6.854 Advanced Algorithms **Petar Maymounkov**

Problem Set 7 (October 28, 2005)

With: Benjamin Rossman, Oren Weimann, and Pouya Kheradpour

---

**Problem 1a.**

$$\{Ax \leqq b, Dx \leqq e\} = \oslash \Leftrightarrow \max \left\{ 0x \middle| \begin{pmatrix} A \\ D \end{pmatrix} x \leqq \begin{pmatrix} b \\ e \end{pmatrix} \right\} \text{ infeasible}$$

$$\Leftrightarrow \max \left\{ x0 | x(A^T \ D^T) \leqq (b^T \ e^T) \right\} \text{ infeasible}$$

$$\Leftrightarrow \min \left\{ (b^T \ e^T)y | (A^T \ D^T)y = 0, y \geqq 0 \right\} \text{ unbounded}$$

$$\Leftrightarrow \exists y = (y_A \ y_D), \text{ such that } y_A A + y_D D = 0 \text{ and } y_A b^T + y_D e^T < 0$$

**Problem 1b.** Set $c = y_A A = -y_D D$, and let $x \in P$ and $w \in Q$:

$$cx - cw = (y_A A)x + (y_D D)w$$

$$= y_A(Ax) + y_D(Dw)$$

Note: $y_A(Ax) \leqq y_A b$, since $y_A \geqq 0$ and $Ax \leqq b$

Note: $y_D(Dx) \leqq y_D e$, since $y_D \geqq 0$ and $Dw \leqq e$

$$\leqq y_A b + y_D e$$

$$< 0$$

**Problem 1c.** Following our discussion from part 1b, the quickly veriafiable witness that $P$ and $Q$ are disjoint is the pair $y_A, y_D$. The verification procedure is: check (a) $y_A \geqq 0$, $y_D \geqq 0$, (b) $y_A b + y_D e < 0$, (c) $y_A A + y_D D = 0$. These conditions ensure that our argument in part 1b is correct.

On the other hand, witness for asserting that $Q$ and $P$ intersect is simply exhibiting a point in common, which can be verified against the polyhedron inequalities.

**Problem 2.** Using the rules for taking duals (from lecture), we get:

$$\begin{aligned} \text{objective: } & \min \sum_e u_e x_e \\ \text{subject to: } & \sum_{e \in p} x_e \geqq 1 \qquad \text{for all paths } p \\ & x_e \geqq 0 \qquad \text{for all edges } e \end{aligned}$$

We are asked to assign weight on each edge, such that every path gets total weight of at least 1, and we also need to minimize sum of edge weights multiplied by edge capacities. I.e. we want to:

1. Put weights on as few edges as possible, i.e. try to put weights on edges that are common to more paths
2. Put less weight on edges with high capacity

The above two conditions suggest a natural solution. A minimum cut defines a set of edges that meet both of the above properties: (1) all $s-t$ paths cross them, and (2) they have minimal aggregate capacity. So, select a minimum cut and set $x_e = 1$ for edges that go forward through the cut, and set $x_e = 0$ for all other edges in the graph.

It is easily verified that this is a feasible solution to the dual LP. It is also optimal (i.e. minimal) because we know that the optimal of the dual LP has to be equal to the optimal of the primal, and we know that the optimal of the primal is exactly equal to the sum of edge capacities through a minimum cut.

**Problem 3a.** We are given a primal:

$$\begin{aligned} & z = \min\{cx | Ax = b, x \geqq 0\}, \text{ with dual:} \\ & w = \max\{yb | yA \leqq c\} \end{aligned}$$

Using approach from lecture, reduce the optimization of the primal LP to LP feasibility, by forming the "gap" problem:

$$\begin{aligned} \text{objective: } & \min 0x \\ \text{subject to: } & Ax = b \\ & x \geqq 0 \\ & yA \leqq c \\ & cx - by = 0 \end{aligned}$$

From lecture we know that:

1. The original LP is feasible if and only if the gap LP is feasible
2. The gap LP is infeaasible if and only if the original LP is either unbounded or infeasible

Therefore, finding a feasible solution to the gap problem gives us a feasible solution to the original problem. On the other hand, finding that there is no feasible solution to the gap problem tells us that the original problem infeasible or unbounded. We can further disambiguate between these two possibilities by solving another feasibility-only problem that contains the original problem's constraints and the trivial $0x$ objective.

Finally, when we convert the gap problem in standard form, we get a problem of the form:

$$\min\{0x|Ax = b, x \geqq 0\}$$

(The $x$ variables are different from the ones above.)

**Problem 3b.**

$$\text{Primal: } \min\{0x|Ax = b, x \geqq 0\}$$

$$\text{Dual: } \max\{yb|yA \leqq 0\}$$

**Problem 3c.** If the primal (of the gap problem) is feasible, then the dual is feasible. But the dual has only onee basic feasible solution, namely $y = 0$, hence it must be the optimal. Hence, we can optimize the dual of the gap problem, simply by checking whether $y = 0$ is feasible.

**Problem 3d.** Implement following algorithm for solving arbitrary LPs:

1. Start with an arbitrary LP
2. Create gap problem of original LP (takes time linear in original LP's size)
3. Take the dual of the gap problem (linear time)
4. Solve the dual of the gap problem in linear time, using trivial algorithm from part 3c
5. Use the black-box algorithm to optimize the gap problem, using the solution to its dual, in time $O((m+n)^k)$
6. Convert the solutions to the gap problems back the solutions of the original LP (linear time)

All steps in the above algorithm are linear, except for the call to the black box algorithm. Also note that the matrix size of the gap and dual problems are only a constant factor bigger than the original LPs matrix size.

**Problem 4a.** LP is:

$$\text{objective: } \min \sum c_{ij} f_{ij} \tag{7.1}$$

$$\text{subject to: } \sum_j f_{ij} - f_{ij} = 0 \quad \forall i \tag{7.2}$$

$$\sum f_{ij} = 1 \tag{7.3}$$

$$f_{ij} \geqq 0 \tag{7.4}$$

Constraints (7.2) and (7.4) tell us that we can interpret $f_{ij}$ as a flow, and since there are no vertices at which there is flow inbalance, $f_{ij}$ is in fact a circulation.

The objective asks that we find the minimum cost flow, subject to the additional constraint (7.3) that the total volume of flow cannot exceed 1. We will use this later.

Because $f$ is a circulation, we can decompose it into cycles of flow. For any such cycle $C$ with $k$ edges, the flow on all edges is equal, call it $h$. The cost of the flow on the cycle is:

$$
\begin{aligned}
h \sum_{e \in C} c_e &= (kh) \left( \frac{1}{k} \sum_{e \in C} c_e \right) \\
&= v_C \mu_C
\end{aligned}
$$

Where the left term, $v_C$, is the volume of flow going through the cycle and it is subject to optimization (as it is represented by variables in the LP). And the right term, $\mu_C$, is the mean edge cost of the cycle, which is a fixed number, property of the cycle itself.

Since the minimum cost circulation (with the volume constraint) minimizes the sum of all cycle costs, we can write it as:

$$
\begin{aligned}
\text{objective: } & \min \sum_{C \text{ cycle}} v_C \mu_C \\
\text{subject to: } & \sum v_C = 1
\end{aligned}
$$

Which is now clearly minimized when all volume is concentrated on the minimum mean cost cycles. Hence the cycle decomposition of the flow that the LP finds, will exhibit one or more minimum mean cycles. Furthermore, since the total volume is 1, the optimized objective will exactly equal the mean edge cost on the minimum mean cycle.

**Problem 4b.** The dual LP is:

$$
\text{objective: } \max \lambda \tag{7.5}
$$

$$
\text{subject to: } y_j - y_i + \lambda \leqq c_{ij} \quad \forall ij \tag{7.6}
$$

Note that the signs of $y_j$ and $y_i$ can be interchanged without changing the LP. We chose the above form for convenience.

**Problem 4c.** Rewrite constraint (7.6) as:

$$y_i + (c_{ij} - \lambda) - y_j \geqq 0 \quad \forall ij$$

This can be interpreted as: *What is the maximum value $\lambda$ that you can subtract from all edge costs of $G$, such that the resulting graph $G'$ has a feasible price function, i.e. has no negative cycles.*

Now notice that the mean edge cost $\mu'_C$ in $G'$ relates to the mean edge cost $\mu_C$ of the same cycle in $G$ as:

$$\mu'_C = \mu_C - \lambda$$

Let $\mu^*$ be the mean edge cost in the minimum mean cycle of $G$. So, in particular:

- When $\lambda > \mu^*$, $G'$ will have at least one negative cost cycle, namely, the minimum mean cycle of $G$
- When $\lambda \leqq \mu^*$ $G'$ will have no negative cost cycles. And, in particular, when $\lambda = \mu^*$ the minimum mean cycles in $G$ will have 0 cost.

Hence $\lambda = \mu^*$. Furthermore notice, that finding $\lambda$ (using the LP) also gives you a way of identifying the actual minimum mean cycle. Simply construct $G'$ and find the cycles of 0 mean edge cost. Even better, examine $G'$ using the reduced edge costs. The minimum mean cycles of $G$ will have only 0-cost edges in $G'$ with reduced costs (provided by the LP). So, consider only the 0-cost edges in $G'$ and find a cycle, using DFS over all connected components e.g.

**Problem 4d.** Let $c_1, k_1$ and $c_2, k_2$ be the total edge cost and number of edges on two cycles with different mean edge costs. Bound the difference of their mean edge costs as follows:

$$\begin{aligned}\left|\frac{c_1}{k_1} - \frac{c_2}{k_2}\right| &= \frac{1}{k_1 k_2}\Big|c_1 k_2 - c_2 k_1\Big| \\ &= \frac{1}{k_1 k_2}\Big|(k_2 - k_1)c_1 + k_1(c_1 - c_2)\Big| \\ &\geqq \frac{1}{k_1 k_2}\Big(|k_2 - k_1||c_1| + |k_1||c_1 - c_2|\Big)\end{aligned}$$

If $k_1 \neq k_2$, this difference is lower-bounded by $1/n^2$, if $c_1 \neq c_2$, this difference is lower-bounded by $1/n$. Altogether, the difference is lower-bounded by $1/n^2$. Also observe that $\lambda \in [-C, +C]$, where $C$ is the maximum value edge cost in $G$.

The algorithm will be as follows. Do a binary search on $\lambda$ in the range $[-C, +C]$ with discrete values separated by $1/n^2$. In each iteration of the binary search, create the graph $G'$ with edge costs $c_{ij} - \lambda$, and check for the existence of a negative cost cycle using the Bellman-Ford shortest paths algorithm, in time $O(mn)$.

When the binary search completes, compute a feasible price function in $G'$ (with edge costs $c_{ij} - \lambda$). Then consider only those edges of $G'$ (with reduced cost now) whose cost is less than $1/n^2$, and find a cycle among them (using Dijkstra's). This cycle is a minimum mean cycle in the original graph. Using the $c_{ij}$ of the original graph, compute the exact value of $\lambda$ as the mean edge cost in this cycle in $G$.

Our binary search takes $\log 2Cn^2$ iterations of $O(mn)$ time each, which amounts to a total time $O(mn(\log n + \log C))$.