FEW

subfaculteit der econometrie

RESEARCH MEMORANDUM

TILBURG UNIVERSITY
DEPARTMENT OF ECONOMICS
Postbus 90153 - 5000 LE Tilburg
Netherlands

FEW
164

Shortest paths for simplicial algorithms

G. van der Laan,[1] A.J.J. Talman,[2]
L. Van der Heyden[3]

December 1984

1) Department of Actuarial Sciences and Econometrics, Free University, Amsterdam, The Netherlands

2) Department of Econometrics, Tilburg University, Tilburg, The Netherlands

3) School of Organization and Management, Yale University, New Haven, Connecticut, U.S.A

Shortest paths for simplicial algorithms*)

by

G. van der Laan, A.J.J. Talman and L. Van der Heyden

Abstract

Several variable dimension simplicial restart algorithms have been proposed for solving systems of nonlinear equations. Each different algorithm is characterized by the number of rays along which the starting point can be left. Computational results thusfar suggest that the 2^n and 3^n-1 ray algorithm are favorable to the 2n ray algorithm whereas the other algorithms including Merrill's homotopy method perform worse than the 2n ray algorithm. In this paper, however, we show that the expected length of the shortest paths for both the 2^n and the 2n ray algorithm is the same when the K' triangulation underlies the algorithms and the solution point is uniformly distributed in the unit cube. These results are confronted with the computational results concluding that the latter must be considered with some caution. Also the expected shortest path length for Merrill's algorithm is calculated and compared with the others.

Keywords: Simplicial subdivision, shortest path, piecewise linear approximation.

*) This research is part of the VF-Program "Equilibrium and disequilibrium in demand and supply"

SHORTEST PATHS FOR SIMPLICIAL ALGORITHMS

1. Introduction

Variable dimension simplicial algorithms for solving an n-dimensional system of nonlinear equations $f(x) = 0$ trace a piecewise linear path in a subdivision of R^n which leads from an abritrarily chosen starting point v to an approximate solution x^*. When the accuracy of the approximate solution is not sufficient, the algorithm can be restarted in x^* with a finer simplicial subdivision of R^n, inducing a refinement of the piecewise linear approximation to f. The algorithms follow the piecewise linear path by generating a path of simplices of varying dimension. The various algorithms are characterized by the number of one-dimensional rays along which the starting point v can be left. Along these rays efficient movements can be made by generating a path of one-dimensional simplices. Building up the dimension of the simplices, not necessarily monotonically, a path of simplices of varying dimension is generated until an n-dimensional simplex is found which yields a zero of the piecewise linear approximation to f. Starting with the n+1 ray algorithm [5] and the 2n ray algorithm [6], [8], algorithms were developed with 2^n rays [12], 2 rays [9], [13], and 3^n-1 rays [3].

The efficiency of the various algorithms has been investigated in e.g. [3], [4] and [1]. Because the 2 ray algorithm can easily exploit separability and sparsity but does not seem to work very well generally, in these studies only the n+1, 2n, 2^n and 3^n-1 ray algorithms have been compared. In general the n+1 ray algorithm is superseded by the other algorithms. Moreover the computational results show that the 2^n and 3^n-1 ray algorithms seem to be favorable to the 2n ray algorithm, although the results do not permit us to be conclusive. The 2^n and 3^n-1 ray algorithms differ only slightly. Whereas in the references mentioned above the number of function evaluations to find an approximate solution is compared, in this paper we make a theoretical comparison between the 2n and 2^n ray algorithms. These two algorithms can be seen as the extreme cases of a class of 3^n-1 ray algorithms. An element of this class is characterized by a variable γ between 0 and 1/n. For $\gamma = 0$ we obtain the 2^n ray algorithm and for $\gamma = 1/n$ the 2n ray algorithm. We only compare the two extreme cases because for the other elements in the class the

calculations become much more complicated and seem to be a hard task. Moreover, for γ not close to 0 or 1/n, the computational results for the 3^n-1 ray algorithms do not differ very much from those for the 2^n ray algorithm. The n+1 ray algorithm is not considered for two reasons. Firstly, also in this case the calculations become rather complicated, and secondly the computational results are much worser than with the other algorithms.

Both the 2n and 2^n ray algorithms usually utilize the K' triangulation of R^n, originally proposed by Todd [10], and obtained by reflecting the well known Freudenthal triangulation over the orthants. Taking this triangulation, Broadie [1,2] calculated for the 2^n ray algorithm the shortest path through this subdivision of R^n to reach an a priori given point w starting at the origin. This shortest path is the number of generated simplices if the underlying problem is given by $f(x) = w-x$. Notice that for (non)linear problems the number of function evaluations equals the number of generated simplices, since at each new simplex a new vertex is coming in for which the function must be evaluated in order to determine which vertex must be replaced at the next step.

In this paper we compare the shortest paths for the 2n and 2^n ray algorithms. By averaging these numbers over all points uniformly distributed in a cube, we obtain a formula for the expected length of the shortest paths. It will appear that this expected length is the same for both algorithms. These calculations are done in the next section. In section 3 we discuss several questions which are related to this analysis. Since the choice of a cube is rather arbitrarily, we are also concerned with the expected length of the shortest path when averaging over points uniformly distributed in a ball or a octahedron. Further the results of the theoretical calculations are confronted with the computational results. This is rather interesting because there seems to be a discrepancy between the shortest path calculations and the computational results reported in the literature. Another point of discussion is the amount of work to be done in a given dimension. In higher dimensions much more work has to be done in updating the system of linear equations at each linear programming step. So, algorithms which are working rather long in low dimensions are more attractive than algorithms which are working most of the time in high dimensions. Further we give the expec-

ted length of the shortest path when using the so-called J' triangulation, recently proposed by Todd [11]. Finally, the shortest path for Merrill's algorithm is calculated.

2. The calculation of the shortest paths

The K' triangulation of R^n proposed in [10], is obtained from the Freudenthal triangulation of R^n restricted to the positive orthant, by reflection across the axes. For a grid size δ, the orthant

$$X(s) = \{x \in R^n \mid s_i x_i \geq 0,\ i = 1,\ldots,n\}$$

with $s_i \in \{+1,-1\}$ for all i, is subdivided in n-dimensional simplices $\sigma(y^1,\pi,s)$ being the convex hull of the n+1 points $y^1,\ldots,y^{n+1}$ in X(s) with all components of y^1 integer multiples of δ, π a permutation of the elements of $I_n = \{1,2,\ldots,n\}$ and $y^{i+1} = y^i + \delta s_{\pi_i} e(\pi_i)$, $i = 1,\ldots,n$ where e(j) is the j-th unit vector. For simplicity we take the origin as the starting point for the algorithms. When a point v is chosen to be the starting point the triangulation is translated such that $\{v\} + X(s)$ is subdivided as above.

Let $f(x) = 0$ be a system of n nonlinear equations and $\bar{f}$ the piecewise linear approximation of f induced by the K' triangulation. To find an approximate zero to f, the 2^n ray algorithm follows a piecewise linear path such that for each x on the path the complementarity conditions

$$\begin{aligned} & x_i \bar{f}_i \geq 0 \\ & \qquad\qquad i = 1,\ldots,n \\ & \bar{f}_i(x)[\max_j |x_j| - |x_i|] = 0 \end{aligned} \tag{2.1}$$

hold. For $s_i = \operatorname{sign} x_i$ if $|x_i| = \max_j |x_j|$ and $s_i = 0$ if $|x_i| < \max_j |x_j|$, let $|s|$ be the number of zero elements of $s = (s_1,\ldots,s_n)^T$. Then a path in the $(|s| + 1)$-dimensional region

$$\{x \in R^n \mid s_i x_i = \max_j |x_j| \text{ if } s_i \neq 0\}$$

is followed as long as $\bar{f}_i(x) \neq 0$ for all i with $s_i \neq 0$ and $|x_i| < \max_j |x_j|$ if $s_i = 0$. As soon as $\bar{f}_i(x)$ becomes equal to zero for some i with $s_i \neq 0$, say index h, then s_h is set equal to zero and the dimension

is increased. As soon as $|x_i|$ becomes equal to $\max_j |x_j|$ for some i with $s_i = 0$, say index k, then s_k is set equal to sign x_k and the dimension is decreased. So a piecewise linear path is followed in regions of varying dimension. In a t-dimensional region the path is followed by generating a path of adjacent t-dimensional simplices.

For the K' triangulation with grid size $\delta = 1$ we will calculate the shortest path to reach an a priori given point w, i.e. the number of simplices encountered by following the path from the origin to w. The last generated simplex is the simplex in which w lies and is denoted by $\sigma(\bar{w},\pi,s)$. Observe that $\bar{w}$ is the integer part of w. Since the K' triangulation is symmetric around the axes and diagonals, we can restrict ourselves to points w in the subset

$$W = \{x \in R^n \mid 0 \leq x_1 \leq x_2 \leq \ldots \leq x_n\} .$$

An approximate formula is derived by observing that on the path the conditions (2.1) must hold with $\bar{f}(x) = f(x) = w-x$. The path leaves the origin $\underline{0}$ in the direction $e = (1,1,\ldots,1)^T$ until the point $w_1 e$ is reached. Since the path from $\underline{0}$ to $w_1 e$ is in the one-dimensional region $\{x \in R^n \mid x_i = \max_j |x_j|, i = 1,2,\ldots,n\}$ a path of one-dimensional simplices is generated and hence w_1 steps have to be done to reach $w_1 e$. Then the path goes in the direction $e-e(1)$ until $(w_1,w_2,\ldots,w_2)^T$ is reached. This takes $2(w_2-w_1)$ steps since now a path of two-dimensional simplices is generated. Generally, for $i = 2,\ldots,n$, it takes $i(w_i-w_{i-1})$ steps to go from $(w_1,w_2,\ldots,w_{i-1},\ldots,w_{i-1})^T$ to the point $(w_1,w_2,\ldots,w_i,\ldots,w_i)^T$. So, the length of the path, i.e. the total number of steps of the algorithm becomes

$$w_1 + \Sigma_{i=2}^n i(w_i-w_{i-1}) = (n+1)w_n - \Sigma_{i=1}^n w_i . \qquad (2.2)$$

Observe that in the t-dimensional region $\{x \in R^n \mid x_i = \max_j |x_j|, i = t,\ldots,n, x_i < \max_j |x_j|, i = 1,\ldots,t-1\}$ the number of steps is equal to $t(w_t-w_{t-1})$ with $w_0 = 0$. As shown in Broadie [1], this approximate number differs from the exact formula by only $n + 1 - \pi^{-1}(n) - \Sigma_{i=1}^n (w_i-\bar{w}_i)$, which can be ignored for w sufficiently far from the origin.

Similarly as above we can derive an approximate formula for the 2n ray algorithm. This algorithm follows a piecewise linear path such that for each x on the path the complementarity conditions

$$\begin{aligned} & x_i \bar{f}_i(x) \geq 0 \\ & \qquad\qquad\qquad\qquad i = 1,\ldots,n \\ & x_i [\max_j |\bar{f}_j(x)| - |\bar{f}_i(x)|] = 0 \end{aligned} \tag{2.3}$$

hold. Observe that the conditions (2.3) are dual to the conditions (2.1). From (2.3) it follows that with $s_i = \text{sign } x_i$ if $x_i \neq 0$ and $s_i = 0$ if $x_i = 0$, a path in the $(n-|s|)$-dimensional region

$$\{x \in R^n \mid s_i x_i \geq 0,\ i = 1,\ldots,n\}$$

is followed as long as $|\bar{f}_i(x)| < \max_j |\bar{f}_j(x)|$ for all i with $s_i = 0$ and $x_i \neq 0$ for all i with $s_i \neq 0$. If $|\bar{f}_i(x)|$ becomes equal to $\max_j |\bar{f}_j(x)|$ for some i with $s_i = 0$, say index h, then s_h is set equal to sign $\bar{f}_i(x)$ and the dimension is increased. If x_i becomes equal to zero for some i with $s_i \neq 0$, then s_i is set equal to zero and the dimension is decreased.

Again, let w be a point in W and $\sigma(\bar{w},\pi,s)$ the simplex containing w with which the algorithm terminates. Now the path followed for $\bar{f}(x) = f(x) = w-x$, leaves the origin by increasing the component h for which $w_h = \max_j w_j$. Since $w_n \geq w_{n-1} \geq \ldots \geq w_1 \geq 0$ and taking the highest component in case of an equality we have that h = n. So, a path of one-dimensional simplices in $\{x \in R^n \mid x_n \geq 0,\ x_i = 0,\ i = 1,\ldots,n-1\}$ is generated until $w_n - x_n$ becomes equal to w_{n-1}, i.e. the path goes in the direction e(n) until the point $(0,\ldots,0,w_n-w_{n-1})^T$ is reached. For $\delta = 1$ this takes $w_n - w_{n-1}$ steps. Then the path goes in the direction e(n)+e(n-1) $w_{n-1}-w_{n-2}$ steps until the point $(0,\ldots,0,w_{n-1}-w_{n-2},w_n-w_{n-2})^T$ is reached. Since now two-dimensional simplices are generated this takes $2(w_{n-1}-w_{n-2})$ steps. In general, for $i = n-1,n-2,\ldots,1$, the path goes from $(0,\ldots,0,w_{i+1}-w_i,\ \ldots,w_n-w_i)^T$ to the point $(0,\ldots,0,w_i-w_{i-1},\ldots,w_n-w_{i-1})^T$ with $w_0 = 0$, i.e. in the (n-i+1)-dimensional region $\{x \in R^n \mid x_j \geq 0,\ j = i,\ldots,n,\ x_j = 0,\ j = 1,\ldots,i-1\}$ the path goes in the direction $\Sigma_{j=i}^n e(j)$. Since all components $j = i,\ldots,n$ are increased with $w_i - w_{i-1}$ this takes $(n-i+1)(w_i-w_{i-1})$ steps. So, the approximate length of the

path becomes

$$\Sigma_{i=1}^{n} (n-i+1)(w_i - w_{i-1}) = \Sigma_{i=1}^{n} w_i \ , \tag{2.4}$$

since $w_0 = 0$. It can be shown that again this differs from the exact length only by $n + 1 - \pi^{-1}(n) - \Sigma_{i=1}^{n} (w_i - \bar{w}_i)$.

We now compare the approximate lengths of the shortest paths given in (2.2) and (2.4). Taking a point w arbitrarily in R^n we obtain that

$$P(2^n) = (n+1) \max_j |w_j| - \Sigma_{i=1}^{n} |w_i|$$

and

$$P(2n) = \Sigma_{i=1}^{n} |w_i|$$

with P(k) the length of the shortest path for the k ray algorithm. So, for some w in R^n

$$P(2^n) \lesseqgtr P(2n) \quad \text{if } \max_j |w_j| \lesseqgtr \frac{2}{n+1} \Sigma_{i=1}^{n} |w_i| \ .$$

This says that if w is close to one of the axes, e.g. $w = (w_1, \varepsilon_2, \ldots, \varepsilon_n)^T$ with $\varepsilon_2, \ldots, \varepsilon_n$ small relative to w_1, the 2n ray algorithm should be used. However, if w is close to one of the diagonals, say $w = (w_1, w_1 - \varepsilon_2, \ldots, w_1 - \varepsilon_n)^T$ it is better to use the 2^n ray algorithm. In general, we do not have any information about w a priori. Therefore we compare the expected path lengths of the two algorithms. Again we may restrict ourselves to W. Now, Let $a_1, \ldots, a_n$ be a random sample from a uniform distribution over the interval [0,a] for some $a > 0$ and let $a_{(1)} \leq a_{(2)} \leq \ldots \leq a_{(n)} \leq a$. Then the expected value $E(w_i)$ of the i-th component of $w = (a_{(1)}, \ldots, a_{(n)})^T$ is equal to $ia/(n+1)$. So, for the expected values of $P(2^n)$ and P(2n) we obtain

$$E\{P(2^n)\} = (n+1)E(w_n) - \Sigma_{i=1}^{n} E(w_i) = \tfrac{1}{2}na \tag{2.5}$$

and

$$E\{P(2n)\} = \Sigma_{i=1}^{n} E(w_i) = \tfrac{1}{2}na \qquad (2.6)$$

and hence for both the 2^n ray and the 2n ray algorithm the expected values of the approximate shortest path to a point w uniformly distributed in the cube $\{x \in R^n \mid \|x\|_\infty \leq a\}$ is equal to $\frac{1}{2}na$, where $\|.\|_\infty$ denotes the max-norm. Since $E\{\pi^{-1}(n)\} = \frac{1}{2}(n+1)$ and $E(w_i - \bar{w}_i) = \frac{1}{2}$, the expected value of the approximate length differs from the expected value of the exact length only by $\frac{1}{2}$. It should be observed that the mesh of the K' triangulation is $\delta\sqrt{n}$ with δ the grid size. Normalizing the mesh to be equal to one by taking $\delta = 1/\sqrt{n}$, the expected value increases with a factor $\delta^{-1} = \sqrt{n}$.

Corollary 2.1. The expected length of the approximated shortest path to a point uniformly distributed in $\{x \in R^n \mid \|x\|_\infty \leq a\}$ is for both the 2^n ray and 2n ray algorithm equal to $\frac{1}{2}an\sqrt{n}$, when taking the K' triangulation with mesh one.

3. Further remarks and conclusions

In the previous section we have calculated the expected length of the shortest path to a point uniformly distributed in a cube. However, the choice of a cube is rather arbitrarily. Some other possible choices are the ball or the octahedron. Doing so we have to take a uniform distribution over a set with the Euclidean norm $\|.\|_2$ respectively the one-norm $\|.\|_1$ bounded by some positive number a, instead of taking the max-norm. In particular the Euclidean norm seems to be a very natural choice. Unfortunately the calculations of the expected length of the shortest paths become much more complicated when taking the Euclidean norm or the one-norm, since then we need the expected values of $w_1, w_2, \ldots, w_n$ with $(w_1, w_2, \ldots, w_n)^T$ a random point from a uniform distribution over the ball or octahedron, yielding the condition $\Sigma_{i=1}^n w_1^2 \leq a^2$ and $\Sigma_{i=1}^n |w_i| \leq a$ respectively. Therefore we restrict ourselves to the case n=2. It should be observed that the ratio between the lengths of the paths for the 2^n ray and the 2n ray algorithm does not change when we take a uniform distribution over the boundary. Recall from the previous section that $P_\infty(2^n)/P_\infty(2n) = 1$, with $P_p(k)$ the expected length of the k ray algorithm for the p-norm.

For n=2 and p=1 and with $w \in W$, we have to take a uniform distribution over the line segment between the points $(0,a)^T$ and $(\frac{1}{2}a, \frac{1}{2}a)^T$. So, $E(w) = \frac{1}{2}a$ and $E(w_2) = \frac{1}{4}a$ and hence from (2.2) and (2.4) it follows that $P_1(2^n) = 5a/4$ and $P_1(2n) = a$. For n=2 and p=2, w is a random point from the uniform distribution over the ball segment $\{x \in R^n | x_1 = a \sin \alpha,\ x_2 = a \cos \alpha,\ 0 \leq \alpha \leq \frac{1}{4}\pi\}$. Clearly, $E(w_1) = a \sin \pi/8$ and $E(w_2) = a \cos \pi/8$ and it follows from (2.2) and (2.4) that $P_2(2^n) = 2a \cos \pi/8 - a \sin \pi/8 \simeq 1.47a$ and $P_2(2n) = a \cos \pi/8 + a \sin \pi/8 \simeq 1.31a$. Concluding we have that for n=2 holds

$$\frac{P_\infty(2n)}{P_\infty(2^n)} = 1 > \frac{P_2(2n)}{P_2(2^n)} = 0.89 > \frac{P_1(2n)}{P_1(2^n)} = 0.80\ , \qquad (3.1)$$

which says that for the 2n ray algorithm the expected length of the shortest path is less than for the 2^n ray algorithm when taking the Euclidean norm or the one-norm. We have seen already in section 2 that

$P(2n) < P(2^n)$ if $\max_j |w_j| < 2(n+1)^{-1} \sum_{i=1}^{n} |w_i|$ and reversely. Taking a uniform distribution over the cube, the area of the region with points for which $P(2^n) < P(2n)$ is equal to the area of the region with points for which $P(2n) < P(2^n)$. However, taking the Euclidean norm there are more points with $P(2n) < P(2^n)$ than the reverse and the difference becomes even larger when taking the one-norm. Therefore we may expect that (3.1) also holds for $n > 2$. In fact we have that taking a finite norm there are more points close to the axes than close to the diagonals. So we may conclude that the 2n ray algorithm seems to perform slightly better than the 2^n ray algorithm, supposing that the analysis done so far for the linear case gives an indication for the amount of work to be done in nonlinear cases. However, from the computational results reported in Kojima and Yamamoto [13], van der Laan and Seelen [4] and Broadie [1,2], the 2^n ray algorithm appears to the slightly more efficient. We may ask whether this should be due to the nonlinearity of the problems and hence whether we have to reject the supposition just stated above. In our opinion this supposition should not be rejected. We think that the results in the references mentioned above are not unbiased since almost all computational experiments have be done with standard test-problems. By the structure of these problems and the starting points chosen in the experiments it occurs that in most of the cases the zero point is on or close to one of the diagonals. For instance, some of the problem have a solution at the point $e = (1,\ldots,1)^T$ whereas the algorithms were started at the origin. As we have seen in the previous section it is advantageous to use the 2^n ray algorithm in such cases. So the computational results do not contradict the theoretical analysis of this paper.

In [1,2] it is observed that for a 10-dimensional version of a problem of Kellogg, Li and Yorke, the 2^n ray algorithm spends over 80% of the time in regions of dimension less than 10 and over 50% of the time in regions of dimension 7 or less. Recall that in regions of low dimension computer time can be saved in updating the system of linear equations. In section 2 we have seen that for the linear problem $f(x) = w-x$ the number of steps in dimension t is equal to $t(w_t - w_{t-1})$. With w a point from a uniform distribution in a cube we have $E(w_t - w_{t-1}) = a/(n+1)$ and hence the expected number of steps in dimension t is $ta/(n+1)$. Since the total number of steps is $\frac{1}{2}na$ we have that the expected amount of

work in dimension t is $2t/n(n+1) \times 100\%$. So, for n=10 the expected amount of work in dimension 10 is 18% and the expected amount of work in dimensions 7 or less is 51%. These numbers confirm the computational results of Broadie. From the analysis in section 2 it follows immediately that for the 2n ray algorithm the expected amount of work in dimension t is also $2t/n(n+1) \times 100\%$. So, again the results are the same for the two algorithms when taking a uniform distribution over the cube. In [4] it is observed that the 2^n ray algorithm consumes considerably less computer time per function evaluation than the 2n ray algorithm, which is ascribed to the fact that the 2^n ray algorithm works rather long in low dimensions. Again this might be caused by the fact that most of the testproblems have zeroes close to the diagonals, in which case the 2n ray algorithm works most of the time in high dimensions and the 2^n ray algorithm works most of the time in low dimensions.

We conclude this paper with two additional results. First, we give the expected length of the paths for the J' triangulation, recently proposed by Todd [11]. For this triangulation we have that in the region W the movement in the direction e(1) is twice as fast as for the K' triangulation, whereas the mesh of the triangulation is equal to the mesh of the K' triangulation. Since also the J' triangulation is symmetric around the axes and the diagonals, we may restrict ourselves to points in W. So, the only difference is that in the direction e(1) we need as half as much steps than for the K' triangulation. It follows from section 2 that for the 2^n ray algorithm the expected number of steps reduces with $\frac{1}{2}na/(n+1)$ and for the 2n ray algorithm with $\frac{1}{2}a/(n+1)$. We see that the J' triangulation is more efficient for the 2^n ray algorithm than for the 2n ray algorithm. However, for the 2^n ray algorithm the expected number of steps reduces only with a factor $n/(n+1)$.

Secondly, we give the length of the path for Merrill's algorithm (see [7]). Starting in (v,0), this algorithm follows a path of zeroes of the homotopy function $H(x,t) = tf(x) + (1-t)(v-x)$, $0 \leq t \leq 1$. Clearly, when $f(x) = w-x$ and $v = \underline{0}$ this path is the straight line from $(\underline{0},0)$ to $\hat{w} = (w,1)$. In Todd [10, theorem 3.2] it is stated that the number of simplices on this line is equal to

$$\Sigma_{i=1}^{n+1} |\hat{w}_i| + \Sigma_{i<j} ||\hat{w}_j| - |\hat{w}_i||.$$

Taking w in the region W we obtain that the number of simplices is approximately equal to

$$\Sigma_{i=1}^{n} w_i + \Sigma_{j=1}^{n} \Sigma_{i=1}^{j} (w_j - w_i) .$$

With $E(w_i) = ia/(n+1)$, it follows that the expected number of simplices is $an + an(n-1)/6$. Since 50% of the generated simplices have a new vertex on the zero-level the expected amount of work on the one-level becomes

$$E\{P(\text{Merrill})\} = \tfrac{1}{2}an + an(n-1)/12 .$$

Compared with (2.5) and (2.6) the expected amount of work has an additional term of the order of n^2.

References

[1] M.N. Broadie, "Subdivisions and antiprisms for PL homotopy algorithms", Technical Report SOL 83-14, Stanford University (Stanford, California, 1983).

[2] M.N. Broadie, "Computational experience with the octahedral algorithm and related results", Stanford University (Stanford, California, 1983).

[3] M. Kojima and Y. Yamamoto, "To the implementation of several restart fixed point algorithms and a new variable dimension algorithm", Mathematical Programming 28 (1984) 288-328.

[4] G. van der Laan and L.P. Seelen, "Efficiency and implementation of simplicial zero point algorithms", Mathematical Programming 30 (1984) 196-217.

[5] G. van der Laan and A.J.J. Talman, "A restart algorithm without an artificial level for computing fixed points on unbounded regions", in: H.-O. Peitgen and H.-O. Walther, eds., Functional differential equations and approximations of fixed points (Springer, Berlin, 1979) 247-256.

[6] G. van der Laan and A.J.J. Talman, "A class of simplicial restart fixed point algorithms without an extra dimension", Mathematical Programming 20 (1981) 33-48.

[7] O.H. Merrill, "Applications and extensions of an algorithm that computes fixed points of certain upper semi-continuous point to set mappings", Ph.D. Thesis, University of Michigan (Ann Arbor, Michigan, 1972).

[8] P.M. Reiser, "A modified integer labelling for complementarity algorithms", Mathematics of Operations Research 6 (1981) 129-139.

[9] R. Saigal, "A homotopy for solving large, sparse and structured fixed point problems", Mathematics of Operations Research 8 (1983) 557-578.

[10] M.J. Todd, "Improving the convergences of fixed point algorithms", Mathematical Programming Study 7 (1978) 151-169.

[11] M.J. Todd, "J': a new triangulation of R^n", Technical Report no. 589, Cornell University (Ithaca, New York, 1983).

[12] A.H. Wright, "The octahedral algorithm, a new simplicial fixed point algorithm", Mathematical Programming 21 (1981) 47-69.

[13] Y. Yamamoto, "A new variables dimension algorithm for the fixed point problem", Mathematical Programming 25 (1983) 329-342.

IN 1983 REEDS VERSCHENEN

126 H.H. Tigelaar
Identification of noisy linear systems with multiple arma inputs.

127 J.P.C. Kleijnen
Statistical Analysis of Steady-State Simulations: Survey of Recent Progress.

128 A.J. de Zeeuw
Two notes on Nash and Information.

129 H.L. Theuns en A.M.L. Passier-Grootjans
Toeristische ontwikkeling - voorwaarden en systematiek; een selectief literatuuroverzicht.

130 J. Plasmans en V. Somers
A Maximum Likelihood Estimation Method of a Three Market Disequilibrium Model.

131 R. van Montfort, R. Schippers, R. Heuts
Johnson S_U-transformations for parameter estimation in arma-models when data are non-gaussian.

132 J. Glombowski en M. Krüger
On the Rôle of Distribution in Different Theories of Cyclical Growth.

133 J.W.A. Vingerhoets en H.J.A. Coppens
Internationale Grondstoffenovereenkomsten.
Effecten, kosten en oligopolisten.

134 W.J. Oomens
The economic interpretation of the advertising effect of Lydia Pinkham.

135 J.P.C. Kleijnen
Regression analysis: assumptions, alternatives, applications.

136 J.P.C. Kleijnen
On the interpretation of variables.

137 G. van der Laan en A.J.J. Talman
Simplicial approximation of solutions to the nonlinear complementarity problem with lower and upper bounds.

IN 1984 REEDS VERSCHENEN

138 G.J. Cuypers, J.P.C. Kleijnen en J.W.M. van Rooyen
Testing the Mean of an Asymetric Population:
Four Procedures Evaluated

139 T. Wansbeek en A. Kapteyn
Estimation in a linear model with serially correlated errors when observations are missing

140 A. Kapteyn, S. van de Geer, H. van de Stadt, T. Wansbeek
Interdependent preferences: an econometric analysis

141 W.J.H. van Groenendaal
Discrete and continuous univariate modelling

142 J.P.C. Kleijnen, P. Cremers, F. van Belle
The power of weighted and ordinary least squares with estimated unequal variances in experimental design

143 J.P.C. Kleijnen
Superefficient estimation of power functions in simulation experiments

144 P.A. Bekker, D.S.G. Pollock
Identification of linear stochastic models with covariance restrictions.

145 Max D. Merbis, Aart J. de Zeeuw
From structural form to state-space form

146 T.M. Doup and A.J.J. Talman
A new variable dimension simplicial algorithm to find equilibria on the product space of unit simplices.

147 G. van der Laan, A.J.J. Talman and L. Van der Heyden
Variable dimension algorithms for unproper labellings.

148 G.J.C.Th. van Schijndel
Dynamic firm behaviour and financial leverage clienteles

149 M. Plattel, J. Peil
The ethico-political and theoretical reconstruction of contemporary economic doctrines

150 F.J.A.M. Hoes, C.W. Vroom
Japanese Business Policy: The Cash Flow Triangle
an exercise in sociological demystification

151 T.M. Doup, G. van der Laan and A.J.J. Talman
The $(2^{n+1}-2)$-ray algorithm: a new simplicial algorithm to compute economic equilibria

IN 1984 REEDS VERSCHENEN (vervolg)

152 A.L. Hempenius, P.G.H. Mulder
Total Mortality Analysis of the Rotterdam Sample of the Kaunas-Rotterdam Intervention Study (KRIS)

153 A. Kapteyn, P. Kooreman
A disaggregated analysis of the allocation of time within the household.

154 T. Wansbeek, A. Kapteyn
Statistically and Computationally Efficient Estimation of the Gravity Model.

155 P.F.P.M. Nederstigt
Over de kosten per ziekenhuisopname en levensduurmodellen

156 B.R. Meijboom
An input-output like corporate model including multiple technologies and make-or-buy decisions

157 P. Kooreman, A. Kapteyn
Estimation of Rationed and Unrationed Household Labor Supply Functions Using Flexible Functional Forms

158 R. Heuts, J. van Lieshout
An implementation of an inventory model with stochastic lead time

159 P.A. Bekker
Comment on: Identification in the Linear Errors in Variables Model

160 P. Meys
Functies en vormen van de burgerlijke staat
Over parlementarisme, corporatisme en autoritair etatisme

161 J.P.C. Kleijnen, H.M.M.T. Denis, R.M.G. Kerckhoffs
Efficient estimation of power functions

162 H.L. Theuns
The emergence of research on third world tourism: 1945 to 1970; An introductory essay cum bibliography

163 F. Boekema, L. Verhoef
De "Grijze" sector zwart op wit
Werklozenprojecten en ondersteunende instanties in Nederland in kaart gebracht