

Course name: Approximation and online algorithms

LTP 3-0-0, 3 credits

Objective

It is well-known that most combinatorial and geometric optimization problems arising out of real world scenarios turn out to be NP-hard. From computational complexity theory, we know that it is impossible to design efficient algorithms for solving such problems exactly unless $P=NP$, which is very unlikely to be the case. This does not obviate the need for designing efficient polynomial time algorithms for these problems. However, we may relax the stringent requirement of computing an exact optimal solution. In practice, it serves our purpose well enough if we can obtain a near optimal solution instead of an exact optimal, especially when the former can be computed much faster. This results in the notion of the “approximate” solution of an optimization problem. As a computer scientist and engineer, it is often worthwhile to focus on designing efficient algorithms that output an approximate solution, but at the same time provide a relative performance guarantee that the value of the solution is within some factor (this factor is called the “approximation ratio”) of the optimal value.

In the same spirit of approximation ratios, the performance of an online algorithm is measured in terms of “competitive ratios”. Many algorithmic problems that arise in practice are of online nature. In these problems, the input is only partially available because some relevant input arrives only in future, and is not available at the time of making certain decisions. An online algorithm must generate output without the knowledge of the entire input.

Contents and Syllabus

Syllabus for proposed elective “Approximation and online algorithms”

Prerequisites CS31005 (Algorithms II) and CS21001 (Discrete Structures).

Approximation algorithms

Introduction and performance measures.

Greedy Algorithms and Local Search: graph algorithms, job scheduling algorithms, set cover, combinatorial problems such as bin packing and knapsack etc., and geometric problems.

Rounding Data and Dynamic Programming: Approximation Schemes for knapsack, binpacking and scheduling problems.

Deterministic Rounding of Linear Programs: Scheduling, facility location and optimization problems such as bin packing.

The Primal-dual Method: Set cover, vertex cover and weighted vertex cover, etc.

Online algorithms

Competitive analysis of online algorithms; The Paging Problem; Amortized analysis of online algorithms; The List Update Problem; Problems of searching for a target in a bounded or an unbounded region; Online graph coloring; the K-server problem; online scheduling algorithms.

Approximation algorithms: Lecture-wise schedule: A sample

[27 hours]

Approximation algorithms

1. Introduction and performance measures [1 hour]
2. Greedy Algorithms and Local Search: [13 hours]

(a) graph algorithms [4 hrs] (b) job scheduling algorithms [2 hours] (c) set cover [1 hour] (d) combinatorial problems such as bin packing and knapsack etc. [4 hours] and (e) geometric problems [2 hours].

3. Rounding Data and Dynamic Programming: [4 hours] Approximation Schemes for knapsack, binpacking and scheduling problems.

4. Deterministic Rounding of Linear Programs: [5 hours]
Scheduling, facility location and optimization problems such as bin packing.

5. The Primal-dual Method: [4 hours]
Set cover, vertex cover and weighted vertex cover, etc.

Online algorithms

[13 hours]

1. Competitive analysis of online algorithms [1 hour]
 2. The Paging Problem [1 hour]
 3. Amortized analysis of online algorithms [1+1/2 hour]
 4. The List Update Problem [1 hour]
 5. Problem of searching for a target in a bounded or unbounded region. [2+1/2 hours]
 6. Online graph coloring [2 hours]
 7. The K-server problem [2 hours]
 8. Online scheduling algorithms [2 hours]
- Total: 40 hours.

Textbooks

1. V. Vazirani, Approximation Algorithms, Springer, 2003.
2. D. P. Williamson and D. B. Shmoys, The Design of Approximation Algorithms, Cambridge, 2011.
3. A. Fiat and G. Woeginger, Online Algorithms, Lecture Notes in Computer Science, no. 1442, Springer, 1998 (Edited Volume).

Reference texts

1. M. R. Garey and D. S. Johnson, Computers and Intractability: A guide to the theory of NP-completeness, W. H. Freeman, 1979.
2. R. Motwani, Lecture Notes on Approximation Algorithms, Volume 1, No. STAN-CS-92-1435, Stanford University, 1992
3. S. K. Ghosh, Approximation algorithms for art gallery problems in polygons and terrains, (Survey Paper), Lecture Notes in Computer Science, No. 5942, pp. 21-34, Springer, 2010.
4. S. Albers, Competitive Online Algorithms, Lecture notes, Max Plank Institute, Saarbrucken, 1996.
5. S. K. Ghosh and R. Klein, Online algorithms for searching and exploration in the plane, (Survey Paper), Computer Science Review, vol. 4, pp. 189-201, 2010.