Advanced Algorithms / Fall 2026

Linear Programming and Optimization

Primal-Dual Algorithms

Vertex cover and metric facility location

Lecture 6: 100 minutes

These notes cover only the linear-programming strand taught by Shi Li. This module comprises nine 100-minute lectures, numbered consistently with the slides. Parts I and II are reading divisions within this lecture; optional appendices are for further study.

Instructor: Shi Li

Nanjing University

Prepared from the instructor's course slides, with AI-assisted expansion of explanations, proofs, and independent-study exercises. The English and Chinese editions have matching mathematical content.

Learning goals and reading route

LP rounding first computes an optimal fractional solution. A primal-dual algorithm instead grows a feasible lower-bound certificate and uses its tight constraints to decide which integral objects to choose. The LP explains and certifies the algorithm even when no LP solver is called.

1. Distinguish a maximal feasible dual vector from an optimal one.
2. Derive and analyze the weighted vertex-cover primal-dual algorithm.
3. Explain equivalence of the basic and star formulations for facility location.
4. Follow dual growth, facility pruning, and the three-hop connection bound.

Part I develops the certificate viewpoint and the facility-location LPs. Part II contains the full facility-location algorithm and analysis. Metric symmetry and the triangle inequality are essential in that analysis.

Contents

1 Part I: Vertex cover and facility-location LPs **1**

1.1 Paying for vertices with edge budgets 1
1.2 Metric uncapacitated facility location 1
1.3 Stars turn the LP into a covering problem 2

2 Part II: Facility-location algorithm and analysis **2**

2.1 Uniform dual growth 2
2.2 Pruning temporarily open facilities 3
2.3 Paying for owned clients and openings 3
2.4 The three-hop bound for remaining clients 3
2.5 The approximation guarantee and implementation 4

3 Exercises for independent study **4**

1 Part I: Vertex cover and facility-location LPs

1.1 Paying for vertices with edge budgets

For weighted vertex cover, omit the redundant upper bounds from the previous unit. The primal and dual are

$$\begin{array}{lll} \min & \sum_v w_v x_v & \\ \text{s.t.} & x_u + x_v \geq 1 & (uv \in E) \\ & x \geq 0 & \end{array} \qquad \begin{array}{lll} \max & \sum_e y_e & \\ \text{s.t.} & \sum_{e \in \delta(v)} y_e \leq w_v & (v \in V) \\ & y \geq 0. & \end{array}$$

Think of each edge as accumulating a budget. A vertex is affordable exactly when the total budget on its incident edges reaches its weight. Dual feasibility prevents us from collecting more money at a vertex than that vertex costs.

Primal-dual weighted vertex cover

1. Set $y = 0$ and $S = \varnothing$.
2. While an edge uv is uncovered, increase y_{uv} by the smaller residual budget of its endpoints.
3. Add a newly tight endpoint to S and mark its incident edges covered. If an endpoint was already tight, a zero increase is allowed and that endpoint is selected.
4. Return S and the certificate y.

Each iteration selects an unselected vertex, so there are at most $|V|$ iterations. The dual remains feasible because only the two endpoint constraints change, and the increase stops at the first tight one. A selected vertex stays tight: no later chosen uncovered edge is incident to it. At termination every edge is covered.

Theorem 1.1. *The algorithm is a two-approximation.*

Proof. Tightness of selected vertices and a change in summation order give

$$\sum_{v \in S} w_v = \sum_{v \in S} \sum_{e \in \delta(v)} y_e = \sum_{uv \in E} y_{uv} |\{u, v\} \cap S| \leq 2 \sum_e y_e \leq 2\,\mathrm{OPT}\,.$$

The last inequality is weak duality, so optimality of the constructed dual vector is unnecessary. □

More generally, take any coordinatewise maximal dual-feasible y and select all tight vertices. If an edge had two slack endpoints, its budget could be increased, contradicting maximality. Thus tight vertices cover every edge and the same charging argument applies. “Maximal” means no coordinate can be increased while fixing the others; it does not mean the objective is globally largest.

1.2 Metric uncapacitated facility location

Let F be candidate facilities and C clients, with a symmetric metric d on $F \cup C$ and opening costs $f_i \geq 0$. Open $S \subseteq F$ and connect every client to an open facility, minimizing

$$\sum_{i \in S} f_i + \sum_{j \in C} d(j, S), \qquad d(j, S) = \min_{i \in S} d(i, j).$$

There are no capacity limits. Assume at least one facility if clients exist; with no clients the empty solution is optimal. The tradeoff is between opening many nearby facilities and paying longer connection distances to a few facilities.

Let y_i represent opening and x_{ij} assignment. The basic relaxation is

$$\begin{aligned} \min \quad & \sum_i f_i y_i + \sum_{i,j} d(i,j) x_{ij} \\ \text{s.t.} \quad & \sum_i x_{ij} \geq 1 \quad (j \in C), \\ & 0 \leq x_{ij} \leq y_i \quad (i \in F, j \in C), \qquad y_i \geq 0. \end{aligned} \tag{1.1}$$

For a fixed feasible opening vector y, each client can buy its unit of assignment from facilities in increasing distance order, with at most y_i from facility i. Different clients do not compete for capacity. The optimum assignment need not be unique when distances tie.

1.3 Stars turn the LP into a covering problem

A star (i, J) consists of facility i and client subset $J \subseteq C$, with cost $f_i + \sum_{j \in J} d(i,j)$. Give it a nonnegative variable z_{iJ}. The star relaxation and dual are

$$\min \quad \sum_{i,J} \left(f_i + \sum_{j \in J} d(i,j) \right) z_{iJ} \qquad \text{s.t.} \quad \sum_{i,J: j \in J} z_{iJ} \geq 1, \quad z \geq 0; \tag{1.2}$$

$$\max \quad \sum_j \alpha_j \qquad \text{s.t.} \quad \sum_{j \in J} \alpha_j \leq f_i + \sum_{j \in J} d(i,j), \quad \alpha \geq 0. \tag{1.3}$$

Every integral facility solution is a collection of stars covering the clients. If a collection uses multiple stars at the same facility, opening that facility once and serving their union costs no more. Thus the integral interpretation is correct. The fractional equivalence also holds.

Proposition 1.2. *The basic and star LPs have the same optimum.*

Proof. From stars to the basic formulation, put $y_i = \sum_J z_{iJ}$ and $x_{ij} = \sum_{J: j \in J} z_{iJ}$. Feasibility and equality of costs follow by summing. Conversely, for a basic solution and each facility i, let a threshold t range over $[0, y_i]$ and define $J_i(t) = \{j : x_{ij} \geq t\}$. Give each resulting star mass equal to the length of thresholds producing it. Total star mass at i is y_i, and mass including client j is x_{ij}. Hence assignment coverage and both objective terms are preserved. Only the breakpoints $0, y_i, x_{ij}$ matter, so at most $|C| + 1$ stars per facility are needed for this conversion. □

The exponentially many dual constraints for one facility are equivalent to one computable inequality:

$$\sum_{j \in C} (\alpha_j - d(i,j))_+ \leq f_i, \qquad (a)_+ = \max\{a, 0\}. \tag{1.4}$$

Indeed, the most violated star chooses exactly the clients with positive surplus $\alpha_j - d(i,j)$. Interpret α_j as client j's budget: first pay the connection distance, then contribute the surplus to opening facility i. Weak duality gives $D := \sum_j \alpha_j \leq \mathrm{OPT}$ for every feasible budget vector.

2 Part II: Facility-location algorithm and analysis

2.1 Uniform dual growth

Initially all budgets are zero and all clients are active. Active budgets grow at unit speed with a global clock t; a frozen client's budget stays fixed. A facility becomes temporarily open when (1.4) first becomes tight, at time t_i. Freeze every active client that can reach a temporarily open facility within its current budget. Future active clients freeze as soon as their budget reaches the distance

to an already temporarily open facility. Process simultaneous events consistently before advancing time. The final budget α_j is exactly the freezing time of client j. Record a witness facility i for each client, satisfying

$$t_i \le \alpha_j, \qquad d(i,j) \le \alpha_j. \tag{2.1}$$

Dual feasibility is preserved. Before a facility becomes tight, stop the growth at its first tightness event. Afterwards all clients already contributing to it are frozen. Any new active client freezes on reaching distance equality and therefore never begins a positive contribution to that facility. Its total contribution remains exactly f_i.

For zero opening costs, facilities may be tight at time zero even with no contributors; include them among the temporarily open facilities. This makes distance-equality freezing well defined and avoids waiting for a nonexistent positive opening payment. All clients eventually freeze because a finite-distance facility would otherwise accumulate enough contribution to become tight.

2.2 Pruning temporarily open facilities

Let $N_i = \{j : \alpha_j > d(i,j)\}$ be the positive contributors to facility i. Equality-budget clients are not in N_i. We cannot simply open every tight facility: the same client might be paying for many of them.

Pruning and final connections

1. Process temporarily open facilities in nondecreasing order of t_i, breaking ties consistently. Start with $S = \varnothing$ and every client unowned.
2. If all clients in N_i are unowned, open i, add it to S, and assign ownership of all N_i to i. Otherwise discard i.
3. Connect owned clients to their owners. Connect every remaining client to its nearest facility in S.

The selected contributor sets are disjoint. Every rejected facility shares a contributor with an earlier selected facility. A facility with no contributors passes the test automatically and has opening cost zero, so this case causes no overpayment.

2.3 Paying for owned clients and openings

Let $C_1 = \bigcup_{i \in S} N_i$ and $C_2 = C \setminus C_1$. Write $f = \sum_{i \in S} f_i$ and c_j for the final connection cost of client j. Tightness gives, for each selected facility,

$$f_i = \sum_{j \in N_i} (\alpha_j - d(i,j)).$$

Its owned clients connect at cost $d(i,j)$, so

$$f + \sum_{j \in C_1} c_j \le \sum_{j \in C_1} \alpha_j. \tag{2.2}$$

Equality holds for the prescribed owner connections; reconnecting to nearer selected facilities could only improve it. The disjointness of contributor sets is what allows one sum of budgets to pay all openings.

2.4 The three-hop bound for remaining clients

Lemma 2.1. *Every $j \in C_2$ has $c_j \le 3\alpha_j$.*

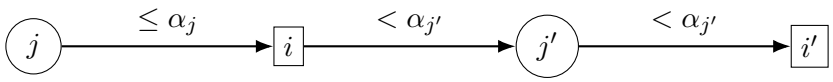
Proof. Take its witness facility i from (2.1). If $i \in S$, connecting to the nearest selected facility costs at most $d(j,i) \le \alpha_j$. Otherwise some $j' \in N_i$ was already owned by a selected i' with $t_{i'} \le t_i$. A positive contributor to i' froze no later than i' opened, so

$$\alpha_{j'} \le t_{i'} \le t_i \le \alpha_j, \qquad d(i,j') < \alpha_{j'}, \quad d(i',j') < \alpha_{j'}.$$

Apply metric symmetry and the triangle inequality along $j \to i \to j' \to i'$:

$$c_j \le d(j,i') \le d(j,i) + d(i,j') + d(j',i') \le \alpha_j + 2\alpha_{j'} \le 3\alpha_j.$$

□



Processing facilities by opening time is not a cosmetic choice: it gives $\alpha_{j'} \le \alpha_j$, which is needed to charge all three segments to the budget of j. The distinction between positive contributors and equality-budget clients is equally important for the payment and pruning arguments.

2.5 The approximation guarantee and implementation

Theorem 2.2. *The algorithm returns a feasible facility-location solution with* $f + \sum_j c_j \le 3\,\mathrm{OPT}$. *In fact, the stronger inequality* $3f + \sum_j c_j \le 3D \le 3\,\mathrm{OPT}$ *holds.*

Proof. Every client has a witness facility. A selected witness supplies a direct connection; a rejected witness supplies an earlier selected facility through a shared contributor. Thus all clients can connect. Multiply (2.2) by three and use nonnegativity of connection costs:

$$\begin{aligned} 3f + \sum_j c_j &\le 3\left(f + \sum_{j \in C_1} c_j\right) + \sum_{j \in C_2} 3\alpha_j \\ &\le 3\sum_{j \in C} \alpha_j = 3D \le 3\,\mathrm{OPT}\,. \end{aligned}$$

Since $f \ge 0$, the ordinary factor-three guarantee follows. □

There is no need to enumerate stars. For a facility that is not yet tight, its total contribution is a continuous piecewise-linear function of the global time, with breakpoints at distances to active clients. Its next tightness time can be computed by sorting or scanning those breakpoints. Also track active clients reaching already-tight facilities. Each major event opens a new facility or freezes a new client, so an event-driven implementation is polynomial. A naive simulation with tiny fixed time steps would not justify a polynomial bound.

Remark 2.3 (Context from the slides). The slides mention Shi Li's 1.488-approximation and a hardness threshold near 1.463, the positive solution of $\rho = 1 + 2e^{-\rho}$. These are advanced results, separate from the factor-three proof here. The historical hardness assumption stated in Li's 2011 paper is NP $\not\subseteq$ DTIME($n^{O(\log\log n)}$). See the original paper for the precise context.

3 Exercises for independent study

1. Run the vertex-cover algorithm on a weighted path under two different edge orders. Compare the returned covers and dual values.

2. Give a coordinatewise maximal dual vector that is not optimal for some vertex-cover instance.
3. For fixed i, explicitly convert a basic LP vector with $y_i = 0.8$ and assignments $0.2, 0.5, 0.8$ into at most four stars.
4. Prove that the family of all star dual constraints for i is equivalent to (1.4).
5. Explain why selecting all tight facilities can charge one client's surplus several times, and why pruning avoids this.
6. Identify exactly where metric symmetry, triangle inequality, opening-time order, and nonnegative opening costs enter the proof.
7. Describe how zero opening costs and simultaneous tightness events are processed without violating dual feasibility.

Source and reading notes

Primary source: the instructor's slide folder

`6.PrimalDual`

`Topics.tex`.

Optional sections follow the optional slides where present. Explanatory examples and exercises are added to support independent reading; they do not change the lecture allocation.

Course website. The early-course notes linked there were used as examples of organization and level of detail, not as the source of this module's subject matter.