Combinatorial Ski Rental and Online Bipartite Matching

Hanrui Zhang Vincent Conitzer
Duke University

Our results:

- <u>optimal</u> (1-1/e)-competitive algorithms (against offline optimum) for combinatorial ski rental & online bipartite matching when <u>costs / capacity constraints can be submodular</u>

- no constant-factor algorithm exists when any part of our assumptions is relaxed

Combinatorial ski rental

- a business analyst's job involves 2 software products: Excel and Powerpoint
- tasks arrive over time, each of which may require <u>either / both</u> of the 2 products
- in order to finish all tasks, analyst may "<u>rent</u>" or "<u>buy</u>" any combination of the products; purchased products <u>cannot be returned</u>
- <u>discounts</u> are available when renting or buying both products

cost of purchasing: Excel $30, Powerpoint $30, both $50

cost of renting: Excel $30, Powerpoint $30, both $50

"upgrading" is allowed: pay $20 for Powerpoint when Excel is already purchased

time

1st task requires: Excel
rent Excel at cost $10, cost so far: $10
product(s) purchased: none

2nd task requires: Excel, Powerpoint
rent both at cost $15, cost so far: $25
product(s) purchased: none

3rd task requires: Excel
buy Excel at cost $30, cost so far: $55
product(s) purchased: Excel

4th task requires: Excel, Powerpoint
rent Powerpoint at cost $10, cost so far: $65
product(s) purchased: Excel

5th task requires: Excel, Powerpoint
buy Powerpoint at <u>(marginal) cost $20</u>, cost so far: $85
product(s) purchased: Excel, Powerpoint

both products purchased
future cost = 0

Primal LP for ski rental

- n products available for purchasing / renting
- $f(S)$: cost of purchasing S
- $g_t(S)$: cost of renting S at time t
- $x(S) \geq 0$: probability of purchasing S
- $y_t(S) \geq 0$: probability of renting S at time t
- minimize:

$$\Sigma_S\, x(S)\, f(S) + \Sigma_{1 \leq t \leq T}\, \Sigma_{S'}\, y_t(S')\, g_t(S')$$

- s.t. for all item i, time t:

$$\Sigma_{S:\, i \in S}\, x(S) + y_t(S) \geq 1$$

Dual LP for online matching

- n offline vertices, one online vertex at each time t
- $\lambda_t(i)$: fraction of online vertex arriving at time t matched to offline vertex i
- maximize:

$$\Sigma_{1 \leq t \leq T}\, \Sigma_{1 \leq i \leq n}\, \lambda_t(i)$$

total (fractional) number of online vertices matched

- s.t. (offline capacity constraints) for every set of items S:

$$\Sigma_{1 \leq t \leq T}\, \Sigma_{i \in S}\, \lambda_t(i) \leq f(S)$$

total load of all offline vertices in S cannot exceed f(S)

- s.t. (online supply constraints) for every set of items S, time t:

$$\Sigma_{i \in S}\, \lambda_t(i) \leq g_t(S)$$

fraction of online vertex at time t matched to offline vertices in S cannot exceed $g_t(S)$

take-home message: online primal-dual analysis can go fully combinatorial

Sketch of algorithm

- minimize: $\Sigma_S\ x(S)\ f(S) + \Sigma_{1 \leq t \leq T}\ \Sigma_{S'}\ y_t(S')\ g_t(S')$
- s.t. for all i, t: $\Sigma_{S:\, i \in S}\ x(S) + y_t(S) \geq 1$

- in constraints: only marginal probabilities matter
- re-parametrize by marginal probabilities
- consider convex envelope = Lovasz extension

primal:

- minimize: $F(x_t(1), \ldots, x_t(n)) + \Sigma_{1 \leq t \leq T}\ G_t(y_t(1), \ldots\ y_t(n))$

dual:

- maximize: $\Sigma_{1 \leq t \leq T}\ \Sigma_{1 \leq i \leq n}\ \lambda_t(i)$
- s.t. for all S: $\Sigma_{1 \leq t \leq T}\ \Sigma_{i \in S}\ \lambda_t(i) \leq f(S)$
- s.t. for all S, t: $\Sigma_{i \in S}\ \lambda_t(i) \leq g_t(S)$

new problem: global objective & constraints

- lemma: under *certain conditions*, local constraints satisfied imply global constraints satisfied & small ratio between primal / dual objectives
- problem boils down to local task: setting $\lambda_t(i)$ (which determine $x_t(i)$) to satisfy these *certain conditions* at each time t
- relatively easy when $x_t(i)$ are all different, highly nontrivial (and combinatorial) otherwise; latter (hard) case is not "zero-measure"

Overview of lower bounds

no constant-factor algorithm if:

- cost of purchasing is allowed to be XOS,
- cost of renting is allowed to be supermodular, or
- upgrading is not allowed (i.e., price of purchasing S given S' is f(S) rather than f(S | S'))

in words: all our assumptions are necessary

symmetrization tecinque: prepare future clauses, choose realization by demanding right copy of item