Special Topics in Transportation Engineering: Transportation Network Optimization

Spring 2024

Course number	CVL851	**No. of credits**	3 credits (3-0-0)
Meeting time:	Slot A (M, Th at 8-9:30A.M.)	**Instructor's name:**	Pramesh Kumar
Email:	pkk@iitd.ac.in	**Instructor's office:**	322, Block-IV
Location:	LH620	**Office Hours:**	M 2:30-3:30 P.M.
Teaching assistant:	-		322, Block-IV

Course website: Access the course material from https://prameshk.github.io/CVL851.html. Submit homework assignments and project deliverable through Moodle.

Course prerequisites: Student should have good knowledge of calculus and linear algebra. Computer programming is required to do homeworks.

Course description: This course is designed to introduce optimization techniques for solving transportation network optimization problems. Topics include introduction to network flow problems, linear programming, simplex method, duality theory, dynamic programming, and integer programming.

Student learning aims/outcomes:

- Modeling transportation network flow problems.
- Solving standard network flow problems
- Formulating and solving linear programs and applying duality theory.
- Formulating integer programming problems
- Using computer programming packages available for optimization

Reference Books:

1. Vanderbei, Robert J., *Linear Pogramming: Foundations and Extensions*, Springer Fifth Edition, 2020. [Link]
2. Ahuja, R., Magnanti, T., and Orlin, J., *Network Flows: Theory, Algorithms, and Applications*, Pearson, 2014. [Free PDF copy].
3. Boyd, Stephen P., and Lieven Vandenberghe. *Convex optimization*, Cambridge university press, 2004. [Free PDF copy]
4. Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to algorithms*, MIT press, 2022. [Link]

Tentative topics to be covered (may not cover all the topics)

1. **Introduction**
   - Introduction to the course
   - Graphs and Network Flow Problems

   - Data structures
   - Computational complexity

2. **Introduction to Linear Programming**
   - Modeling linear programming problems
   - Simplex method
   - Farkas Lemma and LP Duality theory
3. **Introduction to Dynamic Programming**
4. **Network flow problems**
   - Graph search algorithms
   - Minimum spanning tree
   - Shortest path
   - Max-flow and min-cut
   - Min-cost algorithms
   - Matching
5. **Introduction to Integer Programming**
   - Modeling IP problems
   - Totally unimodularity
   - Branch-and-bound method

Homework assignments: You are encouraged (but not required) to use LaTeXto typeset your assignments. Please refer to this website to learn LaTeX. Late submission of assignment will be allowed up to two days after the deadline. For each day, there will be a penalty of 25% deduction in points.

Grading and other class policy: The following is the breakdown for grading:

In-class exercises (10%)
Homework assignments (20%)
Minor exam (15%)
Major exam (25%)
Project (30%)

The major (final) exam will be cumulative, i.e., it will cover all the material taught during the semester. Participation in all components of this course is required to pass the course. Other policies are as follows:

- **Letter grades**: For the description of the letter grades and their cut offs, please refer to this link.
- **Attendance**: If a student's attendance is less than 75%, institute rules will apply. Please refer to Courses of Study.
- **Auditing the course**: If a student is auditing the course, then she has to get at least 30% of the total marks (aggregated) and more than or equal to 75% attendance for obtaining audit pass **NP** grade. Otherwise, the student will be awarded **NF** grade.
- **Re-grade requests**: Requests for re-grading questions on an assignment/exam will be considered if submitted in writing within one week from the time the work is returned in class. Note that the score may change in either direction as a result of a re-grade. The instructor reserves the right to limit the number and scope of re-grades requested by a student.

- **Make-up exams:** Make-up exams can be arranged as per the institute rules.
- **Academic integrity:** All activities in this course must be done independently unless taken permission from the instructor. While solving the problems, you may discuss it your peers but the final answer must be your own. Copying from another student or plagiarizing from other sources will be considered cheating. You may be awarded a Fail **F** grade for academic dishonesty. The case will also be forwarded to student advisor and Dean, Academics. For more information about the honor code, refer to Courses of study.
- **Resources for differently-abled students:** If you require assistance in this regard, please refer and contact Office of accessible education.

Tentative Schedule

Monday	**Thursday**
Jan 1: Intro to the course and graph theory	Jan 4: Intro to graph theory and networkx
Jan 8: Computational complexity	Jan 11: Computational complexity
Jan 15: Data structures	Jan 18: Optimization modeling
Jan 22*: No class: Govt declared holiday	Jan 25: Optimization modeling
Jan 29: Optimization modeling	Feb 1: Optimization modeling
Feb 5: Simplex method	Feb 8: Simplex method/LP duality
Feb 12: LP duality	Feb 15: Dynamic Programming Revision
Feb 19*: No class: Minor exam	**Feb 19***: No class: Institute notification
Feb 26*: No class: Minor exam	**Feb 29***: No class: Minor exam
Mar 4*: No class: Friday Timetable	Mar 7: Dynamic Programming/Minor exam discussion
Mar 11: Graph search	Mar 14: Shortest path
Mar 14: Shortest path	Mar 21: Minimum spanning tree
Mar 25*: No class: Holi	**Mar 28***: No class: Semester break
Apr 1: Max flow min cut	Apr 4: Max flow min cut/Min cost flow
Apr 6: Min cost flow	
Apr 8: Min cost flow/Bipartite matching	**Apr 11***: Idu'l Fitr
Apr 13: Bipartite matching	
Apr 15: Intro to IP/Branch-and-bound	**Apr 18***: On leave
Apr 22: Intro to IP	Apr 25: Final class and Thank you!
Apr 26*: Project presentations	**Apr 29***: Final exam

This is the end of the text