# SEMESTER S7

## APPROXIMATION ALGORITHMS

| | | | |
|---|---|---|---|
| **Course Code** | **PECST749** | **CIE Marks** | 40 |
| **Teaching Hours/Week (L: T:P: R)** | 3:0:0:0 | **ESE Marks** | 60 |
| **Credits** | 3 | **Exam Hours** | 2 Hrs. 30 Min. |
| **Prerequisites (if any)** | **-** | **Course Type** | Theory |

**Course Objectives:**

1. To provide a deep understanding of approximation algorithms, including their design, analysis, and application to various optimization problems.
2. To equip the skills to evaluate and analyze the efficiency and effectiveness of approximation techniques. This includes understanding performance metrics, approximation ratios, and the theoretical limits of approximation algorithms, as well as applying these techniques to complex problems in network design, combinatorial optimization, and other areas.

## SYLLABUS

| Module No. | Syllabus Description | Contact Hours |
|---|---|---|
| 1 | Basics of Approximation Algorithms - Introduction to approximation algorithms, Performance guarantees: approximation ratio and factor, Examples of approximation problems. (Chapter 1)<br>Greedy Algorithms - Introduction to greedy algorithms, Set cover problem, Vertex cover problem. (Chapter 2)<br>Local Search Algorithms - Local search techniques, k-Median and k-Center problems, Analysis of local search algorithms. (Chapter 3) | **9** |
| 2 | Linear Programming Relaxation - Introduction to linear programming (LP), LP relaxation of combinatorial problems, Primal-dual method. (Chapter 4)<br>Rounding Techniques - Randomized rounding, Deterministic rounding, Applications to various problems. (Chapter 5)<br>Integer Programming and Cutting Planes - Integer programming formulation, Cutting plane methods, Applications in network design. (Chapter 6) | **9** |
| 3 | Semi-Definite Programming - Introduction to semi-definite programming (SDP), Goemans-Williamson algorithm for MAX-CUT, Other applications of SDP. (Chapter 8)<br>Approximation Schemes - Polynomial-time approximation schemes (PTAS), | **9** |

| | | |
|---|---|---|
| | Fully polynomial-time approximation schemes (FPTAS), Examples: knapsack problem, Euclidean TSP. (Chapter 9) | |
| **4** | Inapproximability Results - Introduction to inapproximability, Reductions and hardness of approximation, PCP theorem and its implications. (Chapter 10)<br>Network Design Problems - Steiner tree problem, Traveling Salesman Problem (TSP), Multicommodity flow problem. (Chapter 7) | **9** |

**Course Assessment Method**
**(CIE: 40 marks, ESE: 60 marks)**

**Continuous Internal Evaluation Marks (CIE):**

| Attendance | Assignment/ Microproject | Internal Examination-1 (Written) | Internal Examination- 2 (Written) | Total |
|---|---|---|---|---|
| 5 | 15 | 10 | 10 | 40 |

**End Semester Examination Marks (ESE)**

*In Part A, all questions need to be answered and in Part B, each student can choose any one full question out of two questions*

| Part A | Part B | Total |
|---|---|---|
| • 2 Questions from each module.<br>• Total of 8 Questions, each carrying 3 marks<br><br>**(8x3 =24marks)** | • Each question carries 9 marks.<br>• Two questions will be given from each module, out of which 1 question should be answered.<br>• Each question can have a maximum of 3 sub divisions.<br>**(4x9 = 36 marks)** | **60** |

## Course Outcomes (COs)

At the end of the course students should be able to:

| Course Outcome | | Bloom's Knowledge Level (KL) |
|---|---|---|
| **CO1** | Demonstrate a foundational understanding of approximation algorithms, including performance guarantees, approximation ratios, and common examples of approximation problems. | **K3** |
| **CO2** | Illustrate the principles of greedy algorithms and apply them to solve classic problems such as the set cover and vertex cover problems, understanding their efficiency and limitations. | **K3** |
| **CO3** | Show proficiency in local search algorithms and linear programming relaxation methods, including the primal-dual method, and apply these techniques to solve combinatorial optimization problems. | **K3** |
| **CO4** | Understand and implement rounding techniques, both randomized and deterministic, and learn the basics of semi-definite programming (SDP), including algorithms like Goemans-Williamson for the MAX-CUT problem. | **K3** |
| **CO5** | Demonstrate polynomial-time approximation schemes (PTAS) and fully polynomial-time approximation schemes (FPTAS), and explore inapproximability results, including reductions, hardness of approximation, and the PCP theorem. | **K3** |

Note*: K1- Remember, K2- Understand, K3- Apply, K4- Analyse, K5- Evaluate, K6- Create*

## CO-PO Mapping Table (Mapping of Course Outcomes to Program Outcomes)

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 3 | 3 | 3 | | | | | | | | 2 |
| **CO2** | 3 | 3 | 3 | 3 | | | | | | | | 2 |
| **CO3** | 3 | 3 | 3 | 3 | | | | | | | | 2 |
| **CO4** | 3 | 3 | 3 | 3 | | | | | | | | 2 |
| **CO5** | 3 | 3 | 3 | 3 | | | | | | | | 2 |

Note: *1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High), -: No Correlation*

| Text Books | | | | |
|---|---|---|---|---|
| **Sl. No** | **Title of the Book** | **Name of the Author/s** | **Name of the Publisher** | **Edition and Year** |
| 1 | Approximation Algorithms | Vijay V. Vazirani | Springer Nature (SIE) | 2/e, 2013 |

| Reference Books | | | | |
|---|---|---|---|---|
| **Sl. No** | **Title of the Book** | **Name of the Author/s** | **Name of the Publisher** | **Edition and Year** |
| 1 | The design of approximation algorithms | David Williamson and David Shmoys | Cambridge University Press | 1/e, 2011 |
| 2 | Randomized Algorithms | Rajeev Motwani and Prabhakar Raghavan | Cambridge University Press | 1/e, 2004 |
| 3 | Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis | Michael Mitzenmacher and Eli Upfal | Cambridge University Press | 3/e, 2017 |
| 4 | Introduction to Algorithms | Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein | The MIT Press | 4/e, 2023 |
| 5 | The Probabilistic Method | Noga Alon and Joel H. Spencer | Wiley-Blackwell | 4/e, 2016 |
| 6 | Computational Complexity: A Modern Approach | Sanjeev Arora and Boaz Barak | Cambridge University Press | 1/e, 2019 |

| Video Links (NPTEL, SWAYAM…) | |
|---|---|
| **Module No.** | **Link ID** |
| **1** | https://nptel.ac.in/courses/106105471 |
| **2** | https://nptel.ac.in/courses/106105471 |
| **3** | https://nptel.ac.in/courses/106105471 |
| **4** | https://nptel.ac.in/courses/106105471 |

# APPROXIMATION ALGORITHM


**PROF. PALASH DEY**
Department of Computer Science and Engineering
IIT Kharagpur


**PRE-REQUISITES :** Knowledge of algorithm design principles.

**INTENDED AUDIENCE :** Under-graduate and post-graduate students and faculty members.

**INDUSTRY SUPPORT :** All software companies especially Google, Microsoft, etc.

## COURSE OUTLINE :

Many real-world problems are NP-complete. Hence, they are unlikely to admit a polynomial-time algorithm. In this course, we will study various techniques to design efficient algorithms to compute an approximately optimal solutions.

## ABOUT INSTRUCTOR :

Prof. Palash Dey is an Assistant Professor in the Department of Computer Science and Engineering at Indian Institute of Technology, Kharagpur since 2018. Before joining IIT Kharagpur, he was a post-doctoral INSPIRE faculty in TIFR, Mumbai for one year. He finished his Ph.D. and M.E. from the Department of Computer Science and Automation at Indian Institute of Science, Bangalore in 2017 and 2013 respectively. Prior to that, he finished his B.E. from the Department of Computer Science and Engineering at Jadavpur University in 2010. His primary field of research is algorithmic game theory. He is broadly interested in theoretical computer science.

## COURSE PLAN :

**Week 1:** Review of NP-Completeness
**Week 2:** Approximation algorithm for the vertex cover, set cover, and traveling salesman problem
**Week 3:** Approximation algorithm for the Knapsack problem and the job scheduling problems
**Week 4:** Basics of linear programming
**Week 5:** Deterministic rounding: set cover, vertex cover problems
**Week 6:** Deterministic rounding: steiner tree, facility location problems
**Week 7:** Randomized rounding: max-SAT problem
**Week 8:** Randomized rounding: steiner tree, facility location problems
**Week 9:** Primal-dual method: set cover
**Week 10:** Primal-dual method: steiner tree
**Week 11:** Semidefinite programming: max cut
**Week 12:** Hardness of approximation: travelling salesman problem, job scheduling