Algorithms Analysis and Design Project
Discrete Optimization

Team R: Ahana, Anandhini, Ananya, Anmol, Megha, Nikhil, Shashwat, Shivam, Tanvi

Advisors: Dr. Kannan Srinathan, Preet Thakkar

Recap (As of October 22nd)

- Divided into 3 sub-teams:
  - Constraint Programming: Ahana, Anandhini, Ananya, Tanvi
  - Local Search: Anmol, Shashwat, Shivam
  - Linear / Mixed Integer Programming: Megha, Nikhil

- Completed Pascal van Hatenryck Coursera Lectures
- LS team and Nikhil completed the assignment
- Made notes to be formalized into a guide later
- Read theory using external resources and made notes
- Presented progress in mid-discussion

Local Search Subteam - Progress Since

---

1. Both Anmol and Shashwat implemented Euclidean Traveling Salesman Problem Assignment.Best score :**50/60** Summary of results. (Codes can be found here and here)
2. Approaches tried over 2-3 days and improved over several days for both
   a. Swapping adjacent O(N) and all pairs O(N^2)
   b. Initializations: Polar sort, Nearest neighbor, Random with restarts, spanning tree **2-approximation**
   c. 2-opt
   d. Simulated Annealing with Reheats and Restarts
   e. Guided Fast Local Search [Best]

Anmol, Shashwat

Local Search Subteam - Progress Since

---

3. Completed notes (adding Guided Fast Local Search,Simulated annealing, 2-approx algorithm for TSP), converted notes to a formal guide

4. Wrote a report on Euclidean TSP(treating it as a running problem)

5. Our codes have been deployed here.

Input format is number of cities and their coordinates. Output is the ordering of cities. On dragging the output file to the second output, a visualization of the output ordering on 2D space can be obtained here.

Constraint Programming - Progress Since

- Made a guide on constraint programming and its applications with examples like 8 queens, map colouring, sudoku, crypto-arithmetic puzzles,magic series,stable marriage problem,car sequencing problem,perfect square problem,job shop scheduling,character case conversion.
- Added Appendix
- A sudoku solver in python (code is included with the notes)
- A webpage displaying all the information related to constraint programming and a tool within the website to input sudoku values and output the result

Link to the website: https://discreteoptim.herokuapp.com/

Ahana, Anandhini, Ananya, Tanvi

LP - Progress Since

Link to LP guide Made the notes (incl. illustrations and proofs) on Linear Programming for the guide. Read up on MIP as well. Topics covered:

1. Geometric and algebraic perspective of LP
2. Simplex algorithm with detailed proofs.
3. LP Duality and applications like: Max-flow min cut, Minimum Cost flow, Linear Regression, Linear Classifiers, Minimax theorem
4. Google OR Tools and LP Solvers
5. Analysis of methods to solve LP problems: Simplex, Ellipsoid, Integer Point
6. Read up on Karmarkar's algorithm, semi-definite programming, approximation algorithms

Megha

MIP - Progress Since

- Covered branching methods like subtour constraints used for TSP and bounding solution using cutting plane techniques (Goromy, Cover cuts).

- Sensitivity Analysis of Simplex -- extremely useful in practice to re-run Dual Simplex with addition of new constraints/variables.

- Solved Facility Location problem. Received 74/80 score. Report on the same.

Nikhil

Hashcode - Self-Driving Car Fleet Optimization

---

1. Attempted in a timed 4 hour-environment by Nikhil, Shashwat
2. We implemented a simple but interesting greedy simulation
3. Our score set us to achieve Global Rank 13th!
4. Had Local Search formulation ready, but didn’t need to implement due to great performance by our greedy
5. Wrote a detailed report to be published on Medium soon! Will help IIIT and external students prepare too.

Nikhil, Shashwat

Hashcode - Photo Slideshow Optimization

---

1. Attempted in timed 4 hour-environment by Anmol, Nikhil, Shashwat

2. We dissected the problem into Weighted maximum bipartite matching on general graphs and ordering (as in TSP).

3. Coded 4 main approaches:
   a. Data exploitation using DFS sufficed for good score on file B
   b. Greedy + Local search for matching
   c. Greedy + Local search for ordering
   d. Combining b) and c)

4. Not as great results. Should have kept it simple, stupid! :’)

Anmol, Nikhil, Shashwat

Judging System - Minimizing Disagreements using MFAS

- Wrote algorithms for a judging system for subjective contests/grading based on resolving pairwise comparisons by judges into a final ranking. Compiled detailed report.
- Theory work explained in report:
  - Background and applications
  - Reduction to Minimum Feedback Arc Set
  - Proof of NP-Hardness
  - Survey of approx/greedy algorithms
  - MIP Formulation
  - Discrete optimization techniques applied: KL2, Chanas, 2-opt
  - Read Papers

Anmol, Nikhil, Shashwat

Judging System - Minimizing Disagreements using MFAS

- Introduction
  - Humans are bad at regression and numerical scoring, thus the need for pairwise comparisons. Ranking must minimize disagreements.
  - Generally formulated as the Minimum Feedback Arc Set (MFAS) problem or the Linear Ordering problem (LOP).
  - Proof of NP-hardness and survey of approximate/greedy results.

- Use cases (different data sets):
  - Judging AAD projects ~ 40 teams, 5 judges (small but slow)
  - Hackathons ~ 100s of teams, 10-12 judges (large but fast)
  - Digital Art ~ 1000s of submissions, 15-20 judges (medium + medium)

Anmol, Nikhil, Shashwat

Judging System - Minimizing Disagreements using MFAS (2)

- Synthetic Data Generation.
  - Generated by making a model of how judgements happen based on:
    - Varying judge preferences
    - Varying scores of teams on different criteria
    - Planted dream solution

- Approaches
  - Exact approaches like Dynamic Programming, MIP for small cases.
  - Local search heuristics like naive insertion, K-OPT, Lin-Kernighan, Chanas for handling large cases.

- Benchmarks
  - Lolib dataset with normalized matrices (IO, RandA1, RandA2, SGB) (Performance of our algos can be found here)

Anmol, Nikhil, Shashwat

Branch and Bound

Introduced guide using **Branch and Bound**

[Notes on search strategies](#) and [here](#)

[Application to the knapsack problem](#)

This topic serves as an introduction to the concepts of discrete optimization. The goal of a branch-and-bound algorithm is to find a value x that maximizes or minimizes the value of a real-valued function $f(x)$, called an objective function, among some set S of admissible, or candidate solutions.

- Defines notion of a state
- Uninformed search strategies
- Informed search strategies

Ahana, Anmol, Megha, Shashwat, Tanvi

Final Deployments and Conclusion

- Deployed by Shivam, with individual attempts from Ahana, Tanvi and Megha
- Link to guide
- Link to deployments/codes
  - MFAS - Judging System
  - Branch and Bound Knapsack (Plug-and-play search strategies)
  - Traveling Salesman (Multiple approaches using LS)
  - Facility Location Problem (LS + MIP)
  - Hashcode Codes
- Takeaway: Learnt different discrete optimization techniques for NP-Hard problems. Designed and deployed solutions for multiple problems, also dabbling with proofs and approximations.

Shivam

Literature Reviewed

- Some of the many Research Paper/Articles read:
  - Survey of weighted MFAS approaches along with their own approach(MIP)
  - Calculating MFAS on unweighted graphs with clever usage of SCCs
  - Heuristics for ***LOP*** using local search techniques
  - Comprehensive report on guided local search and fast local search
  - On the important details for implementing Simulated Annealing
  - Comprehensive list of different variants of TSP including benchmarks
  - CPAIOR conference on the integration of CP, AI, OR techniques
  - Library of resources for Operations Research
  - Helpful course in Linear Programming with low-level details
  - Designing a better judging system
  - Local Search Techniques for Constrained Portfolio Selection Problems
  - Lectures on Linear Programming
  - Theory and Applications of Linear Programming
  - Guided Fast Local Search - Detailed Guide
  - Algorithms Book - Dasgupta, Papadimitriou, Vazirani
  - A Real-world Application of LP Walkthrough
  - CrowdBT Technique for crowdsourcing judging

Special Thanks to

---

1. Dr. Kannan Srinathan (Professor)
2. Preet Thakkar (TA)
3. Coursera course by Dr. Pascal Van Haternyck
4. Gavel by Anish Athalye
5. LOLIB dataset and Dr. Gerhard Reinelt, Dr. Rafael Marti
6. PicoJr GitHub for Hashcode Checkers used
7. MIT OCW Algorithms Analysis and Design
8. Stanford CS261 on LP & Duality by Tim Roughgarden