

# Efficiency of Simplex Method

- Worst case vs Average Case
- (Dantzig)  $m < 50, m+n < 200$

#(iterations)  $\leq$  (usually)  $\frac{3m}{2}$ , rarely  $\approx 3m$

[C]

TABLE 4.1 Average Number of Iterations Required by the Largest-Coefficient Rule

$m \backslash n$	10	20	30	40	50
10	9.40	14.2	17.4	19.4	20.2
20		25.2	30.7	38.0	41.5
30			44.4	52.7	62.9
40				67.6	78.7
50					95.2

Source: D. Avis and V. Chvátal (1978).

Table 4.2 Average Numbers of Iterations Required by the Largest-Increase Rule

$m \backslash n$	10	20	30	40	50
10	7.02	9.17	10.8	12.1	12.6
20		16.2	20.2	24.2	27.3
30			28.7	34.5	39.4
40				43.3	39.9
50					58.9

Source: D. Avis and V. Chvátal (1978).

# Worst Case Example: Klee-Minty (1970)

[v]

The example is quite simple to state:

$$(4.1) \quad \begin{aligned} & \text{maximize} && \sum_{j=1}^n 2^{n-j} x_j \\ & \text{subject to} && 2 \sum_{j=1}^{i-1} 2^{i-j} x_j + x_i \leq 100^{i-1} && i = 1, 2, \dots, n \\ & && x_j \geq 0 && j = 1, 2, \dots, n. \end{aligned}$$

It is instructive to look closely at the constraints. The first three constraints are

$$\begin{aligned} x_1 & \leq 1 \\ 4x_1 + x_2 & \leq 100 \\ 8x_1 + 4x_2 + x_3 & \leq 10,000. \end{aligned}$$

# Worst Case Example: Klee-Minty (1970)

[v]

$$(4.2) \quad \begin{aligned} & \text{maximize} && \sum_{j=1}^n 2^{n-j} x_j - \frac{1}{2} \sum_{j=1}^n 2^{n-j} \beta_j \\ & \text{subject to} && 2 \sum_{j=1}^{i-1} 2^{i-j} x_j + x_i \leq \sum_{j=1}^{i-1} 2^{i-j} \beta_j + \beta_i && i = 1, 2, \dots, n \\ & && x_j \geq 0 && j = 1, 2, \dots, n. \end{aligned}$$

$$1 \ll \beta_1 \ll \beta_2 \ll \dots \ll \beta_n$$

Largest Coefficient Rule: require  $2^n - 1$  steps

Smallest Coefficient Rule: just 1 steps.

# Worst Case Example: Klee-Minty (1970)

[c]

Using the largest-coefficient rule, we construct the following sequence of dictionaries.  
The initial dictionary:

$$\begin{array}{rcl} x_4 & = & 1 - x_1 \\ x_5 & = & 100 - 20x_1 - x_2 \\ x_6 & = & 10,000 - 200x_1 - 20x_2 - x_3 \\ \hline z & = & 100x_1 + 10x_2 + x_3. \end{array}$$

$2^3 - 1$  steps

In the first iteration, we were led to an unfortunate choice of the entering variable: had we made  $x_3$  rather than  $x_1$  enter the basis, we would have pivoted directly to the final dictionary. In view of this blunder, it is natural to question the expediency of the largest-coefficient rule: perhaps the simplex method would *always* go through only a small number of iterations if it were directed by some other rule. In fact, the largest-coefficient rule is not quite natural. More specifically, it ranks the potential candidates for entering the basis according to their coefficients in the last row of the dictionary: variables with larger coefficients appear to be more promising. But appearances are misleading and the ranking order is easily upset by changes in the scale on which each candidate is measured. For instance, the substitution

# Worst Case Example: Klee-Minty (1970)

[c]

Using the largest-coefficient rule, we construct the following sequence of dictionaries.  
The initial dictionary:

$$\begin{array}{rcl} x_4 & = & 1 - x_1 \\ x_5 & = & 100 - 20x_1 - x_2 \\ x_6 & = & 10,000 - 200x_1 - 20x_2 - x_3 \\ \hline z & = & 100x_1 + 10x_2 + x_3. \end{array}$$

$2^3 - 1$  steps

$$\bar{x}_1 = x_1, \quad \bar{x}_2 = 0.01x_2, \quad \bar{x}_3 = 0.0001x_3$$

converts the Klee-Minty problem (4.3) into the form

$$\begin{array}{rcl} \text{maximize} & \underline{100\bar{x}_1 + 1,000\bar{x}_2 + 10,000\bar{x}_3} & \\ \text{subject to} & \bar{x}_1 & \leq 1 \\ & 20\bar{x}_1 + 100\bar{x}_2 & \leq 100 \\ & 200\bar{x}_1 + 2,000\bar{x}_2 + 10,000\bar{x}_3 & \leq 10,000 \\ & \bar{x}_1, \bar{x}_2, \bar{x}_3 & \geq 0. \end{array}$$

In the first dictionary associated with this new version of (4.3), the nonbasic variable  $\bar{x}_3$  appears most attractive, and so the simplex method reaches the optimal solution in only one iteration.

# Other Pivoting Rules (independent of scale)

[MG] largest increase, steepest edge, Bland Rule - ...

## ALTERNATIVE PIVOTING RULES

Thus we are led to ranking the candidates  $x_j$  for entering the basis according to criteria that are independent of changes of scale. One criterion of this kind is the increase in the objective function obtained when  $x_j$  actually enters the basis. The

resulting rule (always choose that candidate whose entrance into the basis brings about the largest increase in the objective function) is referred to as the largest-increase rule. On the Klee-Minty examples (4.2), the largest-increase rule leads the simplex method to the optimal solution in only one iteration, as opposed to the  $2^n - 1$  iterations required by the previously used largest-coefficient rule. However, the new rule does not always lead to a small number of iterations: R. G. Jeroslow (1973) constructed LP problems that are to the largest-increase rule what the Klee-Minty problems are to the largest-coefficient rule. (More precisely, the number of iterations required by the largest-increase rule grows exponentially with  $m$  and  $n$ .) Again, these examples exploit the myopia inherent in the simplex method. It is conceivable that every easily implemented rule for choosing the entering variable can be tricked in a similar way into requiring very large numbers of iterations.



# Other Pivoting Rules (independent of scale)

Which of the two rules is better? On problems arising from applications, the number of iterations required by the largest increase is *usually* smaller than the number of iterations required by the largest coefficient. Simulation experiments lead to a similar outcome (see Table 4.2).

TABLE 4.1 Average Number of Iterations Required by the Largest-Coefficient Rule

$m \backslash n$	10	20	30	40	50
10	9.40	14.2	17.4	19.4	20.2
20		25.2	30.7	38.0	41.5
30			44.4	52.7	62.9
40				67.6	78.7
50					95.2

Source: D. Avis and V. Chvátal (1978).

Table 4.2 Average Numbers of Iterations Required by the Largest-Increase Rule

$m \backslash n$	10	20	30	40	50
10	7.02	9.17	10.8	12.1	12.6
20		16.2	20.2	24.2	27.3
30			28.7	34.5	39.4
40				43.3	39.9
50					58.9

Source: D. Avis and V. Chvátal (1978).

# Other Pivoting Rules (independent of scale)

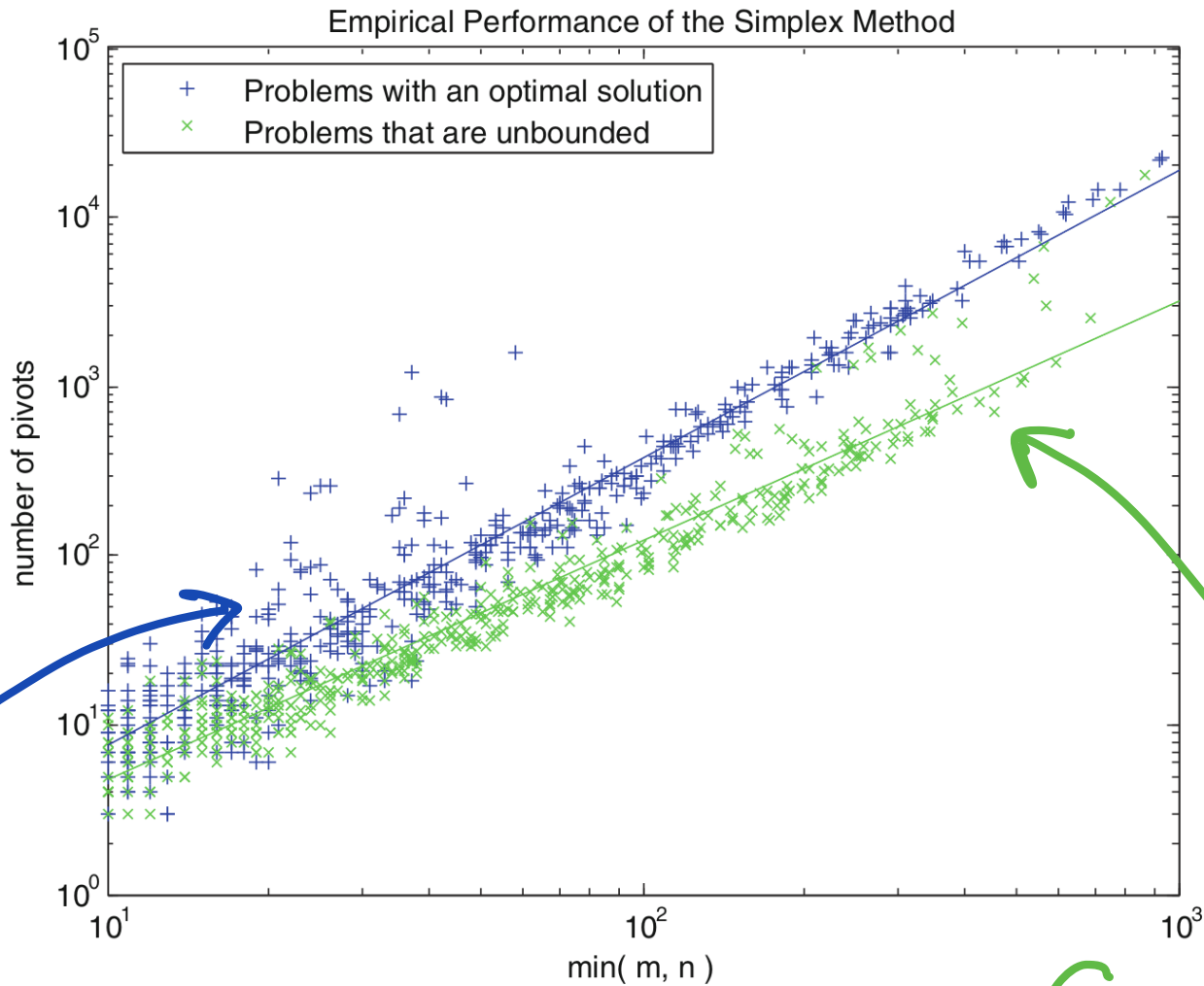
Which of the two rules is better? On problems arising from applications, the number of iterations required by the largest increase is *usually* smaller than the number of iterations required by the largest coefficient. Simulation experiments lead to a similar outcome (see Table 4.2).

Nevertheless, as the largest-coefficient rule takes less time to execute than the largest increase, it is the former that usually wins in terms of total computing time. More generally, the number of iterations is a poor criterion for assessing the efficiency of a rule for choosing the entering variable. It is the total computing time that counts, and rules that tend to reduce the number of iterations often take too much time to execute. In this light, even the largest-coefficient rule is found too time-consuming and therefore rarely, if ever, used in practice. The choice of entering variables in efficient implementations of the simplex method is influenced by the logistics of handling large problems on a computer; this matter will be studied in Chapter 7.



# Some Empirical Studies

[V]



Blue dots

$$\# \approx 0.15 \min(m, n)^{1.70}$$

Green dots

$$\# \approx 1.80 \min(m, n)^{1.42}$$

# Some Theoretical Results

[MG]

Later on, very slow examples of a similar type were discovered for many other pivot rules, among them all the rules mentioned above. Many people have tried to design a pivot rule and prove that the number of pivot steps is always bounded by some polynomial function of  $m$  and  $n$ , but nobody has succeeded so far. The best known bound has been proved for the following simple randomized pivot rule: Choose a random ordering of the variables at the beginning of the computation (in other words, randomly permute the indices of the variables in the input linear program); then use Bland's rule for choosing the entering variable, and the lexicographic method for choosing the leaving variable. For every linear program with at most  $n$  variables and at most  $n$  constraints, the expected number of pivot steps is bounded by  $e^{C\sqrt{n \ln n}}$ , where  $C$  is a (not too large) constant. (Here the expectation means the arithmetic average over all possible orderings of the variables.) This bound is considerably better than  $2^n$ , say, but much worse than a polynomial bound.

This algorithm was found independently and almost at the same time by Kalai and by Matoušek, Sharir, and Welzl. For a recent treatment in a somewhat broader context see

B. Gärtner and E. Welzl: Explicit and implicit enforcing—randomized optimization, in *Lectures of the Graduate Program*

# Some Theoretical Results

[MG]

In spite of the Klee–Minty cube and similar artificial examples, the simplex method is being used successfully. Remarkable theoretical results indicate that these willful examples are rare indeed. For instance, it is known that if a linear program in equational form is generated in a suitable (precisely defined) way at random, then the number of pivot steps is of order at most  $m^2$  with high probability. More recent results, in the general framework of the so-called *smoothed complexity*, claim that if we take an arbitrary linear program and then we change its coefficients by small random amounts, then the simplex method with a certain pivot rule reaches the optimum of the resulting linear program by polynomially many steps with high probability (a concrete bound on the polynomial depends on a precise specification of the “small random amounts” of change). The first theorem of this kind is due to Spielman and Teng, and for recent progress see

R. Vershynin: Beyond Hirsch conjecture: Walks on random polytopes and the smoothed complexity of the simplex method, preprint, 2006.

# Some Theoretical Results

Polynomial bounded algorithms for

Linear Programming does exist!

Interior Method which goes through  
the interior of the feasible set.

(Simplex Method goes through  
the boundary of the feasible set.)