Combinatorial optimization

Graph algorithms

Union-find

Creating and updating an equivalence. Operations UNION (union of two classes) and FIND (detect equivalence class). FIND actually outputs a canonical representative.

Tarjan's UF employs:

- Path compression – compress any path that you go through by linking it at a root.
- Union by rank – every representative remembers its $r(v)$ (rank). When doing UNION, declare higher ranking vertex as representative. If ranks equal, increase rank.

T:With n elements, m operations UNION/FIND take $O((n+m)\log^* n)$.

P: Create buckets of vertices based on their rank. In the bucket k there are vertices of rank $((2\uparrow k-1), 2\uparrow k]$.

O:In k-th bucket, there are at most $n/(2\uparrow k)$ vertices **P(**Obs.):There are at most $n/2^r$ vertices of rank r, because of how rank merges trees. Apply this result to the buckets.

Now, do the accounting.

Minimum spanning tree

Minimum is counted based on the weight function. We say a tree is lighter if T is better than another T'.

D:Light edges $\equiv$ we can swap them in a T to get a lighter spanning tree.

O:We can arrive at any minimum spanning tree from any tree using swap operations.

D:Blue edge $\equiv$ lightest edge of some edge cut.

D:Red edge $\equiv$ heaviest edge on a cycle.

Meta-algorithm: try to avoid red edges while gathering blue edges.

ALGJarnik's algorithm with a Fib. heap

- Put all elements in a heap.
- Extract minimum edge uv ($u \in T, v \notin T$), add to T.
- For all neighbours w of v which aren't in T:
- If edge wv is lighter than any edge to w in heap, add it.
- DECREASEKEY the edge that is heavier than wv, if you added it.

Complexity: $O(m + nlogn)$.

Note: oftentimes we can use Courcelle's Theorem to get fast algorithms on hard problems, if our structure is bounded by treewidth or other parameter.

Algebraic and arithmetic algorithms

Strassen

Euclid's Algorithm

RSA

Working with Q^n

Polytope theory

D:*Polyhedron:* $P = \{x \in R^n | Ax \leq b\}$.

D:*Convex set:* $\forall x, y \lambda_1 x + \lambda_2 y \in C$.

D:*Convex hull:*

Basic property of convex sets: if an element is outside, it can be separated by a hyperplane, but an element on the inside cannot.

D:*Halfspace:* $\{x | c^T x \leq \delta\}$.

D:*Polytope:* Convex hull of a finite set of vertices.

D:*Vertex:* A point in a polytope/polyhedron which isn't a convex combination of any two points.

D:*Simplex:* Convex hull of an affinely independent set of vertices.

D:*Geometric duality:* To a point $a \neq 0$ in R^d we assign the hyperplane $h = \{x | ax = 1\}$ and to a hyperplane not passing through the origin we do the same, assign to it the point a.

D:$A_z \equiv$ the set of vectors of A satisfying $a_i z = b_i$.

T:v vertex of $P \Leftrightarrow r(A_v) = n$

P:$\Rightarrow$: A_z not full rank $\rightarrow$ find $c : A_z c = 0$. v is then a conv. combination of $v + \delta c, v - \delta c$ for some $\delta > 0$.

$\Leftarrow$: v not a vertex $\rightarrow$ find two elements which combine to it; $A_v(x - y) = 0$.

T:A bounded polyhedron with t vertices $\Leftrightarrow P = conv(x_1, \ldots, x_t)$.

P:First implication $P = \bigcap \Gamma$. Hyperplanes are of dimension $d-1$, apply induction. Get set of vertices V_i for each hyperplane. $\cup V$ is then the set of vertices of P. If there is a $x \in P$, it is on a line, and this line is bounded by at least two hyperplanes, but the elements on the hyperplanes can be expressed as a convex combination, as so x is a conv. combination also.

Second implication by duality.

O:Cyclic polytopes have n vertices and roughly $\binom{n-d/2}{d/2}$ facets. Cyclic polytopes also maximize the number of faces in each dimension for n.

D:*Projection* of x onto a a closed, convex, nonempty set K is a point $p \in K$ that minimizes distance $||p - x||$.

T(Projection theorem):K closed, convex, nonempty in R^n, $p(x)$ projection of x. Then $\forall z \in K$, $(z-p)^T(b-p) \leq 0$.

LP

The standard LP setting (everything without indices are vectors):

$$\begin{aligned} &\min c^T x, \\ &Ax = b, \\ &x \geq 0, \end{aligned}$$

We can always move to inequalities, change min to max, as long as we stay linear.

T(Farkas Lemma):Exactly one is true:

- $\exists x \geq 0 : Ax = b$
- $\exists y : A^T y \geq 0, b^T y < 0$.

P:

Not two at the same time:

$$Ax = b \rightarrow 0 \leq x^T A^T y = y^T A x = y^T b < 0.$$

$(\neg 1 \rightarrow 2)$: Assume no $Ax = b$. Look at cone $K = \{Ax | x \geq 0\}$. $b \notin K$. Project b onto K, get p. Use Projection Theorem: $\forall z \in K, (z-p)^T(b-p) \leq 0$. Define $y \equiv p - b$. $\forall x : (Ax-p)^T(y) \geq 0$ (inequality switches due to y). $p = Aw$ in cone, so $\forall x : (Ax - Aw)^T(y) = (x-w)^T(A^T y) \leq 0$. Choose $x = w + (0, 0, 0, \ldots, 1, \ldots, 0)$. This extracts one column of A. $x \geq 0$, because $w \geq 0$. So $(A^T y) \geq 0$.

$y^T b = (p-y)^T y = p^T y - y^T y$. Again, $(Ax - p)^T y \geq 0$, so $p^T y \leq 0$ for $x \equiv 0$. However, $p^T y \neq 0$.

C:Exactly one is true:

- $\exists x \geq 0 : Ax \leq b$,
- $\exists y \geq 0 : A^T y = 0, b^T y < 0$.

Duality

Assume minimazation. We are trying to get a lower bound on the $c^T x$. We can do this if we find a vector y such that $y^T Ax = b^T y$, and we try to make it happen so that $y^T A$ (coefficents of x) are below c^T. Therefore, we have a dual program:

$$\begin{aligned} &\max b^T y \\ &A^T y \leq c \end{aligned}$$

T(Weak duality):Solution of dual $w \leq z \equiv$ solution of primal.

P:Directly from argument above.

T(Strong duality):$w = z$.

P: Suppose primal bounded (and feasible), $x*$ optimum. We now look for y s.t. $A^T y \leq c$ and $b^T y \geq z = c^T x*$. Suppose there is none, then (applying Farkas corollary on modified matrices) there is $x \geq 0, \lambda \geq 0$ such that

$$\begin{aligned} &Ax - \lambda b \geq 0, \\ &c^T x - \lambda z < 0. \end{aligned}$$

If $\lambda \neq 0$, we can normalize it to be $\lambda = 1$ and we have improvement over an optimum. If $\lambda = 0$, we can go to $-\infty$ with cost.

Integrality and ILP

Incidence matrix $\equiv V \times E$.

D:A matrix M is totally unimodular $\equiv$ every square submatrix has determinant -1,0 or 1.

T:Suppose A is totally unimodular, then each vertex of the polyhedron $\{x | Ax \leq b\}$ is integral.

P:Vertex z. Use observation that $A_z = \{col(A) | z_j \neq 0\}$ has full rank. Therefore it is invertible. Since $|detA'| = 1$, all entries of the inverse are integer.

T(Hoffman-Kruskal):A is unimodular $\Leftrightarrow \forall b$ integer vector the polyhedron $Ax \leq b, x \geq 0$ is integer.

T:G bipartite $\Leftrightarrow$ incidence matrix A is totally unimodular.

P:G not bipartite $\rightarrow$ take submatrix of odd cycle, calculate determinant.

G bipartite. Take $t \times t$ matrix M, proceed by induction. if M has a column of zeroes or with just one 1, all done. If each column has two 1s, split the matrix based on the partitions.

Simplex method

We work with a normalized problem, i.e.:

$$\min c_B x_B + c_N x_N$$
$$s.t. A_B x_B + A_N x_N = b, x_B, x_N \geq 0.$$

Idea: in a minimization problem, we can look at the solution we're in as a base x_B (full vertices) plus additional vectors that sum us up to b. We check if one x_N can decrease the cost function. If so, increase contribution of x_N while satisfying equality.

Make sure that $x_B \geq 0$ until it breaks, then we basically added a new vector to the base B.

Note: Many heuristics for pivot choice. Also we need to make sure that removing an element of the basis k which doesn't create a loop in the next step. (The loop may happen if one element is already set to 0 in the basis.) Pivoting rule that works: choose k, j minimal.

Q(Hirsch):For m hyperplanes in d dimensions the length of the shortest path between any two vertices of the arrangement is at most $m - d$.

Not even a proof of the Hirsch conjecture would say much about the Simplex algorithm. Existence of polynomial scheme is still open.

Ellipsoid method

Our problem is a modified problem: we want to find for a given polytope $Ax \leq b$ a solution or answer that it's empty.

D:A symmetric matrix $A \in R^{n+n}$ is positive definite, if $\forall x \neq 0 : x^T Ax > 0$.

T:TFAE:

- All eigenvalues are positive,
- Inversion matrix A^{-1} is also positive defiinite,
- $\exists U : U^T U = A$. U will be denoted as $A^{1/2}$.

D:A set $\{x|(x-a)^T A^{-1}(x-a) \leq 1\}$ is called an *ellipsoid* with center a and defined by the (by the definition positive definite) matrix A.

D:Affine map: $T = Qx + s$.

O:An ellipsoid is an affine map of $E(0, I)$.

P:Translate it by $A^{1/2}$ and a.

T:For affine maps, $vol(T(X)) = |\det Q| vol(X)$.

O:for $vol(E(a, A))$ holds that $|detA^{1/2}|n^{-n} \leq vol(E(a, A)) \leq |detA^{1/2}|2^n$.

O:If we can answer the existence of an element in polynomial time, we can calculate LP in polynomial time.

P:Create a polytope in R^{m+n} which encapsulates both the primal and the dual conditions. If an element exists, it necessarily is an optimum.

Finding initial ellipsoid.

L(Hadamard):$|detC| \leq \prod ||C_i||$.

L:C integral. Then $|detC| \leq 2^{<C>-n^2}$.

T:if $P \equiv \{x|Cx \leq d\} \subseteq Q^n$ is a limited polytope and C, d are integral, then all vertices of P are contained within a ball of origin $(0, 0)$ and radius $R = \sqrt{n}2^{<C,d>+nlogn}$.

Simple iteration step

$$E_k = E(0, I), H_k = \{x|x_1 \leq 0\}.$$

Since the broken condition is in the direction of x_1, we want to create an ellipsoid of the form

$$E' = \{(x - te_1)^T Z^{-1}(x - te_1) \leq 1\}.$$

Z will be diagonal. We want to "shrink" along x_1. Pick $p < 1, d > 1$ and set Z diagonal with the diagonal $(1/p^2, 1/d^2, 1/d^2, \dots)$. We want our ellipsoid to touch points e_1, e_2 and be volume-minimal with this property.

First condition gives: $(1 + t)^2/p^2 = 1$. Second condition gives: $t^2/p^2 + 1/d^2 = 1$.

We know that $vol(E) = \sqrt{|detZ|} vol(B)$. Calculate the volume, you should get $t = -1/n + 1$.

Complex iteration step

Affine transform the ellipsoid to the simple iteration step. You have to do some rotation for the hyperplane, but it will not matter. Harder calculations.

Ending observation

O:If a $P = Ax \leq b$ has full rank, then $vol(P) \geq 2^{-(n+1)<C>+n^3}$.

P:Sketch. (Affinely) transform the polytope into n unit vectors and 0. Vertices are such that a subset of columns where A_x have full rank. Use Cramer rule and calculate total volume.

Special matrices

Totally unimodular matrix.

Positive semidefinite matrix.

TSP

Matching and flow networks

Edmonds

Edmonds' polytope:

$$\sum_{e \in X} \begin{matrix} f \in R^{E(G)} : \sum f(e) \leq 1 \\ f(e) \leq |X| - 1/2 \text{ for odd } X. \end{matrix}$$

Perfect matching polytope has equality in the first condition and the second condition:

$$\sum_{e \in X} f(e) \geq 1.$$

D:r-graph: Graph that is r-regular and all edge cuts of odd size are at of size at least r.

Example of an r-graph: cubic bridgeless graphs.

T:Every r-graph has a uniform cover by perfect matchings.

P:Every element of the perfect matching polytope is a convex combination of perfect matchings. set $f(e) = 1/r$ everywhere, then this f is in the polytope of PMP.

R:Every cubic bridgeless graph has a uniform cover by perfect matchings (and also a perfect matching).

R:Every cubic bridgeless graph has a perfect matching such that it contains no odd cut of size 3.

R:Every cubic bridgeless graph has perfect matchings M_1, M_2 such that $E(M_1) \cup E(M_2) \geq 3E/5$.

P:Take one PM M that contains no odd cut of size 3. Define $f(e) = 1/5$ on it and $f(e) = 2/5$ elsewhere. Then $f(e)$ is in PMP. Since $f(E \setminus M) = 2/5|E \setminus M|$, there is one perfect matching M_i that attains this. Take $M \cup M_i$ and get $1/3 + 2/32/5 = 3/5E$.

Edmonds' algorithm:

Find augmenting paths using an augmenting path BFS tree (Edmonds' tree). If you find an augmenting path, good. If not, you may find an odd cycle that started as an augmenting path. Contact this cycle, find augmenting path in the next graph. Proceed until possible.

Total complexity: $O(n^2(n + m))$.

Dinitz

ALGDinitz: Add augmenting flows, not just augmenting paths. Specifically:

- Create reserve network. Now we work only with that.
- Find shortest $s - t$ path by DFS. Cut all longer $s - t$ paths.
- Find blocking flow on the cleaned network. Whenever adding a $s - t$ path greedily, clean up the network.
- Add blocking flow. Iterate.

Cleanup takes only $O(m)$ time per iteration, searching for blocking flow takes $O(nm)$.

O:Number of iterations is $O(n)$.

P:The only new edges that get added in the next reserve network are backwards edges, which only increase the shortest $s - t$ length. Plus (after cleanup) longer paths may become relevant, but still, shortest $s - t$ path increases before every reserve network construction, and so we have $O(n)$.

Integer capacities **O:**On integer capacities, we can bound the running time of Dinitz by $O(|f|n + mn)$.

P:Every augmenting path extends the flow by at least 1.

T:On integer capacities, we can make Dinitz run in time $O(mnlogC)$.

P:We do it similarly to radix sort – create flows by writing C in binary, and then finding maximum flow using the first k-bits of each capacity. After that, we multiply the flow by 2 (and add the smallest bit) and start with a close-enough flow.

O:$|f_i| - 2|f_{i-1}| \leq m$.

P:Max flow = min cut, but min cut has increased from $|R|$ to at most $2|R| + m$.

Now we use the previous observation on integer Dinitz and flow size to get the result.

Matroid theory