19 Maximum Flow

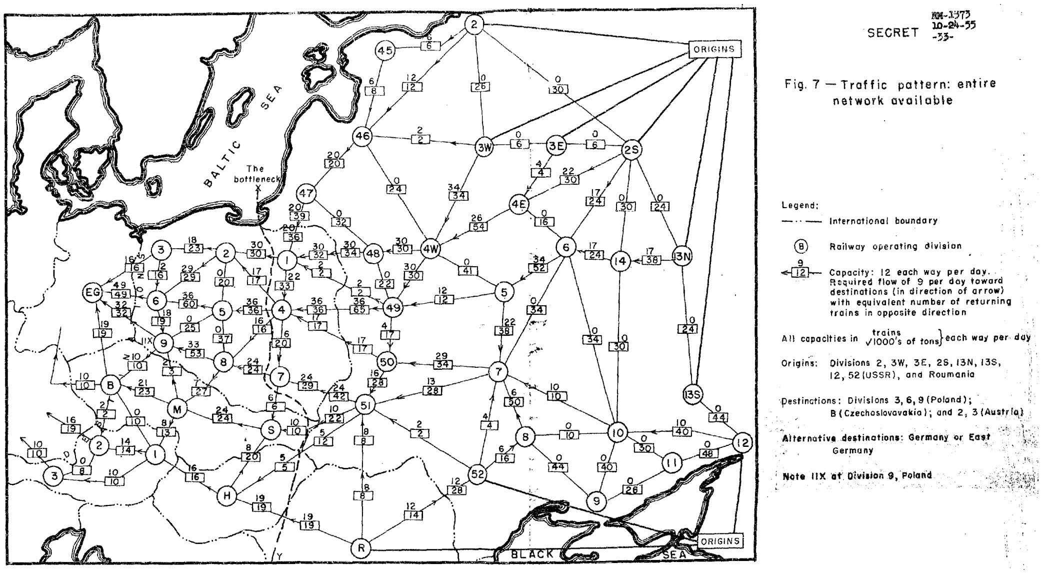
SECRET
RM-1573
10-24-55
-33-
Fig. 7 — Traffic pattern: entire network available
Legend:
International boundary
Railway operating division
Capacity: 12 each way per day. Required flow of 9 per day toward destinations (in direction of arrow) with equivalent number of returning trains in opposite direction
All capacities in trains / √1000's of tons } each way per day
Origins: Divisions 2, 3W, 3E, 2S, 13N, 13S, 12, 52 (USSR), and Roumania
Destinations: Divisions 3, 6, 9 (Poland); B (Czechoslovavakia); and 2, 3 (Austria)
Alternative destinations: Germany or East Germany
Note IIX at Division 9, Poland
ORIGINS
BALTIC SEA
The bottleneck
BLACK SEA
ORIGINS

Figure 1: Harris and Ross's map of the Warsaw-Pact railway network.

19.1 Background

Here I'll provide a brief overview of standard definitions related to the maximum flow problem. For a more thorough and gentler introduction, see chapters 10 and 11 in my algorithms textbook.

Pseudoflows, flows, and circulations

Recall that a *pseudoflow* (or *1-chain*, or *discrete 1-form*) in a graph G is any function $\phi \colon D(G) \to \mathbb{R}$ on the darts of G that is antisymmetric, meaning $\phi(d) = -\phi(\mathsf{rev}(d))$ for every dart d. Intuitively, the value $\phi(u{\to}v)$ represents the rate of flow of some divisible substance like water, train cars, or network packets from u to v along the undirected edge uv. In particular, a negative value indicates that the substance is flowing backward from v to u.[1]

The *boundary* $\partial\phi \colon V(G) \to \mathbb{R}$ of a pseudoflow ϕ that intuitively describes the total net flow into each vertex v:

$$\partial\phi(v) := \sum_u \phi(u{\to}v).$$

A *circulation* is a pseudoflow ϕ whose boundary $\partial\phi$ is identically zero. Intuitively, circulations are pseudoflows that respect conservation of mass; any positive flow into v must be balanced by negative flow into v (that is, positive flow out of v).

For two fixed vertices s and t, an (s,t)-*flow* is a pseudoflow f such that $\partial f(v) = 0$ for all v except possibly s and t. The *value* of an (s,t)-flow ϕ is the total net flow into t or out of s:

$$|\phi| := \partial\phi(t) = -\partial\phi(s).$$

[1] A more common textbook definition of (pseudo)flow is any function $\phi \colon D(G) \to \mathbb{R}$ such that for every dart d, we have $\phi(d) \geq 0$ and either $\phi(d) = 0$ or $\phi(\mathsf{rev}(d)) = 0$. That is, for each edge, instead of choosing an arbitrary values for the darts that sum to 0, we choose both a direction and a non-negative value for the edge. Converting between the antisymmetric formulation and the non-negative formulation is straightforward.

Intuitively, (s,t)-flows model some substance being injected into a network of pipes at s and being extracted at t, with conservation at every other vertex. Every circulation is an (s,t)-flow with value 0.

Lemma: *For any two (s,t)-flows ϕ and ψ in the same graph G and any two real numbers α and β, the function $\alpha\phi + \beta\psi$ is an (s,t)-flow in G with value $\alpha|\phi| + \beta|\psi|$. In particular, if ϕ and ψ are circulations, then $\alpha\phi + \beta\psi$ is also a circulation. Thus, circulations and (s,t)-flows in any graph G define vector spaces.*

We can regard any directed cycle γ as a circulation:

$$\gamma(d) = \begin{cases} 1 & \text{if } d \in \gamma \\ -1 & \text{if } \operatorname{rev}(d) \in \gamma \\ 0 & \text{otherwise} \end{cases}$$

Similarly, we can regard any directed path π from s to t as an (s,t)-flow:

$$\pi(d) = \begin{cases} 1 & \text{if } d \in \pi \\ -1 & \text{if } \operatorname{rev}(d) \in \pi \\ 0 & \text{otherwise} \end{cases}$$

Lemma: *Every circulation is a weighted sum of simple directed cycles. Every (s,t)-flow is a weighted sum of simple directed (s,t)-paths and simple directed cycles.*

Capacities and residual graphs

A *capacity* function for a graph G is any function $c\colon D(G) \to \mathbb{R}$ from the darts to the reals. Capacities are not necessary either symmetric, antisymmetric, or non-negative. A *flow network* is a graph G together with a capacity function c.

A pseudoflow (or circulation or (s,t)-flow) ϕ is *feasible* with respect to c if and only if $\phi(d) \le c(d)$ for every dart d. In particular, we allow the capacity of a dart to be negative; a negative dart capacity is equivalent to a positive *lower bound* on the amount of flow through the reversal of the dart. The zero flow $\phi \equiv 0$ is feasible if and only if every dart has non-negative capacity.

Given a graph G, capacity function c, and two vertices s and t as input, the maximum flow problem asks for a feasible (s,t)-flow in G with the largest possible value.

Fix a graph G and a capacity function c. Any pseudoflow ϕ in G induces a *residual capacity* function $c_\phi : D(G) \to \mathbb{R}$, defined simply as $c_\phi(d) = c(d) - \phi(d)$. A pseudoflow ϕ is feasible if and only if every dart has non-negative residual capacity. The *residual graph* G_ϕ is just the original graph G but with the new capacity function c_ϕ. A *residual path* (respectively, *residual cycle*) is a directed path (respectively, directed cycle) in G_ϕ in which every dart has *positive* residual capacity.

The standard textbook algorithm for maximum flows, proposed by Lester Ford and Delbert Fulkerson in 1953, is the *augmenting path* method. The method starts by finding an initial feasible (s,t)-flow ϕ; typically all capacities are non-negative, so we can start with the zero flow $\phi \equiv 0$. Then we repeatedly *augment* the flow ϕ by *pushing* more flow along paths from s to t. Specifically, at each iteration, we find a residual path π and augment the flow by setting

$\phi' \leftarrow \phi + \min_{d\in\pi} c_\phi(d) \cdot \pi$; here I'm treating π both as a sequence of darts and as an (s,t)-flow. Straightforward definition-chasing implies that if the original flow ϕ is feasible, then the augmented flow ϕ' is also feasible. When G_ϕ contains no more residual paths, ϕ is a maximum (s,t)-flow. More generally:

Lemma: *Let ϕ and ϕ' be any (not necessarily feasible) (s,t)-flows in G.*

(a) *ϕ' is a feasible (s,t)-flow in G if and only if $\phi' - \phi$ is a feasible (s,t)-flow in the residual graph G_ϕ.*
(b) *In particular, if $|\phi| = |\phi'|$, then ϕ' is a feasible (s,t)-flow in G if and only if $\phi' - \phi$ is a feasible **circulation** in G_ϕ.*
(c) *In particular, ϕ' is a **maximum** (s,t)-flow in G if and only if $\phi' - \phi$ is a **maximum** (s,t)-flow in G_ϕ.*

19.2 Planar Circulations

Flows and circulations have particularly nice structure in planar graphs, or more accurately, in planar *maps*.

Fix an arbitrary circulation ϕ in an arbitrary planar map Σ, with a distinguished outer face o. The *winding number* of ϕ around each face f of Σ, denoted $\mathsf{wind}(\phi, f)$ can be defined by extending the definition of the Alexander numbering of a curve:

- $\mathsf{wind}(\phi, o) = 0$
- For every dart d, we have $\mathsf{wind}(\phi, \mathsf{left}(d)) = \phi(d) + \mathsf{wind}(\phi, \mathsf{right}(d))$.

Conservation at each vertex v implies that this st of constraints has a unique solution. Equivalently, for any path (in fact, any *walk*) π in the dual map Σ^* from dual of the outer face o^* to the dual vertex f^*, we have

$$\mathsf{wind}(\phi, f) = \sum_{d^* \in \pi} \phi(d).$$

The second definition is independent of the choice of dual path π, again by conservation. A third equivalent definition uses the fact that ϕ is a weighted sum of simple cycles:

$$\phi = \sum_i \alpha_i \cdot \gamma_i \implies \mathsf{wind}(\phi, f) = \sum_i \alpha_i \cdot \mathsf{wind}(\gamma_i, f);$$

This definition is independent of the chosen decomposition of ϕ into cycles $\gamma_1, \gamma_2, \ldots$.

Alexander numbering is an example of a *face potential* (or 2-chain); more generally, a face potential in Σ is any function $\alpha \colon F(\Sigma) \to \mathbb{R}$ assigning a real number to each face of Σ. The *boundary* of a face potential α is the circulation $\partial\alpha \colon D(\Sigma) \to \mathbb{R}$ defined by setting

$$\partial\alpha(d) = \alpha(\mathsf{left}(d)) - \alpha(\mathsf{right}(d))$$

for every dart d. It should be easy to verify that $\partial\alpha$ is indeed a circulation. Moreover, the boundary operator ∂ is linear; for all face potentials α and β and real numbers a and b, we have $\partial(a \cdot \alpha + b \cdot \beta) = a \cdot \partial\alpha + b \cdot \partial\beta$.

The following lemma is a natural generalization (and consequence) of the Jordan Curve Theorem.

Lemma: *Every circulation in a planar map is a boundary circulation.*

Proof: For any circulation ϕ, routine definition-chasing implies $\phi = \partial(\mathsf{wind}(\phi))$. That is, $\phi = \partial\alpha$, where $\alpha(f) = \mathsf{wind}(\alpha, f)$ for every face f. □

Corollary: *The difference between any two (s,t)-flows with the same value in the same planar map Σ is a boundary circulation in Σ.*

19.3 Feasible Planar Circulations and Shortest Paths

Now suppose we endow our planar map Σ with a capacity function $c\colon D(\Sigma) \to \mathbb{R}$. Every dart d^* in the dual map Σ^* has a *cost* or *length* $c(d^*)$ equal to the capacity of the corresponding primal dart d; in short, we have $c(d^*) = c(d)$.

Lemma [Venkatesan]: *Let Σ be any planar map, and let $c\colon D(\Sigma) \to \mathbb{R}$ be any capacity function for Σ. There is a feasible circulation in Σ if and only if the dual map Σ^* has no negative cycles.*

Proof: First, consider an arbitrary circulation ϕ in Σ and an arbitrary cycle λ^* in the dual map Σ^* with negative total cost. Without loss of generality, assume λ^* is simple and oriented counterclockwise. Whitney's duality theorem implies that the set λ of primal darts whose duals lie in λ^* define a *directed edge cut*. Specifically, let A denote the vertices of Σ whose corresponding dual faces lie inside λ^*. Then λ is the set of all darts in Σ such that $\mathsf{head}(d) \in A$ and $\mathsf{tail}(d) \notin A$. Straightforward calculation implies

$$\begin{aligned}
\sum_{d\in\lambda} \phi(d) &= \sum_{\mathsf{head}(d)\in A} \phi(d) && \text{because } \phi(d) = -\phi(\mathsf{rev}(d))\\
&= \sum_{v\in A} \sum_{\mathsf{head}(d)=v} \phi(d) \\
&= \sum_{v\in A} \partial\phi(v) && \text{by definition of } \partial\\
&= \sum_{v\in A} 0 && \text{because } \phi \text{ is a circulation}\\
&= 0\\
&> \sum_{d\in\lambda} c(d).
\end{aligned}$$

(In the first step, we are adding $\phi(d)$ for all darts with both endpoints in A.) We conclude that $\phi(d) > c(d)$ for at least one dart $d \in \lambda$; in short, ϕ is not feasible.

On the other hand, suppose shortest-path distances are well-defined in Σ^*. For any dual vertex p, let $\mathsf{dist}(p)$ denote the shortest-path distance from the outer face o to p. We can interpret the function dist as a face potential function for Σ. I claim that the boundary circulation $\partial\,\mathsf{dist}$ is feasible. For any dart d, we have

$$\phi(d) \;=\; \mathsf{dist}(\mathsf{left}(d)^*) - \mathsf{dist}(\mathsf{right}(d)^*)$$

Now define the *slack* of every dart d as

$$\begin{aligned}
\mathsf{slack}(d) &:= c(d) - \phi(d)\\
&= c(d) - \mathsf{dist}(\mathsf{left}(d)^*) + \mathsf{dist}(\mathsf{right}(d)^*)\\
&= \mathsf{dist}(\mathsf{tail}(d^*)) + c(d^*) - \mathsf{dist}(\mathsf{head}(d^*))
\end{aligned}$$

The definition of shortest paths implies that $\mathsf{slack}(d) \geq 0$ for every dart d, and thus $\phi(d) \leq c(d)$ for every dart d. We conclude that ϕ is feasible.

Corollary: *Given a planar map Σ with n vertices and arbitrary dart capacities, we can compute either a feasible circulation in Σ or a negative-cost cycle in Σ^* in $O(n \log^2 n)$ time.*

Proof: Run the shortest-path algorithm of Klein, Mozes, and Weimann, starting at the vertex o^* dual to the outer face o. If shortest-path distances in Σ^* are well-defined, set $\phi(d) = \mathsf{dist}(\mathsf{left}(d)^*) - \mathsf{dist}(\mathsf{right}(d)^*)$ for every dart d. Otherwise, the algorithm finds a negative cycle in Σ^*. In both cases, the algorithm runs in $O(n \log^2 n)$ time.

19.4 Our First Planar Max-flow Algorithm

The previous lemma can also be used to find feasible (s,t)-flows with particular values. Fix two vertices s and t in G.

Corollary: *Let ϕ be an arbitrary (not necessarily feasible) (s,t)-flow in Σ. There is a feasible (s,t)-flow in G with value $|\phi|$ if and only if the dual residual map Σ^*_ϕ has no negative cycles.*

Corollary: *Given a planar map Σ with n vertices, arbitrary dart capacities, and a real number λ, we can either compute a feasible (s,t)-flow in Σ with value λ, or correctly report that no such flow exists, in $O(n \log^2 n)$ time.*

Proof: Let π be any path from s to t in Σ with value λ, and let ϕ be the flow $\lambda \cdot \pi$. Then ϕ' is a *feasible* (s,t)-flow with value λ if and only if $\phi' - \phi$ is a feasible circulation in the residual map Σ_ϕ. □

Corollary: *Given a planar map Σ with n vertices, non-negative* ***integer*** *dart capacities $c(d)$, we can compute a maximum (s,t)-flow in Σ in $O(n \log^2 n \log(nU))$ time, where $U = \max_d c(d)$.*

Proof: Suppose every dart in Σ has an integer capacity between 0 and U. Because all capacities are non-negative, we know that the zero circulation is a feasible flow with value 0, and the upper bound on individual capacities implies that every feasible flow has value at most nU. If there is a feasible flow with any value λ, we can scale it down to a feasible flow with any value smaller than λ. Finally, Ford and Fulkerson's augmenting-path algorithm implies by induction that the maximum flow in a network with integer capacities has integer value. Thus, we can compute a maximum flow in Σ by performing a binary search over the nU possible flow values, running the $O(n \log^2 n)$-time decision algorithm at each iteration. □

I find this algorithm deeply unsatisfying, in part because it requires integer capacities, but it does at least serve as a proof of concept. Hassin and Johnson proved that for *undirected* planar graphs, where every dart has the same capacity as its reversal, we can compute a maximum (s,t)-flow by first running Reif's minimum-cut algorithm and then running Dijkstra's algorithm in a modified dual graph. Using Reif's original algorithm, this approach funds maximum flows in $(n \log^2 n)$ time; this running time can be improved to $O(n \log n)$ using either the linear-time shortest-path algorithm of Henzinger et al inside Reif's algorithm, or by replacing Reif's algorithm with multiple-source shortest paths.

Unfortunately, this approach does not extend to directed planar graphs, because we do not have a similar divide-and-conquer minimum-cut algorithm in that setting. In 1997, Karsten Weihe described an algorithm to compute maximum flows in directed planar graphs in $O(n \log n)$ time, generalizing his earlier $O(n)$-time algorithm for undirected unit-capacity planar graphs. However, his algorithm assumes that every dart in the input graph appears in at leas one simple path from s to t. Darts that do not satisfy this criterion can be safely removed from the input graph, but an efficient algorithm to find all such "useless" darts was only found in 2017, by Jittat Fakcharoenphol, Bundit Laekhanukit, and Pattara Sukprasert.

Meanwhile, in 2006, Glencora Borradaile and Philip Klein discovered a much cleaner algorithm to compute planar maximum flows in $O(n \log n)$ time. In the rest of this lecture note I will describe a reformulation of their algorithm that I published in 2010.

19.5 Parametric Shortest Paths

We formulate the planar maximum-flow problem as a *parametric shortest-path* problem, similar to our first multiple-source shortest-path problem. Fix an arbitrary path π from s to t. We are trying to find the largest value λ such that Σ supports and (s,t)-flow with value λ. Equivalently, by the arguments in the last two sections, we are looking for the largest value λ such that the dual residual map $\Sigma^*_{\lambda\cdot\pi}$ does not contain a negative cycle. The algorithm maintains a shortest-path tree in the dual residual map $\Sigma^*_{\lambda\cdot\pi}$ as the parameter λ continuously increases from 0. At critical values of λ, darts $p{\to}q$ in Σ^* become tense and pivot into the shortest-path tree, replacing earlier darts $p'{\to}q$. The algorithm halts when a pivot introduces a cycle into the shortest-path tree, which would become negative if we increased λ any further. (That cycle is dual to the minimum cut!)

Again, we fix an arbitrary path π from s to t; we treat this path as a flow with value 1:

$$\pi(d) = \begin{cases} 1 & \text{if } d \in \pi \\ -1 & \text{if } \mathsf{rev}(d) \in \pi \\ 0 & \text{otherwise} \end{cases}$$

We also fix a vertex o in the dual map Σ^*. Let's establish some notation.

- Σ_λ is just shorthand for the residual graph $\Sigma_{\lambda\cdot\pi}$.
- $c(\lambda,d) = c(\lambda,d^*) = c(d) - \lambda\cdot\pi(d)$ is the capacity of d in the residual graph Σ, and therefore the cost of d^* in the dual residual map Σ^*_λ.
- T_λ is the single-source shortest-path rooted at o in Σ^*_λ.
- $\mathsf{dist}(\lambda,p)$ is the shortest-path distance from o to p in Σ^*_λ.
- $\mathsf{path}(\lambda,p)$ is the shortest path from o to p in Σ^*_λ.
- $\mathsf{pred}(\lambda,p)$ is the second-to-last vertex of $\mathsf{path}(\lambda,p)$.
- $\mathsf{slack}(\lambda,p{\to}q) = \mathsf{dist}(\lambda,p) + c(\lambda,p{\to}q) - \mathsf{dist}(\lambda,p)$
- $\mathsf{cycle}(\lambda,p{\to}q)$ is the closed walk obtained by concatenating $\mathsf{path}(\lambda,p)$, $p{\to}q$, and $\mathsf{rev}(\mathsf{path}(\lambda,q))$.
- A dart d of Σ_λ is *tense* if $\mathsf{slack}(\lambda,d^*) = 0$.
- An edge e of Σ_λ is *loose* if neither of its darts is tense.
- L_λ is the subgraph of all loose edges in Σ_λ.
- A dart d in Σ_λ is *active* if $\mathsf{slack}(\lambda,d^*)$ is decreasing at λ.
- LP_λ is the set of all active darts in Σ_λ.

Except at critical values of λ, subgraph L_λ is a spanning tree of Σ_λ, and in fact (L_λ, T_λ) is a tree-cotree decomposition of Σ.

Lemma: *LP_λ is the unique path from s to t in L_λ.*

Lemma: *LP_λ is the set of all active darts in Σ_λ.*

19.6 Active Darts

19.7 Fast Pivots

19.8 Universal Cover Analysis

19.9 References

1. Therese C. Biedl, Bronă Brejová, and Tomáš Vinař. Simplifying flow networks. *Proc. 25th Symp. Math. Found. Comput. Sci.*, 192–201, 2000. Lecture Notes Comput. Sci. 1893, Springer-Verlag.

2. Glencora Borradaile and Anna Harutyunyan. Maximum st-flow in directed planar graphs via shortest paths. *Proc. 24th Int. Workshop Combin. Algorithms*, 423–427, 2013. Lecture Notes Comput. Sci. 8288, Springer. arXiv:1305.5823.

3. Glencora Borradaile and Philip Klein. An $O(n \log n)$ algorithm for maximum st-flow in a directed planar graph. *J. ACM* 56(2):1–9:1–30, 2009.

4. David Eppstein and Kevin A. Wortman. Optimal embedding into star metrics. *Proc. 11th Algorithms Data Struct. Symp. (WADS)*, 290–301, 2009. Lecture Notes Comput. Sci. 5664, Springer. Another application of parametric shortest paths.

5. Jeff Erickson. Parametric shortest paths and maximum flows in planar graphs. *Proc. 21st Ann. ACM-SIAM Symp. Discrete Algorithms*, 794–804, 2010.

6. Jittat Fakcharoenphol, Bundit Laekhanukit, and Pattara Sukprasert. Finding all useless arcs in directed planar graphs. Preprint, May 2018. arXiv:1702.04786.

7. Lester R. Ford and Delbert R. Fulkerson. Maximal flow through a network. *Canad. J. Math.* 8(399–404), 1956. First published as Research Memorandum RM-1400, The RAND Corporation, Santa Monica, California, November 19, 1954.

8. Lester R. Ford and Delbert R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962. First published as Research Memorandum R-375-PR, The RAND Corporation, Santa Monica, California, August 1962.

9. Theodore E. Harris and Frank S. Ross. Fundamentals of a method for evaluating rail net capacities. Research Memorandum RM-1573, The RAND Corporation, Santa Monica, California, October 24, 1955. Declassified May 13, 1999.

10. Refael Hassin and Donald B. Johnson. An $O(n \log^2 n)$ algorithm for maximum flow in undirected planar networks. *SIAM J. Comput.* 14(3):612–624, 1985.

11. Samir Khuller, Joseph (Seffi) Naor, and Philip Klein. The lattice structure of flow in planar graphs. *SIAM J. Discrete Math.* 477–490, 1993. Removing clockwise residual cycles.

12. Karl Menger. Zur allgemeinen Kurventheorie. *Fund. Math.* 10:96–115, 1927.

13. Shankar M. Venkatesan. *Algorithms for network flows*. Ph.D. thesis, The Pennsylvania State University, 1983.

14. Karsten Weihe. Edge-disjoint (s,t)-paths in undirected planar graphs in linear time. *J. Algorithms* 23(1):121–138, 1997.

15. Karsten Weihe. Maximum (s,t)-flows in planar networks in $O(|V|\log|V|)$ time. *J. Comput. Syst. Sci.* 55(3):454–476, 1997.

19.10 Aptly Not