Deterministic almost-linear time for general matching and network flow

Yilin Liu

19 November 2026

Abstract

We give deterministic almost-linear-time algorithms for maximum-cardinality matching in general graphs and integral minimum-cost flow. With $H = n + m + 2$, both run in $H \exp O((\log H)^{4/5}(\log\log H)^{1/5})$ time for polynomially bounded integer flow data. Their circulation oracle charges every traversal of a returned walk. A clipped logarithmic potential handles flow thresholds under improving updates and maintains strong components under arc insertions. For subgraphs with degree bounds, two matching computations encode a factor of every feasible size. The underlying exchange argument also gives integral cardinality slices for capacitated factors and is compatible with sharp local rounding.

hexagon:2611.00652v1

1 Introduction

Matching in a general graph must respect odd sets, while flow in a network must respect balances and capacities. Both algorithms find improvements by assigning prices and lengths to edges and searching for a circulation of large gain per unit of length. Its length charges every traversal in the returned walks.

Theorem 1.1 (Matching and flow)**.** *Let $H = n + m + 2$ for a graph with n vertices and m edges. A maximum-cardinality matching in an undirected graph and a minimum-cost integral flow with prescribed balances and arc bounds in a directed graph can each be found in deterministic time*

$$H\mathcal{E}_4 = H \exp\Big(O\Big((\log H)^{4/5}(\log\log(H+4))^{1/5}\Big)\Big) = H^{1+o(1)}. \tag{1}$$

Flow data are integers of magnitude $H^{O(1)}$. The outputs are original matching edges and integral values on the original arcs, respectively. The flow algorithm reports infeasibility when appropriate.

An odd path replaces each vertex of the matching instance. On the resulting subcubic graph a phase either enlarges the matching or bounds its distance from optimum. For flow we extend the logarithmic potential by its tangent near zero. The extension measures capacity violations, and an Euler-tour rounding recovers an integral optimum. Sections 2–4 give these arguments.

The flow algorithm also finds the first prefix meeting a cost threshold under insertions, relaxations of arc bounds and cost decreases. Its one-sided potential maintains the strong components of an insertion stream. A single sparse subgraph certifies their entire history (Section 5).

Two matching computations suffice to represent every feasible cardinality of a subgraph with degree intervals. With integral edge capacities, the same alternating-trail argument shows that every integer cardinality slice of the factor polytope is integral (Section 6). Section 7 gives a common reduction for matching and flow using a spanning forest. Table 1 compares the running times with those for distances, spanners and cut trees.

Table 1: Deterministic running times. Dynamic bounds include initialization and all updates. Section 8 gives the accuracy and input assumptions for the metric bounds.

Problem	Total work	Output
Exact algorithms for matching, flow and strong components		
General matching	$H\mathcal{E}_4$	Maximum set of disjoint original edges
Unit edges with degree bounds	$H\mathcal{E}_4$	A factor of every feasible size, in one representation
Static min-cost flow	$H\mathcal{E}_4$	Integral optimum and a description of the optimal face
Improving flow updates	$H\mathcal{E}_4$	First prefix meeting the cost threshold and an integral flow
Incremental strong components	$H\mathcal{E}_4$	Exact labels and one certificate for all prefixes
Distance, spanner and cut algorithms from [12]		
Incremental directed single-source distances	$H\mathcal{E}_4$	$(1+\varepsilon)$-approximate distances and simple paths
Fully dynamic distances	$H\mathcal{E}_4$	Point estimates of stretch $\mathcal{E}_4$ in polylogarithmic query time
Fully dynamic spanners	$H\mathcal{E}_j$, $j = 4, 5$	Stretch $\mathcal{E}_j$, size $n\mathcal{E}_j$, and $H\mathcal{E}_4$ or $(n+D)\mathcal{E}_5$ edge changes, respectively
Decremental strong forest	$H\mathcal{E}_6$	Fixed representatives and approximate paths to the terminal set
Static Gomory–Hu tree	$H\mathcal{E}_6$	All pairwise minimum cut values and cut partitions

Here $H = n + m + 2$. In a static graph m counts input edges. For an update sequence it is the number of initial edges plus the number of updates. For spanners, D counts deletions. With $U = \log(H + 4)$ and $h = \log(U + 4)$, write $\mathcal{E}_j = \exp O(U^{j/(j+1)} h^{1/(j+1)})$.

Prior work. Chen et al. [5] introduced the minimum-ratio-cycle approach to almost-linear-time flow. Van den Brand et al. [4] gave a deterministic algorithm, and Chen et al. [6] extended the approach to incremental flow and strong components. Kyng et al. [10] derive minimum-ratio cycles from a dynamic distance oracle. The general matching algorithm of OpenAI [14] is randomized.

The deterministic routing bound and matching phase used here are proved in the general matching manuscript in [12]. Section 2 gives the parameter calculation behind the four-fifths exponent.

We work on a word RAM with word-sized identifiers and degree bounds. Longer arithmetic is charged at its actual word cost. Parallel edges retain their identifiers. A matching uses edges with distinct ends. For degree constraints a loop contributes two incidences.

2 Circulations and traversal length

Orient a multigraph arbitrarily. Its incidence matrix B records head minus tail, with a zero column for each loop. Given prices $\gamma_e \in \mathbb{Z}$ and lengths $\ell_e \in \mathbb{Z}_{>0}$, write

$$\rho = \max_{Bv=0,\ v\neq 0} \frac{|\gamma^{\mathsf{T}} v|}{\sum_e \ell_e |v_e|},$$

taking $\rho = 0$ for a trivial kernel. Orienting the support by the signs of v decomposes it into directed cycles with additive length and price. Thus a simple cycle attains any positive optimum.

Proposition 2.1 (Routing theorem [12, general matching, Section 8.1, Theorem 11.6 and Eq. (77)]). *For degree at most three, fix polynomial bounds in a size parameter S on graph size, integer labels and the total number of edits and queries. Each operation may depend on all completed operations. There is a deterministic data structure with a computable bound*

$$E = \exp O\Big((\log S)^{4/5}(\log\log(S+4))^{1/5}\Big).$$

A query returns a list of signed edges and forest paths projecting to a circulation v. Write a_e for the number of traversals of edge e in the list, and put $V = \sum_e \ell_e a_e$. If $\rho > 0$, then $V > 0$ and $|\gamma^{\mathsf{T}} v| \geq \rho V/E$.

Initialization costs $O(E)$ per record. Updates and queries cost $O(E)$ amortized, including path pieces and representation changes. Each native edge has at most E forest copies, counting inactive copies. A copy has one fixed signed native edge or an empty image until its retirement, which exposes the old image.

Consume or discard each returned list before the next routing operation. Path aggregates compute V and the price without expanding the list. Out-and-back traversals count twice in V.

The exponent. At one level of routing, replacing an edge e on a cycle by a walk R_e leaves a local correction $e - R_e$. The corrections and the routed cycle have the original total price. Their total traversal length therefore bounds the loss when one keeps the best ratio.

Put $U = \lceil \log_2 S \rceil$ and $h = \lceil \log_2(U+4) \rceil$. In [12, general matching, Eqs. (73)–(77)], set $p = s = t$, repair depth t^2 and logarithmic shrinkage $x = \lceil U/t \rceil$. The logarithmic local cost is $O(U/t + t^3 h)$, and each of $O(t)$ levels contributes $O(U/t^2 + t^3 h)$ to log work and inverse quality. Hence

$$\log E = O(U/t + t^4 h), \qquad t = \lfloor (U/h)^{1/5} \rfloor.$$

This choice gives the asserted exponent. The child-record growth has logarithm $O(U/t^2 + th) = o(x)$ and stored labels use $O(Ut) = o(x^2)$ bits. The cited construction fixes its cap $2^{c_E x}$ before the first update. Bounded sizes are handled directly.

Splitting degrees. In a numerical graph replace each vertex by a path with a port for every incident end. Give its connectors unit length and zero price. An original circulation lifts by putting the signed prefix imbalance on each connector. With M ports in all, the added length is at most $2M \sum_e |v_e|$. Projection preserves the price and the balances. New ends append ports, and repricing reuses the old ones.

3 Maximum matching

Lemma 3.1 (Odd paths [12, general matching, Lemma 3.2]). *Discard loops and isolated vertices. Replace a vertex of degree d by a path of $2d-1$ vertices, attaching its incident edges to distinct odd positions. Distinct ports separate parallel edges. For the resulting simple subcubic graph J,*

$$\nu(J) = \nu(G) + A, \qquad A = \sum_v (\deg_G(v) - 1),$$

where ν denotes the maximum matching size. Construction and decoding take linear time.

Proof. A matching of G leaves at most one occupied port per path. Leave that port available, or the first port if none is occupied. The even segments on either side supply $d-1$ internal edges.

Conversely, let X be the external edges of a matching of J and let $s_v = \deg_X(v)$. Every internal edge uses an even vertex and an unoccupied odd vertex. Hence its path supplies at most $\min(d-1, d-s_v) = d-1-(s_v-1)_+$ edges, where $x_+ = \max(x, 0)$.

Each vertex incident to X chooses one of its edges. Keep those chosen at both ends. This discards at most $\sum_v (s_v-1)_+$ edges, one charge per rejection, and gives a matching of size at least the input matching size minus A. □

Proposition 3.2 (Matching phase [12, general matching, Sections 7, 8.6 and 12.2])**.** *On an N-vertex subcubic graph, a phase with target $K \geq 1$ either gains $\Omega(\max\{1, K/\log(N+2)\})$ matching edges or certifies that the initial gap was less than K. On a certificate the phase keeps its gains. With the valid caps of Proposition 2.1, the cost per phase is $O((N+1)E^4 \operatorname{polylog}(N+2))$.*

The phase contracts factor-critical odd sets and prices circulations in an auxiliary allocation graph. If the initial gap was at least K, after i gains the symmetric difference with an optimum contains at least $K-i$ disjoint augmenting paths. The cited lemma turns their prefixes into a comparison circulation until the promised gain is reached. This is the low-gain certificate. The local projection costs sum over the phase by the volume bound in the cited construction. Its uniform implementation verifies a candidate maximum matching by an independent augmenting-path search.

Descending targets. Start with a greedy matching and the largest power of two $K \leq N/2$ for $N \geq 2$. Repeat until a certificate, then halve K. Stop after the certificate at $K = 1$. Smaller graphs are immediate.

At entry to a scale the gap is less than $2K$. The phase guarantee therefore gives $O(\log(N+2))$ phases per scale. The last certificate gives gap zero. Since J has $O(n+m)$ vertices and edges, Lemma 3.1 proves the matching assertion of Theorem 1.1.

4 Integral flow with a fixed potential

Fix balances b, a cost threshold F, and an integral starting flow within the bounds $[0, u_e]$. New arcs enter at zero. We first solve this threshold problem, then construct the starting flow and optimize over F.

4.1 A feasible direction

Let $N \geq 1$ bound the total number of arcs, including auxiliary arcs. Take a power of two $S \geq 16$ bounding n, N, all numerical data and $|F|$. Put $Q = S^8$, $\delta = 1/Q$ and $A = 4N$. Extend the natural logarithm by its tangent at δ,

$$\phi(s) = \begin{cases} \log s, & s \geq \delta, \\ \log \delta + (s-\delta)/\delta, & s < \delta, \end{cases} \qquad w(s) = \phi'(s) = 1/\max\{s, \delta\}.$$

For positive cost gap $z = c^{\mathsf{T}} f - F$, define

$$P(f) = \sum_e [\phi(f_e) + \phi(u_e - f_e)] - A \log z. \tag{2}$$

At each prefix use the following rule.

1. If $z \le 1/4$, round to an integral witness (Section 4.3).
2. Otherwise, query the priced graph. If the ratio test fails, this prefix has no flow of cost at most F. Retain f for the next insertion.
3. On success, add the scaled circulation of Lemma 4.2 and repeat.

Each update preserves $Bf = b$.

Lemma 4.1 (Comparison). *Write $l_e = w(f_e) + w(u_e - f_e)$ and $\beta_e = w(f_e) - w(u_e - f_e)$. If y is feasible with $c^{\mathsf{T}} y \le F$, then $v = y - f$ is a circulation and*

$$\nabla P(f)^{\mathsf{T}} v \ge \sum_e l_e |v_e| + A - 2N.$$

Proof. For $v_e \ge 0$, the difference $l_e|v_e| - \beta_e v_e$ is $2w(u_e - f_e)v_e \le 2$, since $v_e \le u_e - f_e$. For $v_e < 0$, the same bound follows from $-v_e \le f_e$. The cost term contributes $-Ac^{\mathsf{T}} v/z \ge A$. □

4.2 One step for both potentials

Take D to be the least power of two at least $\max(2, E)$ and put $\eta = 1/(64D)$. Each coordinate is read at an anchor $\widehat{f}_e$. For each arc term $\phi(a + \sigma f_e)$, $\sigma \in \{-1, 1\}$, evaluate $w(a + \sigma \widehat{f}_e)$ at that anchor. Denote the sum of these weights on e by L_e and their signed sum by $\widehat{\beta}_e$. Thus $|\widehat{\beta}_e| \le L_e$.

Freeze a price G_e at each edge. Give it integer price γ_e and length $\ell_e = \lceil QL_e \rceil$, with $|\gamma_e/Q - G_e| \le \eta L_e/32$. Give port connectors price zero and length one. Sign the returned list to make $T = \gamma^{\mathsf{T}} v \ge 0$, where v is its native projection. Recall its full traversal length V from Proposition 2.1.

Lemma 4.2 (Gain per step). *Suppose $L_e|f_e - \widehat{f}_e| \le \eta/1024$ for all e. If $V > 0$ and $T \ge 4\eta V$, choose a power of two μ such that*

$$\eta^2/512 \le t := \mu T/Q \le \eta^2/256. \tag{3}$$

Set $d = \mu v$. Let R be the remaining term of the potential after the arc terms. If R lies above its tangent from f to $f + d$ and

$$\nabla R(f) = \alpha(G - \widehat{\beta}), \qquad |\alpha - 1| \le \eta/16,$$

then the full potential increases by at least $t/2$.

Proof. The total weighted motion on the list is bounded by $M = \mu V/Q \le t/(4\eta) \le \eta/1024$. In particular $\sum_e L_e|d_e| \le M$.

Along the step, each slack s differs from its anchor $\widehat{s}$ by at most $2\eta/(1024 w(\widehat{s}))$. The map $s \mapsto \max(s, \delta)$ is 1-Lipschitz, so $|w(s) - w(\widehat{s})| \le (\eta/32) w(\widehat{s})$. Integration bounds the change in the arc terms below by $\widehat{\beta}^{\mathsf{T}} d - \eta M/32$.

Add the tangent inequality for R. Price rounding gives $|G^{\mathsf{T}} d - t| \le \eta M/32$, while $|\widehat{\beta}^{\mathsf{T}} d| \le M$. The total gain is at least $\alpha t - \eta M \ge t/2$. □

Prices for the threshold test. At the start of an epoch read all coordinates and store $r = z$. Use $G = \widehat{\beta} - Ac/r$. Start a new epoch whenever $|z/r - 1| > \eta/1024$. Between epochs refresh an edge when its accumulated motion requires a new anchor. The global cost term has the exact identity

$$\nabla P(f) - (r/z)G = \beta - (r/z)\widehat{\beta}.$$

As in Lemma 4.2, each true scale is within a factor $1 + O(\eta/1024)$ of L_e, and the right-hand side has magnitude at most $\eta L_e/32$ at edge e.

Round QG_e to the nearest integer. One slack is at most $u_e/2$, so $L_e \geq 1/(S+1)$. As $1/\eta = S^{o(1)}$, the required price accuracy holds. At most $2N$ ports occur, and the connectors and length rounding together add $O(S^{-6})Q\sum_e L_e|v_e|$ to the length of a lifted circulation. Lemma 4.1 therefore gives a rounded ratio at least $1/2$ whenever a feasible comparison exists. In that case the routing theorem guarantees $T \geq V/(2E) > 4\eta V$.

For an accepted list, $|c^{\mathsf{T}}d|/r = |(\widehat{\beta} - G)^{\mathsf{T}}d|/A \leq (M + t + \eta M/32)/A$. The new gap is at least half the old one. Convexity of $-A\log z$ now supplies the tangent inequality, and Lemma 4.2 applies with $\alpha = r/z$.

4.3 Stopping and rounding

Let $v_e(f) = (-f_e)_+ + (f_e - u_e)_+$ and $\mathrm{viol}(f) = \sum_e v_e(f)$. Each pair of arc terms satisfies

$$\phi(f_e) + \phi(u_e - f_e) \leq 2\log(1 + u_e) - v_e(f)/(2\delta). \tag{4}$$

Inside the box this follows from $\phi(s) \leq \log(1+s)$ for $s \geq 0$. At distance $a > 0$ outside the box the slacks are $-a, u_e + a$. Their contributions are at most $-a/\delta$ and $\log(1+u_e) + a$.

At a feasible starting flow above the threshold, $P \geq -O(N\log S)$. A new zero-flow arc reduces P by at most $O(\log S)$, and changing anchors leaves P unchanged. An integral start with $z \leq 1/4$ already meets the threshold. Otherwise the gap stays at least $1/8$ through the last step. Summing Lemma 4.2 and (4) gives at every completed step

$$J = O(N\eta^{-2}\log S), \qquad \mathrm{viol}(f) = O(N\delta\log S), \tag{5}$$

where J counts accepted steps across all arrivals.

Lemma 4.3 (Euler rounding [7])**.** *If $Bx = b$, b is integral and $2^k x$ is integral, one can round x to an integral flow y with the same balances in $O(n + (k+1)m)$ scalar operations. Each y_e lies in $I_e = [\lfloor x_e \rfloor, \lceil x_e \rceil]$, and an affine objective specified by its coefficients on $\prod_e I_e$ can be kept from increasing.*

Proof. At digit $j = k, \ldots, 1$, keep the edges with odd numerator $2^j x_e$. They have even degree at every vertex because $2^j b$ is even. Orient an Euler tour in each component and let $h_e = \pm 1$ record its direction, zero elsewhere. A loop forms a one-edge tour. Both $x \pm 2^{-j}h$ preserve the balances, stay in their integer intervals and lie on the next coarser grid. Their objectives average to the old value. Choose the cheaper sign. Each pass scans the edges and only their incident vertices. □

When $z \leq 1/4$, apply Lemma 4.3 with the objective

$$c^{\mathsf{T}}x + P_0\,\mathrm{viol}(x), \qquad B_0 = \sum_e |c_e|(u_e + 1), \qquad P_0 = B_0 + \max(F, 0) + 1.$$

It is affine on the rounding box because its breakpoints are integers. By (5), $P_0\,\mathrm{viol}(f) + 1/4 < 1$ for large S. The rounded objective is below $F + 1$, and each coordinate lies in $[-1, u_e + 1]$. One unit of violation would already give objective at least $P_0 - B_0 \geq F + 1$. The output is therefore feasible and its integral cost is at most F.

The same rounding applies to the local flow projections in the matching phase [12, general matching, Sections 8.3–8.5]. There one minimizes the total violation alone. An initial violation less than one rounds to zero.

4.4 Cost of maintaining prices

Store signed increments and absolute motion on every forest copy. At retirement transfer these to the native base and a motion counter. Single-edge pieces update both directly. Connectors carry only length. An exact reading adds live signed increments to the base. Signed sums, path additions and minimum remaining motion are dynamic-tree aggregates [2, 12].

For a sufficiently small fixed dyadic $c > 0$, flag an edge if a copy accumulates $c\eta/(EL_e)$ motion or the retirement counter reaches $c\eta/L_e$. Round thresholds down to powers of two. At most E live copies then give drift at most $2c\eta/L_e$ before a flag. Each flagged copy continues accumulating motion and emits just one notice. Complete the list, coalesce its notices by native edge, and clear each reported edge's absolute counters before repricing it at its exact value. Drain retirement notices under the old maps before the next query.

Each report accounts for $\Omega(\eta/E)$ weighted motion on its edge since the last reset. Equation (3) charges only $O(\eta)$ per list before cancellation, so J lists cause $O(EJ)$ reports. A report takes $O(E \operatorname{polylog} S)$ time. The gap estimate gives $|c^{\mathsf{T}}d|/r = O(\eta/N)$, so $\Omega(N)$ steps separate consecutive full epoch scans. With (5), the total time after the vertex scan is $O((N+1)E^4 \operatorname{polylog} S)$.

Lengths, prices and gaps have polynomial magnitude. A returned list has polynomially many traversals, so T, V are polynomially bounded, and every power of two in (3) has $O(\log S)$ bits. The iterates share a dyadic grid of this precision. Rounded reciprocals and exact signed cost sums take $\operatorname{polylog} S$ time per aggregate operation. The prefix work bound and the polynomial size of one further transaction justify a fixed polynomial history bound in Proposition 2.1.

Initialization and optimization. Shift initial lower bounds into the balances and constant cost. Reject inconsistent bounds or balances whose sum is nonzero. Otherwise adjoin a root. With net inflow positive, direct an artificial arc from the root to each vertex of positive balance and from each vertex of negative balance to the root. Give each arc capacity equal to the absolute balance and saturate it. Set the original flows to zero.

Let B_* bound $\sum_e |c_e| u_e$ on the original arcs over the current stream budget. Assign every artificial arc cost $2B_* + |F| + 1$. Any integral augmented flow of cost at most F uses only original arcs. An original witness extends by zero. This proves the incremental threshold assertion, including its integral witness.

For static optimization test B_* for feasibility, then binary-search the integer thresholds in $[-B_*, B_*]$. There are $O(\log S)$ tests, each on an instance of polynomial numerical range and $O(n+m)$ records. This proves the flow assertion of Theorem 1.1. For maximum s-t flow with nonnegative capacities, give the original arcs zero cost and a return arc $t \to s$ cost -1 and capacity $\sum_e u_e$. After deleting the return arc, take the vertices reachable from s in the residual network. They exclude t, since a residual path would increase the flow. Every outgoing arc of this shore is saturated and every entering arc has zero flow, giving a minimum cut.

4.5 Relaxing bounds and prices

Fix the balances and keep lower bounds nonnegative. Allow insertions at lower bound zero, capacity increases, lower-bound decreases, cost decreases and threshold increases. For polynomial-range integer data, the first feasible prefix and its witness take $H\mathcal{E}_4$ total time.

Split each native arc $e = uv$ at a private zero-balance vertex a_e. The gate $u \to a_e$ has initial bounds $[l_e^0, u_e^0]$ and cost zero. From a_e to v place an arc of the current cost and capacity U, an upper bound on all capacities. At an upper-bound increase add its increment to the

gate by a parallel arc. At a lower-bound decrease add the decrement by a reverse gate arc. At a cost decrease append a fresh arc from a_e to v at the new cost.

Conservation at a_e puts the net gate flow x_e within the current bounds. All priced arcs cost at least the current unit price. Conversely, any such x_e lifts with forward flow $\max(x_e, l_e^0)$, reverse flow $(l_e^0 - x_e)_+$, and all x_e on a cheapest arc. Thus projection preserves feasibility at the cost threshold. An increase of Δ in the threshold is represented by a loop of capacity Δ and cost -1, whose saturation subtracts exactly Δ. Process the constant number of insertions for each update as a batch. A budget restart rebuilds the current gates and resets the test to the current threshold.

Optimal potentials and the whole optimal face. Let f be an integral optimum of a feasible flow instance. Its directed residual graph has forward costs c_e and reverse costs $-c_e$ wherever the corresponding residual capacity is positive. A negative cycle would improve f, so the shortest distances p_v from a new zero-cost source joined to all vertices exist.

One more flow call recovers the distances. Give all residual and source arcs capacity n, each old vertex demand one, and the source supply n. One shortest path per vertex attains cost $\sum_v p_v$. Any feasible flow costs at least this much by summing its nonnegative reduced costs against its arc values. Hence every positive arc in an optimum of this auxiliary instance is tight. Any vertices unreachable from the source in the positive support would have positive total demand and no incoming flow. A search of that support finds a shortest-path tree and computes all p_v by summing its arc costs.

Put $\bar{c}_e = c_e - (B^\mathsf{T} p)_e$ on a native arc. Nonnegative residual reduced costs imply $\bar{c}_e(y_e - f_e) \geq 0$ for every feasible y. Since $B(y - f) = 0$, the difference of objective values is the sum of these nonnegative terms. Thus the whole optimal face is specified by the original bounds and balances together with $y_e = f_e$ whenever $\bar{c}_e \neq 0$. In particular f, p certify optimality and the description is obtained in $H\mathcal{E}_4$ time.

5 Strong components under insertions

For a nonnegative circulation, every positive arc lies on a directed cycle. The following version allows a small amount of negative flow.

Lemma 5.1. *Let f be a signed circulation on a directed graph. If $f_{uv} > \sum_e (-f_e)_+$, then v can reach u.*

Proof. Otherwise the vertices reachable from v contain v and exclude u. No arc leaves this set. The signed sum of its entering flows is zero, so f_{uv} is at most their total negative part. □

Theorem 5.2. *Strongly connected component labels can be maintained under arc insertions in $H\mathcal{E}_4$ total time, with constant-time membership tests after initialization and after each insertion.*

Proof. Start with singleton blocks and a zero circulation f on the quotient. Discard internal arcs and insert new arcs at zero. Following the contraction principle of Chen et al. [6, Section 8.3], merge the ends of an arc when its flow reaches $1/4$.

Choose a power of two S of polynomial size large enough to cover the $O(n + m \log(n+2))$ ports counted below. Use Q, δ, η from Section 4.2 and the one-sided potential $\Phi(f) = \sum_e \phi(f_e)$. The frozen length and price of a native arc are both $\lceil Qw(\widehat{f}_e) \rceil$. Queue each native arc whose exact reading reaches $1/4$. Drain the queue before querying, checking that each queued arc is still live, has value at least $1/4$, and has distinct ends.

A surviving anchor is below $1/4$. The drift between reports and the raw motion in a step are together at most $2\eta \max(\widehat{f}_e, \delta)/1024$ in that coordinate. Thus $f_e < 1/2$ up to each

contraction. A directed cycle has rounded price equal to its length on native edges. Each native length is at least $4Q$, whereas port detours cost at most $2S\sum_e |v_e|$. The lifted cycle has ratio at least $1/2$. Failure of the acceptance test therefore certifies an acyclic quotient.

On an accepted step Lemma 4.2 applies with $G = \widehat{\beta} = w(\widehat{f})$, $R = 0$, $\alpha = 1$. All arc terms are nonpositive, and $\phi(s) \le -(-s)_+/\delta$ for $s < 1/2$. A birth contributes $\log\delta - 1$, and deleting internal arcs only increases the potential. With m an upper bound on births, it follows that

$$-m(1-\log\delta) + \Omega(J\eta^2) \;\le\; \Phi(f) \;\le\; -\delta^{-1}\sum_e (-f_e)_+.$$

Consequently $J = O(m\eta^{-2}\log S)$ and the total negative part is less than $1/4$ for large S.

Lemma 5.1 now makes every queued contraction safe. Adding the two old balance equations cancels internal arcs and preserves conservation in the quotient. Inductively each block lies in a true strong component. A quotient path lifts to an original walk by joining successive representatives within their true component. Conversely, original paths project to quotient walks. Hence an acyclic quotient gives precisely the true partition.

At a merge, move the smaller block by number of vertices. Relabel its vertices, retire internal arcs and append a fresh port for each moved surviving end on the larger block's path. Old ports stay unused, so all active ports of a block lie on one path. Test the threshold at each retargeted arc, preserving its exact signed value throughout the edits. Each vertex and arc end moves $O(\log(n+2))$ times, so $O(n + m\log(n+2))$ ports suffice. The motion accounting of Section 4.4 and one failed query per settled prefix give the asserted total time. □

Unknown stream length. For both online algorithms, start with separate power-of-two bounds on the graph size and the largest input magnitude. Whenever an arrival exceeds either bound, raise the exceeded bounds to the least powers of two covering the prefix and rebuild on that prefix before answering. For flow use the artificial starting flow. For components retain the last published partition and initialize the current quotient with zero flow. The budgets stay $O(H)$ for size and $H^{O(1)}$ for magnitudes. The $O(\log H)$ rebuilds preserve (1).

5.1 A sparse certificate at every prefix

The algorithm can maintain an insertion-only subgraph with the same strong components at every prefix. When there are c components, this certificate contains at most $2(n-c)$ native arcs. A bidirected tree requires both arcs of each edge, attaining the bound. The additional total work is $O((n+m)\log(n+2))$.

Use the component merges at the end of each external update, after the quotient has been certified acyclic. If a new component has $r \ge 2$ old components as children, its quotient on these children is strongly connected. Indeed, paths between vertices of a strong component stay inside it. From a largest child find an outward spanning arborescence and an inward spanning arborescence of the quotient. Add their at most $2(r-1)$ native arc identifiers. The earlier certificate was strongly connected inside each child, so the new certificate is strongly connected inside the merged block. It remains a subgraph of the input. Summing $r-1$ over settled merges gives $n-c$. If C_t is the certificate at time t and C_* its final value, then $C_t \subseteq C_* \cap G_t \subseteq G_t$. Thus the final certificate with the original arrival times recovers the strong components at every prefix.

To find the quotient, keep the previous published labels until the end of the update. Group the transient contractions by their final block and scan the original incidence lists of every child except the largest. This sees every interchild arc. Retain only arcs internal to the new block, then relabel the smaller children. Each scanned vertex joins a published component at least twice as large, and an arc scan is charged to its scanned endpoint. The

grouping itself costs at most logarithmic work per contraction using the old labels and their union log. A size-budget restart may start with the preceding published partition and zero quotient flow, leaving the certificate intact. This partition is still a refinement of the true partition, which is all the contraction proof requires. The same construction processes the initial graph at time zero.

For first-prefix queries, give each vertex a leaf at time zero and each settled merge a node with its prefix as timestamp. Its children are the merged components. Join the remaining roots to a virtual root at time $+\infty$. The first common prefix is the timestamp of the lowest common ancestor. An Euler tour of this $O(n)$-node tree reduces that ancestor to a range minimum of depths between the leaves' first visits. Precompute minima on all intervals of power-of-two length and tabulate integer logarithms. Two overlapping intervals then answer each query in constant time after $O(n \log(n+2))$ time and space.

6 Cardinalities of degree-constrained subgraphs

A (g, f)-factor is a set X of edge records satisfying $g_v \le \deg_X(v) \le f_v$ at every vertex. A loop contributes two to the degree. Tutte reduced prescribed degrees to perfect matching [15]. For intervals we use optional matching vertices to make the cardinality of a factor visible in the size of a matching.

Theorem 6.1. *For an undirected multigraph with integral degree intervals, two maximum-matching computations on simple subcubic graphs of size $O(n + m \log(m+2))$ find smallest and largest factors or certify infeasibility, in total time $H\mathcal{E}_4$. A linear-size representation gives constant-time edge membership and $O(s+1)$-time reporting for a factor of each feasible size s. A single prescribed size interval can instead be tested, with a witness, by one perfect-matching computation.*

Encoding a degree interval. At a vertex of degree $d > 0$, intersect its interval with $[0, d]$, rejecting an empty intersection. Check isolated vertices directly. Assign its incidences to the inputs of a Beneš permutation network padded to a power of two [3]. The network has $O(d \log(d+2))$ positions and arcs, and indegree and outdegree at most two. It routes any equally sized sets of inputs and outputs by vertex-disjoint paths.

At size one use an arc. At size $2k$, the two inputs of each pair can enter either of two size-k copies at the corresponding port. Reverse these connections at the outputs. Extend a prescribed bijection to a permutation. Its connections form a 2-regular bipartite multigraph on input and output pairs. Alternating colors on every circuit gives one permutation for each copy.

Split every position into a minus and a plus vertex joined by an idle edge. Each network arc joins its tail's plus to its head's minus. Each original edge joins its two input minus vertices, using separate ports even for a loop. At vertex v, delete the first g_v output plus vertices. Mark the next $f_v - g_v$ output plus vertices as optional. All remaining vertices must be covered.

Each covered internal position is idle or has one active arc in and one out. Acyclicity makes the active arcs disjoint paths from selected incidences to outputs. Deleted plus vertices force active outputs. Optional plus vertices are uncovered exactly at active outputs. All remaining outputs are idle. Conversely, the routing property extends every factor to such a matching by using idle edges off its paths. If $s = |X|$, the total number of uncovered optional vertices is

$$2s - \sum_v g_v. \tag{6}$$

Each split vertex has degree at most three. Distinct incidence ports keep the graph simple for parallel records as well as loops.

Two matching computations. Call the graph just constructed J_-. Take two copies and join corresponding optional vertices by single edges. A perfect matching restricts on either copy to a matching covering all required vertices. Conversely, duplicate any such matching and join the uncovered pairs. Thus the first maximum-matching computation either certifies infeasibility or supplies a matching M_- of J_- that covers the required vertices.

For J_+, attach a private leaf to every optional vertex of J_-. Now require the old vertices to be covered and allow leaves to be free. Extend M_- to a seed M_+ by matching its uncovered vertices to their leaves. Deleting the leaf edges gives the inverse on matchings covering the required vertices. The uncovered leaves correspond to inactive optional outputs. For a factor of size s, their number is $\sum_v f_v - 2s$. Both graphs are simple and subcubic since optional outputs originally had just an idle edge.

Find a maximum matching on $J_- \sqcup J_+$ in the second computation. On either graph let Q be the restriction of the answer and M its seed. Starting from M, flip the paths of $M \triangle Q$ on which Q has one extra edge. Their ends are free in M, so required vertices stay covered. There is no path with an extra M-edge, since it would augment Q. The result has size $|Q|$. By (6) this minimizes the factor size on J_- and maximizes it on J_+.

All intermediate sizes. Let X, Y be the two extremal factors. Color $X \setminus Y$ blue and $Y \setminus X$ red. At each vertex pair opposite-colored incidences as far as possible. The pairs partition the difference into alternating trails and circuits. All unpaired ends at a vertex have the same color, even when the vertex occurs twice on a trail.

Flipping any collection therefore keeps each degree between its values in X and Y. Each trail changes cardinality by 0 or ± 1, and no trail can decrease it, by minimality of X. There are exactly $|Y| - |X|$ increasing trails. Flip these in any order, assigning the neutral trails and circuits to the last flip if the sizes differ. If the sizes agree, retain X. Pairing and tracing take linear time, and each edge changes at most once.

Store the initial state and flip time by original record index. At size s, the factor consists of the permanently selected edges, a prefix of the entering edges ordered by flip time, and a suffix of the departing edges in that order. The trail scan tabulates the two endpoints for all sizes in $O(m + 1)$ space. This proves the membership and reporting bounds.

One prescribed interval. For $a \le s \le b$, the number of active optional outputs in J_- must belong to $[2a - \sum_v g_v, 2b - \sum_v g_v]$ by (6). Intersect it with $[0, p]$, where p is the number of optional outputs, and retain values of parity $\sum_v g_v$. Reject if none remain. Write the endpoints as A, B. Connect each optional plus vertex to a separate input minus vertex of one more permutation network. Delete its first A output plus vertices and pair the next $B - A$ by disjoint edges. Require a perfect matching throughout.

The new network receives precisely the active optional outputs. Its first A outputs must be active, and both members of each output pair are active together. Conversely any allowed count routes to the first A outputs and the appropriate number of pairs. With no optional vertices, test the count zero and use J_- itself. The additional network is simple and subcubic and has $O(m \log(m + 2))$ records. Its logarithmic factor is absorbed by (1), completing the proof of Theorem 6.1.

6.1 Local rounding and integer cardinality slices

The same exchanges control the support of a convex combination. For an integer matrix A, a move is a nonzero integral vector in its kernel. A summand g is *conformal* to h if $g_i h_i \geq 0$ and $|g_i| \leq |h_i|$ in every coordinate. A move is primitive if it has no other conformal move. These moves form the Graver basis [8], finite by Dickson's lemma in each orthant. Successive subtraction expresses every move as a sum of conformal primitive moves. For an integral linear form c, write

$$D_c = \max_{g \text{ primitive}} |c^{\mathsf{T}} g|, \qquad D_i = \max_{g \text{ primitive}} |g_i|,$$

with empty maxima zero.

Theorem 6.2 (Local support and integer levels)**.** *Fix a finite integral box and an integral right-hand side, and let S be the integer solutions of $Ay = b$ in the box. Each $x \in \operatorname{conv} S$ has a convex representation in which the values of coordinate i span at most D_i, and the values of $c^{\mathsf{T}} y$ span at most D_c. Each bound is sharp for its matrix and statistic. Intersecting the integer hull with any integral level of $c^{\mathsf{T}} y$ commutes with taking the integer hull, for every box and right-hand side, exactly when $D_c \leq 1$.*

Proof. Among representations of x, minimize the average of $\|y\|_2^2$. If $h = z - y$ for two support points has $|h_i| > D_i$, take a conformal primitive summand g using i. Exchange equal positive masses at y, z for masses at $y + g, z - g$. Both new points lie between the old ones coordinatewise and have the same sum. The sum of the two squared norms decreases by

$$2 \sum_j |g_j|(|h_j| - |g_j|) > 0.$$

Thus the support fits in a box of the claimed widths. Within that box minimize the average of $(c^{\mathsf{T}} y)^2$. If $c^{\mathsf{T}} h > D_c$, some conformal summand has $0 < c^{\mathsf{T}} g \leq D_c$. The same exchange strictly decreases this second average and stays in the box.

For a primitive $g = g^+ - g^-$, the box between g^- and g^+ with right-hand side Ag^- has just these two solutions, by primitivity. Their midpoint needs both, proving sharpness. If $D_c > 1$, such a segment crosses an intermediate integer level without an integer solution there. If $D_c \leq 1$, the support uses only the floor and ceiling of its mean statistic, proving the assertion for levels and also for intervals with integral endpoints. □

Corollary 6.3 (Capacitated factors)**.** *Allow integral lower and upper bounds on every edge multiplicity and on every vertex degree. Let P be the convex hull of the integral factors and put $s(y) = \sum_e y_e$. For integers $a \leq b$,*

$$P \cap \{a \leq s(x) \leq b\} = \operatorname{conv}\{y \in P \cap \mathbb{Z}^m : a \leq s(y) \leq b\}.$$

The minimum of any linear cost over factors of size k is a discrete convex function on their interval of feasible sizes.

Proof. Write the upper degree requirements as equations with one bounded slack at each vertex. For a move, pair opposite-colored incidences at each vertex as in Section 6, now using one copy per unit of difference and a single incidence per slack copy. Every component of the pairing is a circuit or a trail ending at two slack copies, and gives a conformal move. A primitive move consists of just one component. Its ordinary edge copies alternate in color, so its size change is 0 or ± 1. Theorem 6.2 gives the formula. For optimizers of sizes $k - 1, k + 1$, their midpoint has size k and decomposes into integral factors of size k. Some such factor costs at most the mean of the two optima. □

Theorem 6.4 (Sharp local rounding)**.** *Suppose each column of A consists of at most two signed unit incidences, allowing both at the same row. In Theorem 6.2, each $D_i \le 2$. If qx is integral for an integer $q \ge 1$, the same representation can satisfy*

$$|y_i - x_i| \le D_i(1 - 1/q) \qquad \text{in every coordinate.} \tag{7}$$

For fixed A, i, q the radius is sharp over all such fibers and points. For capacitated factors, taking $c = s$ gives these coordinate bounds together with support sizes $\lfloor s(x) \rfloor$ and $\lceil s(x) \rceil$.

Proof. Put $R_q = 2 - 2/q$. Consider prescribed degrees d in a loopless graph with integral bounds $[l, u]$. Put $v = x - l$, $c = u - l$ and $d' = d - \deg(l)$. For a shore W and $F \subseteq \delta W$, define

$$L = v(\delta W \setminus F) + (c - v)(F), \qquad \pi = d'(W) + c(F).$$

Here a vector evaluated on a set denotes the sum of its entries. In addition to the bounds and degree equations, the capacitated matching polytope is defined by $L \ge 1$ when π is odd [11, Eq. (4)]. The identity $v(\delta W) = d'(W) - 2v(E(W))$ gives $qL \equiv q\pi \pmod 2$. Hence a violated odd inequality has $L \le 1 - 2/q$.

The case $q = 1$ is immediate. For $q \ge 2$, intersect the box with $[\lceil x_e - R_q \rceil, \lfloor x_e + R_q \rfloor]$ on each edge. Every changed bound has slack strictly above $R_q - 1 = 1 - 2/q$ even in the smaller box. In either box a violated inequality must therefore use unchanged crossing bounds. Its parity depends only on $d(W)$ and those bounds, as internal edges count twice. Thus the two boxes have exactly the same violated odd inequalities at x, with the same left sides. Hull membership gives a convex representation in the smaller box.

To pass from prescribed degrees to A, translate each variable to $0 \le t \le c_e$. Replace a negative incidence $-t$ by $(c_e - t) - c_e$, absorbing the constant into the target. Equal contributions at two ends give one ordinary edge. Complementary contributions give a two-edge path with internal degree c_e. A loop with two equal contributions is replaced by a three-edge path back to its vertex, with both internal degrees c_e and values $z, c_e - z, z$. All edges representing this column have bounds $[0, c_e]$. For columns with one end, duplicate the rows and the two-ended edges, then join the two copies of the incident row by an edge carrying its required contribution. A zero column supplies an independent floor-or-ceiling mixture. Duplicating a solution gives a lift, while either copy projects to a native solution. The lifts preserve integrality after multiplication by q and preserve absolute coordinate displacements.

The resulting representation is in the native box of radius R_q. Apply Theorem 6.2 inside this box, with the same linear form c. To see that $D_i \le 2$, apply the radius bound at $q = 2$ to the midpoint in the primitive two-point example from that theorem. A coordinate with $D_i = 1$ uses only the floor and ceiling of x_i, giving radius $1 - 1/q$, and a coordinate with $D_i = 0$ is fixed. For sharpness, take the same two-point example with weights $1/q$ and $1 - 1/q$ when $q \ge 2$. For factors apply the theorem to the degree equations with bounded slacks, using the size statistic of Corollary 6.3. □

For undirected graphs even the nearest factor can be at distance $2 - 2/q$. Join a center to one vertex of each of q disjoint triangles by a spoke of capacity two. Triangle edges have capacity one. Prescribe degree two at the center and its neighbors and degree one at the other vertices. Each triangle uses its spoke zero or two times, so exactly one spoke is doubled. Averaging the q choices gives spoke value $2/q$ and distance at least $2 - 2/q$ from every factor. For a directed flow, the primitive moves are signed simple cycles. Theorem 6.2 then gives floor-or-ceiling support and radius $1 - 1/q$.

7 Removing the forest part

A spanning forest separates the part of an instance that requires a general graph algorithm. Both problems admit constant-size summaries on a tree piece with two attachment vertices. Linear-size static reductions for cardinality matching parameterized by the feedback edge number were given by Mertzios, Nichterlein and Niedermeier [13].

Theorem 7.1 (A common reduction)**.** *Let $r = |E(G)| - n + c$ be the cycle rank of the current graph G, where c counts the components of its underlying undirected graph, including isolates. In $O(n + m)$ time, maximum matching and directed bounded minimum-cost flow each reduce to an instance of the same problem with $O(r+1)$ vertices and edges, or are solved outright. An optimum of the reduced instance lifts to an original optimum, with a fixed offset in the objective. The reduction applies to weighted matching and preserves unit weights in the unweighted case. Under edge insertions, deletions and label changes on a fixed vertex set, the reduction can be maintained in amortized polylogarithmic time per edit and extracted in $O((r + 1)\log(n + 2))$ time, to obtain the reduced instance and its decoding information. All labels have polynomial magnitude in the size bound.*

Proof. Choose a spanning forest and mark the endpoints of its r complementary edges. In each tree containing marks, prune unmarked leaves, then suppress unmarked vertices of degree two. The remaining unmarked vertices have degree at least three, so there are at most as many as marks. The resulting paths, with their pruned branches, have two attachment vertices. Branches at a retained vertex form one piece with a single attachment. A tree without marks contributes just its optimum or an infeasibility flag. Process the pruned branches toward their roots and then sweep each retained path. Each piece needs only a summary at its attachments, called its ports.

Matching. For each subset of available ports, keep the optimum after deleting the other ports. When two pieces meet, assign their shared vertex to at most one child and maximize the sum. For ports a, b, subtract the empty-port value. Write p, q, s for the gains when a, b, or both are available. Since $s \geq \max(p, q) \geq 0$, the triangle on a, b, w with weights p, q, s on aw, bw, ab realizes all four values. One port needs only a private leaf. For unit weights, each port increases the optimum by at most one. If both singleton gains vanish, use either no edge or ab, according to s. A single positive gain needs a leaf at that port. Two positive gains need a common neighbor for $s = 1$ and separate leaves for $s = 2$. Thus the reduced instance can itself be unweighted.

Flow. Impose balances on the interior of a tree piece. In a one-port piece, eliminating leaves fixes all edge flows, the contribution at the port and the cost. In a two-port piece, prescribe the net contribution t at its first port. Leaf elimination again fixes every flow. Off the path between the ports it is constant, and on the path it has the form $d_e \pm t$. All capacities therefore cut out one integral interval for t. The second port receives a constant minus t, and the cost is affine in t. Absorb the constants in the balances and objective. An empty interval signals infeasibility. Otherwise one directed arc represents the interval and cost, shifting its lower bound if needed. At a shared port that becomes internal, the two child contributions sum to its balance. If the port survives, add their contributions. These joins use only unit pivots and interval intersections.

Each join above uses a constant-size summary of each child and saves the choices for decoding. Restoring the complementary edges completes the reduction. A reverse pass recovers a solution on the original records in $O(n + m)$ time. Both algorithms take $O(H) + (r + 1)\mathcal{E}_4$ total time. For cardinality matching the overhead can be evaluated at the reduced size $O(r + 1)$, since the reduced graph is unweighted.

For the dynamic assertion maintain a spanning forest and the list of complementary edges by dynamic connectivity [9], and store these summaries in balanced top trees [2]. A cluster

has at most two ports and a forest edit changes $O(\log(n+2))$ clusters. A change to a vertex balance affects only ancestors of an arbitrary incident forest edge, or just the isolate record if there is no such edge. Maintain the sum of the finite optima and the count of infeasible whole trees. Keep loops outside the forest and bucket parallel edges by endpoint pair. If a deleted forest record has a surviving parallel copy, update its representative and summary. Flow loops stay among the complementary edges.

At extraction, mark one incident forest edge per nonisolated marked vertex. Open the root-to-leaf paths to these edges and keep their two ends. Every other exposed cluster meets the marked vertices it contains only at ports. This produces $O((r+1)\log(n+2))$ pieces. Their two-port paths form a forest. Fold its unmarked leaves and suppress its unmarked degree-two vertices with the same joins as above. Untouched tree values are read from the maintained totals after subtracting each touched tree once, including marked isolates. The resulting instance has the size asserted in the theorem. □

8 Distances, spanners and cut trees

The metric rows of Table 1 use the routing constructions in [12]. Lengths and capacities are nonnegative integers of polynomial magnitude.

Directed distances under insertions. For a fixed source, the incremental shortest-path algorithm of [12, incremental SSSP, Theorem 1.1 and Section 10] gives $(1+\varepsilon)$-estimates in constant query time and reports a simple path of k original edges in $O((k+1)\operatorname{polylog} H)$ time. The total work is

$$H \exp O\left(U^{4/5}h^{1/5} + \sqrt{U\big(h+\log(1/\varepsilon)\big)}\right)$$

for $0 < \varepsilon \le 1/2$ with $\log(1/\varepsilon) \le cU$ for a sufficiently small absolute c. Constant or inverse-polylogarithmic accuracy gives the table entry.

Undirected distances and spanners. For fully dynamic point queries on undirected graphs, total work $H \exp O(U/t + t^3 h)$ gives stretch $\exp O(t^4 h)$ whenever $t \to \infty$ and $t^2 h = o(U)$ [12, APSP frontier, Theorem 2.1]. To extend the positive-length tradeoff to zero lengths, maintain connectivity in the zero-edge subgraph [9] and run the oracle on lengths $n\ell_e + 1$. A simple shortest path uses at most $n-1$ edges, so for a finite distance $d \ge 1$ the transformed distance divided by n lies in $[d, d+(n-1)/n] \subseteq [d, 2d]$. Return zero for a zero-connected pair and otherwise divide the estimate by n, preserving infinity. The added work is $H \operatorname{polylog} H$. Taking $t \asymp (U/h)^{1/5}$ gives the balanced bounds in the table.

The two explicit spanners in [12, dynamic spanners, Parts I and II] support insertions and deletions on a fixed vertex set of arbitrary degrees. Recourse counts changes to their original edge sets. For the second spanner it depends only on the vertex and deletion counts.

Decremental forests and static cuts. For a bounded-degree undirected graph and fixed terminals, the forest of [12, decremental strong forest, Sections 11–14] gives $(1+\varepsilon)$-approximate paths to the terminals for constant or inverse-polylogarithmic ε. Representatives are fixed, paths project to walks, and total recourse is $H\mathcal{E}_6$. Substitution into the Gomory–Hu construction of Abboud et al. [1] gives the cut-tree bound in [12, decremental strong forest, Section 16] for arbitrary degrees. Removing any lightest tree edge on the path between two vertices partitions the original vertices into the shores of a minimum cut for that pair.

Acknowledgment. The author used OpenAI models to formulate results, develop and review proofs, draft the manuscript, and implement finite checks. The source manuscripts [12] were prepared with GPT-6 Astra Pro.

References

[1] A. Abboud, R. Kyng, J. Li, D. Panigrahi, M. Probst Gutenberg, T. Saranurak, Weixuan Yuan and Wuwei Yuan. Deterministic almost-linear-time Gomory–Hu trees. arXiv:2507.20354, 2025.

[2] S. Alstrup, J. Holm, K. de Lichtenberg and M. Thorup. Maintaining information in fully dynamic trees with top trees. *ACM Transactions on Algorithms* 1 (2005), 243–264.

[3] V. E. Beneš. Optimal rearrangeable multistage connecting networks. *Bell System Technical Journal* 43 (1964), 1641–1656.

[4] J. van den Brand, L. Chen, R. Kyng, Y. P. Liu, R. Peng, M. Probst Gutenberg, S. Sachdeva and A. Sidford. A deterministic almost-linear time algorithm for minimum-cost flow. *FOCS* (2023), 503–514.

[5] L. Chen, R. Kyng, Y. P. Liu, R. Peng, M. Probst Gutenberg and S. Sachdeva. Maximum flow and minimum-cost flow in almost-linear time. *FOCS* (2022), 612–623.

[6] L. Chen, R. Kyng, Y. P. Liu, S. Meierhans and M. Probst Gutenberg. Almost-linear time algorithms for incremental graphs: cycle detection, SCCs, s-t shortest path, and minimum-cost flow. *STOC* (2024), 1165–1173.

[7] E. Cohen. Approximate max-flow on small depth networks. *SIAM Journal on Computing* 24 (1995), 579–597.

[8] J. E. Graver. On the foundations of linear and integer linear programming I. *Mathematical Programming* 9 (1975), 207–226.

[9] J. Holm, K. de Lichtenberg and M. Thorup. Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. *Journal of the ACM* 48 (2001), 723–760.

[10] R. Kyng, S. Meierhans, M. Probst Gutenberg and A. Sulser. A simpler and faster min-cost flow solver via min-ratio cycles from distance oracles. arXiv:2609.23852, 2026.

[11] A. N. Letchford, G. Reinelt and D. O. Theis. Odd minimum cut sets and b-matchings revisited. *SIAM Journal on Discrete Mathematics* 22 (2008), 1480–1487.

[12] Yilin Liu and GPT-6 Astra Pro. Graph algorithms for matching and flow. *Algorithm Frontier*, source snapshot `77140c2` (2026).

[13] G. B. Mertzios, A. Nichterlein and R. Niedermeier. The power of linear-time data reduction for maximum matching. *Algorithmica* 82 (2020), 3521–3565.

[14] OpenAI. Almost-linear-time maximum-cardinality matching in general graphs. OpenAI Math Release, 2026.

[15] W. T. Tutte. A short proof of the factor theorem for finite graphs. *Canadian Journal of Mathematics* 6 (1954), 347–352.