Spring School
Finsterau / Borova Lada
17.5.2000 – 27.5.2000

Outline of talks on

Approximation Algorithms, the Primal Dual Method, and Randomized Rounding

supervised by

Clemens Gröpl
Institut für Informatik
Humboldt-Universität zu Berlin
`groepl@informatik.hu-berlin.de`

[May 10, 2000[1]]

The purpose of this text is to give you an idea about the topics, so that you can decide whether you want to take on a talk, and to guide you through the (first steps of) the preparation of the talk itself.

Please do not hesitate to send email to me at all hours of the day. I will try to reply to you a. s. a. p. If you decide to take on one of the talks supervised by me, then the following rule applies:

You must *send me an email at least* twice *a week.*

The content is not so important. A simple life sign suffices. You can say “I didn’t have the time to do anything”, but then I would like to know about that. The idea is that (1.) you do not start looking into the papers during the bus trip to Finsterau, (2.) you don’t get stuck with something where I can help you with a few short explanations, and (3.) I get an idea about your previous knowledge so that we can re-organize or modify the talks whenever unexpected (by me) difficulties arise.

This text is not meant as a self-contained exposition for each talk. The talks are based on textbook articles and original work. Here I give an outline – which parts to skip and where to focus. (And don’t forget to write email.) The references do not indicate the first place where a result appeared.

I assume familiarity with the $O(\ )$-notation.

Talk 1 is the easiest one. A basic understanding of LP duality is necessary.

Talk 2 can be viewed as an improvement of the LP algorithm from Talk 1. You need more involved probabilistic tools (FKG inequality, large deviation inequalities, pessimistic estimators). This talk should best be divided among two persons.

Talk 3 is essentially two talks and should be split among at least two persons. Linear programming is used to guide an ingenious divide-and-conquer approach.

---

[1] Differences to the 2000-04-20 version only concern the references to the talk about vertex feedback sets.

References

[AA] Dorit S. Hochbaum (ed.): Approximation Algorithms for NP-hard problems. PWS Publishing Company, 1996.

[AS] Noga Alon, Joel Spencer: The probabilistic method. John Wiley & sons, 1992.

[BBF] Vineet Bafna, Piotr Berman, Toshihiro Fujito: Constant ratio approximations of feedback vertex sets in weighted undirected graphs. DIMACS Technical Report 96-29, 1996. Available via [www].

[Fuj99] Toshihiro Fujito: Approximating node-deletion problems for matroidal properties. Journal of Algorithms 31:211-227, 1999.

[Fuj00] Toshihiro Fujito: Approximation algorithms for submodular set cover with applications. IEICE Trans. Inf. & Syst. E83-D(3):480–487, 2000. Available via [www] or http://search.ieice.or.jp/

[CLR] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest: Introduction to Algorithms. MIT Press / McGraw Hill, 1989.

[Chv] Vašek Chvátal: A greedy heuristic for the set-covering problem. Math. Oper. Res., 4:857–869, 1979.

[comp] A compendium of NP optimization problems is
`http://www.nada.kth.se/~viggo/problemlist/compendium.html`
This is usually the first place to look at if you want to know what is known about a certain approximation problem.

[ENRS] Guy Even, Joseph (Seffi) Naor, Satish Rao, Baruch Schieber: Divide-and-conquer approximation algorithms via spreading metrics. Available via [www]. To appear in: Journal of the ACM.

[ENSS] Guy Even, Joseph (Seffi) Naor, Baruch Schieber, Madhu Sudan: Approximating minimum feedback sets and multicuts in directed graphs. Available via [www]. Algorithmica, 20:151–174, 1998.

[GW] Michel X. Goemans, David P. Williamson: The primal-dual method for approximation algorithms and its application to network design problems. Chapter 4 in [AA].

[Hoc] Dorit S. Hochbaum: Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems; Chapter 3 in [AA].

[pcp] Ernst W. Mayr, Hans Jürgen Prömel, Angelika Steger: Lectures on proof verification and approximation algorithms. Springer LNCS 1367, 1998. Available via [www].

[Rag] Prabhakar Raghavan: Probabilistic construction of deterministic algorithms: approximating packing integer programs. Journal of Computer and System Sciences 37:130–143, 1988.

[Sla] Petr Slavík: A tight analysis of the greedy algorithm for set cover. Preprint, 1996. Available via [www]. (A revised version has been published at Journal of Algorithms 25, 1997, pp. 237-254.) You can also find some slides (and czech girls) on his homepage `http://www.cs.buffalo.edu/~slavik/`

[Sri] Aravind Srinivasan: Improved approximation guarantees for packing and covering integer programs. Technical report available via [www]. Appeared in SIAM Journal on Computing, Vol. 29, 648–670, 1999.

[VVa] Vijay V. Vazirani: (Draft of a book on) Approximation Algorithms. Available via [www].

[www] Some postscript files will be available from
`http://www.informatik.hu-berlin.de/~groepl/SpringSchool00/`
(user = `spring`, password = `17.may`)
Notice the copyright.

Please let me know immediately if you are unable to obtain a paper you need for your talk!

1 Set cover

1.1 Abstract

In the set cover problem, we are given a set S and a collection of weighted subsets of S. The problem is to find a minimum weight subcollection of these subsets that cover all elements of S. Set cover is one of the classical NP-hard problems, and hence we are interested in approximation algorithms. This talk will focus of two general approaches to solve the set cover problem in an approximate way: the Greedy algorithm and the LP algorithm.

The Greedy algorithm is a local strategy. In each step, it chooses a set that achieves the best ratio $(\text{weight})/(\text{\# points covered for the first time})$. One can show that the Greedy solution is never more than a factor of $H(k)$ worse than the optimum, where k denotes the maximum size of one of the sets in S. The LP algorithm uses the general framework of linear programming (LP). Set cover has a straightforward formulation as an integer linear program (ILP). Since ILPs cannot be solved in polynomial time in general, we relax the integrality constraints and solve the resulting LP instead. This means that we assign values in $[0, 1]$ to the subsets such that for each point in S, the sum of these values over all subsets in which it is contained is at least 1. Then we round the fractional variables to integer values. One can show that the solution of the LP algorithm is never more the a factor of f worse than the optimum, where f denotes the maximum number of times one of the elements of S is covered by the whole subcollection.

The analysis of the LP algorithm is, of course, based on the duality theory of linear programming. The Greedy algorithm can be analysed in an ad hoc fashion, but LP duality provides additional insight.

1.2 References / Outline

The LP algorithm is due to Hochbaum. A good reference for set cover algorithms is her article [Hoc]. You can find a lot of information about related things and background, but you might as well get lost in it, so I recommend to focus on sections 3.1–3,5. (Maybe it's better to read 3.3 before 3.2.) Although still in draft state, I liked Vijay Vazirani's book [VVa] very much. If you want to fresh up your knowledge on LPs, read [VVa, Ch. 10]. Then the LP algorithm is quickly explained in [VVa, Ch.11].

If time permits, include [VVa, Example 11.3] to show that the analysis of the LP algorithm is tight.

The Greedy algorithm for the weighted set cover problem is due to Chvátal. For an LP-free analysis, see [CLR, Ch. 37.3]. But this should not be your last word. For the LP-based analysis of the Greedy algorithm you can rely upon [VVa, Ch. 12] or [Hoc, Sec. 3.2], whichever you like best.

Concerning the quality of these analyses: A tight, but technically involvced analysis was given by [Sla]. (You should not go into details here! I only gave this reference so that you can tell us what a worst-case example looks like. A family of worst-case examples was already given by Chvátal [Chv].) The deeper reason why Greedy cannot be improved easily becomes visible from duality theory: The error introduced by relaxing the integrality constraints is called integrality gap, and [VVa, Example 12.4] shows that the approximation error of Greedy is due to this integrality gap. (This topic is closely related with the talk on randomized rounding, so you might want to come to an agreement with the other group about your presentation.)

You should also point out that vertex cover is a special case of set cover.

Optional: Greedy for partial covers [Hoc, Sec. 3.9].

Don't forget to have a look at [comp]. Also, [GW] is a good reference if you want to learn more about the primal-dual approach.

2 Randomized Rounding

2.1 Abstract

Set cover has a natural formulation as an integer program (see talk 1), but integer linear programs cannot be solved in polynomial time in general. Thus we relax the integrality constraints and obtain a fractional solution. The LP algorithm applied a simple deterministic rounding procedure to obtain a 0-1 solution. But sometimes we can do much better using randomization. This means that we interpret the fractional values as probabilities and flip a biased coin for each subset. In order to obtain a feasible solution with sufficiently high probability, we need to scale up the fractional values before randomized rounding, so that it becomes more likely to choose a subset. Thus we cannot expect to find an optimal set cover in this way. A simple analysis yields an upper bound of $O(\log n)$ on the performance ratio of this randomized approximation algorithm. Using more elaborate probabilistic tools, one can establish a performance ratio that is superior to the Greedy algorithm in many cases, and never much worse in general.

Actually, things are a bit more complicated than just explained. The probabilistic analysis only shows that the rounding procedure produces a set cover that is not too bad with a certain positive, but very small probability. In order to turn this into an efficient algorithm, we make the randomized rounding deterministic as follows. For each variable in turn, we decide whether to round it up or down. We know that for one of the choices, the probability to get a feasible set cover does not decrease (assuming that the remaining variables are still random). The precise probabilities are hard to compute, but we can use a so-called pessimistic estimator instead. Basically, a pessimistic estimator is defined to be just what we need to carry out this de-randomization procedure (that is, the “method of conditional probabilities”).

2.2 References / Outline

Although the paper [Sri] deals with a larger class of problems, you should strictly focus on set cover, at least for the proofs. You should put the results in a wider perspective, though, but in an informal way.

This talk has tree parts. Probably this is all to much for a single talk, so you should split it up (if you are two persons) or skip some, but not all the details (if you are one person).

In the first part, you should explain the basic idea (which is very simple) along the lines of [VVa, Ch. 11]. (Don’t forget to show that “$\mathcal{C}$ is minimized when each of the p_i’s is $1/k$”.) You might also consult [AA, Ch. 11.2.2]. This yields an $O(\log n)$ approximation with probability $1/2$.

In the second part, you should explain how to make use of the fact that the ‘bad’ events (a point remains uncovered) are positively correlated, that is, if you know that one bad event has happened, then another bad event is even more likely to happen (just as in real life). Here is where the FKG inequality [AS, Ch. 6] comes in. You need not prove the FKG inequality, but you should understand how it leads to [Sri, Thm. 2], even if you do not go into the details in your talk. And don’t forget to start with giving us an intuitive feeling why the result should be true. For large deviation inequalities, see the spring school text on the probabilistic method by Matoušek and Vondrák, or [AS, Appendix A].

In the third part, you should explain the application of pessimistic estimators and why they are necessary to get an efficient algorithm. (They weren’t necessary for the simple algorithm from the first part.) The method of conditional probabilities is explained in [AS, Ch. 15.1]. Pessimistic estimators were introduced by Raghavan in [Rag]. It’s a bit technical, but the idea behind it is simple. See Sieling [pcp, Ch. 3] for another introduction to derandomization.

3 Vertex / Edge Feedback Set

3.1 Abstract

Let G be a (weighted) directed or undirected graph. We can always assume that G is connected. A feedback set is a set of vertices or edges whose removal destroys every cycle in G. The feedback set problem asks for a feedback set of minimum weight. Depending on whether we consider undirected or dirrected graphs and feedback sets of vertices or edges, we obtain modifications of the problem with varying degrees of difficulty.

Every inclusion-minimal edge feedback set in a connected undirected graph is the complement of a spanning tree. Using this trivial observation, one can devise an algorithms that finds an optimal edge feedback set very efficiently.

The vertex feedback set problem in undirected graphs is more difficult. A simple approximation-preserving reduction shows that it is at least as difficult as the vertex cover problem, for which the best known approximation algorithms (see e. g. talk 1) achieve a ratio of $2 - |o(1)|$. The algorithm of Bafna, Berman, and Fujito finds a vertex feedback set whose weight is at most twice the optimal. There is an interesting connection with a generalization of the set cover problem to polymatroids, called submodular set cover (don't be afraid of names).

In directed graphs, simple reductions show that the edges and vertex versions are equivalent. This problem seems to be much harder. The algorithm of Even, Naor, Schieber, and Sudan (based upon work of Seymour) uses linear programming to guide a divide-and-conquer approach and was the first one to break a 'log-square' barrier for the approximation ratio.

3.2 References / Outline

This 'talk' has two parts, and each part has enough material enough for a whole talk.

The vertex feedback set problem in undirected graphs is discussed by Hochbaum in [AA, Ch. 9.2.1]. There you also find the reduction from vertex cover to vertex feedback set in undirected graphs [AA, Ch. 9.2.1.1]. Originally the algorithm is due to Bafna, Berman, and Fujito [BBF], but in your presentation you should follow the recent papers of Fujito [Fuj99, Fuj00], which deal with the vertex feedback set problem in the more general framework of node deletion problems (find a minimal set to delete from a graph to make sure it has a certain property – e.g. circuit free for VFS). See [Fuj00, Sec. 4.2] for a brief proof sketch and [Fuj99] for details. Don't deal with "uniformly k-sparse" for $k > -1$. Even for $k = -1$ ("circuit free"), [Fuj99, Lemma 6] seems a bit technical to prove. Just explain the Dual Greedy Algorithm for VFS and prove its performance ratio. The so-called reverse-delete step is used to ensure that the final solution is inclusion-minimal. This is a standard technique and explained in [GW, especially Thm. 4.1]. (If you find it necessary to go back to [BBF], you should definitely skip [BBF, Sec. 4]. For the local-ratio theorem of Bar-Yehuda and Even, see also [Hoc, Ch. 3.7.4].)

The edge feedback set problem is discussed by Shmoys in [AA, Ch. 5.4]. You should mention its motivation from electrical engineering. For the equivalence of edge and vertex versions in directed graphs see [AA, Intro. Ch. 9.2.1] or [ENSS, Sec. 2]. You should follow Schmoys's outline and start by showing how a good partitioning technique leads to a good approximation ratio, and after that explain the sphere growing technique used for finding such a partition. You may mention the subset feedback vertex / edge problems, but I guess you will not have much time to go into details on that. The topics of the second part have the numbers 1. and 3. on [ENSS, p. 4]. They also give a fast approximate algorithm to solve the LPs which are used in their algorithm [ENSS, Sec. 5], but that seems to be too technical for a presentation.