Simplex Report

Boyuan Yao 19307110202

October 11, 2021

1 Primal Simplex

My primal simplex method contains three stages: presolve, phaseI and phaseII. The first part is to do a simple processing of the original problem, in my implementation it involves two operations: delete some rows to make the matrix has full row rank(through QR decomposition) and eliminate the rows that only has one nonzero entry(which means that the corresponding decision variable is stationary). The second part is to find the initial BFS while the second one is for the optimal solution. The followings are the outcomes with the given testing instances.

	optvalue	tpresolve(s)	tphaseI(s)	tphaseII(s)	iter1	iter2	error_solution	error_outcome
25fv47	5.5018e+03	0.1762	267.8798	301.2520	98894	107553	8.6058e-16	1.3225e-15
brandy	1.5185e+03	0.0120	0.2934	0.1382	1767	793	1.5217e-15	1.4973e-16
cre_a	2.3595e+07	12.0023	1.5075e+03	1.4276e+03	16011	14180	3.5187e-15	3.5176e-16
cre_d	2.4455e+07	852.3142	6.4671e+03	3.7481e+04	10700	67499	6.5879e-15	7.4643e-15
e226	-18.7519	0.0092	0.4611	0.8002	1235	2075	1.2793e-15	2.0840e-15
scrs8	904.2970	0.0568	2.5736	1.8096	2091	1474	1.1714e-16	0

Optvalue means the optimal value attained by my primal simplex code, tpresolve, tphaseI and tphaseII mean the time spends in the corresponding stages. iter1 and iter2 mean the number of iterations in phaeI and phaseII. error_solution means the relative error of the solution, which is calculated by $\frac{||Ax-b||_2}{||x||_2}$. error_outcome compares the optimal value attained by my implementation with the one attained by MATLAB standard function linprog, which is calculated by $\frac{|optvalue-f^*|}{|f^*|}$.

When testing nug08, I met degeneracy problem that phaseI failed to complete.

2 Dual simplex

My dual simplex method only contains one stage, assuming that the initial basis is given, and the matrix A has full row rank. I use randomize cases to test the function, and compare it with my primal simplex:

	primal	tphaseII(s)	iter2	dual	optvalue	t	iter	error_solution	error_outcome
case1		0.7120	6611		9.2184	0.0519	632	3.2434e-16	1.9270e-16
case2		395.9219	317177		31.0542	2.3556	2231	2.9422e-16	1.1440e-16
case3		7.2580	876		26.5685	2.7413	174	3.1100e-17	4.0116e-16

Three testing instances are generating by my code test_instance. Matrix has the structure $A = (\tilde{A}, I)$. Here are some information about the three instances:

	row	column	density
case1	100	300	0.1
case2	500	1500	0.01
case3	1000	2000	0.002

You could find all the codes in the folder "src", all the testing outcomes in the folder "outcome".