IE 670 Selected Topics in Logistics Optimization
TSP Assignment

Seung Hyun Kwon

November 2020

1 Introduction

There is a breadth of research pertaining to the Traveling Salesman Problem. In Part 1 of this assignment, we looked at five mathematical formulations of the TSP and measured their overall performance by looking at run time, the optimal Linear Relaxation solution, the best integer solution and the optimality gap. We generated various (x,y) coordinates of a specified size by using a TSP instance generator developed by Professor Walteros. We generated two instances for each size of 15, 20, 25, 30, 50 and 70 where size represents the number of nodes (ex. city coordinates) in the tour. Below is a table measuring the performance of each formulation. Note: We set the run-time limit to 10 minutes or 600 seconds.

Formulation	Instance	LR Sol	Integer Sol	Gap (Int)	Run time
Miller-Tucker-Zemlin	n 15 inst 1	55.679	65.132	0%	0.303 sec
	n 15 inst 2	51.450	71.824	0%	1.850 sec
	n 20 inst 1	47.258	68.378	0%	1.792 sec
	n 20 inst 2	61.233	77.848	0%	1.648 sec
	n 25 inst 1	57.361	79.487	0%	8.719 sec
	n 25 inst 2	64.262	83.086	0%	3.498 sec
	n 30 inst 1	68.575	89.623	0%	15.710 sec
	n 30 inst 2	57.694	88.607	0%	10.144 sec
	n 50 inst 1	94.773	114.188	0.0072%	144.470 sec
	n 50 inst 2	86.515	107.904	2.472%	Limit Reached
	n 70 inst 1	110.820	131.322	0%	231.629 sec
	n 70 inst 2	109.004	132.039	0%	567.627 sec
Multicommodity Flow	n 15 inst 1	55.599	65.132	0%	0.543 sec
	n 15 inst 2	55.251	71.824	0%	0.467 sec
	n 20 inst 1	46.934	68.378	0%	1.574 sec
	n 20 inst 2	61.264	77.848	0%	1.456 sec
	n 25 inst 1	57.090	79.487	0%	2.720 sec
	n 25 inst 2	64.004	83.086	0%	10.344 sec

Multicommodity Flow	n 30 inst 1	68.195	89.623	0%	30.673 sec
	n 30 inst 2	57.010	88.607	0%	13.925 sec
	n 50 inst 1	94.582	114.188	0%	371.078 sec
	n 50 inst 2	86.570	107.904	0%	182.504 sec
	n 70 inst 1	110.706	131.322	15.69%	Limit Reached
	n 70 inst 2	108.790	157.368	30.869%	Limit Reached
Shortest Path	n 15 inst 1	58.553	65.132	0%	0.791 sec
	n 15 inst 2	64.051	71.824	0%	1.853 sec
	n 20 inst 1	56.495	68.377	0%	32.884 sec
	n 20 inst 2	69.818	77.848	0%	12.335 sec
	n 25 inst 1	64.197	79.487	0%	293.049 sec
	n 25 inst 2	71.439	83.086	0%	92.372 sec
	n 30 inst 1	71.665	89.623	6.217%	Limit Reached
	n 30 inst 2	68.653	88.607	0%	405.412 sec
	n 50 inst 1	98.218	118.403	13.694%	Limit Reached
	n 50 inst 2	92.848	136.231	30.227%	Limit Reached
	n 70 inst 1	114.080	161.774	26.453%	Limit Reached
	n 70 inst 2	114.053	171.141	31.513%	Limit Reached
Quadratic Assignment	n 15 inst 1	65.132	65.132	0%	4.893 sec
	n 15 inst 2	71.824	71.824	0%	7.935 sec
	n 20 inst 1	77.834	68.378	9.693%	Limit Reached
	n 20 inst 2	97.742	78.555	13.461%	Limit Reached
	n 25 inst 1	99.824	84.190	18.988%	Limit Reached
	n 25 inst 2	132.051	83.610	16.911%	Limit Reached
	n 30 inst 1	124.707	127.997	40.486%	Limit Reached
	n 30 inst 2	151.191	115.503	42.818%	Limit Reached
	n 50 inst 1	373.303	356.645	98.735%	Limit Reached
	n 50 inst 2	367.129	443.375	98.834%	Limit Reached
	n 70 inst 1	640.712	645.705	98.956%	Limit Reached
	n 70 inst 2	659.446	652.407	99.295%	Limit Reached
Dantzig Fulkerson Johnson	n 15 inst 1	60.360	65.132	0%	0.084 sec
	n 15 inst 2	63.221	71.824	0%	0.093 sec
	n 20 inst 1	59.083	68.378	0%	0.100 sec
	n 20 inst 2	66.361	77.848	0%	0.104 sec
	n 25 inst 1	73.367	79.487	0%	0.158 sec
	n 25 inst 2	75.317	83.086	0%	0.158 sec
	n 30 inst 1	84.828	89.623	0%	0.190 sec
	n 30 inst 2	81.216	88.607	0%	0.148 sec
	n 50 inst 1	106.732	114.188	0%	0.511 sec
	n 50 inst 2	93.310	107.904	0%	0.445 sec
	n 70 inst 1	126.045	131.322	0%	0.632 sec
	n 70 inst 2	127.079	132.039	0%	0.720 sec

2 Some Thoughts

The performance for the Dantzig Fulkerson and Johnson (DFJ) formulation with subtour elimination should catch your eye. It obtains the optimal solution for all our instances regardless of the size, all in less than one second! The optimality gap is 0% across the board for DFJ. This makes sense since we first obtain a very simple integer solution and only put the subtour constraints sequentially, allowing the formulation to run super efficiently. After that, the Miller-Tucker-Zemlin Formulation was able to obtain optimal and near optimal integer solutions for all instances. We do see the run time increase noticeably to a few minutes when the size gets up to 50 and larger. While the consrtaints containing the 'u' variables are good, we do have to implement all the cases which could get tedious with larger graphs. The MultiCommodity Flow Formulation performs pretty well while run-time increases with larger graphs. Similar to the MTZ formulation, we have to implement the flow capacity constraints every time which gets more time consuming with larger graphs. The shortest path method has more constraints than the formulations discussed so far and likewise, we see the run-time increase much more significantly for the same instances. Lastly, the Quadratic Assignment formulation was not useful as the run-time limit of 10 minutes was reached for all except for graphs of size 15. As we discussed in class, the run-time is order n cubed so with size 20, already the order is 8000. Overall, we see that the Dantzig Fulkerson and Johnson formulation with lazy cuts is extremely efficient and a useful insight as we learn to model more optimization problems.

3 Conclusion and Incomplete Portions

Overall, we see that the Dantzig Fulkerson and Johnson formulation with lazy cuts is extremely efficient. The idea of adding constraints when needed is a key insight rather than adding all the constraints at once. It was extremely interesting to see how exactly the formulation impacts the run time. Few things to note. I was not able to implement the plain loop due to running out of time. I am looking forward to finishing the assignment at my own pace and analyzing the results. Additionally, I wasn't quite sure how to implement a linear relaxation on the quadratic assignment problem. If I eased the x variables as continuous between 0 and 1, the solution was always 0. So I kept the x variables binary and w variables continuous between 0 and 1, which made it less efficient than obtaining the integer solution.

4 Additional Help

For the DFJ formulation, I consulted the Gurobi website where they gave a good explanation on subtour elimination and the code to implement it. After spending about four hours trying to get my own subtour elimination to work, I revised the Gurobi one to match my model in order to implement the DFJ.

With regards to the other formulations, I followed a former UB student youtube video Hernan Caceres on a Capacitated vehicle routing formulation and was able to see how to set up the variables, arcs and model. I am new to Gurobi so it took me about a day to understand how to formulate constraints with multiple summations.

5 Note on rest of assignment

In part 2, I completed the first portion where we build the minimum cost 1 tree after building the minimum spanning tree from the Prims Algorithm. I coded a Lagrangian relaxation but was not able to get it to work. I was not able to complete the Subgradient Algorithm in part 2 and all of part 3 due to running out of time. I am looking to do it on my own time as this exercise has been extremely interesting.