Pipage Rounding[1]

- In a previous lecture, we described a rounding algorithm to convert a fractional solution to the bipartite matching into an integral solution with the same or larger value. In this lecture, we build on that idea and show its applicability to obtaining approximation algorithms for NP-hard problems. This style of rounding has been called *pipage rounding* in the literature. We illustrate this on a problem which we have seen before: MAX COVERAGE. Recall that in this problem we are given a universe U and a collection of subsets of U: $\mathcal{S} = \{S_1, \ldots, S_m\}$. The objective is to choose a collection of k sets so as to maximize the cardinality of their union. We know that a greedy algorithm obtains an $1 - \left(1 - \frac{1}{k}\right)^k$-approximation which tends to $1 - \frac{1}{e}$ as $k \to \infty$.

  Suppose every element was present in at most f sets. That is, the *degree* of the set system $\mathcal{S}$ is $\leq f$. In this note we describe an $1 - \left(1 - \dfrac{1}{f}\right)^f$-approximation for the MAX-COVERAGE problem. In particular, this implies a $\frac{3}{4} = 0.75$-approximation for the MAX VERTEX COVERAGE problem, where we are given an undirected graph $G = (V, E)$ and objective is to choose a subset $U \subseteq V$ with $|U| = k$ which maximizes the number of edges having at least one endpoint in U. This is because, $f = 2$ for this set family; every edge contains 2 vertices. It is not hard to show an example where the greedy algorithm for MAX-COVERAGE when tailored to MAX VERTEX COVERAGE only be as good as a $\approx 1 - \frac{1}{e} \approx 0.632$ approximation.

- ***LP Relaxation.*** We begin with a LP relaxation for the problem.

$$\texttt{opt} \leq \mathsf{lp}(U, \mathcal{S}) := \text{maximize} \quad \sum_{i \in U} z_i \tag{MaxCov-LP}$$

$$z_i \leq \sum_{j : i \in S_j} x_j, \qquad \forall i \in U \tag{1}$$

$$\sum_{j=1}^{m} x_j = k, \tag{2}$$

$$0 \leq z_i, x_j \leq 1, \qquad \forall i \in U, 1 \leq j \leq m \tag{3}$$

  Above, x_j indicates whether set j is picked, z_i indicates to what extent element i is covered. (1) captures the notion that an element is covered only if a set containing it is picked, and (3) captures the notion that any element can't be covered more than once.

  Before moving ahead, for reasons which will soon become clear, we rewrite the above program by eliminating the z-variables. Given x_j's, the value z_i will be set to $\min(1, \sum_{j : i \in S_j} x_j)$.Therefore, (MaxCov-LP) is equivalent to

$$\texttt{opt} \leq \max \; \{L(\mathbf{x}) : \; \mathbf{x} \in \mathcal{P}\} \tag{4}$$

[1] *Lecture notes by Deeparnab Chakrabarty. Last modified : 4th Mar, 2022*
These have not gone through scrutiny and may contain errors. If you find any, or have any other comments, please email me at deeparnab@dartmouth.edu. Highly appreciated!

where

$$L(\mathbf{x}) = \sum_{i \in U} \min\left(1, \sum_{j: i \in S_j} \mathbf{x}_j\right) \quad \text{and} \quad \mathcal{P} := \{\mathbf{x} \in [0,1]^m : \sum_j \mathbf{x}_j = k\}$$

Although the function $L : \mathbb{R}^m \to \mathbb{R}$ is not linear, (4) can be solved via the linear program (MaxCov-LP) since it is equivalent to it. However, the solution to (4) can have *fractional entries*, that is, needn't be in $\{0,1\}^m$

- ***Another non-linear function.*** We now describe another non-linear function $F : \mathbb{R}^m \to \mathbb{R}$ which is the key definition. The *nice* properties of F would be (a) the *maximum* value of $F(\mathbf{x})$ over $\mathbf{x} \in \mathcal{P}$ would also be an upper bound on $\mathtt{opt}$, and (b) the maximum value of F would be obtained on *integral* points. The *not-nice* property of F would be that maximizing F over $\mathcal{P}$ is NP-hard (yes, I understand that it seems no progress is being made; hold on).

  Here's the function.

$$F(\mathbf{x}) := \sum_{i \in U} \left(1 - \prod_{j: i \in S_j} (1 - \mathbf{x}_j)\right) \tag{Continous Coverage}$$

  Observe that whenever $\mathbf{x} \in \{0,1\}^m$, $F(\mathbf{x})$ equals the value of the algorithm which picks sets with $\mathbf{x}_j = 1$. Thus, as promised, the following math program is also an upper bound on $\mathtt{opt}$.

$$\mathtt{opt} \leq \max \ \{F(\mathbf{x}) : \ \ \mathbf{x} \in \mathcal{P}\} \tag{5}$$

  As mentioned above, the beauty of the function $F(\cdot)$ defined in (Continous Coverage) is the following : given any solution $\mathbf{x} \in \mathcal{P}$, there is a rounding algorithm called PIPAGE ROUNDING, which returns a solution $\mathbf{x}^{\mathsf{int}} \in \{0,1\}^m \cap \mathcal{P}$ with the property that $F(\mathbf{x}^{\mathsf{int}}) \geq F(\mathbf{x})$. In other words, the program (5) has *no integrality gap*! We describe this pipage rounding in greater detail soon, but before doing so we need to address what use is this rounding algorithm if (5) cannot be solved in polynomial time (which we believe one can't unless P=NP).

- ***Comparing F and L.*** The main point is that although F can't be maximized over $\mathcal{P}$, L can, and the following analytic claim shows that F and L are point-wise related. For brevity's sake, let $\rho(x) := 1 - \left(1 - \frac{1}{x}\right)^x$. It is not too hard to see $d \leq f$ implies $\rho(d) \geq \rho(f)$.

Lemma 1. For all $\mathbf{x} \in \mathcal{P}$, $F(\mathbf{x}) \geq \rho(f) \cdot L(\mathbf{x})$, where f is the degree of the set system $\mathcal{S}$.

Proof. Fix $i \in U$ and suppose it is contained in $d \leq f$ sets. By the AM-GM inequality,

$$\prod_{j: i \in S_j} (1 - \mathbf{x}_j) \leq \left(\frac{\sum_{j: i \in S_j} (1 - \mathbf{x}_j)}{d}\right)^d = \left(1 - \frac{\sum_{j: i \in S_j} \mathbf{x}_j}{d}\right)^d \tag{6}$$

Let $g(t) := 1 - (1 - \frac{t}{d})^d$. Then, for $d \geq 1$ and $t \in [0,1]$ we have $g(t) \geq \rho(d) \cdot t$. This follows because in that interval, g is concave, and thus $g(t) \geq (1-t)g(0) + tg(1) = \rho(d) \cdot t$. Now, we can use this in (6) to say

$$1 - \prod_{j: i \in S_j} (1 - \mathbf{x}_j) \geq \min\left(1, \sum_{j: i \in S_j} \mathbf{x}_j\right) \cdot \rho(d) \geq \rho(f)$$

where we used the monotonicity of ρ in the last inequality. Summing for all $i \in U$ proves the lemma. $\square$

- ***Approximation Algorithm.*** To summarize, we know how to maximize L, that is solve (4), but the solution may not be integral. We don't know how to maximize F, that is solve (5), but we know there is an integral optimum. The previous lemma tells us $F(\mathbf{x}) \geq \rho_{\mathtt{f}} \cdot L(\mathbf{x})$ for all $x \in \mathcal{P}$. Putting all three of these together implies the following $\rho_{\mathtt{f}}$-approximation for MAX COVERAGE.

1: **procedure** MAX COVERAGE ROUNDING($\mathcal{S}$):
2: Solve (MaxCov-LP) to get $(\mathbf{x}, \mathbf{z})$.
3: Run PIPAGE ROUNDING($\mathbf{x}$) to obtain $\mathbf{x}^{\mathsf{int}}$ with $F(\mathbf{x}^{\mathsf{int}}) \geq F(\mathbf{x})$. ▷ *We describe this next.*
4: Pick sets with $\mathbf{x}_j^{\mathsf{int}} = 1$ covering $F(\mathbf{x}^{\mathsf{int}})$ elements.
5: ▷ *Since $\mathbf{x}^{\mathsf{int}} \in \mathcal{P} \cap \{0,1\}^m$, there will be exactly k sets picked.*

Theorem 1. MAX COVERAGE ROUNDING is a $\rho_{\mathtt{f}}$ approximation.

To see why, note that $\mathtt{opt} \leq L(\mathbf{x}) \leq \frac{1}{\rho_{\mathtt{f}}} F(\mathbf{x}) \leq \frac{F(\mathbf{x}^{\mathsf{int}})}{\rho_{\mathtt{f}}}$.

- ***Pipage Rounding.*** The setting where pipage rounding applies is more general than the one described above. Abstractly, suppose we want to maximize a function F on m variables in $\{0,1\}^m$ intersected with a polytope $\mathcal{P}$

$$\max\{F(\mathbf{x}) : \quad \mathbf{x} \in \mathcal{P} \cap \{0,1\}^m \} \tag{7}$$

Suppose the following two conditions hold.

Pa. For any non-integral $\mathbf{x} \in \mathcal{P}$, one can efficiently find a vector $\mathbf{v}_x \in \mathbb{R}^m$ and scalars $\alpha_x, \beta_x > 0$ such that $\mathbf{x} + \alpha_x \mathbf{v}_x$ and $\mathbf{x} - \beta_x \mathbf{v}_x$ have strictly more integral coordinates that $\mathbf{x}$.

Pb. For all $\mathbf{x} \in \mathcal{P}$, the function $F(\cdot)$ is convex in the direction of the above vector $\mathbf{v}_x$. More precisely, the function $g_x(t) := F(\mathbf{x} + t\mathbf{v}_x)$ is a convex function over the variable $t \in \mathbb{R}$.

Theorem 2. If conditions (Pa) and (Pb) are satisfied, then given any $\mathbf{x} \in \mathcal{P}$ there exists an efficient algorithm PIPAGE which returns $\mathbf{x}^{\mathsf{int}} \in \mathcal{P} \cap \{0,1\}^m$ with $F(\mathbf{x}^{\mathsf{int}}) \geq F(\mathbf{x})$.

Remark: *In fact, one doesn't require the first condition (Pa) very strongly. It suffices if one can show $\mathbf{x} + \alpha_x \mathbf{v}_x$ and $\mathbf{x} - \beta_x \mathbf{v}_x$ "make progress" towards an integral solution. One possible measure of progress is that both these points lie on a face of $\mathcal{P}$ of smaller dimension.*

Proof. The proof is the following obvious while loop

procedure PIPAGE ROUNDING($\mathbf{x}$):
  while $\mathbf{x} \notin \{0,1\}^m$ **do**:
    Use (Pa) to obtain $\alpha_x, \beta_x, \mathbf{v}_x$.
    if $F(\mathbf{x} + \alpha_x \mathbf{v}_x) \geq F(\mathbf{x})$ **then**:
      $\mathbf{x} \leftarrow \mathbf{x} + \alpha_x \mathbf{v}_x$
    else:
      $\mathbf{x} \leftarrow \mathbf{x} - \beta_x \mathbf{v}_x$
    $\triangleright$ *Note that the number of integral coordinates in $\mathbf{x}$ increases in either case.*

We now show that in every while loop the value of $F(\mathbf{x})$ can only increase. Using the fact that $F(\cdot)$ is convex in direction of $\mathbf{v}_x$, we assert that $\max\left(F(\mathbf{x} + \alpha_x \mathbf{v}_x), F(\mathbf{x} - \beta_x \mathbf{v}_x)\right) \geq F(\mathbf{x})$ which would give us what we need. To see the inequality, write $0 = \frac{\beta_x}{\alpha_x+\beta_x} \cdot \alpha_x + \frac{\alpha_x}{\alpha_x+\beta_x} \cdot (-\beta_x)$ and use convexity of g_x to say

$$\underbrace{g_x(0)}_{=F(\mathbf{x})} \leq \frac{\beta_x}{\alpha_x + \beta_x} \underbrace{g_x(\alpha_x)}_{=F(\mathbf{x}+\alpha_x\mathbf{v}_x)} + \frac{\alpha_x}{\alpha_x + \beta_x} \underbrace{g_x(-\beta_x)}_{=F(\mathbf{x}+\alpha_x\mathbf{v}_x)}$$

And so, $g_x(0) \leq \max(g(\alpha_x), g(-\beta_x))$ proving what we asserted. Since the number of integral coordinates increases, the above procedure terminates in at most m steps. Thus the time taken is at most m times the time taken to implement (Pa). $\square$

- *(Pa) and (Pb) for coverage.* Recall, for the coverage problem $\mathcal{P} := \{x \in [0,1]^m : \sum_{j=1}^m x_j = k\}$.

Claim 1. (Pa) and (Pb) are true for the above polytope.

Proof. Suppose $\mathbf{x}$ is a non-integral vector in $\mathcal{P}$. Since $\sum_{j=1}^m \mathbf{x}_j$ is an integer, there must be at least two coordinates, call them x_p and x_q, such that both are in $(0,1)$. Our vector $\mathbf{v}_x$ is then the vector $(\mathbf{e}_p - \mathbf{e}_q)$, where $\mathbf{e}_i$ is the unit-vector with 1 in the tth coordinate and 0 elsewhere. Set $\alpha_x = \min(1 - \mathbf{x}_p, \mathbf{x}_q)$ and $\beta_x = \min(1 - \mathbf{x}_q, \mathbf{x}_p)$. Note that $\mathbf{x} + \alpha_x \mathbf{v}_x$ and $\mathbf{x} - \beta_x \mathbf{v}_x$ are vectors in $\mathcal{P}$ with at least one less fractional coordinate. The whole process above was efficient.

Now we establish the convexity of $F(\mathbf{x})$ in the direction $(\mathbf{e}_p - \mathbf{e}_q)$. Indeed, $g_x(t) := F(\mathbf{x} + t(\mathbf{e}_p - \mathbf{e}_q))$ can be written as using (Continous Coverage)

$$g(t) = \sum_{i \in U} h_i(t)$$

where $h_i(t)$ is independent of t if the element i is neither in the set S_p nor in the set S_q, is a linear function if i is in exactly one of S_p or S_q, or a quadratic in t with a positive coefficient for t^2 if $i \in S_p$ and $i \in S_q$. Indeed, if we define $C_x := \prod_{j: i \in S_j, j \neq p, j \neq q} (1 - \mathbf{x}_j)$ which is a *non-negative* constant independent of t, then in the first case $h_i(t) = 1 - C_x$, in the second case $h_i(t) = 1 - (1 - \mathbf{x}_p - t)C_x$ or $h_i(t) = 1 - (1 - \mathbf{x}_q + t)C_x$ which are both linear functions of t, or (most interestingly perhaps)

$$h_i(t) = 1 - (1 - \mathbf{x}_p - t)(1 - \mathbf{x}_q + t)C_x = t^2 C_x + \text{linear function of } t$$

if $i \in S_p \cap S_q$. In sum, in all cases $h_i(t)$ is a convex function, and thus $g(t)$ is a convex function. $\square$

Exercise: ☕☕ *Show that the* MAX COLORFUL COVERAGE *problem has a* $(1-\frac{1}{e})$*-approximation. Indeed, a* $(1-(1-\frac{1}{f})^f)$ *approximation.*

Notes

The algorithm in this note, and indeed the nomenclature of pipage rounding, is from the paper [1] by Ageev and Sviridenko. See the paper for other applications such as graph and hypergraph partitioning problems, and a job scheduling problem as well. Pipage rounding was used in the influential paper [2] by Calinescu, Chekuri, Pál, and Vondrák to give an $(1-\frac{1}{e})$-approximation for maximizing any monotone submodular function over a matroid constraint. Randomized versions of pipage rounding have been studied in the paper [4] by Gandhi, Khuller, Parthasarathy, and Srinivasan, for many kinds of problems with "hard constraints", and more recently explored for submodular objectives in the paper [3] by Chekuri, Vondrák, and Zenklusen.

References

[1] A. A. Ageev and M. I. Sviridenko. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8(3):307–328, 2004.

[2] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing (SICOMP)*, 40(6):1740–1766, 2011.

[3] C. Chekuri, J. Vondrák, and R. Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *Proc., IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 575–584, 2010.

[4] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM*, 53(3):324–360, 2006.