More flows and cuts

M is incidence matrix of Directed graph

$$\begin{array}{c|c} & \xrightarrow{\text{edges}} \\ \text{verts} \downarrow & M_{va} = \begin{cases} 0 & \text{no incidence} \\ -1 & a \text{ leaves } v \\ +1 & a \text{ enters } v. \end{cases} \end{array}$$

Thm: the incidence matrix M is Totally Unimodular

Proof: Induction:

- consider a minor B of dim K. ($K=1$; trivial)
- 3 cases.
  - some col is all zeros, $\det(B)=0$
  - B has at least 1 col that has exactly 1 non-zero
    $B = \begin{bmatrix} \pm 1 & b^T \\ 0 & B' \end{bmatrix}$; induction; $\det B = \pm \det B'$
  - B has all cols with 2 non-zeros then $1^T B = 0 \Rightarrow \det B = 0$ $\square$

Max-Flow, Min cut via LP.

<u>Thm</u>: $D=(V,A)$ a directed graph, c a capacity $s,t \in V$ $c: A \to R^+$. The max value of an s-t flow is equal to the min capacity of an s-t cut

<u>Proof</u>:
we need only show that there is a flow $x \le c$ and a cut whose capacity is not > flow

M is incidence matrix
M' is reduced incidence matrix, delete rows csp to s, t

$M'x = 0$ ← Kirchoffs law.

w is row of ~~the~~ M csp to t.
for arcs, $w_a = +1$ enters t, -1 exit,
max value of s-t flow is

$$\max \{ w'x \ \text{st} \ M'x = 0,\ 0 \le x \le c \}$$

From LP duality, this is

$$\cancel{\text{arg}}\ \min\ y^T c \quad \text{st} \quad y \geq 0$$
$$y^T + z^T M' \geq w^T$$

$$\begin{bmatrix} I & M'^T \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \geq w$$

This is TUM because M' is TUM

w is <u>integer</u> so y, z are <u>integer</u> at minimum.

now ~~[illegible]~~ write $\tilde{z} = [z;\ \underset{t\ \text{posn}}{-1};\ \overset{s\ \text{posn}}{0}]$.

this means

$$y^T + \tilde{z}^T M \geq 0$$

Now consider $U = \{v \in V \mid \tilde{z}_v \geq 0\}$

U contains s, not t

now we want to show that the max flow is = the min cut

$$w^T x = \text{max flow} = y^T c$$

we already have

min cut $\geq$ max flow

so all we need is

$$c(\delta^{out}(U)) \leq \text{max flow} = y^T c$$

↑ value of this cut

to show this, need only show

$(u,v) =$ edge $a \in \delta^{out}(U) \Rightarrow y_a \geq 1$

(because there are other y's, $y \geq 0$)

now $\tilde{z}_u \geq 0$, $\tilde{z}_v \leq -1$

$y_a + \tilde{z}_v - \tilde{z}_u \geq 0$ from $y^T + \tilde{z}^T M \geq 0$

so $y_a \geq z_u - z_v \geq 1$ $\square$

Proof. Consider the proof of Theorem 10.6. We do at most ϕ iterations, while each iteration takes $O(n+t)$ time, where t is the number of arcs deleted. ∎

Hence, similarly to Corollary 10.6a one has:

Corollary 10.11a. *For integer capacities, a maximum flow can be found in time $O(n(\phi+m))$, where ϕ is the maximum flow value.*

Proof. Similar to the proof of Corollary 10.6a. ∎

Therefore,

Corollary 10.11b. *For integer capacities, a maximum flow can be found in time $O(nm \log C)$.*

Proof. In the proof of Theorem 10.10, a maximum flow with respect to c' can be obtained from $2f''$ in time $O(nm)$ (by Corollary 10.11a), since the maximum flow value in the residual graph $D_{f''}$ is at most m. ∎

10.8b. Complexity survey for the maximum flow problem

Complexity survey ($*$ indicates an asymptotically best bound in the table):

$O(n^2mC)$	Dantzig [1951a] simplex method
$O(nmC)$	Ford and Fulkerson [1955,1957b] augmenting path
$O(nm^2)$	Dinits [1970], Edmonds and Karp [1972] shortest augmenting path
$O(n^2m \log nC)$	Edmonds and Karp [1972] fattest augmenting path
$O(n^2m)$	Dinits [1970] shortest augmenting path, layered network
$O(m^2 \log C)$	Edmonds and Karp [1970,1972] capacity-scaling
$O(nm \log C)$	Dinits [1973a], Gabow [1983b,1985b] capacity-scaling
$O(n^3)$	Karzanov [1974] (preflow push); cf. Malhotra, Kumar, and Maheshwari [1978], Tarjan [1984]
$O(n^2\sqrt{m})$	Cherkasskiĭ [1977a] blocking preflow with long pushes
$O(nm \log^2 n)$	Shiloach [1978], Galil and Naamad [1979,1980]
$O(n^{5/3}m^{2/3})$	Galil [1978,1980a]

≫

continued

	$O(nm \log n)$	Sleator [1980], Sleator and Tarjan [1981,1983a] dynamic trees
$*$	$O(nm \log(n^2/m))$	Goldberg and Tarjan [1986,1988a] push-relabel+dynamic trees
	$O(nm + n^2 \log C)$	Ahuja and Orlin [1989] push-relabel + excess scaling
	$O(nm + n^2\sqrt{\log C})$	Ahuja, Orlin, and Tarjan [1989] Ahuja-Orlin improved
$*$	$O(nm \log((n/m)\sqrt{\log C}+2))$	Ahuja, Orlin, and Tarjan [1989] Ahuja-Orlin improved + dynamic trees
$*$	$O(n^3/\log n)$	Cheriyan, Hagerup, and Mehlhorn [1990,1996]
	$O(n(m + n^{5/3} \log n))$	Alon [1990] (derandomization of Cheriyan and Hagerup [1989,1995])
	$O(nm + n^{2+\varepsilon})$	(for each $\varepsilon > 0$) King, Rao, and Tarjan [1992]
$*$	$O(nm \log_{m/n} n + n^2 \log^{2+\varepsilon} n)$	(for each $\varepsilon > 0$) Phillips and Westbrook [1993,1998]
$*$	$O(nm \log_{\frac{m}{n \log n}} n)$	King, Rao, and Tarjan [1994]
$*$	$O(m^{3/2} \log(n^2/m) \log C)$	Goldberg and Rao [1997a,1998]
$*$	$O(n^{2/3}m \log(n^2/m) \log C)$	Goldberg and Rao [1997a,1998]

Here $C := \|c\|_\infty$ for integer capacity function c. For a complexity survey for unit capacities, see Section 9.6a.

Research problem: Is there an $O(nm)$-time maximum flow algorithm?

For the special case of *planar* undirected graphs:

	$O(n^2 \log n)$	Itai and Shiloach [1979]
	$O(n \log^2 n)$	Reif [1983] (minimum cut), Hassin and Johnson [1985] (maximum flow)
	$O(n \log n \log^* n)$	Frederickson [1983b]
$*$	$O(n \log n)$	Frederickson [1987b]

For *directed* planar graphs:

	$O(n^{3/2} \log n)$	Johnson and Venkatesan [1982]
	$O(n^{4/3} \log^2 n \log C)$	Klein, Rao, Rauch, and Subramanian [1994], Henzinger, Klein, Rao, and Subramanian [1997]
$*$	$O(n \log n)$	Weihe [1994b,1997b]

Minimum cost flows:

$D = (V, A)$ directed graph
$k : A \to R$ cost function

for *any* f,

$$\text{cost}(f) = \sum_{a \in A} k(a) f(a)$$

$s, t \in V$

minimum cost flow

given these, a capacity fn c, demand d, value ϕ, find:

$d \leq$ s-t flow $\leq c$, value $= \phi$, with minimum cost

includes:

Find max-value flow $\leq c$ that has min cost.

We will work with <u>circulations</u>:

<u>Circulation</u>:

directed graph: function f: edges $\to R$ such that

$$\underset{f}{\text{excess}}(v) = \sum_{e \in \text{incoming at } v} f(e) + \sum_{e \in \text{outgoing at } v} f(e) = 0$$

for all verts v

<u>Useful fact</u>:

<u>Theorem</u>: $D = (V, A)$ a directed graph, $f: A \to R_+$

then f is a nonnegative linear comb. of at most $|A|$ vectors X^P, where P is a directed path or circuit. If P is a <u>path</u>, it starts at a vertex v with $\text{excess}_f(v) < 0$, ends at vert with $\text{excess}_f > 0$.

Proof:

1) we add a vertex u, edges to make $excess_f = 0$ everywhere.

a) add u

b) for each v such that $excess_f(v) > 0$, add $\vec{v,u}$ with flow $= excess_f(v)$.

c) for each v s.t. $excess_f(v) < 0$, add $\vec{u,v}$ with flow $= -excess_f(v)$

d) we now have $excess_f(w) = 0 \ \forall w$ [Why?]

2) Now prove for $excess_f(w) = 0$.

$A' = \{a \mid f(a) > 0\}$ induction

$A' \neq \emptyset$ (otherwise easy)

so A' contains a directed circuit, C.

[why?]

let τ be $\min_{a \in C} f(a)$.

- we have one x, which is x^C
- set $f' = f - \tau$ and go again

Notice:
- we have removed at least 1 edge, so there can be no more than $|A|$ x^P's
- This x^C appears w/ non negative coeffs.
- The u construction yields the constraints on paths

$\square$

This theorem reads, for circulations:

Each non-negative circulation is a non-negative linear combination of incidence vectors of directed circuits

Important

Back to minimum cost. We work with circulations.

- Formalize our approach to flow-augmenting paths

For a digraph, $D = (V, A)$ with $f : A \to \mathbb{R}$, capacity function c, demand function d

Define a <u>residual graph</u>:

$$D_f = (V, A_f)$$

$$A_f = \{a \mid a \in A,\ f(a) < c(a)\}$$

edges which can take more forward flow

$$\cup$$

$$\{a^{-1} \mid a \in A,\ f(a) > d(a)\}$$

edges where we can reduce flow i.e. edge can take more backward flow.

reverse the direction of a.

Notation:

- extend cost $k(a)$ to A^{-1} by
  $k(a^{-1}) = -k(a)$.

- Directed circuit C in D_f gives an undirected circuit in D

- $\chi^C(a) = \begin{cases} 1 & \text{if } C \text{ traverses } a \\ -1 & \text{if } C \text{ traverses } a^{-1} \\ 0 & \text{otherwise.} \end{cases}$

- f is feasible if $d \leq f \leq c$.

Theorem: $D = (V,A)$ digraph; $d, c, k : A \to \mathbb{R}$

$f : A \to \mathbb{R}$ is a feasible circulation. ~~Th~~ Then:

f has min cost among all feasible circs $\iff$ Each directed circuit of D_f has non-neg cost.

Proof:

$\Rightarrow$ Let C be directed circuit in D_f of negative cost. Then there is some ε such that $f + \varepsilon\chi^C$ is feasible.

But $f + \varepsilon\chi^C$ has lower cost than f $\square$

$\Leftarrow$ Suppose each directed circuit in D_f has non-negative cost. Let f' be any feasible circ. Then $f' - f$ is a circulation, so

$$f' - f = \sum_{j=1}^{m} \lambda_j \chi_j^{C_j} \qquad \lambda_j > 0$$

for some directed circuits C_j in D_f

hence

$$\text{cost}(f') - \text{cost}(f) = \text{cost}(f' - f)$$

$$= \sum_j \lambda_j \, k(C_j) \geq 0$$

$\square$

Algorithm to improve circulation

- choose a negative cost directed circuit C in D_f and reset

$$f \longrightarrow f + \tau \chi^C$$

where τ is max subject to $0 \le f \le c$.

- If there is no such circuit, we are finished.

Notes:

- arbitrary choice of circuits gives exponential alg
- choose C of minimum mean cost where mean cost is

$$\frac{k(C)}{|C|}$$

gives strongly polynomial time alg.

$:= -c(a)$. Moreover, define a cost function k' on the arcs of D' by $k'(u,u_a) := k(a)$, $k'(v_a,u_a) := 0$, $k'(v_a,v) := 0$ if $k(a) \geq 0$, and $k'(u,u_a) := 0$, $k'(v_a,u_a) := -k(a)$, $k'(v_a,v) := 0$ if $k(a) < 0$. Then a minimum-cost b-transshipment $x \geq \mathbf{0}$ in D' gives a minimum-cost b-transshipment x satisfying $\mathbf{0} \leq x \leq c$ in the original digraph D.

By Theorem 12.7, a minimum-cost b-transshipment $x \geq \mathbf{0}$ in D' can be found by finding $O(n \log n)$ times a shortest path in a residual graph D'_x. While this digraph has $2m+n$ vertices, it can be reduced in $O(m)$ time to finding a shortest path in an auxiliary digraph with $O(n)$ vertices only. Hence again it takes $O(m + n\log n)$ time by using Fibonacci heaps (Corollary 7.7a) and maintaining a potential as in Theorem 12.5. ∎

12.5. Further results and notes

12.5a. Complexity survey for minimum-cost circulation

Complexity survey for minimum-cost circulation (* indicates an asymptotically best bound in the table):

	$O(n^4CK)$	Ford and Fulkerson [1958b] labeling
	$O(m^3C)$	Yakovleva [1959], Minty [1960], Fulkerson [1961] out-of-kilter method
	$O(nm^2C)$	Busacker and Gowen [1960], Iri [1960] successive shortest paths
*	$O(nC \cdot \mathrm{SP}_+(n,m,K))$	Edmonds and Karp [1970], Tomizawa [1971] successive shortest paths with nonnegative lengths using vertex potentials
	$O(nK \cdot \mathrm{MF}(n,m,C))$	Edmonds and Karp [1972]
*	$O(m \log C \cdot \mathrm{SP}_+(n,m,K))$	Edmonds and Karp [1972] capacity-scaling
	$O(nm \log(nC))$	Dinits [1973a] capacity-scaling
	$O(n \log K \cdot \mathrm{MF}(n,m,C))$	Röck [1980] (cf. Bland and Jensen [1992]) cost-scaling
	$O(m^2 \log n \cdot \mathrm{MF}(n,m,C))$	Tardos [1985a]
	$O(m^2 \log n \cdot \mathrm{SP}_+(n,m,K))$	Orlin [1984a], Fujishige [1986]
	$O(n^2 \log n \cdot \mathrm{SP}_+(n,m,K))$	Galil and Tardos [1986,1988]
	$O(n^3 \log(nK))$	Goldberg and Tarjan [1987], Bertsekas and Eckstein [1988]
*	$O(n^{5/3}m^{2/3} \log(nK))$	Goldberg and Tarjan [1987] generalized cost-scaling

≫

continued

	$O(nm \log n \log(nK))$	Goldberg and Tarjan [1987] generalized cost-scaling; Goldberg and Tarjan [1988b,1989] minimum-mean cost cycle-cancelling
*	$O(m \log n \cdot \mathrm{SP}_+(n,m,K))$	Orlin [1988,1993]
*	$O(nm \log(n^2/m) \log(nK))$	Goldberg and Tarjan [1990] generalized cost-scaling
*	$O(nm \log\log C \log(nK))$	Ahuja, Goldberg, Orlin, and Tarjan [1992] double scaling
*	$O(n \log C(m + n\log n))$	*circulations with lower bounds only* Gabow and Tarjan [1989]
*	$O((m^{3/2}C^{1/2} + \gamma \log \gamma) \log(nK))$	Gabow and Tarjan [1989]
*	$O((nm + \gamma \log \gamma) \log(nK))$	Gabow and Tarjan [1989]

Here $K := \|k\|_\infty$, $C := \|c\|_\infty$, and $\gamma := \|c\|_1$, for integer cost function k and integer capacity function c. Moreover, $\mathrm{SP}_+(n,m,K)$ denotes the running time of any algorithm finding a shortest path in a digraph with n vertices, m arcs, and nonnegative integer length function l with $K = \|l\|_\infty$. Similarly, $\mathrm{MF}(n,m,C)$ denotes the running time of any algorithm finding a maximum flow in a digraph with n vertices, m arcs, and nonnegative integer capacity function c with $C = \|c\|_\infty$.

Complexity survey for minimum-cost nonnegative transshipment:

*	$O(n \log B \cdot \mathrm{SP}_+(n,m,K))$	Edmonds and Karp [1970,1972]
	$O(n^2 \log n \cdot \mathrm{SP}_+(n,m,K))$	Galil and Tardos [1986,1988]
*	$O(n \log n \cdot \mathrm{SP}_+(n,m,K))$	Orlin [1988,1993]

Here $B := \|b\|_\infty$ for integer b.

12.5b. Min-max relations for minimum-cost flows and circulations

From Corollary 12.1a, the following min-max equality for minimum-cost circulation can be derived. The equality also follows directly from linear programming duality and total unimodularity. Both approaches were considered by Gallai [1957,1958a, 1958b].

Theorem 12.8. *Let $D = (V,A)$ be a digraph and let $c,d,k : A \to \mathbb{R}$. Then the minimum of $\sum_{a\in A} k(a)f(a)$ taken over all circulations f in D satisfying $d \leq f \leq c$ is equal to the maximum value of*

$$\sum_{a\in A}(y(a)d(a) - z(a)c(a)), \tag{12.51}$$

where $y,z : A \to \mathbb{R}_+$ are such that there exists a function $p : V \to \mathbb{R}$ with the property that

Matchings:

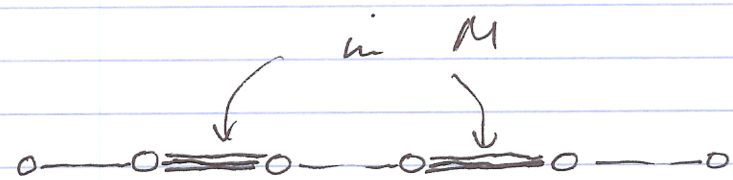
$G = (V, E)$ undirected

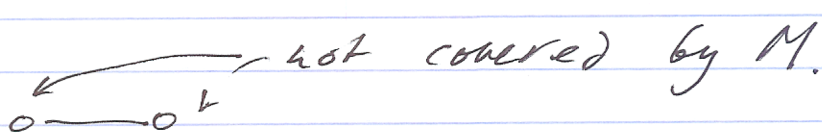
A **matching** is a set of disjoint edges

An **augmenting path** for a matching M is a path with

- odd length
- first, last vert not covered by M
- alternate edges in M

Examples:





Define symmetric difference Δ by

$$A \Delta B = (A - B) \cup (B - A)$$

$$= \{\text{everything in one, but not Both}\}$$

if P is an M-augmenting path

then $M' := M \,\Delta\, \underbrace{EP}_{\text{edges of } P}$

is a matching with $|M'| = |M| + 1$

Thm: $G = (V, E)$ a graph, M a matching. Then either M is a matching of max size, or there exists an M-augmenting path.

Proof:

if M is max size matching, cannot have M-augmenting P because

$M \triangle E P$ would be bigger than M

if M' is a matching, larger than M, consider

$$G' = (V, M' \cup M)$$

This has max degree 2

So each component is a path (perhaps length 0) or a circuit.

$|M'| > |M|$ so one component must have more M' edges than M edges

this is an M augmenting path. $\square$

Maximum size bipartite matching

(we did this w/ flow already)

$G = (V, E)$, bipartite
M matching] input

Output: M', such that $|M'| > |M|$

Alg: - G has color classes U, W.

- D_M is directed.
    - verts are verts of G
    - edges are edges of G

      $e \in M$, e goes $W \to U$

      $e \notin M$, e goes $U \to W$

- U_M is elements in U not covered by M

  W_M " W "

- find a <u>directed</u> path from U_M to W_M in D_M
- this is M augmenting, so gives a matching larger than M.

<u>Weighted</u> bipartite matching. (Hungarian alg)

- ~~obt~~ each edge has a weight $w(e)$
- we have $G = (V, E)$, color classes U, W, $w: E \to \mathbb{Q}$

<u>Method</u>

- Start with $M = \phi$
- construct D_M, directed by
  - orienting each edge in M to go $W \to U$, with length $= w_e$
  - all others go $U \to W$, length $= -w_e$

- Write U_M for U verts not covered by M

  W_M " W "

- find shortest $U_M - W_M$ path (if it exists), say P

- ~~form~~ $M \longrightarrow M \triangle EP$

  and iterate.

- Stop when no P can be found.

<u>Thm</u>: call a matching <u>extreme</u> if it has max weight among all size $|M|$ matchings. Each M found by this method is extreme.

Proof: (Induction)

- True for $M = \emptyset$
- Suppose M is extreme. P and M' from matching alg. Show M' is extreme
- Consider any N, a matching, extreme,
  $|N| = |M| + 1$.
- $|N| > |M|$, so $M \cup N$ has a
  component Q that is M-augmenting
- P is shortest such, so $\ell(P) \leq \ell(Q)$
- $N \triangle Q$ is a matching, $|N \triangle Q| = |M|$
  $\rightarrow$ | Why | $\leftarrow$
- but M is extreme, so $w(M) \geq w(N \triangle Q)$
- $w(N) \overset{\rightarrow|\text{Why}|\leftarrow}{=} w(N \triangle Q) - \ell(Q) \leq w(M) - \ell(P) = w(M')$
- So M' is extreme

$\square$

... suffices to show that each vertex cover C has $|C| \geq |U|$. This indeed is the case, since $N(U \setminus C) \subseteq C \cap W$, and hence

(16.8) $$|C| = |C \cap U| + |C \cap W| \geq |C \cap U| + |N(U \setminus C)| \geq |C \cap U| + |U \setminus C| = |U|.$$

∎

This can be extended to general subsets of V. First, Hoffman and Kuhn [1956b] and Mendelsohn and Dulmage [1958a] showed:

Theorem 16.8. *Let $G = (V, E)$ be a bipartite graph with colour classes U and W and let $R \subseteq V$. Then there exists a matching covering R if and only if there exist a matching M covering $R \cap U$ and a matching N covering $R \cap W$.*

Proof. Necessity being trivial, we show sufficiency. We may assume that G is connected, that $E = M \cup N$, and that neither M nor N covers R. This implies that there is a $u \in R \cap U$ missed by N and a $w \in R \cap W$ missed by M. So G is an even-length $u - w$ path, a contradiction, since $u \in U$ and $w \in W$. ∎

(This theorem goes back to theorems of F. Bernstein (cf. Borel [1898] p. 103), Banach [1924], and Knaster [1927] on injective mappings between two sets.)

Theorem 16.8 implies a characterization of sets that are covered by some matching:

Corollary 16.8a. *Let $G = (V, E)$ be a bipartite graph with colour classes U and W and let $R \subseteq V$. Then there is a matching covering R if and only if $|N(S)| \geq |S|$ for each $S \subseteq R \cap U$ and for each $S \subseteq R \cap W$.*

Proof. Directly from Theorems 16.7 and 16.8. ∎

It also gives the following exchange property:

Corollary 16.8b. *Let $G = (V, E)$ be a bipartite graph, with colour classes U and W, let M and N be maximum-size matchings, let U' be the set of vertices in U covered by M, and let W' be the set of vertices in W covered by N. Then there exists a maximum-size matching covering $U' \cup W'$.*

Proof. Directly from Theorem 16.8: the matching found is maximum-size since $|U'| = |W'| = \nu(G)$. ∎

Notes. These results also are special cases of the exchange results on paths discussed in Section 9.6c. Perfect [1966] gave the following linear-algebraic argument for Corollary 16.8b. Make a $U \times W$ matrix A with $a_{u,w} = x_{u,w}$ if $uw \in E$ and $a_{u,w} := 0$ otherwise, where the $x_{u,w}$ are independent variables. Let U' be any maximum-size subset of U covered by some matching and let W' be any maximum-size subset of W covered by some matching. Then U' gives a maximum-size set of linearly independent rows of A and W' gives a maximum-size set of linearly independent columns of A. Then the $U' \times W'$ submatrix of A is nonsingular, hence of nonzero determinant. It implies (by the definition of determinant) that G has a matching covering $U' \cup W'$.

(Related work includes Perfect and Pym [1966], Pym [1967], Brualdi [1969b,1971b], and Mirsky [1969].)

16.7. Further results and notes

16.7a. Complexity survey for cardinality bipartite matching

Complexity survey for cardinality bipartite matching ($*$ indicates an asymptotically best bound in the table):

	$O(nm)$	Kőnig [1931], Kuhn [1955b]
	$O(\sqrt{n}\, m)$	Hopcroft and Karp [1971,1973], Karzanov [1973a]
$*$	$\tilde{O}(n^{\omega})$	Ibarra and Moran [1981]
	$O(n^{3/2}\sqrt{\frac{m}{\log n}})$	Alt, Blum, Mehlhorn, and Paul [1991]
$*$	$O(\sqrt{n}\, m \log_n(n^2/m))$	Feder and Motwani [1991,1995]

Here ω is any real such that any two $n \times n$ matrices can be multiplied by $O(n^{\omega})$ arithmetic operations (e.g. $\omega = 2.376$).

Goldberg and Kennedy [1997] described a bipartite matching algorithm based on the push-relabel method, of complexity $O(\sqrt{n}\, m \log_n(n^2/m))$. Balinski and Gonzalez [1991] gave an alternative $O(nm)$ bipartite matching algorithm (not using augmenting paths).

16.7b. Finding perfect matchings in regular bipartite graphs

By Kőnig's matching theorem, each k-regular bipartite graph has a perfect matching (if $k \geq 1$). One can use the regularity also to find quickly a perfect matching. This will be used in Chapter 20 on bipartite edge-colouring.

First we show the following result of Cole and Hopcroft [1982] (which will not be used any further in this book):

Theorem 16.9. *A perfect matching in a regular bipartite graph can be found in $O(m \log n)$ time.*

Proof. We first describe an $O(m \log n)$-time algorithm for the following problem:

(16.9) given: a k-regular bipartite graph $G = (V, E)$ with $k \geq 2$,
find: a nonempty proper subset F of E with (V, F) regular.

Proof. See above.

17.5. Further results and notes

17.5a. Complexity survey for maximum-weight bipartite matching

Complexity survey for the maximum-weight bipartite matching (* indicates an asymptotically best bound in the table):

	$O(nW \cdot \mathrm{VC}(n,m))$	Egerváry [1931] (implicitly)
	$O(2^n n^2)$	Easterfield [1946]
	$O(nW \cdot \mathrm{DC}(n,m,W))$	Robinson [1949]
	$O(n^4)$	Kuhn [1955b], Munkres [1957][28] Hungarian method
	$O(n^2 m)$	Iri [1960]
	$O(n^3)$	Dinits and Kronrod [1969]
*	$O(n \cdot \mathrm{SP}_+(n,m,W))$	Edmonds and Karp [1970], Tomizawa [1971]
	$O(n^{3/4} m \log W)$	Gabow [1983b,1985a,1985b]
*	$O(\sqrt{n}\, m \log(nW))$	Gabow and Tarjan [1988b,1989] (cf. Orlin and Ahuja [1992])
	$O(\sqrt{n}\, mW)$	Kao, Lam, Sung, and Ting [1999]
*	$O(\sqrt{n}\, mW \log_n(n^2/m))$	Kao, Lam, Sung, and Ting [2001]

Here $W := \|w\|_\infty$ (assuming w to be integer-valued). Moreover, $\mathrm{SP}_+(n,m,W)$ is the time needed to find a shortest path in a directed graph with n vertices and m arcs, with nonnegative integer lengths on the arcs, each at most W. Similarly, $\mathrm{DC}(n,m,W)$ is the time required to find a negative-length directed circuit in a directed graph with n vertices and m arcs, with integer lengths on the arcs, each at most W in absolute value. Moreover, $\mathrm{VC}(n,m)$ is the time required to find a minimum-size vertex cover in a bipartite graph with n vertices and m edges.

Dinits [1976] gave an algorithm for finding a minimum-weight matching in $K_{p,q}$ of size p, with time bound $O(|p|^3 + pq)$ (taking $p \leq q$).

17.5b. Further notes

Simplex method. Finding a maximum-weight matching in a bipartite graph is a special case of a linear programming problem (see Chapter 18), and hence linear programming methods like the simplex method apply.

[28] Munkres showed that Kuhn's 'Hungarian method' takes $O(n^4)$ time.

signment problem. Using the 'strongly feasible' trees of Cunningham [1976], Roohy-Laleh [1980] showed that a version of the simplex method solves the assignment problem in less than n^3 pivots (cf. Hung [1983], Orlin [1985], and Akgül [1993]; the last paper gives a method with $O(n^2)$ pivots, yielding an $O(n(m + n\log n))$ algorithm).

Balinski [1985] (cf. Goldfarb [1985]) showed that a version of the dual simplex method (the *signature method*) solves the assignment problem in strongly polynomial time ($O(n^2)$ pivots, yielding an $O(n^3)$ algorithm). More can be found in Dantzig [1963], Barr, Glover, and Klingman [1977], Balinski [1986], Ahuja and Orlin [1988,1992], Akgül [1988], Paparrizos [1988], and Akgül and Ekin [1991].

For further algorithmic studies of the assignment problem, consult Flood [1960], Kurtzberg [1962], Hoffman and Markowitz [1963], Balinski and Gomory [1964], Tabourier [1972], Carpaneto and Toth [1980a,1983,1987], Hung and Rom [1980], Karp [1980], Bertsekas [1981,1987,1992] ('auction method'), Engquist [1982], Avis [1983], Avis and Devroye [1985], Derigs [1985b,1988a], Carraresi and Sodini [1986], Derigs and Metz [1986a], Glover, Glover, and Klingman [1986], Jonker and Volgenant [1986], Kleinschmidt, Lee, and Schannath [1987], Avis and Lai [1988], Bertsekas and Eckstein [1988], Motwani [1989,1994], Kalyanasundaram and Pruhs [1991, 1993], Khuller, Mitchell, and Vazirani [1991,1994], Goldberg and Kennedy [1997] (push-relabel), and Arora, Frieze, and Kaplan [1996,2002].

For computational studies, see Silver [1960], Florian and Klein [1970], Barr, Glover, and Klingman [1977] (simplex method), Gavish, Schweitzer, and Shlifer [1977] (simplex method), Bertsekas [1981], Engquist [1982], McGinnis [1983], Lindberg and Ólafsson [1984] (simplex method), Glover, Glover, and Klingman [1986], Jonker and Volgenant [1987], Bertsekas and Eckstein [1988], and Goldberg and Kennedy [1995] (push-relabel). Consult also Johnson and McGeoch [1993].

Linear-time algorithm for weighted bipartite matching problems satisfying a quadrangle or other inequality were given by Karp and Li [1975], Buss and Yianilos [1994,1998], and Aggarwal, Bar-Noy, Khuller, Kravets, and Schieber [1995].

For generating *all* minimum-weight perfect matchings, see Fukuda and Matsui [1992]. For studies of the 'most vital' edges in a weighted bipartite graph, see Hung, Hsu, and Sung [1993].

Aráoz and Edmonds [1985] gave an example showing that iterative dual improvements in the linear programming problem dual to the assignment problem, need not converge for irrational data.

For the 'bottleneck' assignment problem, see Gross [1959] and Garfinkel [1971]. An algebraic approach to assignment problems was described by Burkard, Hahn, and Zimmermann [1977].

For surveys on matching algorithms, see Galil [1983,1986a,1986b]. Books covering the weighted bipartite matching and assignment problems include Ford and Fulkerson [1962], Dantzig [1963], Christofides [1975], Lawler [1976b], Bazaraa and Jarvis [1977], Burkard and Derigs [1980], Papadimitriou and Steiglitz [1982], Gondran and Minoux [1984], Rockafellar [1984], Derigs [1988a], Bazaraa, Jarvis, and Sherali [1990], Cook, Cunningham, Pulleyblank, and Schrijver [1998], Jungnickel [1999], Mehlhorn and Näher [1999], and Korte and Vygen [2000].