is the feasible domain
- $\mathcal{W}_T(i)$ is the best worst score at T for storage level i
- gain(x,i) is the worst gain when going from storage level x to i



Known prices Unknown Rt

now we can write the dynamic programming equation

Theorem , Best worst-case gain

To obtain such W_T it must obey the following equation:

$$\mathcal{W}_T(i) = \min_{x \in E} \left\{ \begin{aligned} \max \Big[& \mathcal{W}_{T-1}(x) + (x - R_m - i)P_T + P_{t-1} \cdot R_m, \\ & \mathcal{W}_{T-1}(x) + (x + R_m - i)P_T - P_{t-1} \cdot R_m \Big] \end{aligned} \right\}$$

.

Known prices Unknown Rt

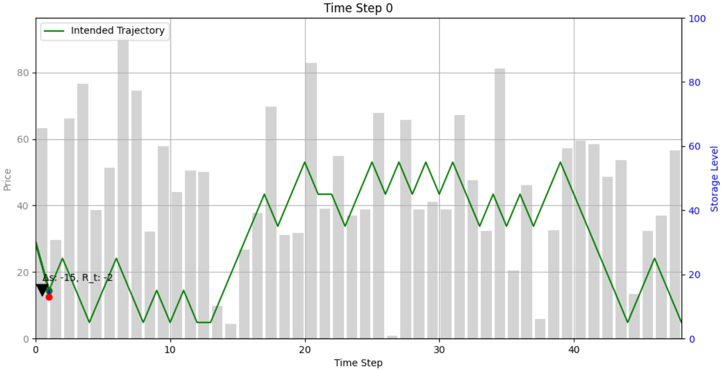


Figure: buy/sell strategy

Known prices Unknown Rt

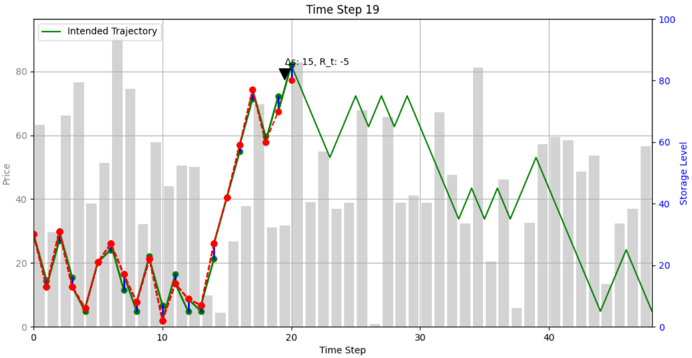


Figure: update as new Rt gets revealed

Known prices Unknown Rt

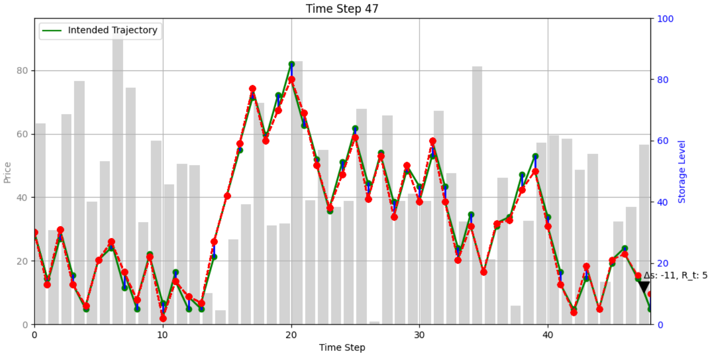


Figure: End of simulation

Plan

No demand, unknown prices

Theorem 2, Competitve ratio of OTA

Suppose that the prices are lower and upper bounded by L and U, respectively. Let $\theta := U/L$. In the case where no demand is observed, suppose that $S_0 + TC^{max} - S \leqslant 0$ and that $S_0 \leqslant C^{max}$.

- Algorithm 1 with ϕ^* as the threshold function achieves an optimal competitive ratio of $(1 + \ln(\theta))$.

Online Threshold Algorithm

Consider the following algorithm:

Algorithm 1 OTA with Threshold Function ϕ (OTA_ϕ) for DC

Input: Threshold function $\phi := \{\phi(\cdot)\}$, and $w^{(1)} = 0$;
While item i arrives and verify the constraints
For each $j \leqslant i$ (do one step in the solving process of SubPj)

$$y(j,i) = \begin{cases} \phi_j^{-1}(P_i) - w_j^{(i)} & P_i \geqslant \phi_j(w_j^{(i)}) \\ 0 & P_i < \phi_j(w_j^{(i)}) \end{cases}$$

update the utilization $w_j^{(i+1)} = w_j^{(i)} + y(j,i)$,
$y_i = \sum_{j=1}^{i} y(j,i)$
$x_i = C^{max} - y_i$

Algorithm

$$
\begin{array}{cccc|c}
y_{1,1} & & & & y_1 \\
y_{1,2} & y_{2,2} & & & y_2 \\
y_{1,3} & y_{2,3} & y_{3,3} & & y_3 \\
\vdots & & \ddots & & \\
y_{1,n} & \cdots & & y_{n,n} & y_n
\end{array}
$$

Threshold function

ϕ^*

Let ϕ^* defined as :

$$\phi^*(w) = \begin{cases} L & w \in [0, \beta^*) \\ Le^{(1+\ln\theta)w/b-1} & \text{if } w \in [\beta^*, b] \\ \infty & \text{if } w > b \end{cases}, \qquad (1)$$

where $\beta^* = \frac{b}{\alpha_{\phi^*}}$ is the utilization threshold, where $b = C^{max}$

Plan

1. Problem formulation and chosen approach

2. Contributions
   - Constant prices
   - Known prices, unknown demand
   - No demand, unknown prices
   - Worst-case cost

3. Proofs
   - Constant prices
   - No demand, unkown prices
   - Known prices , Unknown demand
   - Worst-case value

Worst-case cost

Definition Worst-case cost

For a minimization problem and a given algorithm ALG, the worst-case cost is

$$\sup\{ALG(I);\ I \text{ an instance of the online problem}\}$$

The algorithm is guaranteed to do better (i.e lower) than its worst-case cost.
The better the algorithm, the lower the worst-case cost is.
Concept close to competitive ratio but different and easier to work with.

Dynamic programming formulation for best worst-case cost

Define by induction the functions $W_1, \ldots, W_{T+1}$:

$$W_t(s) = \sup_{p \in [p^{\min}, p^{\max}]} \inf_{|x| \leqslant C^{\max} - R^{\max}} \sup_{|r| \leqslant R^{\max}} \big(p(x+r) + W_{t+1}(s+x+r) \big)$$

$$W_t(s \notin [0, S]) = +\infty \quad \forall t; \quad W_{T+1}(s \in [0, S]) = 0$$

Theorem 3, Best worst-case cost

An algorithm choosing at time t

$$x_t \in \operatorname*{arg\,min}_{|x| \leqslant C^{\max} - R^{\max}} \sup_{|r| \leqslant R^{\max}} \big(P_t(x + r) + W_{t+1}(s_{t-1} + x + r) \big)$$

has a worst-case cost of $W_1(s_0)$. This is the lowest worst-case cost that can be achieved.

Propositions on recursive functions W

We can have the explicit expression of W_T (after that it is getting difficult)

$$W_T(s) = \begin{cases} p^{\max}(2R^{\max} - s) & \text{if } s \leqslant 2R^{\max} \\ p^{\min}(2R^{\max} - s) & \text{if } s \in [2R^{\max}, C^{\max}] \\ p^{\min}(2R^{\max} - C^{\max}) & \text{if } s \geqslant C^{\max} \end{cases}$$

Proposition

For all $t \leqslant T$ and $s \in [0, S]$,

$$W_t(s) \geqslant p^{\min}(2R^{\max} - s)$$

Propositions on recursive functions W

Proposition Convexity of W

For all $t \leqslant T + 1$, W_t is convex.

Several remarks can be made from this:

- $r \mapsto P_t(x + r) + W_{t+1}(s_{t-1} + x + r)$ is convex, so the worst case demand R_t is always either $-R^{\max}$ or $R^{\max}$
- $x \mapsto \sup_{|r| \leqslant R^{\max}} \big(P_t(x + r) + W_{t+1}(s_{t-1} + x + r)\big)$ is convex too, so choosing x_t at time t amounts to solving a convex optimization problem (provided that W_{t+1} is known).

Plan

1. Problem formulation and chosen approach
2. Contributions
3. Proofs
   - Constant prices
   - No demand, unkown prices
   - Known prices , Unknown demand
   - Worst-case value

Plan

Elements of Theorem 1's proof

Theorem 1, Optimal algorithm for constant prices

If $\min(C^{\max}, S) \geqslant 2R^{\max}$, then the algorithm choosing at time t

$$x_t = \max(R^{\max} - C^{\max}, R^{\max} - s_{t-1})$$

is acceptable, and has a competitive ratio of

$$c = \frac{\max(2R^{\max} - s_0, -T(C^{\max} - 2R^{\max}))}{\max(-s_0, -TC^{\max})}$$

This is the best competitive ratio that can be achieved.

Elements of Theorem 1's proof

- The algorithm is acceptable.
- The offline problem has an optimal value of

$$OPT(R) = \max(-TC^{\max}, -s_0)$$

  regardless of the values taken by R.
- The algorithm described previously can always do better than

$$ALG(R) \leqslant \max(2R^{\max} - s_0, -T(C^{\max} - 2R^{\max}))$$

- For every acceptable algorithm ALG', the instance $R^* = (R^{\max})_{1 \leq t \leq T}$ satisfies

$$ALG'(R^*) \geqslant \max(2R^{\max} - s_0, -T(C^{\max} - 2R^{\max}))$$

Plan

Transformed Problem

After a variable transformation $y_t = C^{max} - x_t$, we obtain an equivalent formulation for the initial problem :

$$
\begin{aligned}
\text{maximize} \quad & \sum_{t=1}^{T} P_t y_t \\
\text{subject to} \quad & S_0 + tC^{max} - S \leqslant \sum_{i=1}^{t} y_i \leqslant S_0 + tC^{max}, \quad \forall t \quad \text{(SPeq)} \\
& 0 \leqslant y_t \leqslant 2C^{max}, \quad \forall t \\
& y_t \text{ depends only on past}, \ \forall t \in [T].
\end{aligned}
$$

Simplified Problem Under Assumptions

Under the assumptions $S_0 + TC^{max} - S \leqslant 0$ and $S_0 \leqslant C^{max}$, the problem reduces to:

$$\begin{aligned} \text{maximize} \quad & \sum_{t=1}^{T} P_t y_t \\ \text{subject to} \quad & \sum_{i=1}^{t} y_i \leqslant S_0 + tC^{max} =: b(t), \ \forall t \\ & 0 \leqslant y_t, \ \forall t \\ & y_t \text{ depends only on past, } \forall t \in [T]. \end{aligned} \tag{SP'}$$

Decoupling the Problem

We consider the following decoupled problem (DC):

$$
\begin{aligned}
\text{maximize} \quad & \sum_{j=1}^{n} \sum_{i=j}^{n} p(i) y(j,i) \\
\text{subject to} \quad & \sum_{i=j}^{n} y(j,i) \leqslant b(j) - b(j-1), \quad j = 1, \ldots, n \\
\text{variables} \quad & y(j,i) \geqslant 0, \; i \geqslant j.
\end{aligned}
\tag{DC}
$$

Equivalence

Lemma

Any optimal solution to (DC) yields an optimal solution to (SP')
by the construction:

$$y^*(i) = \sum_{j=1}^{i} y^*(j, i).$$

Moreover, both problems share the same optimal objective value.

Resulting Subproblems

Each subproblem for a fixed j is:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=j}^{n} p(i) y(j,i) \\ \text{subject to} \quad & \sum_{i=j}^{n} y(j,i) \leqslant b(j) - b(j-1), \\ & y(j,i) \geqslant 0, \ i \geqslant j. \end{aligned} \tag{SubPj}$$

Each of these subproblems corresponds to a "Generalized One-Way Trading" (GOT) problem, which we know how to solve with an online threshold algorithm with an optimal competitve ratio of $(1 + \ln(\theta))$ [2].

From the subproblems to the original

(GOT)

$\max_{y_n} \quad \sum_{n\in\mathcal{N}} g_n(y_n), \quad \text{s.t. } \sum_{n\in\mathcal{N}} y_n \le C, \quad 0 \le y_n \le D_n$

Algorithm Online Threshold-Based Algorithm

---

input: threshold function $\phi := \{\phi_m(\cdot)\}_{m\in\mathcal{M}}$, and $w_m^{(1)} = 0$

While item n arrives determine $\boldsymbol{y}_n^*$ by solving the pseudo-utility maximization problem

$$\boldsymbol{y}_n^* = \arg\max_{\boldsymbol{y}_n\in\mathcal{Y}_n} \quad g_n(\boldsymbol{y}_n) - \int_{w^{(n)}}^{w^{(n)}+y_n} \phi(u)du; \tag{2}$$

update the utilization $w^{(n+1)} = w^{(n)} + y_n^*$.

Competitive Ratio

Theorem

Suppose that $S_0 + TC^{max} - S \leqslant 0$ and that $S_0 \leqslant C^{max}$. When the threshold function of OTA_ϕ for SubPj is

$$\phi_j^*(w) = \begin{cases} L & w \in [0, \beta_j^*) \\ Le^{(1+\ln\theta)w/b_j - 1} & \text{if } w \in [\beta_j^*, b_j] \\ \infty & \text{if } w > b_j \end{cases}, \tag{3}$$

where $\beta_j^* = \frac{b_j}{\alpha_{\phi^*}}$ is the utilization threshold, where $b_j = b(j) - b(j-1)$, the competitive ratio of OTA_{ϕ^*} is $\alpha_{\phi^*} = 1 + \ln\theta$.

Online Threshold Algorithm

We thus construct the algorithm 1 for DC

Extensions and Conjectures

If the assumptions do not hold, the problem can still be decoupled but will involve bounds from both sides. In such a scenario, we conjecture a competitive ratio under $2(1 + \ln \theta)$.

Paths of progress:

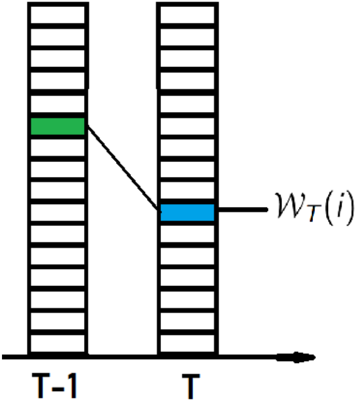
- Improve bound estimates step-by-step if price structure is can be learned.
- Use online algorithms for fractional knapsack in **random order model** to achieve competitive ratios under certain conditions.

Plan

Numeric implementation of Known prices, Unknown Rt

Let's specify that $E = A_i \cap F$:

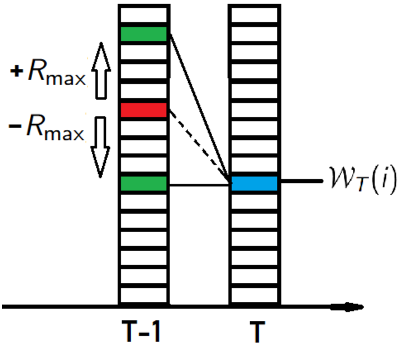
- F : acceptable ending the feasible storage levels adjusted for random perturbations: $[R_{\max}, S - R_{\max}]$
- A_i : valid x positions where $\{\,|x - i| \leqslant C_{\max} - 2R_{\max}\}$



Numeric implementation of Known prices, Unknown Rt

Let's specify that $E = A_i \cap F$:

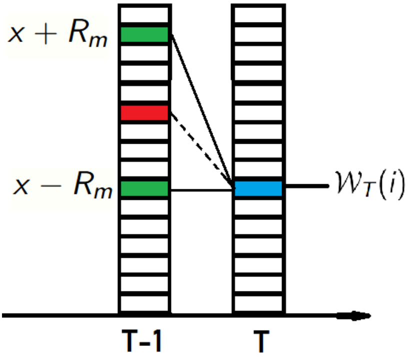
- F : acceptable ending the feasible storage levels adjusted for random perturbations: $[R_{\max}, S - R_{\max}]$
- A_i : valid x positions where $\{\, |x - i| \leqslant C_{\max} - 2R_{\max}\}$



Numeric implementation of Known prices, Unknown Rt

Let's specify that $E = A_i \cap F$:

- F : acceptable ending the feasible storage levels adjusted for random perturbations: $[R_{\max}, S - R_{\max}]$
- A_i : valid x positions where $\{\, |x - i| \leqslant C_{\max} - 2R_{\max} \}$



Implementation Details

- A table was used to store the maximum worst-case gain $W_T(i)$ for each storage level i and time step T.
- Working forward simplifies the implementation:
  - At each time step t, all values for W_{t+1} are already computed.
  - we save the previous optimal state to recursively search for the optimal decision path later.
- Transitions between states were validated against E
- The table-filling process proceeds iteratively until all time steps are covered.

Implementation Details

Algorithm:

1. For each time step $t = 0$ to $T - 1$:
   1. For each storage level s:
      1. For each possible next storage level i such that $|i - s| \leqslant C_{\max} - 2R_{\max}$:
      2. Compute the two possible gains:

$$\text{gain1} = \text{gain\_matrix}[t][s] + (s - R_{\max} - i)P_t + P_{t-1} \cdot R_{\max}$$

$$\text{gain2} = \text{gain\_matrix}[t][s] + (s + R_{\max} - i)P_t - P_{t-1} \cdot R_{\max}$$

      3. Determine the worst-case gain:

$$\text{worse\_gain} = \max(\text{gain1}, \text{gain2})$$

      4. If $\text{worse\_gain} < \text{gain\_matrix}[t+1][i]$:
      5. Update $\text{gain\_matrix}[t+1][i] = \text{worse\_gain}$
      6. Update $\text{prev\_state}[t+1][i] = s$
2. Trace back the optimal path from $t = T$ to $t = 0$ using *prev_state*.

Output: - Optimal storage levels and maximum worst-case gain.

Plan

Elements of Theorem 3's proof

Define by induction the functions $W_1, \ldots, W_{T+1}$:

$$W_t(s) = \sup_{p \in [p^{\min}, p^{\max}]} \inf_{|x| \leqslant C^{\max} - R^{\max}} \sup_{|r| \leqslant R^{\max}} \big(p(x+r) + W_{t+1}(s+x+r)\big)$$

$$W_t(s \notin [0, S]) = +\infty \quad \forall t; \quad W_{T+1}(s \in [0, S]) = 0$$

Theorem 3, Best worst-case cost

An algorithm choosing at time t

$$x_t \in \operatorname*{arg\,min}_{|x| \leqslant C^{\max} - R^{\max}} \sup_{|r| \leqslant R^{\max}} \big(P_t(x+r) + W_{t+1}(s_{t-1} + x + r)\big)$$

has a worst-case cost of $W_1(s_0)$. This is the lowest worst-case cost that can be achieved.

Elements of Theorem 3's proof

The key idea is to write W_t as

$$W_t(s) = \inf_{\substack{X_i \in \mathcal{X}_i^t \\ i \in [t,T]}} \sup_{\substack{R_t,\dots,R_T \in [\pm R^{\max}] \\ P_t,\dots,P_T \in [p^{\min}, p^{\max}]}} \left(\sum_{i=t}^{T} P_i(X_i(P_{[t,i]}, R_{[t,i-1]}, s) + R_i) \right)$$

where $P_{[t,i]} = (P_t, \dots, P_i)$ and
$\mathcal{X}_i^t \subset \{X : \mathbb{R}^{i-t+1} \times \mathbb{R}^{i-t} \times \mathbb{R} \to [\pm(C^{\max} - R^{\max})\}$
We use induction reasoning and the following lemma

Lemma Minimax

Let X, Y be sets and $F\colon X \times Y \to \mathbb{R}$. Then

$$\inf_{h\colon X \to Y} \sup_{x \in X} F(x, h(x)) = \sup_{x \in X} \inf_{y \in Y} F(x, y)$$

Proof of the convexity of W

We use two convexity lemmas

Lemma Danskin's lemma

Let X be a convex set, Y be a set and $F : X \times Y \to \mathbb{R}$. If for all $y \in Y$, $x \mapsto F(x, y)$ is convex, then $x \mapsto sup_{y \in Y} F(x, y)$ is convex.

Lemma Convexity of the optimal value function

Let X and Y be convex sets and $F : X \times Y \to \mathbb{R}$. If F is jointly convex, then $x \mapsto \inf_y F(x, y)$ is convex.

Proof of the convexity of W

Proof.

We proceed by reverse induction. Let t be such that W_{t+1} is convex. Then, for all p, r, $(s, x) \mapsto p(x + r) + W_{t+1}(s + x + r)$ is jointly convex.

By Danskin's lemma, for all p,

$(s, x) \mapsto \sup_r p(x + r) + W_{t+1}(s + x + r)$ is convex.

By convexity of the optimal value function lemma, for all p,

$s \mapsto \inf_x \sup_r p(x + r) + W_{t+1}(s + x + r)$ is convex.

Finally, by Danskin's lemma, $s \mapsto W_t(s)$ is convex. $\square$

References I

Yang, X. & Wang, S., "Online Linear Programming: Competitive Algorithms for Real-Time Decisions."

Sun, X., Qi, Q., Wierman, A., "Competitive Algorithms for Online Multiple-Choice Knapsack and One-Way Trading."