Technical Report CS75001-R

MACHINE-INSPIRED ENHANCEMENTS OF THE SIMPLEX ALGORITHM

James E. Kalan

January 1975

Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, Virginia

ABSTRACT

Using spare storage and over-lapping overhead available during the generation of the pricing form in the revised-simplex algorithm allows the production of additional reduced costs of variables which may be effectively used to avoid redundant pivots and to predict actual gains. Results from a simulation indicate a dramatic improvement over all other meaningful selection methods.

Keywords and Phrases: linear programming, pricing strategies, degeneracies, pricing tactics.

1.0 Introduction

The simplex algorithm has enjoyed startling enhancements within its short life. Theoretical improvements (e.g., revised-simplex, factored inverse, implicit constraints) and practical discoveries (e.g., sparsity, super-sparsity, Hellerman-Rarick reinversion) have electrified the performance of simplex codes allowing solution of an ever-increasing class of linear (and non-linear) programs.

Of great interest is the attempt to improve in a computationally-efficient way the quality of iterations. A popular and fertile area under current scrutiny is that of candidate selection. Multiple-pricing, scaling, pivot prediction, etc., as reported by Orchard-Hays [3,p.112], Tomlin [5], and Harris [1], among others, have led to more judicious pricing and, in turm, a more effictive algorithm. In particular, the 'inertial gradient' pricing technique developed by Harris [1] has demonstrated its effectiveness and is in use in a number of commercial systems.

In this paper we approach the goal of candidate-selection improvement in an inverse fashion. Beginning with the identification of a portion of the simplex algorithm wherein surplus computer resources are available or can be diverted cheaply, we examine a number of functionals which can be computed cheaply. As it turns out, at least two of these functionals can be utilized to forecast redundant pivots and the resulting gain.

2.0 Algorithm Sketch

The standard LP (linear program) may be considered to be in the form

(2.1) maximize L_f (some f: $1 \leq f \leq m$) subject to

(2.2) $$\sum_{j=1}^{n} x_j a^j + L = b$$

$$x_j \geq 0 \qquad (1 \leq j \leq n)$$
$$L_i \geq 0 \qquad (1 \leq i \leq m,\ i \neq f)$$
$$L_f \text{ unconstrained}$$

where $L = (L_1, \ldots, L_m)$ is the vector of logical variables, $b = (b_1, \ldots, b_m)$ is popularly known as the right-hand side, and the n vectors a^j are m-dimensional and may be thought of as the column vectors of the constraint matrix.[1]

The revised simplex algorithm for the solution of 2.1 may be superficially described by the following:

Step 0. Identify a basic solution $\beta = B^{-1}b$ where B is an m x m matrix composed of columns corresponding to the basic variables.

Step 1. Define the vector C by

1a. $$\left.\begin{array}{ll} C_i = -1 & \text{if } \beta_i < 0 \\ = 0 & \text{if } \beta_i \geq 0 \end{array}\right\} \quad (i \neq f)$$

1b. if all $C_i = 0$ in 1_a define $C_f = 1$. Else $C_f = 0$.

(This is the construction of the cost form and generates a cost vector for either Phase I--current solution infeasible--or Phase II--current solution feasible.)

Step 2. Compute the pricing form π by

$$\pi = B^{-T}C$$

(If B^{-1} is given explicity, then this step is either cheap--Phase I--or free--Phase II. If B^{-1} is specified otherwise, then this step is very expensive.)

---

[1] Note: we are not recommending that notions of equality, simple-bounds, etc. be cast into 2.2; indeed such non-structural constraints should be handled implicitly. This uncluttered version of the LP is given merely to expedite understanding of the following considerations.

3.0 Candidate Selection

The selection of a candidate (or candidates) for basis entry is made on the notion of promise--d_w. In fact, the gain, G_w, once the variable is pivoted in is given by

$$(3.1) \quad G_w \equiv d_w \Theta_w = \alpha_f^w \frac{\beta_p}{\alpha_p^w}$$

The choice of w yielding the most negative d_j is within the strategy of insuring the best gain. However, since Θ_w is unknown at time of pricing, it is not clear how this selection approaches this goal. Furthermore, scaling has a direct effect on candidate selection.

If the ith row of the LP is multiplied by R_i and the jth variable is divided by C_j, then the LP takes the form

$$RACC^{-1}x + \tilde{L} = Rb$$

where R is the m x m diagonal matrix with diagonal $(R_1, \ldots, R_m)$ and C is the n x n diagonal matrix $(C_1, \ldots, C_n)$. For this scaled problem we have

$$\tilde{A}\,\tilde{x} + \tilde{L} = \tilde{b}$$

where $\tilde{A} = RAC$, $\tilde{x} = C^{-1}x$, and $\tilde{b} = Rb$. Given a basis B and the scaled basis $\tilde{B}$, we have

$$\tilde{B} = R\,B\,C_B$$

where C_B is the principal submatrix of C $\sim$ the basis variables. Since

$$\tilde{B}^{-1} = C_B^{-1} B^{-1} R^{-1} \quad ,$$

then

$$(3.2) \quad \tilde{\alpha}^j \equiv \tilde{B}^{-1}\,\tilde{a}^j = C_B^{-1} \cdot B^{-1}R^{-1} \cdot (C_j R a^j) = C_j \cdot C_B^{-1}\,\alpha^j .$$

and

$$\tilde{\beta} = \tilde{B}^{-1} b \tag{3.3}$$

$$= C_B^{-1} \beta \quad .$$

Since $d_j = \alpha_f^j$,

$$\tilde{d}_j = c_j d_j \tag{3.4}$$

and the choice of 'best d_j' is clearly seen to be dependent upon the scaling of the problem, whether done explicitly or inherent in the original model.

4.0 Pricing Functionals

Let $C^k = (C_1^k, \ldots, C_m^k)$ be arbitrary. If we define π^k via

$$\pi^k \equiv B^{-T} C^k \tag{4.1}$$

and d_j^k by

$$d_j^k \equiv (\pi^k , a^j) \tag{4.2}$$

then

$$\begin{aligned} d_j^k &= (B^{-T} C^k , a^j) \\ &= (C^k , B^{-1} a^j) \\ &= (C^k , \alpha^j) \\ &= \sum_{i=1}^{m} C_i^k \quad \alpha_i^j \end{aligned} \tag{4.3}$$

where $\alpha^j \equiv B^{-1} a^j$. Thus, the usual reduced cost is simply a linear combination of coordinates of the α-vector. Various such linear combinations appear attractive. E.g., if the pivot row, p, were known in advance at Step 2, the gain of each variable could be computed in the following way:

Define $\tilde{\pi} = B^{-1} e^{p}$ [3];

$$\tilde{d}_j = (\tilde{\pi}, a^j)$$

$$= \alpha_p^j ;$$

$$G_j = d_j \cdot \frac{\beta_p}{\alpha_p^j}$$

We explore various reduced costs.

Case 1

Let C^1 be defined as in Step 1. Then d_j^1 is the usual reduced cost for candidate selection.

Case 2

Define C^2 by

$$C_i^2 = 1 \quad \text{if } \beta_i = 0$$

$$= 0 \quad \text{otherwise}$$

Then

$$d_j^2 \equiv (\pi^2, a^j)$$

$$= \sum_{\beta_i=0} \alpha_i^j \quad .$$

If we assume that subsequent pivoting is such that feasible basic values are to be maintained then $d_j^2 > 0$ means that a subsequent pivot must be on a row for which $\beta_p = 0$. This information may be used to exclude otherwise attractive variables from candidate selection if possible. Unfortunately, $d_j^2 \leq 0$ does not imply that pivoting will not occur on zero values of β_p. However, we can get some idea of how well this will work by a probabilistic argument.

Assume that $\beta_i = 0$ for $1 \leq i \leq N$ and that $\beta_i \neq 0$ for $i > N$. (No generality is lost by this assumption.) Suppose that the first N coordinates of α independently

---

[3] e^k is the k-th unit vector.

obey the same continuous distributions symmetric about 0, but that there is probability 1- σ(σ is the density of the α-vector) that each α_i is zero. The following probabilities are then obvious:

$$P(\alpha_i < 0) = \sigma/2$$

$$P(\alpha_i > 0) = \sigma/2$$

$$P(\alpha_i = 0) = 1-\sigma$$

$$P(\alpha_i \leq 0) = \frac{\sigma}{2} + 1 - \sigma = 1 - \frac{\sigma}{2}$$

$$P(\alpha_i \leq 0, \text{ all } i) = (1 - \frac{\sigma}{2})^N$$

$$P(\alpha_i > 0, \text{ some } i) = 1 - (1 - \frac{\sigma}{2})^N$$

$$P(\alpha_i = 0, \text{ all } i) = (1-\sigma)^N$$

$$P(\Sigma\alpha_i < 0) = P(\Sigma\alpha_i > 0)$$

The use of d_j^2 to catch degeneracy fails when $\alpha_i > 0$ for some i but $\Sigma\alpha_i \leq 0$.

Let A stand for "$\Sigma\alpha_i \leq 0$" and B for "$\alpha_i > 0$ some i". Then

$$P(A) = P_c(A \mid B)\, P(B) + P_c(A \mid \sim B)\, P(\sim B)$$

where P_c represents a conditional probability and $\sim B$ is the complement of B (Parzen, [4,p.60]). It follows that

$$P_c(A \mid B) = \frac{P(A) - P_c(A \mid \sim B)\, P(\sim B)}{P(B)}$$

Clearly, $P_c(A \mid \sim B) \equiv 1$. Using the previously given probabilities,

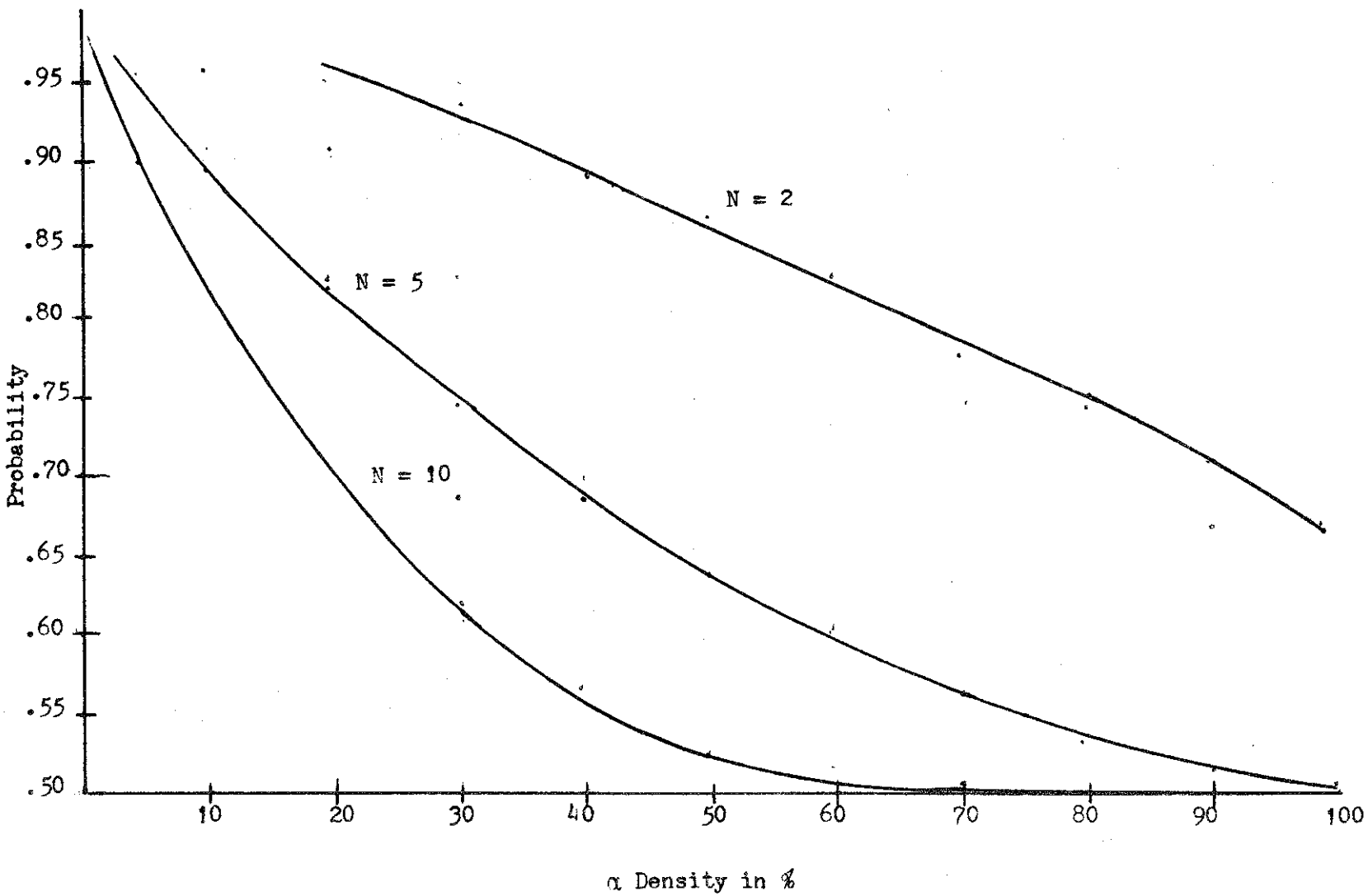
$$P_c(A \mid B) = \frac{P(A) - (1 - \frac{\sigma}{2})^N}{1 - (1 - \frac{\sigma}{2})^N}$$

Since this is the probability of failure, that of success, P_s, is.

$$P_s = \frac{1 - P(A)}{1 - (1 - \frac{\sigma}{2})^N}$$

Graph 1

Probability of Success Using d_j^2



Now,

$$P(A) = P(\Sigma\alpha_i \leq 0)$$

$$= P(\Sigma\alpha_i < 0) + P(\Sigma\alpha_i = 0)$$

$$= \frac{1 + P(\Sigma\alpha_i = 0)}{2} \quad .$$

where we use the fact that $1 = P(\Sigma\alpha_i < 0) + P(\Sigma\alpha_i = 0) + P(\Sigma\alpha_i > 0)$ and the equality of the 1st and 3rd terms. Therefore,

$$P_s = \frac{1 - P(\Sigma\alpha_i = 0)}{2\left[1 - \left(1 - \frac{\sigma}{2}\right)^N\right]}$$

Clearly, $P(\Sigma\alpha_i = 0) \geq P(\alpha_i = 0, \text{ all } i)$. However, strict inequality cannot hold since the α_i otherwise obey a continuous distribution. Thus

$$P_s = \frac{1 - (1 - \sigma)^N}{2\left[1 - \left(1 - \frac{\sigma}{2}\right)^N\right]}$$

Graph 1 gives P_s for a variety of N and α densities ranging from 0 to 1.

For $N > 10$ and $\sigma > .6$, $P_s \sim .5$, and in all cases $P_s \geq 1/2$. Thus, the use of d_j^2 to avoid degenerate pivoting shows great promise. (There are, at present, no alternatives!)

Case 3

Define C^3 by

$$C_i^3 = 1 \quad \text{all } i.$$

Then

$$d_j^3 = \Sigma\alpha_i^j \quad .$$

Certainly, the more positive d_j^3 the more positive some of the α^j must be. In this way the "size" of the α-vector may be estimated in selecting candidates.

Moreover, if column factors C_j had been applied to every column then for the scaled problem

$$\tilde{d}_j^1 = C_j d_j^1$$

and $$\tilde{d}_j^3 = C_j d_j^1 \quad .$$

Thus the ratio

$$\frac{\tilde{d}_j^1}{|\tilde{d}_j^3|} = \frac{d_j^1}{|d_j^3|}$$

is <u>invariant</u> to non-basic column scaling and may be used in place of d_j^1 for column selection. (Unfortunately, by 3.2, column scales of basic variables still has effect on this functional, but at least a portion of the capricousness is removed.)

<u>Case 4</u>

Define C^4 by

$$C_i^4 = \frac{1}{\beta_i} \qquad (\beta_i \neq 0)$$

$$= 0 \qquad (\beta_i = 0)$$

Then

$$d_j^4 = (\pi^4, a^j)$$

$$= \sum_{\beta_i \neq 0} \frac{\alpha_i^j}{\beta_i}$$

As with d_j^3, d_j^4 may be used to estimate the "size" of the α-vector. Furthermore, the ratios α_i^j/β_i offer a better estimate of this size then the possibly meaningless simple sum of α_i^j.

Of interest is the ratio d_j^1/d_j^4 regarding the effect of row and column scaling. In fact, if row scales R_i and column scales C_j have been applied, then for the scaled problem by (3.2) and (3.3)

$$\frac{d_i^1}{d_4^1} = \frac{R_f\ C_j\ \alpha_f^j}{\sum_{\beta_i \neq 0} \frac{R_i\ C_i\ C_{B_i}^{-1}\ \alpha_i^j}{R_i\ C_{B_i}^{-1}\ \beta_i}}$$

$$= R_f \cdot \frac{\alpha_f^j}{\sum_{\beta_i \neq 0} \alpha_i^j / \beta_i}$$

$$= R_f\ \frac{d_j^1}{d_j^4} \quad .$$

Since R_f involves only multiplying the cost row and this multiplier is the same for every such ratio we may without loss of generality assume that $R_f = 1$. (This is in accordance with the practice within proprietary systems that free rows are never scaled). Thus, the ratio $\frac{d_j^1}{d_j^4}$ is invariant to row and column scaling and on that basis alone deserves consideration as a pricing functional.

This ratio merits further exploration. If we define

$$\Theta_j^+ = \min_{\alpha_i^j > 0} \frac{\beta_i}{\alpha_i^j}$$

$$= \frac{\beta_p}{\alpha_p^j}$$

and $$\mu_j = \sum_{\substack{i \neq p \\ \beta_i > 0}} \frac{\alpha_i^j}{\beta_i}$$

then

$$\frac{d_j^1}{d_j^4} = \frac{d_j^1}{1/\Theta_j^+ + \mu_j}$$

$$= \frac{\frac{1}{d_j}\ \Theta_j^+}{1 + \Theta_j^+ \mu_j}$$

If $\Theta_j = 0$, then the gain $G_j = 0$.

If, however, $\Theta_j > 0$ (as is made more probable if d_j^2 is used effectively) then $\Theta_j = \Theta^+_j$ and

$$\frac{d_j^1}{d_j^4} = \frac{G_j}{1 + \Theta_j \mu_j} .$$

If d_j^1/d_j^4 is highly attractive (i.e., negative and of 'great' magnitude) then either G_j is highly attractive or $\Theta_j \mu_j = -1 + \varepsilon$, ε 'small'. By the definition of μ_j, the number of constraining pivots (i.e., number of β-values which will decrease) is matched by the number of β-values which will increase. Thus, the feeling is that even if the gain is not that great, the increase in future pivots will subsequently compensate. The same considerations applied to the case when $d_j^4 < 0$ lead us to the conclusion that

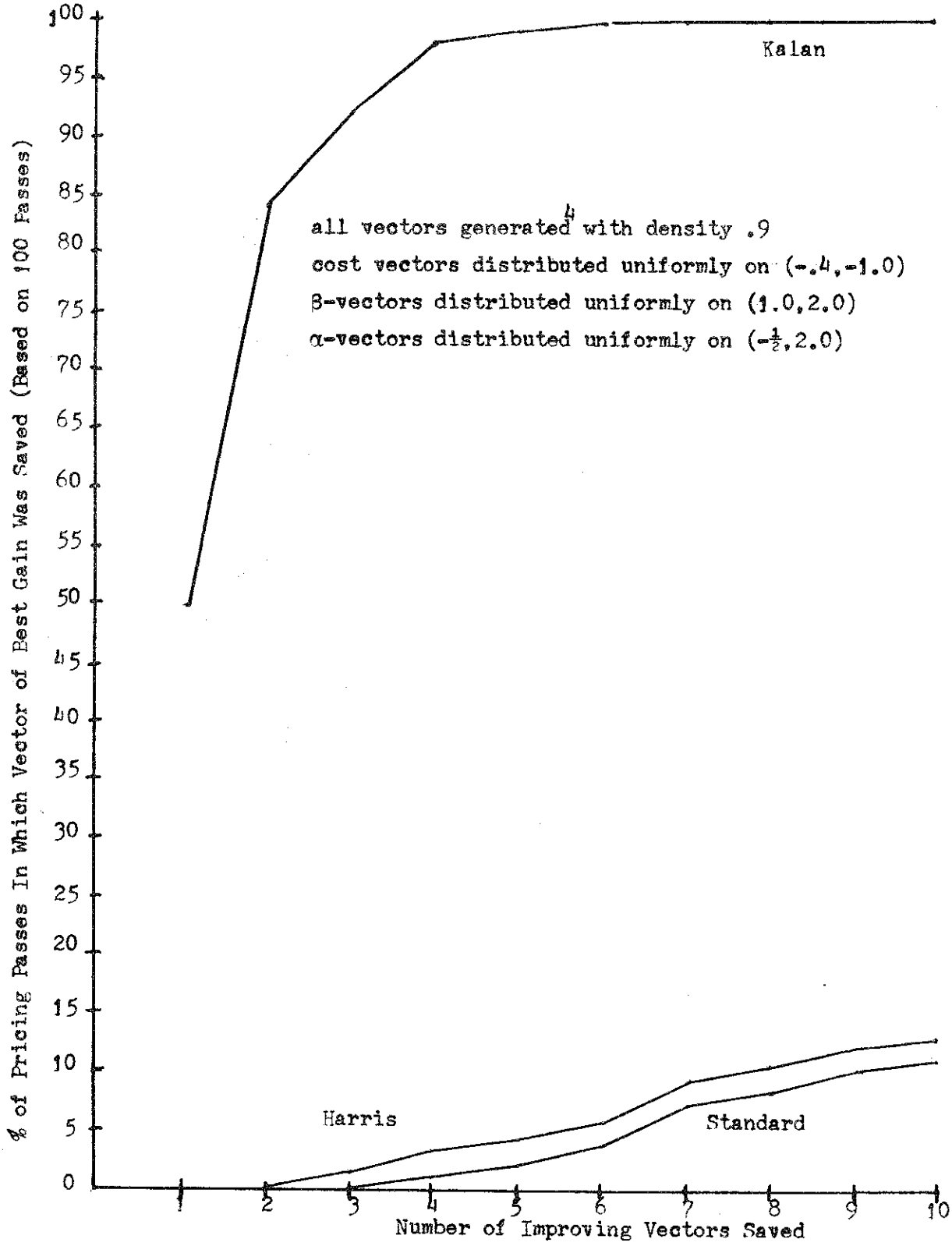
$$\left| \frac{d_j^1}{d_j^4} \right|$$

should be a 'good' indication of the resulting pivot.

5.0 Preliminary Results - Continuations

The foregoing strongly suggest the use of d_j^2 and d_j^4 in pricing strategies as an alternative to the usual selection method. To obtain a feeling for the effectiveness of d_j^2 and d_j^4 we wrote a quick-and-dirty simulator which generates cost vectors, β-vectors, and α-vectors with given densities and distributions and feeds them into a pricing step and subsequent pivot. The statistic measured over a number of 'iterations' was the number of best candidates to save in order to insure that the vector of best possible gain was among them. Graph 2 records these results in terms of the (normalized) frequency with which the vector of greatest gain was so included, as a function of the number of vectors saved.

Graph 2



[4] Psuedo-random number generation implemented as in Nance and Overstreet [2].

In each case pricing was continued until 50 vectors were saved. The vectors were ranked, for purpose of comparison, with 3 strategies:

Standard: Reject any for which $d_j^1 \geq 0$. Order 50 saved on increasing d_j^1.

Harris: Reject any for which $d_j^1 \geq 0$. Order 50 saved on increasing $d_j^1 / \sqrt{\Sigma(\alpha_i^j)^2}$ [5]

Kalan: Reject any for which $d_j^1 \geq 0$. Order 50 saved on increasing $d_j^1/|d_j^4|$. Then 1st consider those for which $d_j^2 \leq 0$.

As can be seen, the third method is overwhelmingly best.

There are many and extensive tests to be performed before an enduring evaluation can be made. The true measure of what is best is the cost of an optimization, not number of iterations, etc. There are many knobs to turn in fine-tuning the performance of an optimizer in obtaining maximal advantage of a sub-step. Towards an exhaustive comparative study of what is proposed here, we are engaged in an extensive simulated performance using all known pricing techniques together with experimental ones (which are tested simply because they can be computed cheaply; e.g., the use of d_j^3) and a possibly less-thorough analysis on an LP system modified to experiment with different strategies.

6.0 Conclusions

The recommendations here offer great promise and should be considered as an alternative to pricing techniques in vogue. The success reported so far should also act as a stimulus to encourage future research into the use of other pricing functionals where they have as the initial origin the fact that they can be computed cheaply.

---

[5] A promising scheme recommended by Harris [1].

REFERENCES

1. Harris, P.M.J., "Pivot Selection Methods of the Devex LP Code", British Petroleum Company Limited, 1972.

2. Nance, R. E. and Overstreet, C. L., "The Mixed Method of Random Number Generation: A Tutorial", Virginia Polytechnic Institute and State University, Technical Report CS74013-T, 1974.

3. Orchard-Hays, W., Advanced Linear Programming Computing Techniques, McGraw-Hill, 1968.

4. Parzen, E., Modern Probability Theory and Its Applications, John Wiley, 1960.

5. Tomlin, J.A., "Scaling Factors in Linear Programming", Stanford University, Operations Research House, Technical Report No. 73-20, 1973.