CSCI 699: Topics in Discrete Optimization 
and Learning

Units: 4
Spring 2020 – Thursday – 2:00-5:20

Thursday 2:00- 5:20pm
Location: TBD

Instructors: Bistra Dilkina
Office: SAL 304
Office Hours: Wednesday 11am-12 or By Appointment

Contact Info: dilkina@usc.edu
D Clearance Form: https://forms.gle/wihPhqn8TbVaDCv4A
Teaching Assistant: TBD

Course Description
This course will examine recent research that leverages synergies between machine learning and discrete optimization, including 1) leveraging machine learning in computational methods for solving NP-hard discrete optimization problems, 2) learning in decision-focused way, and 3) leveraging combinatorial algorithmic ideas for machine learning tasks. The course will introduce students to computational discrete optimization and in particular integer programming, branch and bound, SAT, local search, submodular optimization and empirical evaluation of algorithms, as well as applications of machine learning methods, such as deep learning, graph convolutional networks and reinforcement learning. The course will study in depth recent papers focusing on a data-driven algorithm design for combinatorial problems, where ML techniques such graph convolutional network, reinforcement learning, and deep learning are used to improve existing methods or design new ones altogether. The second part will focus on the combinatorial approaches in the context of machine learning, such as binarized NN, optimal decision trees, interpretability, and adversarial examples. This class is targeted at PhD students. Background in ML is expected. Mathematical maturity, as well as research experience in computer science and/or data science is strongly recommended.

Learning Objectives
1. Learn solution techniques for discrete optimization
2. Learn how to empirically evaluate computational performance of solvers.
3. Learn how to use different ML techniques in the context of discrete optimization
4. Learn how to critically read a technical research paper on this topic.

Prerequisite(s): sufficient mathematical background; some background in AI, machine learning and discrete optimization, with research experience in at least one, is strongly recommended; good programming skills

Recommended Preparation: CS Algorithms class (preferably graduate level) is strongly advisable, ML and optimization course work or background is advisable, not mandatory
Course Notes
Lecture slides/notes will be available online after class.

Required Readings and Supplementary Materials
The course will not have any official textbook but will instead use assigned papers and book chapters reading. This is a preliminary list of the reading list:

Part 1: ML for Discrete Optimization

Graph Optimization problems (and RL)
- Learning heuristics for the TSP by policy gradient, Deudon, M., Cournut, P., Lacoste, A., Adulyasak, Y., & Rousseau, L. M. CPAIOR 2019
- Neural Large Neighborhood Search for the Capacitated Vehicle Routing Problem. Arxiv 2019

Attention

Integer Programming

MIP Branching:
- Learning to branch in mixed integer programming, E. B. Khalil, P. Le Bodic, L. Song, G. Nemhauser, B. Dilkina, AAAI 2016
- Learning to Branch, M-F Balcan, T Dick, Tuomas Sandholm, E Vitercik, ICML 2018
- Exact Combinatorial Optimization with Graph Convolutional Neural Networks. Maxime Gasse, Didier Chetelat, Nicola Ferroni, Laurent Charlin, Andrea Lodi. ArXIV 2019. (code)
- Neural Network Branching for Neural Network Verification. J Lu, M. Pawan Kumar. ArXiV 2019

MIP Node Selection:
- **Learning to Search via Retrospective Imitation**, J Song, R Lanka, A Zhao, Yisong Yue, M Ono. *ArXiv*, 2018

**Other MIP topics:**
- Learning to Solve Large-Scale Security-Constrained Unit Commitment Problems. Alinson S. Xavier, Feng Qiu, Shabbir Ahmed. *ArXiv* 2019
- Accelerating Primal Solution Findings for Mixed Integer Programs Based on Solution Prediction. Jian-Ya Ding, Chao Zhang, Lei Shen, Shengyin Li, Bing Wang, Yinghui Xu, Le Song. *ArXiv*, 2019

**SAT and CSP**
- **Learning Robust Search Strategies Using a Bandit-Based Approach**, Wei Xia, Roland H. C. Yap, *AAAI* 2018
- **Learning a SAT Solver from Single-Bit Supervision**, Daniel Selsam, Matthew Lamm, Benedikt Bunz, Percy Liang, Leonardo de Moura, David L. Dill. *ICLR* 2019

**RL for Discrete Optimization tasks**
- **Improving optimization bounds using machine learning: decision diagrams meet deep reinforcement learning**, Cappart, Q., Goutiere, E., Bergman, D., & Rousseau, L. M. *AAAI* 2019
- **Learning to Perform Local Rewriting for Combinatorial Optimization**, Xinyun Chen and Yuandong Tian. *NeurIPS* 2019
- **Causal Discovery with Reinforcement Learning**, Zhu S., Ng I., Chen Z., *ICLR* 2020

**PART 2: Decision-focused Learning**
PART 3: Discrete approaches in ML

- Training Binarized Neural Networks Using MIP and CP. RT Icarte, L Illanes, MP Castro, AA Cire, SA Mcclraith, JC Beck. CP 2019
- **Mapping images to scene graphs with permutation-invariant structured prediction**, Herzig, R., Raboh, M., Chechik, G., Berant, J., & Globerson, A. NeurIPS 2018 ([Structured Prediction](https://doi.org/10.1007/s10601-018-9285-6))
- **Optimized Pre-Processing for Discrimination Prevention**, Calmon, F., Wei, D., Vinzamuri, B., Ramamurthy, K. N., & Varshney, K. R., NeurIPS 2017 ([Fairness](https://doi.org/10.1007/s10601-018-9285-6))

**Exact Decision Trees**
- Learning Optimal Decision Trees with SAT. Nina Narodytska, Alexey Ignatiev, Filipe Pereira, João Marques-Silva. IJCAI 2018

**Interpretability**
- Compiling Neural Networks into Tractable Boolean Circuits. Arthur Choi and Weijia Shi and Andy Shih and Adnan Darwiche. AAAI 2019
- Compiling Bayesian Networks into Decision Graphs. Andy Shih and Arthur Choi and Adnan Darwiche. AAAI 2019

**Adversarial examples for NNs via MIP**
- Combinatorial Attacks on Binarized Neural Networks. Elias B. Khalil, Amrita Gupta, Bistra Dilkina. ICLR 2019


Background

- **Integer Programming: The Branch and Bound Method**, Detailed example of branch-and-bound applied to a small integer programming problem.
- CH. 5, **STOCHASTIC LOCAL SEARCH: FOUNDATIONS AND APPLICATIONS** by Holger H. Hoos and Thomas Stützle

Other Class Pages

William Cook’s class “Deep Learning in Discrete Optimization”:
http://www.ams.jhu.edu/~wcook12/dl/index.html Spring 2019

Description and Assessment of Assignments

1. **Reviews**: We will explore the course topics through a series of assigned readings in the form of papers (and book chapters). Students will be expected to read the papers before class and submit a one page review for the assigned reading papers as homework. Students will get credit for every review submitted.

2. **Paper Presentations**: Students will present a recent research papers relevant to the course topic and lead discussions with the class.

3. **Project**: A key component of the course will be to get a hands on experience in using ML in the context of discrete optimization. Projects will be alone or in small teams (1-3). Projects will be graded based on their novelty and technical results. Students will be expected to prepare project proposals early in the course, and give presentations of their projects/papers at the end of the course. The final project paper should have the structure of a conference paper with problem statement, lit review, approach, empirical results and discussion. A statement of author contributions (i.e. who did what) must be turned in with the final draft. Rough drafts and partial
Grading Breakdown

<table>
<thead>
<tr>
<th></th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class participation</td>
<td>5%</td>
</tr>
<tr>
<td>Paper Reviews (10 @ 1% each)</td>
<td>10%</td>
</tr>
<tr>
<td>Paper Presentations</td>
<td>30%</td>
</tr>
<tr>
<td>Final Project</td>
<td>55%</td>
</tr>
<tr>
<td>Project Proposal (5%)</td>
<td></td>
</tr>
<tr>
<td>Preliminary Paper (10%)</td>
<td></td>
</tr>
<tr>
<td>Final Presentation (10%)</td>
<td>Last week</td>
</tr>
<tr>
<td>Project/Final Paper (30%)</td>
<td>Final exam time</td>
</tr>
</tbody>
</table>

Assignment Submission Policy
Paper reviews and other deliverables should be submitted the day before class at midnight. Assignments will be administered through Blackboard.

Statement on Academic Conduct and Support Systems

Academic Conduct
Plagiarism – presenting someone else’s ideas as your own, either verbatim or recast in your own words – is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in SCampus in Section 11, Behavior Violating University Standards https://scampus.usc.edu/1100-behavior-violating-university-standards-and-appropriate-sanctions. Other forms of academic dishonesty are equally unacceptable. See additional information in SCampus and university policies on scientific misconduct, http://policy.usc.edu/scientific-misconduct.

Discrimination, sexual assault, and harassment are not tolerated by the university. You are encouraged to report any incidents to the Office of Equity and Diversity http://equity.usc.edu or to the Department of Public Safety http://capsnet.usc.edu/department/department-public-safety/online-forms/contact-us. This is important for the safety of the whole USC community. Another member of the university community – such as a friend, classmate, advisor, or faculty
member – can help initiate the report, or can initiate the report on behalf of another person. The Center for Women and Men http://www.usc.edu/student-affairs/cwm/ provides 24/7 confidential support, and the sexual assault resource center webpage http://sarc.usc.edu describes reporting options and other resources.

**Note on Collaborative Work**

For collaborative projects, students are expected to have equal distribution. If there is any perceived imbalance in the collaborative project, the student should bring this to the attention of the instructor or the teaching assistant.

**Support Systems**

A number of USC’s schools provide support for students who need help with scholarly writing. Check with your advisor or program staff to find out more. Students whose primary language is not English should check with the American Language Institute http://dornsife.usc.edu/ali, which sponsors courses and workshops specifically for international graduate students. The Office of Disability Services and Programs http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html provides certification for students with disabilities and helps arrange the relevant accommodations. If an officially declared emergency makes travel to campus infeasible, USC Emergency Information http://emergency.usc.edu will provide safety and other updates, including ways in which instruction will be continued by means of blackboard, teleconferencing, and other technology.