Advanced Algorithms / Fall 2026

Linear Programming and Optimization

LP Rounding

Vertex cover and unrelated-machine scheduling

Lecture 4: 100 minutes

These notes cover only the linear-programming strand taught by Shi Li. This module comprises eight 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

An LP need not be integral to be useful. A fractional optimum supplies a benchmark; a rounding algorithm turns it into a feasible discrete solution while controlling the increase in cost. This unit contrasts a coordinatewise threshold with a coordinated matching-based rounding.

1. State the approximation proof chain and interpret an integrality gap correctly.
2. Prove the factor-two weighted vertex-cover rounding theorem.
3. Construct scheduling slots, find a job-covering matching, and prove the $2T$ load bound.

Parts I and II are the main reading. The appendix develops independent randomized rounding for congestion, using the concentration tools from the earlier course strand. All minimization costs and processing times are nonnegative.

Contents

1 Part I: LP rounding and vertex cover **1**
1.1 The benchmark and the proof chain . . . 1
1.2 Weighted vertex cover . . . 1
1.3 A sharp LP benchmark . . . 2

2 Part II: Unrelated-machine scheduling **2**
2.1 Guessing a makespan . . . 2
2.2 Splitting machines into slots . . . 2
2.3 Charging each later job to the previous slot . . . 3
2.4 Obtaining a polynomial-time two-approximation . . . 3

3 Exercises for independent study **3**

A Optional: congestion minimization **4**
A.1 Fractional routing and sparse path distributions . . . 4
A.2 Independent rounding and a union bound . . . 4
A.3 Two useful refinements . . . 4

1 Part I: LP rounding and vertex cover

1.1 The benchmark and the proof chain

For a minimization problem, let OPT be the integral optimum and LP the optimum of a relaxation containing all feasible integral solutions. If x^* is an optimal fractional solution and rounding returns a feasible discrete S with cost at most $\alpha c^\top x^*$, then

$$\text{cost}(S) \le \alpha c^\top x^* = \alpha \text{LP} \le \alpha\, \text{OPT}\,. \tag{1.1}$$

There are two separate obligations: the rounded object must be feasible, and its cost must be bounded. A coordinatewise cost inequality is useless if an edge remains uncovered or a job remains unassigned. Conversely, a feasible rounding without a cost analysis is not an approximation theorem.

Definition 1.1 (Integrality gap)**.** For an instance with positive LP optimum, its gap is OPT /LP. The worst-case gap of a relaxation is the supremum of this ratio over instances. Zero-optimum cases require separate interpretation rather than division by zero.

If an instance has gap g, no feasible integral solution can cost less than gLP. Hence an analysis of the form $\text{cost}(S) \le \alpha\text{LP}$ cannot hold for $\alpha < g$. This limits that LP-relative proof method; it does not by itself prove hardness of achieving a better approximation using a stronger formulation or another benchmark.

1.2 Weighted vertex cover

For an undirected graph $G = (V, E)$ with vertex weights $w_v \ge 0$, a vertex cover S contains at least one endpoint of every edge. Its natural relaxation is

$$\min \sum_{v \in V} w_v x_v \quad \text{s.t.} \quad x_u + x_v \ge 1 \ (uv \in E), \qquad 0 \le x_v \le 1.$$

The upper bounds can be omitted without changing the optimum: truncating a nonnegative feasible coordinate to 1 preserves every covering constraint and cannot increase a nonnegative objective. Keeping them makes the relaxation explicitly bounded.

Threshold rounding for vertex cover

Solve the LP to obtain x^*. Return $S = \{v : x^*_v \ge 1/2\}$.

Theorem 1.2. *The returned set is a vertex cover and has weight at most* $2\text{LP} \le 2\,\text{OPT}$.

Proof. If neither endpoint of an edge were chosen, both coordinates would be below $1/2$, contradicting $x^*_u + x^*_v \ge 1$. For the cost, every selected vertex satisfies $1 \le 2x^*_v$, so

$$\sum_{v \in S} w_v \le 2 \sum_{v \in S} w_v x^*_v \le 2 \sum_{v \in V} w_v x^*_v = 2\text{LP}.$$

Nonnegative weights justify extending the sum to all vertices. □

The threshold must include equality. On a single edge with $x_u = x_v = 1/2$, selecting only coordinates strictly greater than $1/2$ would return the empty set. Independent Bernoulli rounding with these probabilities also fails feasibility with probability $1/4$; preserving the expected objective alone is not enough.

1.3 A sharp LP benchmark

For the unit-weight complete graph K_n, $n \geq 2$, an integral cover has size at least $n-1$. Setting every coordinate to $1/2$ is feasible with value $n/2$. Summing all edge constraints gives $(n-1)\sum_v x_v \geq n(n-1)/2$, so LP $= n/2$. The gap is

$$\frac{n-1}{n/2} = 2 - \frac{2}{n} \longrightarrow 2.$$

Together with the rounding theorem, this proves that the worst-case gap of the basic relaxation is 2. The gap example concerns the quality of the LP lower bound; it should not be confused with the ratio of this particular rounded output to OPT on the same instance.

2 Part II: Unrelated-machine scheduling

2.1 Guessing a makespan

There are jobs J and machines M. Job j takes time p_{ij} on machine i; a forbidden assignment has $p_{ij} = \infty$. An assignment $\sigma : J \to M$ has makespan $\max_i \sum_{j:\sigma(j)=i} p_{ij}$. Unlike identical machines, the processing time can vary arbitrarily with both indices. For a trial value $T \geq 0$, exclude every pair with $p_{ij} > T$, since no schedule of makespan T can use it. On $E_T = \{(i,j) : p_{ij} \leq T\}$ solve the feasibility LP

$$\sum_{i:(i,j)\in E_T} x_{ij} = 1 \ (j \in J), \qquad \sum_{j:(i,j)\in E_T} p_{ij}x_{ij} \leq T \ (i \in M), \qquad x \geq 0. \tag{2.1}$$

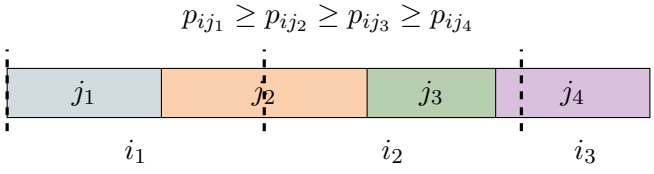
Infeasibility certifies that T is too small. Feasibility need not certify a schedule of makespan T, but we will recover one of makespan at most $2T$. The exclusion of large pairs is essential: the final bound charges one entire job to each machine.

2.2 Splitting machines into slots

Slot construction and matching

1. For each machine i, sort its positive fractions in nonincreasing order of p_{ij}. Lay out consecutive pieces of widths x_{ij} on a line.
2. Partition this line into unit-length slots $i_1, \ldots, i_{a_i}$, where $a_i = \lceil \sum_j x_{ij} \rceil$. The final slot may be partially filled.
3. Let $z_{i_t j}$ be the overlap of job j's piece with slot i_t. Create a bipartite edge whenever this overlap is positive.
4. Find an integral matching covering every job. Assign a matched job to the original machine owning its matched slot.

Every job has total incident fractional mass 1, and every slot has mass at most 1. Therefore z is a fractional bipartite matching of total size $|J|$. By bipartite matching integrality, a maximum-cardinality matching has at least that size, so it covers all jobs. One can also apply Hall's theorem: for any job subset U, its mass $|U|$ must fit into neighboring slots of capacity one, giving $|N(U)| \geq |U|$.



In this illustrative local construction, widths are $0.6, 0.8, 0.5, 0.6$. Job j_2 crosses a slot boundary, and j_4 crosses another. Splitting fractions does not split the final job: the matching selects exactly one slot for each job.

2.3 Charging each later job to the previous slot

Theorem 2.1 (Shmoys–Tardos slot rounding)**.** *Any job-covering matching in the constructed support graph yields load at most $2T$ on every original machine.*

Proof. Fix machine i and let k_t be the job matched to slot i_t; if it is unmatched, use a dummy job of processing time zero. The first job has $p_{ik_1} \le T$. For $t \ge 2$, slot i_{t-1} is full. Every processing time in that earlier slot is at least p_{ik_t} because the pieces were sorted in nonincreasing order and k_t has positive overlap with the later slot. Thus

$$p_{ik_t} \le \sum_j z_{i_{t-1}j} p_{ij}.$$

Summing over slots gives

$$\begin{aligned} \mathrm{load}(i) &= \sum_{t=1}^{a_i} p_{ik_t} \\ &\le T + \sum_{t=1}^{a_i - 1} \sum_j z_{i_t j} p_{ij} \\ &\le T + \sum_j x_{ij} p_{ij} \le 2T. \end{aligned}$$

If there are no slots, the load is zero; if there is one, only the first-job bound is needed. □

The ordering is what makes this argument work. Merely splitting each machine into enough arbitrary slots would preserve the matching constraints but would not justify comparing a rounded job with the previous slot's fractional load.

2.4 Obtaining a polynomial-time two-approximation

For integer processing times, feasibility of (2.1) is monotone in the integer target T. Binary search between zero and the makespan of any feasible assignment returns the smallest feasible integer target T_0. Since an optimal integral schedule makes the LP feasible at OPT, we have $T_0 \le \mathrm{OPT}$. Rounding then gives makespan at most $2T_0 \le 2\,\mathrm{OPT}$. If a job has no finite machine, the instance is infeasible. A target of zero is handled by assigning only zero-time pairs. For rational input, scale by a common denominator before this argument. Its numeric value can be large, but the product of input denominators has bit length at most their total bit length, and binary search uses logarithmically many targets. Alternatively, numerical target search must explicitly account for its tolerance. The rounding graph remains small: $\sum_i \sum_j x_{ij} = |J|$, hence $\sum_i a_i \le |J| + |M|$. Each positive $x_{ij} \le 1$ crosses at most two slots. Sorting and finding a bipartite maximum matching therefore take polynomial time. No exponentially large convex decomposition needs to be computed.

3 Exercises for independent study

1. Consider a binary covering problem and its relaxation $\min\{c^{\mathsf{T}} x : \sum_{j \in S_i} x_j \ge 1 \ (i \in [m]),\ 0 \le x \le 1\}$, where $c \ge 0$, $1 \le |S_i| \le k$, and integer solutions require $x \in \{0,1\}^n$. Prove that rounding at threshold $1/k$ gives a feasible solution of cost at most k times the LP optimum, and hence a k-approximation.

2. Verify the K_n integrality-gap calculation and explain why a threshold above $1/2$ can destroy feasibility.
3. For one machine with fractions $0.6, 0.8, 0.5, 0.6$ and processing times $9, 7, 4, 2$, calculate every slot overlap and fractional slot load. Verify the previous-slot bound for each supported rounded job.
4. Show directly from Hall's theorem that the slot graph has a matching covering every job.
5. Find an example showing why keeping pairs with $p_{ij} > T$ can invalidate the $2T$ proof.
6. Optional: derive the congestion tail bound below from the exponential-moment inequality, without replacing the true mean by C inside an unjustified equality.

A Optional: congestion minimization

A.1 Fractional routing and sparse path distributions

Given a simple directed graph and terminal pairs (s_i, t_i), choose one path for each pair to minimize the maximum number of paths using any edge. Remove trivial pairs with $s_i = t_i$ first. If no nontrivial pair remains, the optimum is zero; if a pair has no path, the instance is infeasible. Otherwise the integral optimum is at least one. For each commodity i, assign nonnegative masses z_{iP} to its simple paths, summing to one. Minimize C_{LP} subject to $\sum_i \sum_{P \ni e} z_{iP} \le C_{\mathrm{LP}}$ for every edge. Although paths are exponentially numerous, a compact formulation uses commodity arc flows $f_{ie} \ge 0$, unit net supply at s_i, unit net demand at t_i, conservation elsewhere, and $\sum_i f_{ie} \le C_{\mathrm{LP}}$. Decompose each commodity flow into paths and cycles and discard cycles. The resulting sparse path distribution has total mass one and no larger edge usage. Thus the two LPs have the same optimum, and sampling does not require listing every path. Define $C = \max\{1, C_{\mathrm{LP}}\} \le \mathrm{OPT}$; the maximum with one is needed because fractional congestion can be below one.

A.2 Independent rounding and a union bound

Independently for each commodity, sample path P_i with probabilities z_{iP}. Fix an edge e and let X_i indicate $e \in P_i$. The variables are independent across commodities, and $\mu_e = \sum_i \mathbb{E}X_i \le C$. For every $\lambda > 0$,

$$\mathbb{E}\exp\left(\lambda \sum_i X_i\right) = \prod_i \left(1 + \mathbb{E}X_i(e^\lambda - 1)\right) \le \exp\left(C(e^\lambda - 1)\right).$$

Markov's inequality with $\lambda = \ln \alpha$ gives, for $\alpha > 1$,

$$\mathbb{P}[\mathrm{cong}(e) \ge \alpha C] \le \exp\{-C(\alpha \ln \alpha - \alpha + 1)\}. \tag{A.1}$$

Choose α so that $\alpha \ln \alpha - \alpha + 1 \ge \ln(2m)$, where $m = |E|$. Since $C \ge 1$, each failure probability is at most $1/(2m)$, and a union bound gives success probability at least $1/2$. For sufficiently large m, $\alpha = O(\log m / \log\log m)$ suffices. In a simple directed graph $m \le n^2$, this is the slide bound $O(\log n / \log\log n)$, with a constant chosen for small instances.

Because the rounded congestion can be computed, repeat independent trials until the target bound holds. The expected number of trials is at most two. If $C = \Omega(\log m)$, a constant $\alpha > 1$ already makes (A.1) small enough. Thus a large fractional load improves the relative rounding loss. No independence between different edges is required for the union bound.

A.3 Two useful refinements

If all sources coincide, the problem is exactly solvable by integral max flow. For a trial integer congestion B, give every original edge capacity B and add a supersink with one unit-capacity

arc for each demand from its sink t_i. A flow of value k saturates all demand arcs. Its integral decomposition gives one path for each demand, including repeated sinks represented by parallel demand arcs.

For derandomization, choose integer $\alpha > 1$ meeting the preceding bound and integer threshold $L = \lceil \alpha C \rceil$. Consider the sum of exponential penalties $\sum_e \alpha^{\mathrm{cong}(e)-L}$. Its initial expectation is at most $1/2$. Fix commodities one at a time, choosing a supported path whose conditional expectation does not increase; such a path exists by averaging. The remaining expectation factors over independent commodities and is computable from their edge marginals. At the end the sum is below one, so no edge can have congestion at least L. For rational path weights this calculation uses rational products and integer powers, making the conditional-expectation idea concrete.

Source and reading notes

Primary source: the instructor's slide folder

`4.LPRounding`

`Topics.tex` and `Optional.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.