CSC2411 - Linear Programming and Combinatorial Optimization*
Lecture 12: Approximation Algorithms using Tools from LP

Notes taken by Matei David

April 12, 2005

Summary: In this lecture, we give three example applications of approximating the solutions to IP by solving the relaxed LP and rounding. We begin with Vertex Cover, where we get a 2-approximation. Then we look at Set Cover, for which we give a randomized rounding algorithm which achieves a $\mathcal{O}(\log n)$-approximation with high probability. Finally, we again apply randomized rounding to approximate MAXSAT.

In the associated tutorial, we revisit MAXSAT. Then, we discuss primal-dual algorithms, including one for Set Cover.

LP relaxation It is easy to capture certain combinatorial problems with an IP. We can then relax integrality constraints to get a LP, and solve the latter efficiently. In the process, we might lose something, for the relaxed LP might have a strictly better optimum than the original IP. In last class, we have seen that in certain cases, we can use algebraic properties of the matrix to argue that we do not lose anything in the relaxation, i.e. we get an exact relaxation.

Note that, in general, we can add constraints to an IP to the point where we do not lose anything when relaxing it to an LP. However, the size of the inflated IP is usually exponential, and this procedure is not algorithmically doable.

1 Vertex Cover revisited

In the IP/LP formulation of VC, we are minimizing $\sum x_i$ (or $\sum w_i x_i$ in the weighted case), subject to constraints of the form $x_i + x_j \geq 1$ for all edges i, j. In the IP, x_i is 1 when we have to pick set i, and 0 otherwise. In the LP relaxation, we only ask that $0 \leq x_i \leq 1$. Actually, the upper bound on x_i is redundant, as the optimum solution cannot have any $x_i > 1$.

* Lecture Notes for a course given by Avner Magen, Dept. of Computer Sciecne, University of Toronto.

Let $OPT(G)$ denote the optimum to the IP (which is what we want), and let $OPT_f(G)$ denote the (fractional) optimum solution of the LP (which we can get efficiently). To see that for some G, $OPT_f(G) < OPT(G)$, look at K_3. In that case, $OPT(K_3) = 2$, but $x_1 = x_2 = x_3 = 1/2$ is a feasible LP solution with cost $3/2$, so $OPT_f(K_3) < 3/2$. In fact, by symmetry we can argue that $OPT_f(G_3) = 3/2$.

So we can get an optimal fractional solution to the LP with cost OPT_f, but what we want is a good integral solution to the IP, with cost close to OPT. If we were able to always find the best integral solution, then we would solve VC, so $P = NP$. Therefore, we'll settle for an integral solution which is "not too far" from OPT. The idea we are going to use is to *round* certain fractional x_i's to integers. We will then argue about the factor by which the objective function grows during the rounding. Before describing the algorithm for VC, note the following Lemma.

Lemma Every vertex of the LP relaxation of VC is half-integral, meaning that all its coordinates have values 0, 1/2 or 1.

Algorithm Let x^* denote an optimal fractional solution. Given the fact above, the most natural rounding is to take the ceiling, i.e. we set $x_i = 0$ if $x_i^* = 0$, and $x_i = 1$ if $x_i^* \in \{1/2, 1\}$. In order to argue that we get a 2-approximation, we need to show that (i) x is still feasible, and that (ii) at most a factor of 2 is lost in the rounding. The ratio of the approximation algorithm is a measure of how far the integral optimum OPT is from the integral solution we are building. In general, we do not know OPT, so we use OPT_f instead, which is a lower bound on OPT.

To argue (i), assume that some constraint $x_i + x_j \geq 1$ for some edge (i, j) is now unsatisfied. Then $x_i = x_j = 0$. By the rounding, $x_i^* = x_j^* = 0$. But then $x_i^* + x_j^* = 0 < 1$, contradicting the assumption that x^* is feasible. As for (ii), $\sum x_i \leq \sum 2 \cdot x_i^* = 2 \cdot OPT_f \leq 2 \cdot OPT$.

Note that even without the Lemma, we could round anything below 1/2 to 0, anything equal to or greater than 1/2 to 1, and the argument would still work.

Proof of Lemma Let x be a feasible solution which contains other than half-integral values. So, for certain i, $x_i \in (0, 1/2)$ and for certain j, $x_j \in (1/2, 1)$. We claim that x is not a BFS of the LP. To prove this, we show that x is not extreme.

The first thing to note is that there can be no edges in between vertices i, j with both $x_i < 1/2$ and $x_j \leq 1/2$, because the constraint $x_i + x_j \geq 1$ would not be satisfied. For some small but positive ϵ, consider the solutions

$$x_i^+ = \begin{cases} x_i + \epsilon & x_i \in (1/2, 1) \\ x_i - \epsilon & x_i \in (0, 1/2) \\ x_i & \text{otherwise} \end{cases} \qquad x_i^- = \begin{cases} x_i - \epsilon & x_i \in (1/2, 1) \\ x_i + \epsilon & x_i \in (0, 1/2) \\ x_i & \text{otherwise} \end{cases}$$

We can choose ϵ small enough so that values of all vertices do not leave the intervals $(0, 1/2)$ and $(1/2, 1)$, i.e. $x_i^+ \in (1/2, 1)$ iff $x_i \in (1/2, 1)$ etc.

Clearly, $x = (x^+ + x^-)/2$. We claim both x^+, x^- are still feasible. The only edges (i, j) which x^+ might violate are those which have at least one vertex, say i, with $x_i \in$

$(0, 1/2)$. But then $x_j > 1/2$, for otherwise x would not be feasible. If $x_j = 1$, then $x_i^+ + x_j^+ \geq 1$. Otherwise, $x_j \in (1/2, 1)$ and $x_i^+ + x_j^+ = (x_i - \epsilon) + (x_j + \epsilon) = x_i + x_j \geq 1$ by feasibility of x. A similar argument shows that x^- is feasible. Hence, x is a convex combination of two different feasible solutions, therefore it's not an extreme and not a BFS.

Notes There are trivial 2-approximation algorithms for VC, the entire machinery of IP/LP solving and rounding wasn't really necessary. However, it does make for a nice application. The algorithm consists of the following steps.

- relax the IP to an LP;
- solve the LP;
- map the fractional optimum to the LP to a feasible integral solution;
- analyze that not much is lost in the translation.

Approximation ratios An algorithm A for a minimization problem is called an α-approximation if $\alpha = \sup_I \frac{A(I)}{OPT(I)} \geq 1$. Note that α might depend on the size of the input I (e.g. $\alpha = \log n$). For VC, we've seen a constant factor 2-approximation. For a maximization problem, an algorithm is an α-approximation if $\alpha = \inf_I \frac{A(I)}{OPT(I)} \leq 1$.

Integrality Gap The integrality gap for a certain IP/LP formulation of a problem is a bound on the ratio between the true integral optimum and the fractional optimum. The IG of a minimization problem is $\sup_I \frac{OPT(I)}{OPT_f(I)}$. and the IG of a maximization problem is $\inf_I \frac{OPT(I)}{OPT_f(I)}$. Intuitively, the closer the IG is to 1, the better we hope the approximation will be.

To analyze the IG for the VC formulation we've seen before, consider K_n. For this instance, we have $OPT(K_n) = n - 1$. But $x_1 = \cdots = x_n = 1/2$ is a feasible solution to the LP, so $OPT_f(K_n) \leq n/2$. Hence $IG \geq 2 - 2/n$. Since on any instance I, the cost of the integral solution we produce is at least $OPT(I)$ and at most $2 \cdot OPT_f(I)$, we get $IG \leq 2$. Therefore, for this formulation of VC, taking the supremum we get $IG = 2$.

Tightening an IP Suppose we have an IP for a minimization problem, and we have to relax it to an LP. The LP might achieve smaller optimum cost than the IP by using fractional solutions. Consider some constraint that *any* integral solution must satisfy. We can safely add any such constraint to the IP/LP formulation, without affecting the integral optimum. However, the new constraint might eliminate certain fractional solutions, so the fractional optimum can only increase, bringing it closer to the integral optimum and lowering the IG.

For example, consider any triangle (i, j, k) in a graph instance of VC. No (integral) vertex cover of the graph can include less than 2 vertices from that triangle. Hence, we can safely add a constraint of the form $x_i + x_j + x_k \geq 2$ to the IP/LP formulation.

We do not change OPT, since any feasible solution of the IP satisfies this constraint. However, we might potentially increase OPT_f, thus reducing the IG.

What can we hope to get in this way? Can we get a better than 2-approximation? What other LP formulation can we consider? For example, if we apply the general tightening technique by Lovasz-Schrijver [LS91], we get in one step constraints about all odd cycles in the graph. Namely, if C is an odd cycle containing vertices $x_1, \ldots, x_l$, we can add the constraint $\sum_{i=1}^{l} x_i \geq \lfloor \frac{l}{2} \rfloor$. This is strictly stronger than what we can derive from individual edges, which is $\sum_{i=1}^{l} \geq \frac{l}{2}$. Arora, Bollobos and Lovasz [ABL02] consider all LP formulations which involve a constant fraction of the vertices and that use the "natural" 0/1 variables, i.e. x_i for each vertex i indicating if it is taken in the cover. They also contain formulations that contain all odd cycle constraints among others. They show that despite these relatively strong formulations, the integrality gap is still $2 - o(1)$.

2 Randomized Rounding for Set Cover

In the Set Cover problem, we have a universe U of n elements, and m subsets $S_1, \ldots, S_m \subseteq U$. In the unweighted case, we simply want to minimize the number of sets we pick in order to cover U. Notice that VC is a special instance of SC, where the elements of the universe are the edges, and we have a set for every vertex, containing the edges incident to that vertex.

IP/LP formulation For the IP/LP formulation os SC, we consider the $n \times m$ 0/1 matrix A, which has $a_{ij} = 1$ if $i \in S_j$ and 0 otherwise. The sets correspond to columns in A, and elements of the universe to rows of A. The variable x_j is 1 if we have to take S_j and 0 otherwise. So, the IP/LP formulation is

$$\min \sum x_j$$

subject to

$$A \cdot x \geq 1$$

In the IP, $x_j \in \{0, 1\}$. In the LP, we only ask $x \geq 0$. So, given the IP, we relax it to an LP and solve the LP to get optimal fractional solution x^*. How do we map this to an integral solution?

The idea is to think of the quantity x_i^* as a probability. So, instead of directly mapping x^* to an integral solution, we define a stochastic procedure to do this step. For example, if the optimal fractional solution is $x^* = (0, 1/10, 9/10)$, we would like to be more likely to select S_3 than S_2, and we will never select S_1. So, in one pass of randomized rounding, we (independently) pick the set S_i with probability x_i^*. Note that this rounding has the nice property of leaving integer values unchanged. Suppose we do several (around $\log n$) such passes, and we let our final cover be the union of all the covers in the individual passes. In other words, we take S_i in the final cover, if we take S_i in any one pass. We would like to say that with high probability we get a cover of U, and that with high probability, the objective function does not increase much.

We do not cover a certain element a in one pass if none of the sets a belongs to are taken in that pass. So, the probability a is not covered is exactly $\prod_{j:a\in S_j}(1-x_j^*)$. The constraint corresponding to a in the LP says that $\sum_{j:a\in S_j} x_j^* \geq 1$. So,

$$\Pr[a \text{ not covered in one pass}] = \prod_{j:a\in S_j}(1-x_j^*) \leq \prod e^{-x_j^*} = e^{-\sum x_j^*} \leq e^{-1} = \frac{1}{e}$$

After $(\log n + 2)$ iterations, we haven't covered a iff we haven't covered it in any iteration, i.e. with probability at most $e^{-(\log n+2)} \leq \frac{1}{4n}$. By union bound, the probability that there exists an uncovered element after $(\log n + 2)$ passes is at most 1/4.

On the other hand, $E[\sum x_j] = (\log n + 2)\cdot \sum x_i^* \leq (\log n + 2)\cdot OPT_f$. By Markov's inequality,

$$\Pr[\sum x_j > 4\cdot(\log n + 2)\cdot OPT_f] \leq \frac{1}{4}$$

Therefore, the probability that after $(\log n + 2)$ iterations we have a cover of cost at most $4\cdot(\log n+2)\cdot OPT_f$ is at least $1 - 1/4 - 1/4 = 1/2$. We can apply the same algorithm several times to amplify the probability of success.

This is probably the easiest algorithm using randomized rounding. Note that if OPT is the true integral optimum, OPT_f is the fractional optimum and A is the result of rounding OPT_f to an integral solution, we have $OPT_f \leq OPT \leq A$. The ratio OPT/OPT_f is controlled by the IG, and the ratio A/OPT_f is controlled by the rounding process. Ultimately, we are interested in bounding the ratio A/OPT. In particular, if we have a small IG, we also need a small loss in rounding to get a good ratio.

For Set Cover, the IG is $\Omega(\log n)$. Feige [F??] showed that if $P \neq NP$, no other technique will do better. For Vertex Cover, the trivial 2-approximation algorithms are essentially the best known.

3 Randomized Rounding for MAXSAT

The input to MAXSAT is a CNF formula, which is a collection of clauses. Each clause is a disjunction of literals, and each literal is a boolean variable or the negation of a variable. For example, $C = x_1 \vee \overline{x}_3 \vee x_5 \vee x_7$ is a clause. The output in the MAXSAT problem is an assignment that maximizes the number of satisfied clauses.

IP formulation In the IP formulation, we introduce 0/1 variables x_i for every variable in the formula, and 0/1 variables Z_C for every formula C. The semantics is that x_i is 1 in the IP iff x_i should be set to $true$ in the formula, and Z_C is 1 iff C is satisfied by the assignment to x variables.

For every clause C, let L_C^+ be the set of indices of variables appearing positively in C, and let L_C^- be the set of indices of variables appearing negatively. For example, if $C = x_1 \vee x_5 \vee \overline{x}_7$, we have $L_C^+ = \{1, 5\}$ and $L_C^- = \{7\}$.

With this notation, the IP/LP is

$$\max \sum_{C} Z_C$$

such that

$$\sum_{i \in L_C^+} x_i + \sum_{i \in L_C^-} (1 - x_i) \geq Z_C$$

The IP requires $x_i, Z_C \in \{0, 1\}$, while the LP relaxation requires $0 \leq x_i, Z_C \leq 1$. To see that the IP captures the problem, note that there is no point in Z_C being 0 when C is satisfied, for we are maximizing the sum of the Z_C's. Unlike in the case of VC, we do need the ≤ 1 constraints. The only constraint we do not actually need is $Z_C \geq 0$.

Let (x^*, Z^*) denote an optimal fractional solution to the LP relaxation, with cost OPT_f. Next, set x_i to $true$ in the formula with probability x_i^*. There's no need to set Z_C, we simply set it to 1 if C is satisfied. Note that with MAXSAT, there are no issues about feasibility of the constructed solution, as in the case of VC and SC.

Let us now study the probability a certain clause C is not satisfied. Wlog, $C = x_1 \vee \cdots \vee x_k$. We are assuming all literals are different, for two opposite occurences of the same variable make the clause always true, and two occurences of the same sign are redundant. Furthermore, we are assuming C contains only positive literals, for otherwise we can replace $\overline{x}_i$ by y_i, and $1 - x_i^*$ by y_i^*. For every clause C, we seek to relate Z_C^* with the probability we satisfied clause C.

$$\begin{aligned}
\Pr[C \text{ not satisfied}] &= \prod_{i=1}^{k} (1 - x_i^*) \\
&\leq \left(\frac{\sum_{i=1}^{k} (1 - x_i^*)}{k} \right)^k, \text{ geometric mean at most arithmetic mean} \\
&= \left(1 - \frac{\sum_{i=1}^{k} x_i^*}{k} \right)^k \\
&\leq \left(1 - \frac{Z_C^*}{k} \right)^k, \text{ as } \sum_{i=1}^{k} x_i^* \geq Z_C^*
\end{aligned}$$

The function $h(z) = 1 - \left(1 - \frac{z}{k}\right)^k$ on the interval $z \in [0, 1]$ is concave, as

$$\frac{d}{dz} h(z) = (-1) \cdot k \cdot \left(1 - \frac{z}{k}\right)^{k-1} \cdot \left(-\frac{1}{k}\right),$$

$$\frac{d^2}{dz^2} h(z) = (k-1) \cdot \left(1 - \frac{z}{k}\right)^{k-2} \cdot \left(-\frac{1}{k}\right) < 0.$$

So $h(z) = h((1-z) \cdot 0 + z \cdot 1) \geq (1-z) \cdot h(0) + z \cdot h(1) = z \cdot h(1)$. Therefore,

$$\Pr[C \text{ satisfied}] \geq 1 - \left(1 - \frac{Z_C^*}{k}\right)^k = h(Z_C^*) \geq Z_C^* \cdot h(1) = Z_C^* \cdot \left(1 - \left(1 - \frac{1}{k}\right)^k\right)$$

Since $\sum_C Z_C^* = OPT_f$ and $OPT \leq OPT_f$, we now have

$$\begin{aligned}
E[\text{number of satisfied clauses}] &= \sum_C \Pr[C \text{ satisfied}] \\
&\geq \sum_C Z_C^* \left(1 - \left(1 - \frac{1}{|C|}\right)^{|C|}\right) \\
&\geq OPT_f \cdot \left(1 - \frac{1}{e}\right) \\
&\geq OPT \cdot \left(1 - \frac{1}{e}\right)
\end{aligned}$$

So, if the objective function "gains" Z_C^*, the rounded solution "gains" $(1-1/e) \cdot Z_C^*$ in expectation. We thus get an assignment that satisfies at least $(1-1/e) \cdot OPT_f$ clauses in expectation. As before, the analysis involved bounding individually every Z_C.

Derandomizing the rounding Can we derandomize this algorithm, that is, decide how to set the variables deterministically in such a way that we obtain the same number of satisfied clauses, this time no longer in expectation? It turns out that for this problem, we can. We know that the expectation over all choices of x_1 is at least some T. What we can do is to set x_1 to $true$, simplify the formula, count the number T_1 of clauses we satisfy directly and recompute the number T_2 of clauses we expect to satisfy over the choices over the remaining variables. If $T_1 + T_2 \geq T$, we leave x_1 set to $true$ and continue. Otherwise, we set x_i to $false$. We know that at least one of these choices gives the number of clauses at least T.

$$\begin{aligned}
E[\text{\# satisfied}] &= \sum_C \Pr[C \text{ satisfied}] \\
&= \sum_C (\Pr[x_i = 1] \cdot \Pr[C \text{ satisfied}|x_i = 1] + \Pr[x_i = 0] \cdot \Pr[C \text{ satisfied}|x_i = 0]) \\
&= x_i^* \cdot \sum_C \Pr[C \text{ satisfied}|x_i = 1] + (1 - x_i^*) \cdot \sum_C \Pr[C \text{ satisfied}|x_i = 0] \\
&= x_i^* \cdot E[\text{\# satisfied}|x_i = 1] + (1 - x_i^*) \cdot E[\text{\# satisfied}|x_i = 0]
\end{aligned}$$

At least one of the two conditional expectations must be at least the unconditional one.

4 Tutorial - Half-Approximation for MAXSAT

We are given a CNF formula with variables $x_1, \ldots, x_n$, and clause weights w_C. We want to find a truth assignment that maximizes the sum of the weights of the satisfied clauses.

Suppose we randomly and independently assign each x_i to either $true$ and $false$ with probabilities 1/2 and 1/2. A clause of size $|C|$ is not satisfied with probability

$2^{-|C|}$ (we assume each variable occurs exactly once in every clause, either positively or negatively). Let W be the random variable denoting the weight of the satisfied clauses. Than

$$E[W] = \sum_C w_C \Pr[C \text{ satisfied}] = \sum_C w_C \left(1 - 2^{-|C|}\right)$$

Since $|C| \geq 1$, we have $E[W] \geq 1/2 \cdot \sum_C w_C$. Furthermore, the best we can hope for is to satisfy all clauses, so $\sum_C w_C \geq OPT$, and therefore $E[W] \geq 1/2 \cdot OPT$. This is an easy 2-approximation.

We can derandomize this algorithm, so we can get a half-approximation deterministically. Consider a complete binary tree of depth n. We can label every node by the set of literals that are set to true everywhere in the subtree below that node. So label the root node with $\emptyset$, label the two children of the root with $\{x_1\}$ and $\{\overline{x_1}\}$, respectively. In all nodes below the child labelled $\{x_1\}$, the variable x_1 is set to $true$. Conversely, below $\{\overline{x}_1\}$, x_1 is set to $false$. The labels on level 2 below root are $\{x_1, x_2\}, \{x_1, \overline{x}_2\}, \{\overline{x}_1, x_2\}, \{\overline{x}_1, \overline{x}_2\}$. Our algorithm can be seen as selecting a certain leaf of this tree, where all vaiables are assigned to either $true$ or $false$.

At the root, we have $E[W]$. Set x_1 to $true$. Some clauses will be satisfied (those that contain x_1), say a of them. Other clauses will be simplified (those that contain $\overline{x}_1$), say b of them. For the simplified clauses, we can get a 1/2-approximation. So if continue along this branch, we can get at least $a+b/2$ clauses satisfied in total, which is a lower bound on $E[W|x_1 = 1]$. Since $E[W] = 1/2 \cdot E[W|x_1 = 1] + 1/2 \cdot E[W|x_1 = 0]$, we know at least one of the conditional expectations is at least $E[W]$. We can compute lower bounds on both $E[W|x_1 = 1]$ and $E[W|x_1 = 0]$, and simply follow the path along which the expectation is larger.

5 Tutorial - Complementary Slackness

Consider the primal IP/LP

$$\min \sum_{j=1}^{n} c_j x_j$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i, \text{ for all } 1 \leq i \leq m$$

$$x_j \geq 0$$

with its dual

$$\max \sum_{i=1}^{m} b_i y_i$$

subject to

$$\sum_{i=1}^{m} a_{ij} y_i \leq c_j, \text{ for all } 1 \leq j \leq n$$

$$y_i \geq 0$$

Let (x, y) be optimum solutions to the primal and the dual. The slackness conditions are:

$$\begin{aligned} \text{primal} &: \quad x_j \neq 0 \Rightarrow \sum_{i=1}^{m} a_{ij} y_i = c_j \\ \text{dual} &: \quad y_i \neq 0 \Rightarrow \sum_{j=1}^{n} a_{ij} x_j = b_i \end{aligned}$$

The idea of a primal-dual algorithm for an IP is to use primal-dual slackness to solve the IP. This is not possible without breaking some hardness assumptions, but we can get an approximation algorithm. For example, we can relax dual slackness conditions and only require, for some $\alpha > 1$, that

$$y_i \neq 0 \Rightarrow \sum_{j=1}^{n} a_{ij} x_j \leq \alpha \cdot b_i$$

Note that by feasibility of x, we get $b_i \leq \sum_{j=1}^{n} a_{ij} x_j$.

Theorem If we can find feasible primal and dual solutions satisfying primal slackness conditions and α-relaxed dual slackness conditions, then x is an α-approximation.

$$\begin{aligned} \sum_{j=1}^{n} c_j x_j &= \sum_{j:x_j>0} c_j x_j, \text{ since } x_j \geq 0 \\ &= \sum_{j:x_j>0} x_j \left(\sum_{i=1}^{m} a_{ij} y_i \right), \text{ by primal slackness} \\ &= \sum_{j:x_j>0} \sum_{i=1}^{m} a_{ij} x_j y_i \\ &= \sum_{i:y_i>0} y_i \left(\sum_{j:x_j>0} a_{ij} x_j \right), \text{ since } y_i \geq 0 \\ &= \sum_{i:y_i>0} y_i \left(\sum_{j=1}^{n} a_{ij} x_j \right), \text{ added 0 terms} \\ &\leq \sum_{i:y_i>0} y_i \cdot \alpha \cdot b_i, \text{ by relaxed dual slackness} \\ &= \alpha \left(\sum_{i=1}^{m} b_i y_i \right), \text{ added 0 terms} \\ &\leq \alpha \cdot OPT, \text{ since } y \text{ feasible.} \end{aligned}$$

The problem, then, is how to chose α, and how to take steps making sure these conditions are satisfied. A general primal-dual algorithm look like the following.

start with integral (generally non-feasible) primal solution $x = 0$
and with feasible dual solution $y = 0$
while there exists a primal constraint not satisfied for some i
 increase y_i until one or many dual constraints j become tight
 set x_j to 1 for all tight dual constraints

To prove a bound on the approximation ratio, we have to chose α well.

Unweighted Set Cover The primal LP is

$$\min \sum_S x_S$$

subject to

$$\sum_{S:i \in S} x_S \geq 1, \text{ for all elements } i$$

$$x \geq 0$$

and the dual LP is

$$\max \sum_i y_i$$

subject to

$$\sum_{i \in S} y_i \leq 1, \text{ for all sets } S$$

$$y \geq 0$$

Relaxed dual slackness can be easily satisfied with $\alpha = f$. Consider the following primal-dual algorithm, following the general paradigm above:

start with $x = 0$, which is integral but not primal feasible
and with $y = 0$, which is dual feasible
while there exists some element i not covered (i.e. $\sum_{S:i \in S} x_S < 1$)
 set y_i to 1
 set x_S to 1 for all sets S containing i (i.e. pick all these sets as part of the cover)

The primal slackness condition $x_S \neq 0 \Rightarrow \sum_{i \in S} y_i = 1$ is satisfied because in the while loop we set $y_i = 1$ for exaclty one element i from each S. The f-relaxed dual slackness condition $y_i \neq 0 \Rightarrow \sum_{S:i \in S} x_S \leq f \cdot 1$ holds because for every element i, we can select at most all sets containing it, and there are at most f of those. So this algorith is an f-approximation to SC.

References

[ABL02] Arora, Bollobas and Lovasz. *Proving integrality gaps without knowing the linear program*. FOCS, 2002.

[LS91] Lovasz and Schrijver. *Cones of matrices and set-functions and 0-1 optimization*. SIAM Journal on Optimization, 1991.

[F??] Feige.