XI Balkan Conference on Operational Research
BALCOR 2013
Belgrade, Zlatibor, 07-10 September 2013
UNIVERSITY OF BELGRADE
FACULTY OF ORGANISATIONAL SCIENCES

COMPUTATIONAL COMPARISON OF PIVOTING RULES FOR THE REVISED SIMPLEX ALGORITHM

NIKOLAOS PLOSKAS[1], NIKOLAOS SAMARAS[2]

[1] Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece, ploskas@uom.gr
[2] Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece, samaras@uom.gr

Abstract: *The simplex algorithm is the most widely-used method for solving linear programming problems. Pricing is the operation of the selection of an improving non-basic variable in order to enter the basis in each iteration of the algorithm. Pricing is one of the most time-consuming steps in simplex type algorithms and it is essential to reduce the time taken to perform this step. In this paper, we perform a computational comparison in which the pricing operation is computed with eight different pivoting rules: (i) Bland's rule, (ii) Dantzig's rule, (iii) Greatest Increment Method, (iv) Least Recently Considered Method, (v) Partial Pricing Rule, (vi) Queue Rule, (vii) Stack Rule, and (viii) Steepest Edge Rule; and incorporate them with the revised simplex algorithm. All pivoting rules have been implemented in MATLAB and the test set used in the computational study is the Netlib (Optimal, Kennington, and Infeasible) set of linear problems.*

Keywords: *Linear Programming, Revised Simplex Method, Pricing, Pivoting Rules, MATLAB.*

1. INTRODUCTION

Linear Programming (LP) is the process of minimizing or maximizing a linear objective function $z = \sum_{i=1}^{n} c_i x_i$ to a number of linear equality and inequality constraints. Simplex algorithm is the most widely used method to solve linear programming problems (LPs). We assume that the problem is in its standard form. Formulating the linear problem, we can describe it as shown in (LP-1):

$$\begin{aligned} \min \quad & c^T x \\ \textit{subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned} \tag{LP-1}$$

where $A \in R^{mxn}$, $(c, x) \in R^n$, $b \in R^m$, and T denotes transposition. We assume that A has full rank, rank(A)=m, m<n. Simplex algorithm searches for the optimal solution by moving from one feasible solution to another, along the edges of the feasible set. The dual problem associated with the linear problem (LP-1) is shown in (DP-1):

$$\begin{aligned} \min \quad & b^T w \\ \textit{subject to} \quad & A^T w + s = c \\ & s \geq 0 \end{aligned} \tag{DP-1}$$

where $w \in R^m$ and $s \in R^n$. A_B is an *mxm* non-singular sub-matrix of A, called basic matrix or basis. The columns of A which belong to subset B are called basic and those which belong to N are called non basic. A column A_l is selected in each step in order to enter the basis and a column A_r to leave the basis. The variable l is called entering variable and is computed according to a pivoting rule.

A crucial step in solving an LP with the simplex algorithm is the selection of the entering variable and it is performed in each iteration. Good choices can lead to a fast convergence to the optimal solution, while poor choices lead to worst execution times or even no solutions of the LPs. A pivoting rule is one of the main factors that will determine the number of iterations that simplex algorithm performs (Maros & Khaliq 1999). Many pivoting rules have been proposed in the literature. Eight of these are presented and compared in this paper; namely, (i) Bland's rule, (ii) Dantzig's rule, (iii) Greatest Increment Method, (iv) Least Recently Considered Method, (v) Partial Pricing Rule, (vi) Queue Rule, (vii) Stack Rule, and (viii) Steepest Edge Rule.

Thomadakis (1994) has compared five pivoting rules: (i) the Dantzig's rule, (ii) the Bland's rule, (iii) the Least-Recently Considered Method, (iv) the Greatest-Increment Method, and (v) the Steepest-Edge rule. The author examined the trade-off between the number of iterations and the execution time per iteration and concluded that: (i) Bland's rule requires the shortest execution time per iteration, but it usually needs many more iterations than the other methods to converge to the optimal solution, (ii) Dantzig's rule and Least Recently Considered Method perform comparably, but the latter requires fewer iterations in cases where degenerate pivots exist, (iii) the computational cost per iteration in the Greatest Increment Method is greater than the aforementioned methods, but it usually leads to fewer iterations, and (iii) the Steepest-Edge rule requires fewer iterations than all other pivot rules and the computational cost of this method is lower than Greatest Increment Method but higher than the other three methods.

To the best of our knowledge, Thomadakis' report (1994) is the only paper that compares some of the most widely-used pivoting rules. This paper is an extension of the work of Thomadakis (1994) where eight well-known pivoting rules are compared. Furthermore, Thomadakis (1994) has focused on the number of iterations and the execution time per iterations, while we also investigate the total execution time of the simplex algorithm relating to the pivoting rule that is used.

The structure of the paper is as follows: in Section 2, a brief description of the revised simplex algorithm is presented. In Section 3, eight widely-used pivoting rules are presented and analyzed. In Section 4, the computational comparison of the pivoting rules is presented over the Netlib set (Optimal, Kennington and Infeasible LPs). Finally, the conclusions of this paper are outlined in section 5.

2. REVISED SIMPLEX METHOD

Using a basic partition A=[B N], the linear problem (LP-1) can be written as:

$$\begin{aligned} \min \quad & c_B^T x_B + c_N^T x_N \\ \textit{subject to} \quad & A_B x_B + A_N x_N = b \\ & x_B, x_N \geq 0 \end{aligned} \qquad \text{(LP-2)}$$

The solution of the linear problem $x_B = A_B^{-1} b, x_N = 0$ is called a basic solution. A solution $x = (x_B, x_N)$ is feasible if $x > 0$. Otherwise the solution is infeasible. The solution of the linear problem (LP-2) is computed by the relation $s = c - A^T w$, where $w = (c_B)^T A_B^{-1}$ are the simplex multipliers and s are the dual slack variables. The basis B is dual feasible if $s \geq 0$. A formal description of the revised simplex algorithm (Dantzig *et al.* 1953) is given below.

Table 1: Revised Simplex Algorithm

Step 0. (*Initialization*).

Start with a feasible partition (A_B, A_N). Compute A_B^{-1} and vectors x_B, w and s_N.

Step 1. (*Test of optimality*).

if $s_N \geq 0$ then STOP. The linear problem (LP-2) is optimal.

Step 2. (*Choice of the entering/leaving variable*).

Choose the index l of the entering variable using a pivoting rule. Variable x_l enters the basis.

Compute the pivot column $h_l = A_B^{-1} A_l$.

if $h_l \leq 0$ then STOP. The linear problem (LP-2) is unbounded.

else

Choose the leaving variable $x_{B[r]} = x_k$ using the following equation:

$$x_{B[r]} = \frac{x_{B[r]}}{h_{il}} = \min\left\{ \frac{x_{B[i]}}{h_{il}} : h_{il} < 0 \right\}$$

Step 3. (*Pivoting*).

Swap indices k and l. Update the new basis inverse $\overline{A_B}^{-1}$. Go to Step 1.

3. PIVOTING RULES

Eight pivoting rules are presented in this section: (i) Bland's rule, (ii) Dantzig's rule, (iii) Greatest Increment Method, (iv) Least Recently Considered Method, (v) Partial Pricing Rule, (vi) Queue Rule, (vii) Stack Rule,

and (viii) Steepest Edge Rule. Some necessary notations should be introduced, before the presentation of the aforementioned pivoting rules. Let l be the index of the entering variable and $\overline{c_l}$ be the difference in the objective value when the non-basic variable x_l is increased by one unit and the basic variables are adjusted appropriately.

3.1. Bland's Rule

Bland's rule (Bland 1977) selects as entering variable the first among the eligible ones, that is the leftmost among columns with negative relative cost coefficient. Although Bland's rule avoids cycling, it has been observed in practice that this pivoting rule can lead to stalling, a phenomenon where long degenerate paths are produced.

3.2. Dantzig's Rule

The first pivoting rule that was used in the simplex algorithm is Dantzig's rule or largest coefficient rule (Dantzig 1963). This pivoting rule selects the column A_l with the most negative $\overline{c_l}$. It guarantees the largest reduction in the objective value per unit of non-basic variable $\overline{c_l}$ increase. Although its worst-case complexity is exponential (Klee & Minty 1972), Dantzig's rule is claimed as simple but powerful enough to guide simplex algorithm into short paths (Thomadakis 1994). This is the first proposed pivoting rule for the simplex algorithm and has been widely-used in simplex implementations (Bazaraa *et. al.* 1990, Papadimitriou & Steiglitz 1982).

3.3. Greatest Increment Method

According to the Greatest Increment Method (Klee & Minty 1972), the variable with the largest total objective value improvement is selected as the incoming variable. Greatest Increment Method calculates the improvement of the objective value for each non-basic variable and then selects the variable that offers the largest improvement in the objective value. Although this pivoting rule can lead to fast convergence to the optimal solution, this advantage is eliminated by the additional computational cost per iteration. Finally, Gärtner (1995) constructed LPs that Greatest Increment Method showed exponential complexity.

3.4. Least Recently Considered Method

In the first iteration of the algorithm, the incoming variable l is selected according to Bland's rule, that is the leftmost among columns with negative relative cost coefficient. In the next iterations, the Least Recently Considered Method (Zadeh 1980) starts searching for the first eligible variable with index greater than l. If $l = n$ then Least Recently Considered Method starts searching from the first column again. Least Recently Considered Method prevents stalling and it performs fairly well in practice Thomadakis (1994). However, its worst-case complexity has not been proved yet.

3.5. Partial Pricing Rule

Partial Pricing methods are variants of the standard rules that take only a part of non-basic variables into account. In the computational study presented in Section 4, we have implemented the partial pricing rule as variant for Dantzig's rule. In static partial pricing, non-basic variables are divided into equal segments with predefined size and the pricing operation is carried out segment by segment, until a reduced cost is found. In dynamical partial pricing, the segments' size is determined dynamically during the execution of the algorithm.

3.6. Queue Rule

Queue is a FIFO (First-In-First-Out) data structure, where the first element added to the queue is the first one to be removed. In the pivoting rule of queue, two queues are constrcuted; the first one holds the indices of the basic variables, while the other the indices of the non-basic variables. The entering and leaving variables are selected from the front of the corresponding queue. The variable, which is extracted from the front of the queue that holds the basic variables, is inserted to the end of the queue that holds the non-basic variables.

Respectively, the variable, which is extracted from the front of the queue that holds the non-basic variables, is inserted to the end of the queue that holds the basic variables.

3.7. Stack Rule

Stack is a LIFO (Last-In-First-Out) data structure, where the last element added to the stack is the first one to be removed. In the stack rule, the entering and leaving variables are selected from the top of the corresponding stack. The variable, which is extracted from the top of the stack that holds the basic variables, is inserted to the top of the stack that holds the non-basic variables. Respectively, the variable, which is extracted from the top of the stack that holds the non-basic variables, is inserted to the end of the stack that holds the basic variables.

3.8. Steepest Edge Rule

Steepest Edge Rule or All-Variable Gradient Method (Goldfarb & Reid 1977) selects as entering variable the variable with the most objective value reduction per unit distance. Although this pivoting rule can lead to fast convergence to the optimal solution, this advantage is debatable due to the additional computational cost. Approximate methods have been proposed in order to improve the computational efficiency of this method (Świętanowski, 1988, Vanderbei 2001).

4. COMPUTATIONAL RESULTS

Computational studies have been widely-used in order to examine the practical efficiency of an algorithm or even compare algorithms. The computational comparison of the aforementioned eight pivoting rules has been performed on a quad-processor Inter Core i7 3.4 GHz with 32 Gbyte of main memory and 8 cores. The revised simplex method and the pivoting rules have been implemented using MATLAB Professional R2012b. MATLAB is a powerful programming environment and is especially designed for matrix computations.

The test set used in the computational study was the Netlib set (Optimal, Kennington and Infeasible LPs) (Gay 1985, Carolan *et al.* 1990). The Netlib library is a widely-used suite for testing the computational performance of linear programming algorithms and contains many real world LPs. Ordóñez and Freund (2003) have shown that 71% of the Netlib LPs are ill-conditioned. Hence, numerical difficulties may occur. Table 1 presents some useful information about the Netlib set. The first column includes the name of the problem, the second the number of constraints, the third the number of variables, the fourth the non-zero elements of matrix A and the fifth the optimal objective value. The test bed includes 40 optimal and 5 infeasible LPs from Netlib and 3 Kennington LPs that do not have ranges and bounds sections in their mps files.

Table 2: Statistics of the Netlib set (Optimal, Kennington and Infeasible LPs)

Name	Constraints	Variables	Non-zeros A	Objective value
25FV47	822	1,571	11,127	5.50E+03
ADLITTLE	57	97	465	2.25E+05
AFIRO	28	32	88	-4.65E+02
AGG	489	163	2,541	-3.60E+07
AGG2	517	302	4,515	-2.02E+07
AGG3	517	302	4,531	1.03E+07
BANDM	306	472	2,659	-1.59E+02
BEACONFD	174	262	3,476	3.36E+04
BLEND	75	83	521	-3.08E+01
BNL1	644	1,175	6,129	1.98E+03
BNL2	2,325	3,489	16,124	1.81E+03
BRANDY	221	249	2,150	1.52E+03
CRE_A	3,517	4,067	19,054	2.36E+07
CRE_C	3,069	3,678	16,922	2.53E+07
DEGEN2	445	534	4,449	-1.44E+03
E226	224	282	2,767	-1.88E+01
FFFFF800	525	854	6,235	5.56E+05

ISRAEL	175	142	2,358	-8.97E+05
ITEST2	10	4	17	Infeasible
ITEST6	12	8	23	Infeasible
KLEIN1	55	54	696	Infeasible
KLEIN2	478	54	4,585	Infeasible
KLEIN3	995	88	12,107	Infeasible
LOTFI	154	308	1,086	-2.53E+01
OSA-07	1,119	23,949	167,643	5.36E+05
SC50A	51	48	131	-6.46E+01
SC50B	51	48	119	-7.00E+01
SC105	106	103	281	-5.22E+01
SC205	206	203	552	-5.22E+01
SCAGR7	130	140	553	-2.33E+06
SCFXM1	331	457	2,612	1.84E+04
SCFXM2	661	914	5,229	3.67E+04
SCFXM3	991	1,371	7,846	5.49E+04
SCORPION	389	358	1,708	1.88E+03
SCRS8	491	1,169	4,029	9.04E+02
SCTAP1	301	480	2,052	1.41E+03
SCTAP2	1,091	1,880	8,124	1.72E+03
SCTAP3	1,481	2,480	10,734	1.42E+03
SHARE1B	118	225	1,182	-7.66E+04
SHARE2B	97	79	730	-4.16E+02
SHIP04L	403	2,118	8,450	1.79E+06
SHIP04S	403	1,458	5,810	1.80E+06
SHIP08L	779	4,283	17,085	1.91E+06
SHIP08S	779	2,387	9,501	1.92E+06
SHIP12L	1,152	5,427	21,597	1.47E+06
SHIP12S	1,152	2,763	10,941	1.49E+06
STOCFOR1	118	111	474	-4.11E+04
STOCFOR2	2,158	2,031	9,492	-3.90E+04

Table 3 presents the results from the total execution time of the revised simplex algorithm combined with the aforementioned pivoting rules, while Table 4 the iterations needed. The results are also graphically illustrated in Figures 1 – 2, where the average execution time and the average number of iterations are presented, respectively. In Tables 3 – 4 and Figures 1 – 2, the following abbreviations are used: (i) Bland's rule – BR, (ii) Dantzig's rule – DR, (iii) Greatest Increment method – GIM, (iv) Least Recently Considered method – LRCM, (v) Partial Pricing rule – PPR, (vi) Queue rule – QR, (vii) Stack rule – SR, and (viii) Steepest Edge rule – SER. For each instance we averaged times over 10 runs. All times in Table 3 and Figure 1 are measured in seconds. A limit of 70,000 iterations was set that explains why there are no measurements for some pivoting rules on specific instances. Finally, the objective value calculated using each pivoting rule was accurate with a precision of 8 decimal digits.

From the following results, we observe that only Dantzig's rule has solved all instances, while Bland's rule solved 45 out of 48 instances, Greatest Increment method 46 out of 48, Least Recently Considered method 45 out of 48, partial pricing 45 out of 48, Queue's rule 41 out of 48, Stacks' rule 43 out of 48, and Steepest Edge rule 46 out of 48. Furthermore, Dantzig's rule requires the shortest execution time both on average and on almost all instances. On the other hand, Steepest Edge rule has the worst execution time both on average and on almost all instances. Despite its computational cost, Steepest Edge rule needs the fewest number of iterations than all the other pivoting rules, while Bland's rule is by far the worst pivoting rule in terms of the number of iterations.

Table 3: Total Execution Time

Name	BR	DR	GIM	LRCM	PPR	QR	SR	SER
25FV47	-	63.26	-	-	-	-	-	6,504.45
ADLITTLE	0.03	0.06	0.02	0.04	0.02	0.03	0.03	0.04

AFIRO	0.01	0.004	0.005	0.01	0.01	0.01	0.01	0.004
AGG	0.06	0.05	0.10	0.07	0.07	0.07	0.07	0.20
AGG2	0.14	0.11	0.41	0.11	0.11	0.13	0.15	1.24
AGG3	0.22	0.13	0.49	0.26	0.22	0.20	0.29	1.39
BANDM	1.33	0.48	0.80	1.28	1.08	-	1.98	1.88
BEACONFD	0.02	0.02	0.02	0.02	0.03	0.02	0.02	0.02
BLEND	0.05	0.03	0.07	0.77	0.06	2.06	0.09	0.04
BNL1	181.40	20.92	30.20	95.43	53.20	-	-	264.11
BNL2	-	211.51	-	-	-	-	-	-
BRANDY	1.69	0.16	0.34	0.51	0.67	0.39	0.78	0.56
CRE_A	1,205.65	100.33	4,156.39	3,109.87	145.63	287.45	210.48	5,567.89
CRE_C	830.45	84.59	255.67	325.41	224.20	320.35	165.39	2,801.39
DEGEN2	15.89	2.48	5.01	39.64	15.67	-	9.20	16.86
E226	1.31	0.25	0.95	0.69	0.76	-	0.86	2.21
FFFFF800	4.04	0.49	3.31	1.39	1.98	-	2.48	15.90
ISRAEL	0.12	0.12	0.16	0.17	0.14	0.15	0.19	0.41
ITEST2	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002
ITEST6	0.002	0.002	0.003	0.002	0.003	0.003	0.003	0.003
KLEIN1	0.02	0.03	0.07	0.05	0.04	0.06	0.05	0.05
KLEIN2	2.25	0.45	0.46	1.42	1.07	0.77	1.63	1.29
KLEIN3	24.05	6.80	3.68	17.75	19.33	7.34	71.12	15.22
LOTFI	0.27	0.12	0.39	0.16	0.20	0.25	0.23	0.75
OSA-07	8.95	6.31	14.07	3.86	22.11	14.11	9.54	-
SC50A	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
SC50B	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
SC105	0.03	0.02	0.03	0.03	0.03	0.03	0.04	0.04
SC205	0.10	0.11	0.16	0.15	0.11	0.16	0.16	0.30
SCAGR7	0.10	0.05	0.11	0.10	0.08	0.10	0.12	0.08
SCFXM1	1.73	0.40	1.43	0.82	1.32	0.84	2.03	2.92
SCFXM2	17.81	2.54	7.86	8.01	11.29	12.34	9.87	15.64
SCFXM3	45.65	7.98	48.13	28.67	33.16	29.65	-	80.67
SCORPION	0.28	0.24	0.30	0.26	0.27	0.27	-	0.41
SCRS8	-	1.30	4.83	-	-	3.73	1.62	29.40
SCTAP1	0.50	0.16	0.77	0.37	0.49	0.34	0.48	2.77
SCTAP2	5.28	3.60	38.04	6.92	7.08	5.29	6.97	58.28
SCTAP3	9.64	6.20	92.85	10.50	17.10	10.72	13.70	335.03
SHARE1B	0.58	0.09	0.22	0.26	0.32	0.20	0.41	0.37
SHARE2B	0.07	0.03	0.06	0.05	0.06	0.06	0.06	0.06
SHIP04L	0.89	0.85	3.22	0.95	0.99	1.44	1.41	2.43
SHIP04S	0.33	0.32	1.19	0.38	0.37	0.54	0.48	1.06
SHIP08L	6.39	4.01	13.14	4.65	5.86	7.99	6.25	15.79
SHIP08S	1.21	0.59	2.36	0.68	0.78	1.26	0.99	5.91
SHIP12L	9.84	9.14	33.19	9.76	11.08	13.79	12.00	39.77
SHIP12S	1.35	1.13	5.54	1.38	1.42	2.21	1.75	5.32
STOCFOR1	0.07	0.03	0.06	0.06	0.06	0.05	0.08	0.05
STOCFOR2	140.14	34.04	81.46	84.34	98.13	85.65	120.56	130.13
AVERAGE	**56.00**	**11.91**	**104.51**	**83.49**	**15.04**	**19.76**	**15.20**	**346.14**

Table 4: Number of Iterations

Name	BR	DR	GIM	LRCM	PPR	QR	SR	SER
25FV47	-	6,522	-	-	-	-	-	1,510
ADLITTLE	198	118	82	239	209	172	205	100
AFIRO	24	11	9	18	20	18	20	10
AGG	106	73	69	124	111	104	108	69

AGG2	184	141	127	167	132	157	172	122
AGG3	300	153	145	386	293	234	359	134
BANDM	1,654	544	279	1,561	1,346	-	2,467	252
BEACONFD	40	22	22	30	23	31	31	22
BLEND	170	102	104	3,069	198	7,280	271	60
BNL1	36,078	4,447	1,643	25,096	12,673	-	-	866
BNL2	-	8,517	-	-	-	-	-	-
BRANDY	4,401	360	275	1,368	1,876	999	2,165	330
CRE_A	64,132	5,487	3,098	14,856	4,867	13,985	12,045	1,801
CRE_C	52,134	5,126	2,854	21,098	14,378	18,654	11,345	1,530
DEGEN2	7,415	844	569	19,392	6,678	-	4,014	441
E226	2,687	522	285	1,585	1,664	-	1,673	235
FFFFF800	6,499	457	403	1,890	2,517	-	3,523	253
ISRAEL	371	363	150	492	416	394	505	141
ITEST2	4	4	4	4	4	4	4	4
ITEST6	4	4	4	4	4	4	4	4
KLEIN1	90	190	102	276	182	267	225	117
KLEIN2	2,515	554	231	1,623	1,245	832	1,972	466
KLEIN3	7,805	2,411	602	6,114	6,840	2,408	24,398	1,263
LOTFI	708	351	191	431	527	570	577	144
OSA-07	6,405	1,059	860	3,546	7,011	4,015	3,330	-
SC50A	33	30	28	33	30	29	38	27
SC50B	29	31	30	34	31	31	34	30
SC105	73	66	53	69	81	70	106	56
SC205	141	139	124	221	155	211	235	115
SCAGR7	241	87	87	236	164	207	268	73
SCFXM1	1,756	403	388	1,025	1,486	836	2,007	286
SCFXM2	5,378	786	1,198	3,101	4,013	4,421	2,689	541
SCFXM3	7,745	1,227	2,453	4,854	4,578	4,801	-	789
SCORPION	155	112	117	157	141	154	-	111
SCRS8	-	658	348	-	-	2,639	1,155	373
SCTAP1	814	284	284	614	675	491	684	167
SCTAP2	2,283	1,132	1,378	2,653	2,348	2,300	2,329	333
SCTAP3	2,501	2,862	1,733	2,582	3,426	2,683	2,972	619
SHARE1B	1,860	200	92	762	955	493	1,134	162
SHARE2B	294	104	100	220	276	212	249	77
SHIP04L	368	227	241	500	326	508	572	211
SHIP04S	243	208	167	412	240	337	282	152
SHIP08L	3,788	469	443	1,479	2,130	2,093	1,317	377
SHIP08S	1,696	237	237	522	457	774	571	224
SHIP12L	1,795	733	731	1,751	1,668	1,892	1,460	633
SHIP12S	929	378	432	990	665	1,032	758	324
STOCFOR1	235	69	70	206	200	131	212	70
STOCFOR2	11,034	1,617	2,013	6,010	5,890	5,541	9,145	1,386
AVERAGE	**5,273.67**	**1,050.85**	**540.33**	**2,928.89**	**2,069.98**	**2,000.34**	**2,270.47**	**369.78**

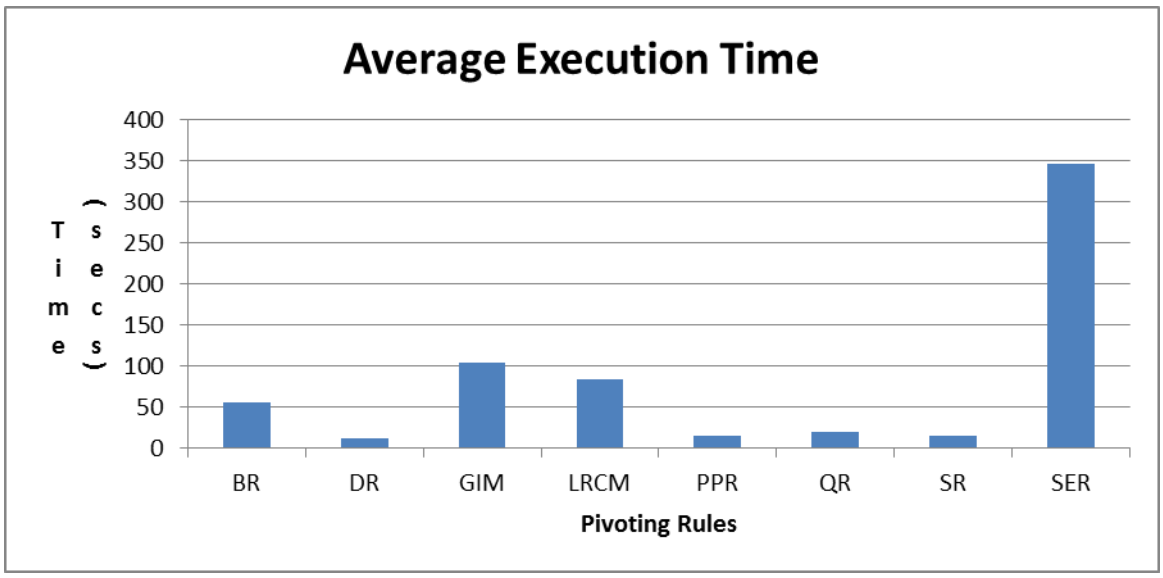

Figure 1: Average Execution Time

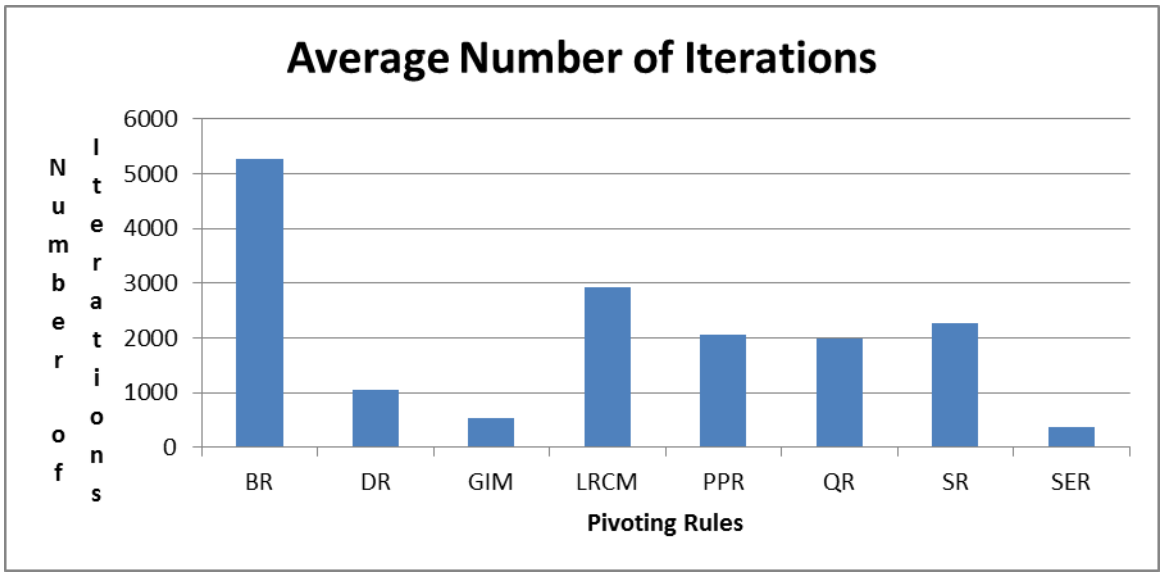

Figure 2: Average Number of Iterations

5. CONCLUSION

The selection of the entering variable is a crucial step in the revised simplex algorithm and should be carefully designed in order to economize this operation. Eight well known pivoting rules have been reviewed and compared in this paper. The computational study over 48 Netlib LPs showed that only Dantzig's rule solved all instances. Furthermore, Dantzig's rule performs better than the other pivoting rules in terms of execution time. Finally, the Steepest Edge rule requires the fewest number of iterations, but it has the worst execution time compared to the other pivoting rules.

As future work, we aim to include some other well-known pivoting rules, like Devex and some multiple pricing techniques. Furthermore, we aim to implement revised simplex algorithm in a GPU (Graphical Processing Unit). Dantzig's rule, which showed the best performance in computational terms, will be included in this implementation.

REFERENCES

[1] Bazaraa, M.S., Jarvis, J.J., & Sherali, H.D. (1990). Linear Programming and Network Flows. John Wiley & Sons, Inc.

[2] Bland, R.G. (1977). New finite pivoting rules for the simplex method. Mathematics of Operations Research, 2(2), 103–107.

[3] Carolan, W.J., Hill, J.E., Kennington, J.L., Niemi, S., & Wichmann, S.J. (1990). An Empirical Evaluation of the KORBX® Algorithms for Military Airlift Applications. Operations Research, 38(2), 240–248.

[4] Dantzig, G.B. (1963). Linear programming and extensions. Princeton, NJ: Princeton University Press.

[5] Dantzig, G.B., Orden, A., & Wolfe, P. (1953). The Generalized Simplex Method. RAND P-392-1.

[6] Gärtner, B. (1995). Randomized optimization by Simplex-type methods. PhD thesis, Freien Universität, Berlin.

[7] Gay, D.M. (1985). Electronic mail distribution of linear programming test problems. Mathematical Programming Society COAL Newsletter, 13, 10–12.

[8] Goldfarb, D., & Reid, J.K. (1977). A Practicable Steepest-Edge Simplex Algorithm. Mathematical Programming, 12(3), 361–371.

[9] Klee, V., & Minty, G.J. (1972). How good is the simplex algorithm. In O. Shisha (Ed.), Inequalities - III. New York and London: Academic Press Inc.

[10] Maros, I., & Khaliq, M.H. (1999). Advances in design and implementation of optimization software. European Journal of Operational Research, 140(2), 322–337.

[11] Ordóñez, F., & Freund, R. (2003). Computational experience and the explanatory value of condition measures for linear optimization. SIAM J. Optimization, 14(2), 307–333.

[12] Papadimitriou, C.H., & Steiglitz, K. (1982). Combinatorial Optimization: Algorithms and Complexity. Prentice-Hall, Inc.

[13] Świętanowski, A. (1998). A new steepest edge approximation for the simplex method for linear programming. Computational Optimization and and Application, 10 (3), 271–281.

[14] Thomadakis, M.E. (1994). Implementation and Evaluation of Primal and Dual Simplex Methods with Different Pivot-Selection Techniques in the LPBench Environment. A Research Report. Texas A&M University, Department of Computer Science.

[15] Vanderbei, R.J. (2001). Linear Programming: Foundations and Extensions (2nd ed.). Kluwer Academic Publishers.

[16] Zadeh, N. (1980). What is the worst case behavior of the simplex algorithm? Technical report, Department of Operations Research, Stanford University.