Parallel Algorithms for the Traveling Salesperson Problem (Checkpoint)

Andrew Singh (andrewsi), Ria Pradeep (rpradeep)

April 24, 2020

Project Summary

We are going to parallelize and compare three different approaches to solving the Traveling Salesperson Problem (TSP). We will implement an exact algorithm and two approximation algorithms, and analyze each algorithm for accuracy, efficiency, and scalability.

Progress

We have finished both the sequential and parallel implementations of the Held-Karp algorithm, the exact algorithm for solving the TSP. We have benchmarked the algorithm on 4 TSP instances of different sizes and evaluated how well the parallel implementation scales for each of the instances (see Preliminary Results).

We have also started implementing the two approximation algorithms, the Lin-Kernighan heuristic and genetic algorithm and identified 4 test cases from TSPLIB for testing and benchmarking our code so far.

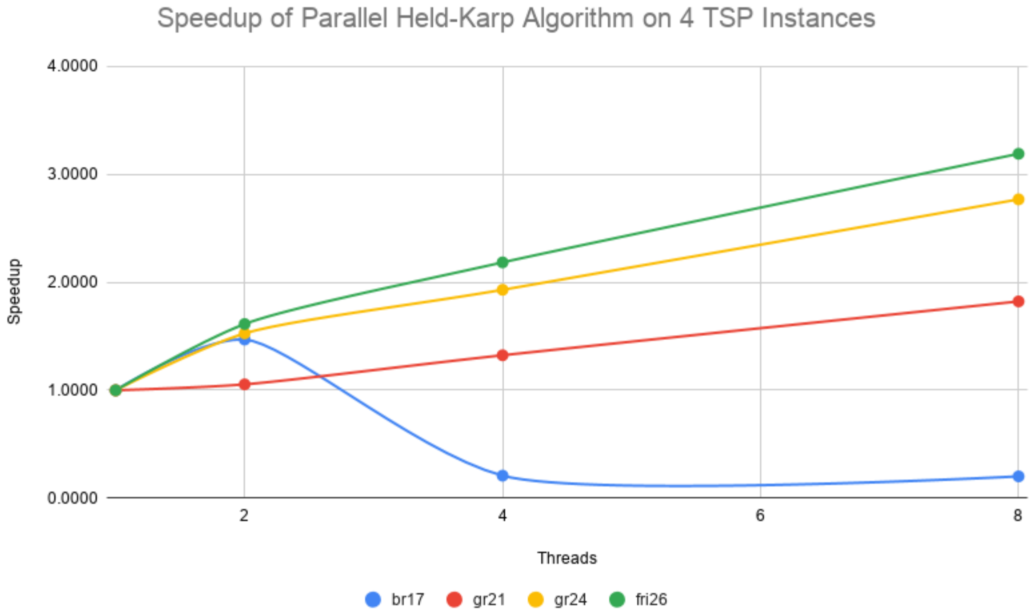
We believe we will be able to complete all our deliverables, which were to paralellize all the above algorithms and measure and compare their performance on GHC machines. We also plan on exploring using the Latedays machines to get more consistent results. The following is our updated list of goals, with the addition bolded:

1. Implement a sequential and parallel version of the three proposed algorithms for solving TSP: Held-Karp, Lin-Kernighan heuristic, and genetic.
2. Evaluate each algorithm on a test suite of TSP instances selected from TSPLIB. For each algorithm, benchmark its sequential version, and benchmark its parallel version on several thread counts to measure scalability. **Do this on GHC machines and explore benchmarking on Latedays machines.**
3. Compile the evaluation data into plots and tables that compare the accuracy, speed, and scalability of each algorithm. Analyze each algorithm for its relative strengths and weaknesses, and how amenable each one is to parallelism.

Preliminary Results

We have benchmarked the Held-Karp algorithm on the GHC machines and obtained some preliminary results.

Instance	Sequential	2 Threads	4 Threads	8 Threads
br17	0.0804	0.0547	0.3835	0.3960
gr21	1.9800	1.8775	1.4954	1.0865
gr24	20.1860	13.2243	10.4594	7.2958
fri26	93.1457	57.7422	42.6566	29.2109



Presentation Plans

We plan on presenting charts and graphs detailing the results of parallelizing the algorithms, and how they scale from sequential to the maximum number of threads available to us.

Concerns

When benchmarking the Held-Karp algorithm on the GHC machines, we noticed that the timings significantly varied day-to-day. While the results shown here were all measured in a single session, we would like our results in future sessions to be consistent with our current results. For evaluation going forward, we will look into running the algorithms on the Latedays machines, which ensures that our job is the only one running on a worker node. In addition, Latedays will allow us to scale our parallel algorithm to 12 threads instead of just 8.

Schedule

Due Date	**Item**	**Assigned To**
4/24	Project checkpoint	Both
4/24	Implement sequential version of genetic algorithm	Ria
4/26	Implement sequential version of Lin-Kernighan heuristic	Andrew
4/27	Parallelize Lin-Kernighan heuristic	Andrew
4/27	Parallelize genetic algorithm	Ria
4/28	Benchmark all three algorithms on TSPLIB instances at different thread counts	Both
4/30	Compile evaluation data and generate plots and tables that compare the accuracy, speed, and scalability of each algorithm	Both
5/4	Write up final report, analyzing each algorithm for its relative strengths and weaknesses, and how amenable each one is to parallelism	Both