# Math 466/566: Network Optimization (Fall 2026)

| | |
|---|---|
| Crosslisting | CPT_S 456/556 |
| Time | Tue, Thu 2:55–4:10 PM |
| Location | Pullman: CUE 318; Vancouver: VECS 125 |
| Instructor | Bala Krishnamoorthy |
| Office | VLIB 210P; Zoom |
| Check-In Hours | Wed, Thu 10:30–11:30 AM |
| Email | kbala@wsu.edu |
| Course web page | http://bala-krishnamoorthy.github.io/Math566.html |
| Book | Ahuja, Magnanti, and Orlin —<br>Network Flows: Theory, Algorithms, and Applications.<br>Prentice-Hall. ISBN: 013617549X |

## Description of the Course

Welcome to Network Optimization! Network flow problems constitute an important class of optimization problems with applications to several areas including big data analytics, chemistry, computer networking, engineering, public policy, scheduling, telecommunications, transportation, and more. We will present an integrated view of the theory, algorithms, and applications of key network optimization problems including the shortest path, maximum flow, minimum cost flow, minimum spanning tree, and matching problems. We will present most arguments from first principles, and we will adopt a network (or graph) view point. We will use linear optimization approaches on some occasions, and will provide the necessary background so as to keep the discussion self-contained. **Previous knowledge of linear optimization (at the level of Math364 in particular) will not be required. The only (pre)requirement is mathematical sophistication and intuition.** We will emphasize powerful algorithm strategies, rigorous analysis of the algorithms, and data structures for their implementation. Apart from problems involving proofs (in homework and the midterm exam), the student will produce working implementations of some of the algorithms using Matlab or Python (or another package/language). Depending on student interest, we will discuss specific applications and topics from data sciences and/or combinatorial optimization.

## Organization and Grading

There will be around eleven homework assignments. Each assignment will be posted at least one week before the day on which it will be due. Dates are given in the tentative schedule (they may be subject to some changes). Some of the problems will involve writing programs in a computer language/package (Matlab, by default) to produce running versions of algorithms discussed in class. **All homework and project submissions should be made electronically by email.**

There will be one in-class, closed-book, midterm exam on **Thursday, October 8**. Students are encouraged to not miss the exam. Requests for taking a **make-up exam** at an alternative time could be considered under extreme circumstances. But such a make-up exam will be different from the one taken by the rest of the class at the regular time.

There will be a final exam that will given as a take-home exam with access to limited resources. The total score for the course will be calculated using the following weights:

- homework: 60 %
- mid-term: 20 %
- final: 20 %

The least homework grade **from among those turned in** will be dropped. In each homework and in the exam, some problems will be marked as "advanced", indicated by **[G]**. Students registered for Math 466 will not be responsible for solving these problems. They are strongly encouraged to attempt the advanced problems, though. Points gained on these problems by Math 466 students will be counted as extra credit.
The total score will be *rounded up* to the first decimal place, e.g., $87.42 \to 87.5$, $79.91 \to 80$.
The **grade for the course** will be determined by the total score based on the following scales.
**Math 466:** $\geq$92: A; 88–91.9: A$-$; 83–87.9: B$+$; 78–82.9: B; 73–77.9: B$-$; 68–72.9: C$+$; 63–67.9: C; 58–62.9: C$-$; 53–57.9: D$+$; 45–52.9: D; and $\leq$44.9: F.
**Math 566:** $\geq$92: A; 86–91.9: A$-$; 80–85.9: B$+$; 73–79.9: B; 66–72.9: B$-$; 59–65.9: C$+$; 53–58.9: C; 48–52.9: C$-$; 43–47.9: D$+$; 38–42.9: D; and $\leq$37.9: F.

**Expectations for student effort:** For each hour of lecture equivalent, students should expect to have a minimum of two hours of work outside class. Given the conjoint setting for this class, this expectation applies to students in both Math 466 and Math 566. Also see the note about expected effort required for the programming assignments below (under Software).

**Attendance Policy:** I will not be recording your attendance on a regular basis. At the same time, it is essential to not miss the lectures if you want to do well in this course. As such, you are encouraged to attend class regularly.

## Software

Students will be asked to implement several of the algorithms discussed in class. This implementation/programming part will be an integral part of the learning process in this course.

Programming assignments could be done using Matlab (or another package or language of your choice, e.g., Python). WSU has a systemwide Matlab license, using which students could get Matlab on their PC (see the Matlab Portal for more info). You will not need powerful computers—an ordinary laptop or desktop could be used to run Matlab.

Students should expect to spend a nontrivial amount of time on programming assignments, especially if they are doing such programming for the first time. It is highly recommended that each student installs Matlab and gets familiar with basic commands early in the semester.

## Academic Integrity

I encourage discussion of homework problems with others. But each student should submit their own (hand or type) written solutions and/or computer programs. You might search the internet for finding materials to enhance your understanding. If you use such material to assist in your homework submission, you **should** cite the relevant sources. Plagiarism or cheating will **not** be tolerated. In particular, do not copy blindly from internet sources! Such behavior is easy to detect, and will result in a zero grade for the item in question and possibly a failing grade for the entire course.

**AI Use Permitted with Acknowledgment:** You are welcome to try AI tools based on generative models such as Claude, Chat-GPT, Gemini, and others. Any such use in your homework assignments or projects should be properly acknowledged in your submission. One way to list such an acknowledgment is as follows:
*"This response is based on the solution provided by* ChatGPT *when queried with* the exact text of the problem statement *on* Sep 12, 2026*."* (change the entries in red appropriately).

At the same time, be warned that most of these tools are not guaranteed to give you the correct response each time! Also, you will not be able to rely on them for the midterm and final exams.

## University Syllabus

**Students are responsible for reading and understanding all university-wide policies and resources pertaining to all courses (for instance: accommodations, care resources, policies on discrimination or harassment), which can be found in the University Syllabus.**

### Student Learning Outcomes (SLOs) and Assessment

**Quantitative skills:** A student completing this course should be able to do the following tasks.

- Formulate problems in application domains as appropriate network flow problems;
- employ existing, or devise new efficient, algorithms to solve the network flow problems;
- prove results on the finiteness and efficiency of these algorithms; and
- produce working implementations of several algorithms in a software package.

These skills will be evaluated through the student's responses to problems—both proof-type as well as computational ones involving software programming—in homework assignments as well as the midterm and final exams. Implementations of algorithms will be assessed also as part of selected homework assignments.

**Written communication skills:** A student completing this course should be able to

- provide sound mathematical arguments including proofs,
- justify choices of network models used for problems in a logically clear manner, and
- write a cohesive report for the final project in the form of a short research paper.

These skills will be evaluated through the student's responses to problems in homework assignments as well as the midterm and final exams.

# Tentative Schedule for Math 466/566 (Fall 2026)

| Week | Lec # | Date | Details |
|---|---|---|---|
| 1 | 1 | Tue, Aug 25 | syllabus, logistics, intro to network models |
| | 2 | Thu, Aug 27 | various network optimization problems |
| 2 | 3 | Tue, Sep 1 | network representations, data structures |
| | 4 | Thu, Sep 3 | network representations, data structures **[HW 1 Due]** |
| 3 | 5 | Tue, Sep 8 | transformations and flow decompositions |
| | 6 | Thu, Sep 10 | computational complexity, search algorithms **[HW 2 Due]** |
| 4 | 7 | Tue, Sep 15 | breadth- and depth-first search, intro to shortest paths |
| | 8 | Thu, Sep 17 | shortest paths (SP): label setting algos **[HW 3 Due]** |
| 5 | 9 | Tue, Sep 22 | label setting algos, label correcting algos |
| | 10 | Thu, Sep 24 | complexity, implementation of SP algos **[HW 4 Due]** |
| 6 | 11 | Tue, Sep 29 | more on SP algos |
| | 12 | Thu, Oct 1 | all-pairs shortest paths **[HW 5 Due]** |
| 7 | 13 | Tue, Oct 6 | review for mid-term |
| | 14 | Thu, Oct 8 | **MIDTERM** |
| 8 | 15 | Tue, Oct 13 | maximum flow, augmenting paths (AP) |
| | 16 | Thu, Oct 15 | max-flow min-cut (MFMC), shortest AP (SAP) algo **[HW 6 Due]** |
| 9 | 17 | Tue, Oct 20 | more on SAP, excess scaling algo |
| | 18 | Thu, Oct 22 | preflow-push algorithms **[HW 7 Due]** |
| 10 | 19 | Tue, Oct 27 | more on preflow-push algos |
| | 20 | Thu, Oct 29 | min-cost flows (MCF), residual network for MCF **[HW 8 Due]** |
| 11 | 21 | Tue, Nov 3 | MCF optimality conditions |
| | 22 | Thu, Nov 5 | cycle canceling algo for MCF **[HW 9 Due]** |
| 12 | 23 | Tue, Nov 10 | successive SP algo, complementary slackness |
| | 24 | Thu, Nov 12 | minimum spanning trees (MST) **[HW 10 Due]** |
| 13 | 25 | Tue, Nov 17 | Kruskal's and Prim's algo for MST |
| | 26 | Thu, Nov 19 | matching problems, assignment problems |
| | | Tue, Nov 24 | *No class: thanksgiving* |
| | | Thu, Nov 26 | *No class: thanksgiving* |
| 14 | 27 | Tue, Dec 1 | stable marriage problem, non-bipartite matching |
| | 28 | Thu, Dec 3 | augmenting paths, more on matching **[HW 11 Due]** |
| 15 | 29 | Tue, Dec 8 | combinatorial optimization/other topics |
| | 30 | Thu, Dec 10 | combinatorial optimization/other topics |
| 16 | | Tue, Dec 10 | **[Final Exam] by 10:00 PM.** |