Online Algorithms and the Primal-Dual Approach
Road Map

1) Online Algorithms
2) Performance Evaluation of Online Algorithms
3) Ski Rental Problem
4) Linear Programming Framework
5) Duality Principle
6) Ski Rental via Primal Dual Approach
7) References
Online Algorithms

- Input is given in pieces over time, where each piece is called request.

- Request sequence: \( \{p_1, p_2, \ldots, p_n\} \).

- Upon arrival of request:
  - The algorithm has to serve the request.
  - Decision for previous requests can’t be changed/revoked.
  - Make the decision with uncertainty (future is unknown!)
Offline algorithm has all the input pieces beforehand.

Online problems:
Ski rental, set-cover, routing, k-server load balancing and weighted caching problem.
Road Map

1) Online Algorithms
2) Performance Evaluation of Online Algorithms
3) Ski Rental Problem
4) Linear Programming Framework
5) Duality Principle
6) Ski Rental via Primal Dual Approach
7) References
For every input sequence $p_1, p_2, \ldots, p_n$:

Compare the cost of the online algorithm to the cost of an optimal offline algorithm.

**Competitive factor** of online algorithm $A$ is $\alpha$ if for every input sequence $p_1, p_2, \ldots, p_n$:

$$A(p_1, p_2, \ldots, p_n) \leq \alpha \cdot \text{OPT} (p_1, p_2, \ldots, p_n) + O(1)$$
Road Map

1) Online Algorithms
2) Performance Evaluation of Online Algorithms
3) Ski Rental Problem
4) Linear Programming Framework
5) Duality Principle
6) Ski Rental via Primal Dual Approach
7) References
The Ski Rental Problem

- Buying costs $B.
- Renting costs $1 per day.

**Problem:**
Number of ski days is not known in advance.

**Goal:** Minimize the total cost.
Ski Rental: Analysis

- Online algorithm $\to$ rent for $m$ days then buy the skies.

- The **optimal** choice of $m$? $m=B$

- The competitive factor ?
  - if the number of ski days $\leq B$, cost $A = \text{cost OPT}$
  - if the number of ski days $> B$, cost $A = 2 \times \text{cost OPT}$.

- Consider **oblivious** adversary:
  - he knows $A$ but doesn’t know the decision made by $A$. 
Road Map

1) Online Algorithms
2) Performance Evaluation of Online Algorithms
3) Ski Rental Problem
4) Linear Programming Framework
5) Duality Principle
6) Ski Rental via Primal Dual Approach
7) References
Linear Programming (LP)

- An optimization technique.
- Canonical form:
  \[ \text{Min } c^T x \]
  Subject to \[ Ax \geq b \]
  and \[ x \geq 0 \]
- Example:
  \[ \text{Min } x_1 - 3x_2 + 2x_3 \]
  \[ x_1 - x_2 + 7x_3 \leq 2 \]
  \[ x_2 - x_3 \geq 0 \]
  \[ x_1 + 3x_2 + x_3 \geq -3 \]
- Polynomial time solvable.
Solving LP problems

1. The basic algorithm (Guesses !)
2. The graphical solution: https://www.youtube.com/watch?v=gbL3vYq3cPk
3. The simplex algorithm: https://www.youtube.com/watch?v=F6myxAIW0wU
4. LINDO! http://www.lindo.com
Road Map

1) Online Algorithms
2) Performance Evaluation of Online Algorithms
3) Ski Rental Problem
4) Linear Programming Framework
5) Duality Principle
6) Ski Rental via Primal Dual Approach
7) References
Duality Principle

\[
\begin{align*}
\text{Min } c^T x & \quad \text{(primal)} \\
Ax & \geq b \\
x & \geq 0
\end{align*}
\]

\[
\begin{align*}
\text{Max by } & \quad \text{(dual)} \\
A^T y & \leq c \\
y & \geq 0
\end{align*}
\]

Minimize \( z = 3x_1 + 2x_2 - 5x_3 + x_4 \)
subject to \( x_1 + x_2 + x_3 + x_4 = 1 \)
\(-x_1 + x_2 - 3x_3 - x_4 \leq 2 \)
\( x_2 \geq 0 \)

\( w = \) maximize \( y_1 - 2y_2 \)
subject to
\( y_1 + y_2 = 3 \)
\( y_1 - y_2 \leq 2 \)
\( y_1 + 3y_2 = -5 \)
\( y_1 + y_2 = 1 \)
\( y_2 \geq 0 \).
Primal Dual Relationship: Weak Duality

\[ \sum_{i=1}^{n} c_i x_i \geq \sum_{j=1}^{m} b_j y_j. \]

Proof:

\[ \sum_{i=1}^{n} c_i x_i \geq \sum_{j=1}^{m} \left( \sum_{i=1}^{n} a_{ij} y_j \right) x_i \]

\[ = \sum_{j=1}^{m} \left( \sum_{i=1}^{n} a_{ij} x_i \right) y_j \]

\[ \geq \sum_{j=1}^{m} b_j y_j, \]

(1)
The gap between the optimal primal solution and the optimal dual solution is zero:

\[ \text{the optimal primal} = \text{the optimal dual}. \]
Let \( x = (x_1, x_2, \ldots, x_n) \)
\[ y = (y_1, y_2, \ldots, y_m) \]

Suppose \( \exists \alpha, \beta \) s.t:

- if \( x_i > 0 \), then \( c_i/\alpha < A^T y_i < c_i \).
- if \( y_i > \) then \( b_i < A x_i < \beta b_i \).

\[ c^T x \leq \alpha \beta b^T y \]

Feasible Solution
Online Primal-Dual Approach

**Advantages:**

1. **Generic** ideas and algorithms applicable to many online problems.
2. **Linear Program** helps detecting the difficulties of the online problem.
3. **General recipe** for the design and analysis of online algorithms.
4. **Competitiveness** with respect to a fractional optimal solution.
Road Map

1) Online Algorithms
2) Performance Evaluation of Online Algorithms
3) Ski Rental Problem
4) Linear Programming Framework
5) Duality Principle
6) Ski Rental via Primal Dual Approach
7) References
Ski Rental – Via Primal Dual Approach

Where:

\[ x = \begin{cases} 
1 & \text{Buy} \\
0 & \text{Don't Buy} 
\end{cases} \]

\[ z_i = \begin{cases} 
1 & \text{Rent on day } i \\
0 & \text{Don't rent on day } i 
\end{cases} \]

\[
\min Bx + \sum_{i=1}^{k} z_i
\]

Subject to:

For each day \( i \):

\[ x + z_i \geq 1 \quad \text{(either buy or rent)} \]

\( x, z_i \in \{0, 1\} \)
Formulating Primal (Cover) Linear Program

- Identify decision variables.
- Write the objective function.
- Write the constraints.
- Write the non negative restrictions.

Maximize $z = \sum_{j=1}^{n} c_j x_j$,

subject to:

$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad (i = 1, 2, \ldots, m),$

$x_j \geq 0 \quad (j = 1, 2, \ldots, n).$
Motivate the Dual (Packing) Program

- In primal constraint, multiply each inequality by $y_i$:
  $$y_i \sum_{j=1}^{n} a_{ij} x_j \leq b_i y_i$$
  $$y^T A x \leq y^T b$$
  $$(A^T y)^T x \leq y^T b$$

- Try to find the best lower bound you can find:

  Minimize $v = \sum_{i=1}^{m} b_i y_i,$

  subject to:
  $$\sum_{i=1}^{m} a_{ij} y_i \geq c_j \quad (j = 1, 2, \ldots, n),$$
  $$y_i \geq 0 \quad (i = 1, 2, \ldots, m).$$
Ski Rental Problem – Integer Program

\[ x = \begin{cases} 
1 & \text{Buy} \\
0 & \text{Don't Buy} 
\end{cases} \]

\[ z_i = \begin{cases} 
1 & \text{Rent on day } i \\
0 & \text{Don't rent on day } i 
\end{cases} \]

\[
\min Bx + \sum_{i=1}^{k} z_i
\]

Subject to:

For each day \( i \):

\[ x + z_i \geq \begin{cases} 
1 & \text{(either buy or rent)} \\
0 & \text{otherwise} 
\end{cases} \]

\[ x, z_i \in \{0,1\} \]
To fully capture the problem, we need integrality constraints.

Integral constraints are not allowed in LP.

Relax the LP by allowing $x$ and $z_i$ take fractional values (0,1).

The algorithm will give fractional solution online.

Need to show we can maintain an integral solution online.

Please! We need a solution that will not cost much worse than the fractional one.
Ski Rental – Relaxation

P: Primal Covering

\[
\min Bx + \sum_{i=1}^{k} z_i \\
\text{For each day } i: x + z_i \geq 1 \\
x, z_i \geq 0
\]

D: Dual Packing

\[
\max \sum_{i=1}^{k} y_i \\
y_i \leq 1 \\
\text{For each day } i: \sum_{i=1}^{k} y_i \leq B
\]

Online setting:

• Primal: New constraint arrive each day.
• Dual: New variable added each day.
• Monotonicity: Variables can only be increased.
Ski Rental – Algorithm

P: Primal Covering
\[
\min Bx + \sum_{i=1}^{k} z_i \\
\text{For each day } i: \quad x + z_i \geq 1 \\
x, z_i \geq 0
\]

D: Dual Packing
\[
\max \sum_{i=1}^{k} y_i \\
y_i \leq 1 \\
\text{For each day } i: \quad \sum_{i=1}^{k} y_i \leq B
\]

Initially \( x \leftarrow 0 \)
Each new day (new constraint):
if \( x < 1 \):
\[
\begin{align*}
zi & \leftarrow 1 - x \\
x & \leftarrow x(1 + 1/B) + 1/(c*B)
\end{align*}
\]
\[
yi \leftarrow 1
\]
Approximate Complementary Slackness

- Let $x = (x_1, x_2, \ldots, x_n)$
  $y = (y_1, y_2, \ldots, y_m)$

  Suppose $\exists \alpha, \beta$ s.t:
  - if $x_i > 0$, then $c_i / \alpha < A^T y_i < c_i$
  - if $y_i > 0$, then $b_i < A x_i < \beta b_i$

  $\Rightarrow c^T x \leq \alpha \beta b^T y$

Feasible Solutions
Approximate Complementary Slackness

How can this help?

\[ b^T y \leq \text{OPT}_D \leq \text{OPT}_P \leq c^T x \leq \alpha \beta b^T y \leq \alpha \beta \text{OPT}_P \]

We can say \( c^T x \) is \( \alpha \beta \) competitive.
Analysis of Online Algorithm

Proof of competitive factor:
1. Primal solution is feasible.
2. In each iteration, $\Delta P \leq (1 + 1/c) \Delta D$.
3. Dual is feasible.

Conclusion: Algorithm is $(1 + 1/c)$-competitive

Initially $x \leftarrow 0$

Each new day (new constraint):

if $x < 1$:

- $z_i \leftarrow 1 - x$
- ‘c’ later.
- $x \leftarrow x(1 + 1/B) + 1/(c*B)$
- $y_i \leftarrow 1$
Analysis of Online Algorithm

1. Primal solution is feasible.
   If $x \geq 1$ the solution is feasible.
   Otherwise set: $z_i \leftarrow 1-x$.

2. In each iteration, $\Delta P \leq (1+ 1/c)\Delta D$:
   If $x \geq 1$, $\Delta P = \Delta D = 0$
   Otherwise:
   - Change in dual: 1
   - Change in primal:
     $B\Delta x + z_i = x + 1/c + 1-x = 1+1/c$

Algorithm:
When new constraint arrives, if $x < 1$:
- $z_i \leftarrow 1-x$
- $x \leftarrow x(1+1/B) + 1/c*B$
- $y_i \leftarrow 1$
3. Dual is feasible:

Need to prove \( \sum_{i=1}^{k} y_i \leq B \)

We prove that after B days \( x \geq 1 \)

\( x \) is a sum of geometric sequence

\[ a_1 = \frac{1}{(cB)} , \quad q = 1 + \frac{1}{B} \]

\[ x = \frac{1}{cB} \left( \left(1 + \frac{1}{B}\right)^B - 1 \right) = \frac{\left(1 + \frac{1}{B}\right)^B - 1}{c} \]

Algorithm:

When new constraint arrives, if \( x < 1 \):

\[ z_i \leftarrow 1 - x \]

\[ x \leftarrow x(1 + 1/B) + 1/c*B \]

\[ y_i \leftarrow 1 \]
Randomized Algorithm

- Choose $d$ uniformly in $[0,1]$
- Buy on the day corresponding to the “bin” $d$ falls in
- Rent up to that day

Analysis:

- Probability of buying on the $i$-th day is $x_i$
- Probability of renting on the $i$-th day is at most $z_i$
Cost of the Randomized algorithm

- \( E \text{ cost} = E(\text{ } \$ \text{ spent buy } ) + E(\$\text{spent renting}). \)
- \( E(\$ \text{ spent buy}) = B \cdot P(\text{buy}) \)
- \( P(\text{buy}) = p(x), \text{ where } x = \sum x_i \)
- \( E(\$\text{spent renting}) = \sum(\text{over } i=1 \text{ to } k) \text{ p(rent on day } i) \)
  \[= \sum(\text{over } i=1 \text{ to } k) \left(1 - (x_1, x_2, \ldots , x_{i-1})\right) = \]
  \[= Bx + \sum(\text{over } i=1 \text{ to } k) \ z_i \]
The Online Set-Cover Problem

- Elements: e₁, e₂, …, en
- Set system: s₁, s₂, … sm
- Costs: c(s₁), c(s₂), … c(sm)

Online Setting:

- Elements arrive one by one.
- Upon arrival elements need to be covered.
- Sets that are chosen cannot be “unchosen”.

Goal: Minimize the cost of the chosen sets.
Set Cover – Linear Program

P: Primal Covering
\[ \min \sum_{s \in S} c(s)x(s) \]
\[ \forall e \in E \sum_{s \in s} x(s) \geq 1 \]
\[ X(s) \geq 0 \]

D: Dual Packing
\[ \max \sum_{e \in E} y(e) \]
\[ \forall s \in S \sum_{e \in s} y(e) \leq c(s) \]

- Online setting:
  Primal: constraints arrive one by one.
  Requirement: each constraint is satisfied.
  Monotonicity: variables can only be increased.
Conclusion

- The dual is feasible with cost $1/O(\log m)$ of the primal.
- The algorithm produces a fractional set cover that is $O(\log m)$-competitive.
- Remark: No online algorithm can perform better in general.

What about an integral solution?
- Round fractional solution. (With $O(\log n)$ amplification.)
- Can be done deterministically online [AAABN03].
- Competitive ratio is $O(\log m \log n)$. 
References


3– Lecture Series on Advanced Algorithms by Prof.J.Nelson, Department of Computer Science, Harvard University. https://www.youtube.com/watch?v=0JUN9aDxVmI&list=PL2SOU6wwxB0uP4rJgf5ayhHWgw7akUWSf

4– Lecture Series on Operations Research by Prof.Salimian. https://www.youtube.com/watch?v=muLLnPrRdf8