DSA Lab Project

APPROXIMATION ALGORITHMS

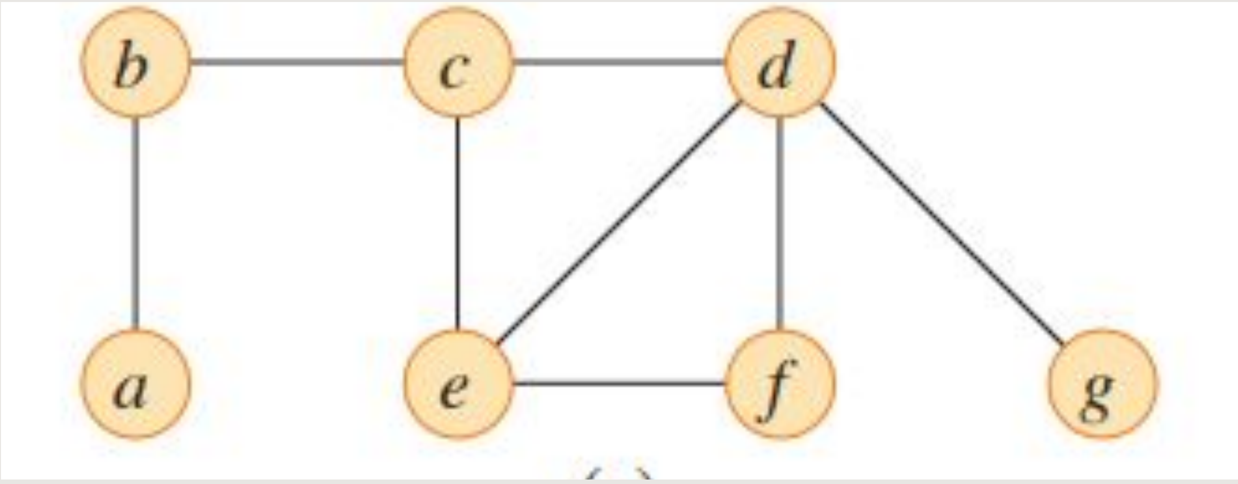
because something is better than nothing :)

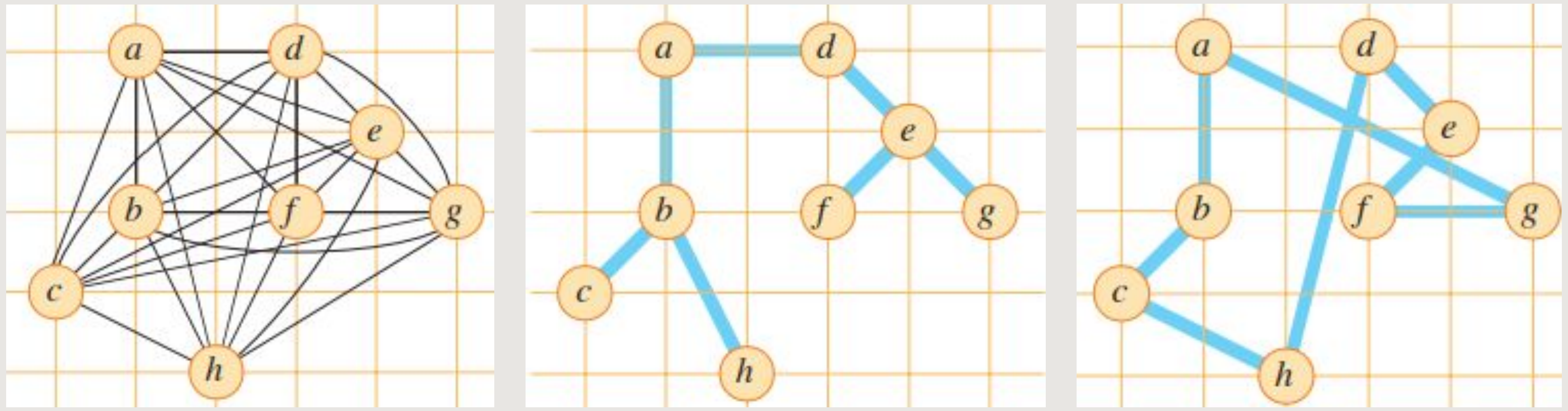
TEAM 4

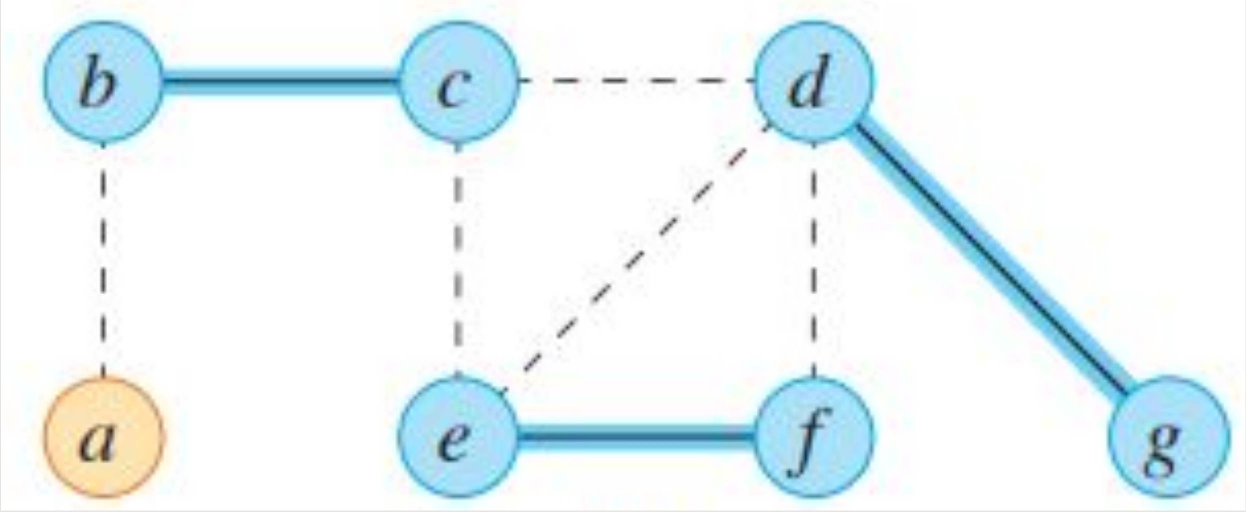
Siddarth Ankita Ankush Praneeth

PREVIOUS WORK

- Read about P and NP type problems.
- Vertex Cover
- Travelling Salesman Problem





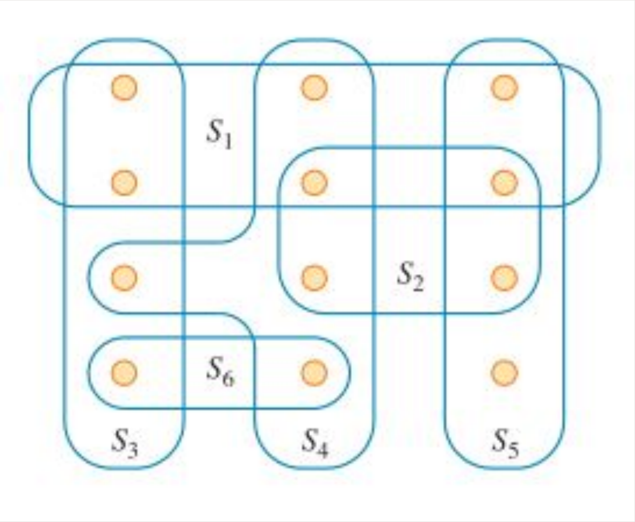


Set Cover Problem

Given a universal set $U = \{e1,e2,\ldots,en\}$ and a collection of subsets $S=\{S1,S2,\ldots,Sm\}$, where $Si \subseteq U$,
The goal is to identify the minimum number of subsets from S whose union equals U, i.e., $\cup\, i \in I\ Si=U$, where $I \subseteq \{1,2,\ldots,m\}$.

Subset sum problem

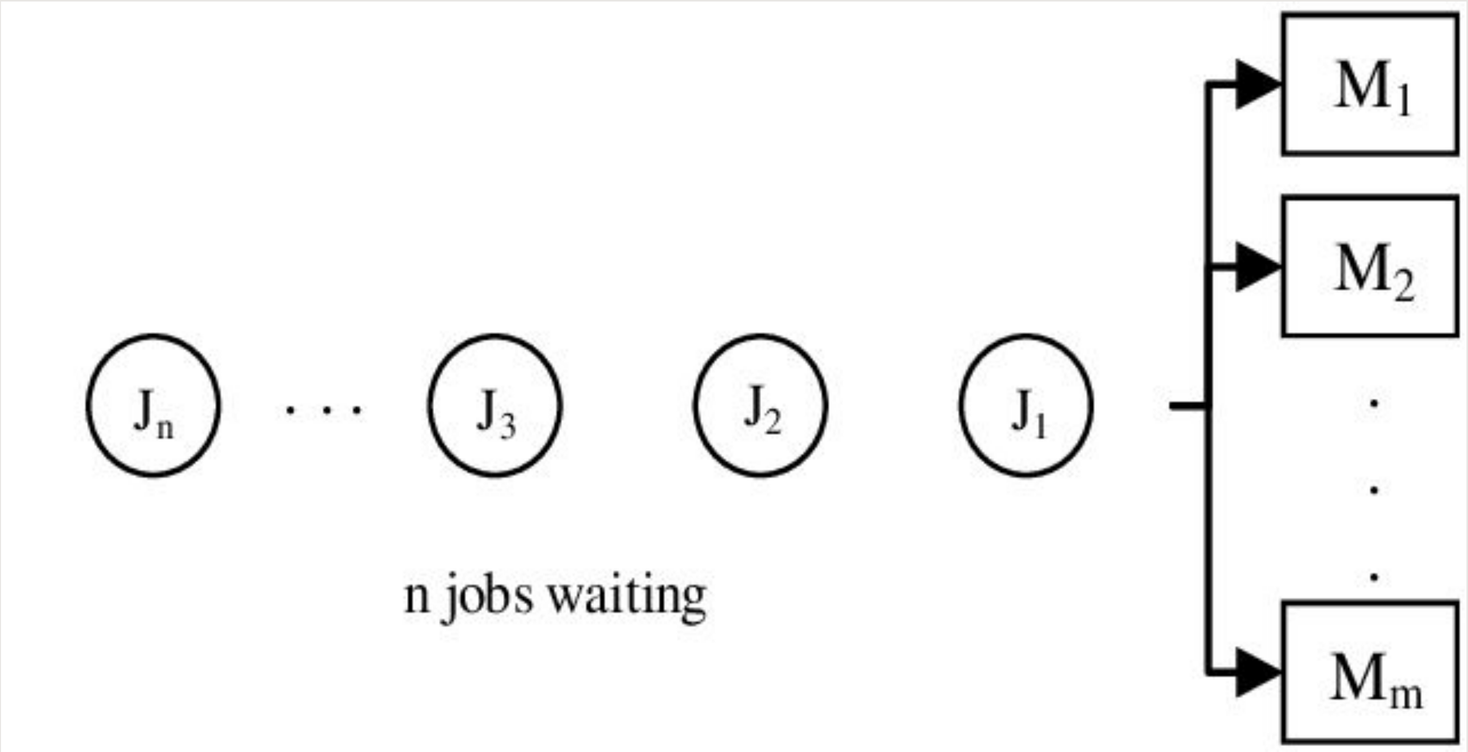
Given a set of integers $S=\{a1,a2,\ldots,an\}$ and a target sum T, the task is to decide if there exists a subset $S' \subseteq S$ such that the sum of the elements in S' is as large as possible but not greater than T.



Parallel machine scheduling

The objective is to schedule a set of jobs on multiple identical machines to minimize the makespan, which is the time by which all jobs are completed.

Given n jobs J1,J2,…,Jn, each with a processing time pk≥0, m identical machines M1,M2,…,Mm_mM1,M2,…,Mm with each job requiring pk consecutive time units to complete, the problem is to plan a scheduling such that each of the jobs are completed running in minimal time. Some constraints are to be met.



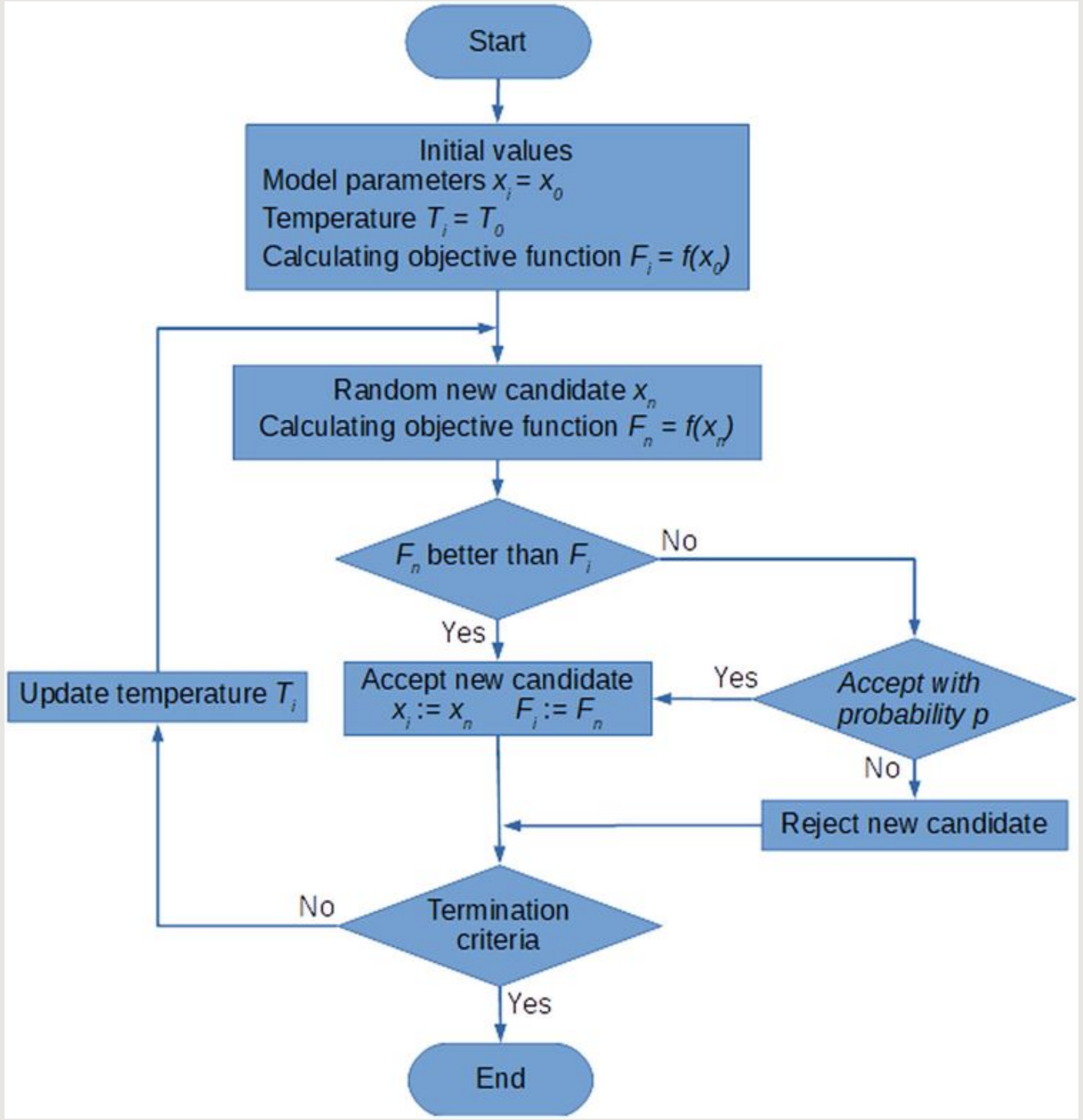
SIMULATED ANNEALING

- **Inspired by metallurgical annealing.**
- **Solves optimization problems by exploring the solution space probabilistically.**

$$P = e^{-\Delta \mathrm{cost}/\mathrm{temperature}}$$

Steps:

- **Start with an initial solution and high "temperature."**
- **Iteratively modify the solution and evaluate cost changes.**
- **Accept worse solutions with decreasing probability to escape local minima.**
- **Gradually cool the system to refine the solution.**



- **Pros: Escapes local optima, efficient for large problems.**
- **Cons: Performance depends on tuning parameters.**

COMPARING DIFFERENT ALGORITHMS

The following three algorithms were applied to the previously discussed problems, and their solutions were compared and analyzed:

- Brute Force
- Basic Approximation Algorithms
- Simulated Annealing

COMPARISON RESULTS

Brute Force:

- Guarantees optimal solution.
- Time complexity grows exponentially.
- Feasible only for small graphs due to high computational cost.
- Brute force is ideal for validation but impractical for large inputs.

Approximation Algorithm

- Provides a near-optimal solution
- Efficient.
- Accuracy depends on graph structure (typically within 2x).
- Fast and reliable for larger graphs.
- Approximation is fast and suitable for consistent near-optimal results.

Simulated Annealing:

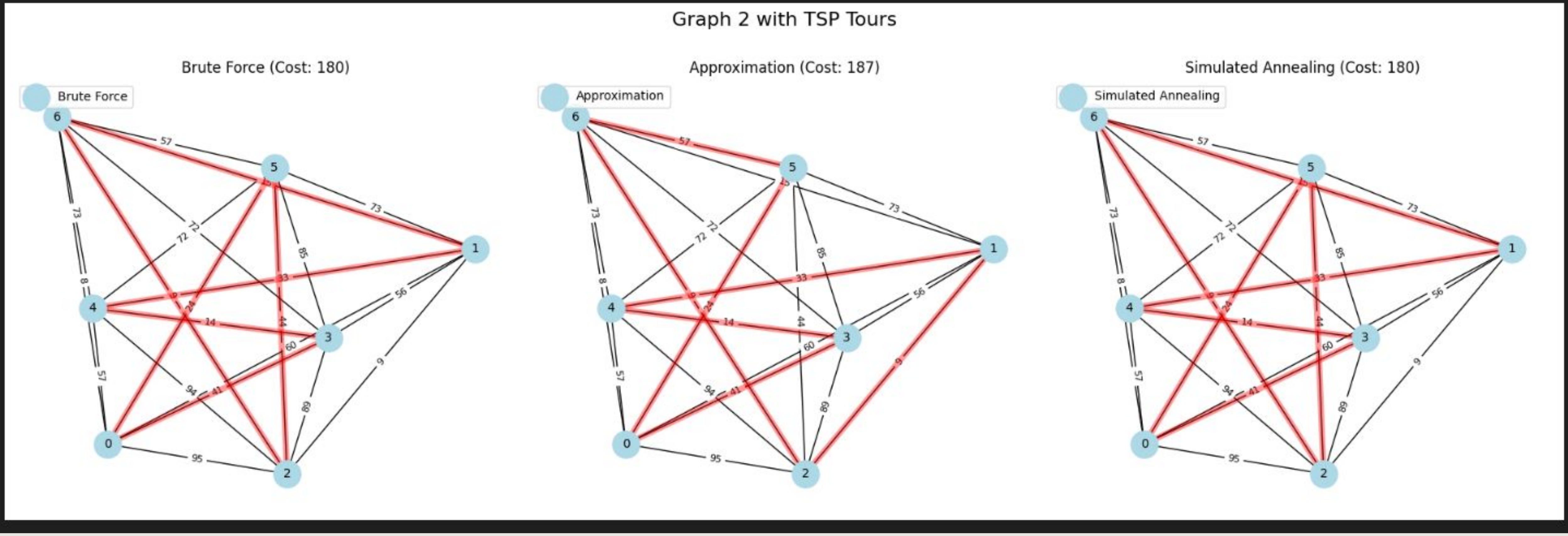
- Balances solution quality and runtime.
- Requires tuning (e.g., cooling rate, initial temperature).
- Typically achieves high accuracy.
- Simulated annealing combines flexibility with competitive accuracy.

Graph Size	Brute Force		2- Approximation Algorithm			Simulated Annealing		
	Cost	Time (ms)	Cost	Time (ms)	Accuracy (%)	Cost	Time (ms)	Accuracy (%)
5	147	0.000000	180	0.000000	81.67	147	4.000000	100.00
7	180	0.000000	187	0.000000	96.26	180	0.000000	100.00
9	198	25.000000	198	0.000000	100.00	208	0.000000	95.19
11	217	3378.000000	407	0.000000	53.32	232	0.000000	93.53
4	-1	0.000000	-1	0.000000	100.00	-1	3.000000	100.00

Note: -1 indicates no solution found.

Tour results along with test cases have been saved as file 'results.txt'.

```
Plotting tour for Brute Force: [(0, 3), (3, 4), (4, 1), (1, 6), (6, 2), (2, 5), (5, 0)]
Plotting tour for Approximation: [(0, 5), (5, 6), (6, 2), (2, 1), (1, 4), (4, 3), (3, 0)]
Plotting tour for Simulated Annealing: [(0, 3), (3, 4), (4, 1), (1, 6), (6, 2), (2, 5), (5, 0)]
```



THANK YOU

TEAM 4

Siddarth Ankita Ankush Praneeth