easily extended to yield optimal Bayesian stopping rules [4].

3. Multi Level Single Linkage.

In spite of the scope that Multistart offers for analysis, the procedure is lacking in efficiency. The main reason is that it will inevitably cause each local minimum to be found several times. To avoid all these time consuming local searches, P should be applied no more than once, or better still exactly once, in every region of attraction. For this purpose, the Multi Level Single Linkage method has been developed. Unlike the Single Linkage method described in [1], Multi Level Single Linkage focuses on the function values of the sample points to obtain an extremely simple but powerful method.

In the Multi Level Single Linkage method the local search procedure P is applied to every sample point except if there is another sample point or a previously detected local minimum within some critical distance which has a smaller function value.

Actually, the method is implemented in an iterative fashion, where points are sampled in groups of fixed size, say N. In each iteration the above rule is applied to the points of the expanded sample to determine from which sample points P should be started.

In spite of its simplicity, the theoretical properties of Multi Level Single Linkage are quite strong [16]. If, for some $\sigma > 0$, the critical distance in iteration k is chosen to be

$$r_k = \pi^{-\frac{1}{2}}[(\Gamma+\frac{n}{2})m(S)\ \sigma\ \frac{\log kN}{kN}]^{1/n} \qquad (3)$$

(where m(.) and Γ(.) denote the Lebesque measure and the gamma function respectively), then any local minimum x* will be found by Multi Level Single Linkage within a finite number of iterations with probability 1. However, at the same time we can prove that, if $\sigma > 4$, the total number of local searches ever started by Multi Level Single Linkage is finite with probability 1 even if the sampling continues forever. In [16] it is indicated that these results are in some sense the strongest possible ones. (Actually, the above results were obtained for a slightly different version of Multi Level Single Linkage, modified to ensure that P is never applied in a point which is very close to

the boundary of S or to a stationary point detected previously.)

Since Multi Level Single Linkage and Multistart result in the same set of minima with a probability that tends to 1 with increasing sample size, we can simply use the stopping rules which were designed for Multistart [4].

One might believe that it is unlikely that the global minimum will be found by applying P to a sample point with a relatively high function value. It is possible to reduce the sample by removing a certain fraction, say $1-\gamma$, of the sample points with the highest function values and apply Multi Level Single Linkage to the reduced sample points only. Such a reduction of the sample does not significantly affect the theoretical properties of Multi Level Single Linkage. In [16] it is observed that in case of a reduction of the sample to γkN sample points in iteration k, the critical distance should still equal (3), but that in (2) M should equal γkN and not kN.

Because of the extreme simplicity of the Multi Level Single Linkage method it can be very efficiently implemented. Of course, it is not advisable to start the calculations necessary for applying the method from scratch in every iteration. Since the sample of iteration k-1 is a subset of the sample of iteration k, and since it is known in what way the critical distance varies with k, it turns out to be possible to develop an efficient dynamic implementation of the method [16]. In this dynamic implementation, the information which is necessary to determine the starting points of the local search procedure in iteration k is determined by updating the corresponding information from iteration k-1. It turns out to be possible to implement Multi Level Single Linkage in such a way that the running time needed up to iteration k is only O(k) in expectation. Hence, the calculations needed to update the information in iteration k do not vary with the size of the complete sample, but only with the number of newly sampled points.

4. Computational results.

To examine the computational behaviour of Multi Level Single Linkage it has been coded in Fortan IV and run on the DEC 2060 computer of the Computer Institute Woudestein.

We tested Multi Level Single Linkage on the standard set of test functions [11] which in commonly used in global optimization. These test functions are listed in Table 1.

Table 1

TEST FUNCTIONS [11]

GP	Goldstein and Price
BR	Branin (RCOS)
H3	Hartman 3
H6	Hartman 6
S5	Shekel 5
S7	Shekel 7
S10	Shekel 10

In this section Multi Level Single Linkage will be compared with a few leading contenders whose computational behaviour is described in [11]. In this reference methods are compared on the basis of two criteria: the number of function evaluations and the running time required to solve each of the seven test problems. To eliminate the influence of the different computer systems used, the running time required is measured in units of standard time, where one unit corresponds to the running time needed for 1000 evaluations of the S5 test function in the point (4,4,4,4).

Since both the number of function evaluations and the units of standard time required are sensitive to the peculiarities of the sample at hand, the results reported for Multi Level Single Linkage represent the average outcome of four independent runs. We applied Multi Level Single Linkage to 20% of the sample points (γ=0.2) and choose σ to be equal to 4. After an initial sample of size 100, we increased the sample and applied Multi Level Single Linkage iteratively until the expected number of minima (2) exceeded the number of different minima found by less than 0.5. We did not implement the method as efficiently as possible since this is not really necessary if the sample size is moderate. Since all test functions are twice differentiable, we could use the VA10AD variable metric routine from the Harwell subroutine library as local search procedure.

In Table 2 and Table 3 we summarize the computational results of Multi Level Single Linkage and compare them to those obtained for a few leading contenders as reported in [11].

value y_* with probability 1 (i.e. almost surely) (cf [15]).

2. Multistart.

Most successful methods for global optimization involve local searches from some or all of the sample points. This presupposes the availability of some local search procedure P which starting from an arbitrary point $x \in S$, produces a local minimum x^*. Depending on what may be assumed about f, a large number of such procedures is available from the nonlinear programming literature. We assume that P is strictly descent [16], such that if P is started from any point $x \in S$ and converges to a local minimum x^*, there exists a path from x to x^* along which the function values are nonincreasing. We also assume that this path is completely contained in S. Finally we assume that the number of stationary points of f, i.e. points where the gradient of f is zero, is finite.

The simplest way to make use of the local search procedure P occurs in a folklore method known as Multistart. Here, P is applied to every point in a sample, drawn from a uniform distibution over S, and the local minimum with the lowest function value found in this way is the candidate value for y_*.

An interesting analysis of Multistart was initiated in [19] and extended in [2,3,5]. It is based on a Bayesian estimate of the number of local minima W and of the relative size of each region of attraction $\Theta_\ell = m(R(x^*))/m(S)$, $\ell=1,\ldots,W$, where a region of attraction $R(x^*)$ is defined to be the set of all points in S starting from which P will arrive at x^*.

In [5] a so-called non-informative prior distribution is specified for the unknowns W, $\Theta_1,\ldots,\Theta_W$. Given the outcome of an application of Multistart, Bayes's rule is then used to compute the posterior distribution, which incorporates both the prior beliefs and the sample information.

After lengthy calculations, surprisingly simple expressions emerge for the posterior disrtibution and posterior expectation of several interesting parameters [5]. For instance, if w different local minima have been found as the result of M local searches started in uniformly distributed points, then the posterior expectation of the number of local minima is

$$\frac{w(M-1)}{M-w-2} \qquad (2)$$

This Bayesian analysis is quite an attractive one, the more so since it can be

THE MULTI LEVEL SINGLE LINKAGE METHOD FOR GLOBAL OPTIMIZATION

by

Alexander H.G. Rinnooy Kan and Gerrit T. Timmer
Econometric Institute
Erasmus University Rotterdam
3000 DR Rotterdam
The Netherlands

1. Introduction.

Written at the request of the Editor of this Newsletter, this is a short presentation of the theoretical properties and the computational performance of the recently developed Multi Level Single Linkage method for global optimization. This method has been discussed extensively in [16].

The global optimization problem considered is to find the global optimum (say, the global minimum) x_* of a real valued twice continuously differentiable objective function $f: \mathbb{R}^n \to \mathbb{R}$. For computational reasons one usually assumes that a set $S \subset \mathbb{R}^n$ which is convex, compact and contains the global minimum as an interior point, is specified in advance. Nevertheless, the problem to find

$$y_* = \min_{x \in S}\{f(x)\} \tag{1}$$

remains essentially one of unconstrained optimization.

Only a few solution methods for this problem have been developed so far, certainly in comparison with the multitude of nonlinear programming methods that aim for an arbitrary local minimum. For a survey of global optimization methods we refer to [9,10,14,16]. It appears not to be possible to design methods which offer an absolute guarantee of success for abritrary f. Therefore, most methods are of a stochastic nature and provide an asymptotic guarantee in a stochastic sense. For instance, if the function is evaluated in points which are drawn from a uniform distribution over S, then it can be shown that the smallest function value found converges to the global minimum

Table 2

NUMBER OF FUNCTION EVALUATIONS

	FUNCTION						
METHOD	GP	BR	H3	H6	S5	S7	S10
Branin [6,12]	-	-	-	-	5500	5020	4860
Bremmerman [7,11]	300	160	420L	515	375L	405L	336L
Price [13]	2500	1800	2400	7600	3800	4900	4400
Törn [17,18]	2499	1558	2584	3447	3649	3606	3874
De Biase [8]	378	597	732	807	620	788	1160
Multi Level Single Linkage [16]	148	206	197	487	404	432*	564

L: the method did not find the global minimum
*: the global minimum was not found in one of the four runs

Table 3

NUMBER OF UNITS STANDARD TIME

	FUNCTION						
METHOD	GP	BR	H3	H6	S5	S7	S10
Branin [6,12]	-	-	-	-	9	8.5	9.5
Bremmerman [7,11]	0.7	0.5	2L	3	1.5L	1.5L	2L
Price [13]	3	4	8	46	14	20	20
Törn [17,18]	4	4	8	16	10	13	15
De Biase [8]	15	14	16	21	23	20	30
Multi Level Single Linkage [16]	0.15	0.25	0.5	2	1	1*	2

L: the method did not find the global minimum
*: the global minimum was not found in one of the four runs.

Acknowledgement. This research was partially supported by the Netherlands Foundation for Mathematics SMC with financial aid from the Netherlands Organization for Advancement of Pure Research (ZWO).

Note. The theoretical properties of Multi Level Single Linkage described briefly in Section 2 and the presentation of some related methods will shortly appear in the form of technical reports that can be obtained from the authors.

References

[1] C.G.E. Boender, A.H.G. Rinnooy Kan, L. Stougie and G.T. Timmer, "A Stochastic Method for Global Optimization", Mathematical Programming 22 (1982), 125-140.

[2] C.G.E. Boender and R. Zielinski, "A Sequential Bayesian Approach to Estimating the Dimension of a Multinomial Distribution", Technical Report, The Institute of Mathematics of the Polish Acadamy of Sciences (1982).

[3] C.G.E. Boender and A.H.G. Rinnooy Kan, "A Bayesian Analysis of the Number of Cells of a Multinomial Distribution", The Statistician 32 (1983), 240-248.

[4] C.G.E. Boender and A.H.G. Rinnooy Kan, "Bayesian Stopping Rules for a Class of Stochastic Global Optimization Methods", Technical Report, Erasmus University Rotterdam, The Netherlands (1983).

[5] C.G.E. Boender, The Generalized Multinomial Distribution: A Bayesian Analysis and Applications, Ph. D. Dissertation, Erasmus University Rotterdam (Centrum voor Wiskunde en Informatica, Amsterdam, 1984).

[6] F.H. Branin and S.K. Hoo, "A Method for Finding Multiple Extrema of a Function of n Variables", in: F.A. Lootsma, ed., Numerical Methods of Nonlinear Optimization (Academic Press, London, 1972).

[7] H. Bremmerman, "A Method of Unconstrained Global Optimization", Mathematical Biosciences 9 (1970), 1-15.

[8] L. De Biase and F. Frontini, "A Stochastic Method for Global Optimization: Its Structure and Numerical Performance", in: L.C.W. Dixon and S.P. Szegö, eds., Towards Global Optimization 2 (North-Holland, Amsterdam 1978).

[9] L.C.W. Dixon and G.P. Szegö, eds., Towards Global Optimization (North-Holland, Amsterdam 1975).

The factor L^2 in the complexity of Karmarkar's algorithm comes from two sources. The number of steps of the algorithm is O(nL), each step requires $O(n^{2.5})$ arithmetic operations and each arithmetic operation requires a precision of O(L) bits.

If one is interested in finding a solution whose objective function value is a certain fixed fraction, say 99.99% of the optimum value, then the algorithm requires only O(n) steps thus saving a factor of L.

Regarding the second factor of L, any algorithm - such as the simplex algorithm - that requires representation of inverse of a submatrix of the constraint matrix during the course of computation requires at least as much precision in arithmetic operations as this algorithm, which is O(L) bits in the worst case. As compared to this worst-case bound, the simplex algorithm seems to work with much less precision in practice and the same amount of precision is sufficient for this algorithm, thus saving the second factor of L.

Each step Karmarkar's algorithm requires optimization of a linear function over an ellisoid or equivalently solution of a linear system of equations of the type $(ADA^r)x=b$, where A is a fixed matrix and D is a diagonal matrix with positive entries which changes by a small amount from step to step. He devises a method based on succesive rank-one modifications and proves a worst-case bound of $O(n^{2.5})$ arithmetic operations per step.

In practice, the matrices encountered are sparse, hence more efficient methods for solving the above problem are possible. Another feature of the algorithm which can lead to computational saving is that it is not necessary to find the exact solution to the optimization problem stated above.

Regarding the practical performance of this algorithm the first reactions are undecided: savings of a factor of 50 in comparison to a commercial LP code, are reported on one of George Dantzig's energy problems (!), but apparently the optimum was used as a first guess for the optimal objective function value (!).

We hope to report more on this algorithm in the next issue.

A NEW POLYNOMIAL-TIME ALGORITHM FOR LINEAR PROGRAMMING

In the last few months we have been excited by the news of a new polynomial-time algorithm for linear programming, designed by Narendra Karmarkar of AT & T Bell Laboratories in Murray Hill, New Jersey 07974.

At the time of this writing it was announced that Mr. Karmarkar will give a lecture on this method at the Dallas ORSA/TIMS meeting. For those readers who were not there we give a brief summary of the method based on a publication in the Proceedings of the 16th annual ACM Symposium on the Theory of Computing (Washington DC, April 30 - May 2, 1984) pp. 302-310.

The simplex algorithm for linear programming has been shown to require an exponential number of steps in the worst-case. A polynomial-time algorithm for linear programming was published by Khachiyan in 1979. The complexity of his algorithm is $O(n^6L^2)$ where n is the dimension of the problem and L is the number of bits in the input [3] . Karmarkar presents a new polynomial-time algorithm for linear programming whose time- complexity is $O(n^{35}L^2)$.

He proves a theorem about polytopes which seems to be interesting in its own right. Given a polytope $P \subseteq R^r$ and a strictly interior point $a \in P$. there is a projective transformation of the space that maps P,a to P',a' having the following property: The ratio of the radius of the smallest sphere with center a', containing P' to the radius of the largest sphere with center a', contained in P' is O(n).

The algorithm for linear programming is based on repeated application of such projective transformations followed by optimization over the inscribed sphere to create a sequence of points which converges to the optimal solution in polynomial time.

Since representing the output of a linear programming problem requires O(L) bits per variable in the worst case, the factor L must appear in the worst-case complexity of any algorithm for linear programming.

[10] L.C.W. Dixon and G.P. Szegö, eds., Towards Global Optimization 2, (North-Holland, Amsterdam 1978).

[11] L.C.W. Dixon and G.P. Szegö, "The Global Optimization Problem: an Introduction", in: L.C.W. Dixon and G.P. Szegö, eds., Towards Global Optimization 2 (North-Holland, Amsterdam 1978).

[12] J. Gomulka, "Two implementations of Branin's Method: Numerical Experience", in: L.C.W. Dixon and G.P. Szegö, eds., Towards Global Optimization 2 (North-Holland, Amsterdam 1978).

[13] W.L. Price, " A Controlled Random Search Procedure for Global Optimization", in: L.C.W. Dixon and G.P. Szegö, eds., Towards Global Optimization 2 (North-Holland, Amsterdam 1978).

[14] A.H.G. Rinnooy Kan and G.T. Timmer, "Stochastic Methods for Global Optimization", American Journal of Mathematical and Management Sciences to appear (1984).

[15] R.Y. Rubenstein, Simulation and the Monte Carlo Method (John Wiley and Sons, New York 1981).

[16] G.T. Timmer, Global Optimization: A Stochastic Approach, Ph.D. Dissertation, Erasmus University Rotterdam (Centrum voor Wiskunde en Informatica, Amsterdam 1984).

[17] A.A. Törn, "Cluster Analysis Using Seed Points and Density Determined Hyperspheres with an Application to Global Optimization", in: Proceedings of the Third International Conference on Pattern Recognition, Coronado, California (1976).

[18] A.A. Törn, "A Search Clustering Approach to Global Optimization", in L.C.W. Dixon and G.P. Szegö, eds., Towards Global Optimization 2 (North-Holland, Amsterdam 1978).

[19] R. Zielinski, "A Stochastic Estimate of the Structure of Multi-Extremal Problems", Mathematical Programming 21 (1981), 348-356.

THE NUMBER OF SIMPLEX ITERATIONS FOR A GENERAL LP CODE ON NETWORK PROBLEMS

C.A. Haverly, Haverly Systems Inc.
PO Box 919, Denville, New Jersey 07834 U.S.A.

This computational study is in response to the informative article on the number of simplex iterations by Professors Olafsson and Lindberg (1). They used PNET, a network code, to examine assignment and transportation models and provided formulas for the number of iterations expected for various size models. This study was prompted by their reports of iteration numbers higher than I expected for a modern general LP code.

Previous experiences

Based on many years experience with practical models, I use as a rough estimation of the number of iterations in an LP model:

number of iterations = FR where R= number of rows
F= a proportionality factor

This relationship has been long known and used. Two recent references are (2) and (3). F is typically between 1 and 5. Smaller, easier to solve models have F's in the range 1-2. I would expect most assignment and transportation models to have F's between 1 and 1.5.

It is easy to formulate transportation models for which F would be less than 1. A transportation model could be formulated with all source availabilities as maximums and all destination requirements as equalities. If the values of availabilities are sufficiently large, the number of iterations to solve the model will be equal to the number of destinations. The minimum value of F will be F=D/ (D+S+1) where D and S are the number of Destinations and Sources, respectively.
In my experience, the number of columns (C) normally has little effect on the number of iterations. Similary, I would not expect the number of significant places carried in the objective row and right hand side (RHS) to have much effect.

CODE - BREAKER MODELS

Larry Haverly
Haverly Systems Inc.
PO Box 919, Denville, New Jersey 07834 U.S.A.

The following note was received by the editors on May 29, 1984.

Editor:

In experimenting with the number of simplex iterations for the blending model structure, a code-breaker model was found. It is only 103 rows, 801 columns, 61,703 elements but took over 3000 iterations to solve. A copy of the matrix will be supplied to anyone requesting it.
Larry Haverly, Haverly Systems, P.O. Box 919, Denville, NJ 07834

Some months later Larry informed us that he now has a smaller verzion of this problem too. For more information contact him directly.

consequence, the performance of optimization methods must be evaluated by experiments. By defining a sample, i.e. by searching for suitable test problems, and by designing a computational experiment, the efficiency, robustness, and reliability of an algorithm can be investigated. Such experimental optimization will become even more important in future years as modeling efforts attempt to solve problems related to evermore complex systems and consequently, the mathematical algorithms used to solve these problems become more and more sophisticated. The need to disseminate information about computational mathematical programming procedures was the main impulse for organizing the Advanced Study Institute.

Consequently, the general purpose of the Advanced Study Institute was to link new algorithmic developments for optimization models with practical applications. From the organizational point of view, the ASI consisted of tutorials, research seminars, and a software fair. The tutorials covered all major types of optimization models currently used in practice. Leading experts gave an introduction to the subject, provided any mathematical background, presented suitable algorithms stressing information about their computer implementations, usage and numerical performance. The topics of the tutorials ranged from linear and integer programming, networks, and various aspects of nonlinear programming to optimal control, stochastic and nonsmooth optimization. They were accomplished by about 60 research seminars of other participants leading to a more detailed insight in special individual research projects. Moreover, information material about existing optimization software was displayed at the ASI e.g. in form of user's guides or code descriptions.

Manuscripts of the tutorials and code descriptions will be published in the proceedings of the ASI (NATO ASI Series, Springer). In addition selected papers, presented in form of a research seminar, will be published in a special volume of Mathematical Programming Study. The editors are K. Hoffmann, R. Jackson and J. Telgen

Tests

These expectations were tested by running DxD assignment models for D=20, 50, 100, 125, and transportation models with dimensions(sources x destinations) : 20x30, 40x60, 10x90, 80x120, 500x30 and 980x15. The objective row and transportation RHS values were taken randomly from a uniform ditribution. All runs were made using our general LP code, HSLP. Runs were started from an all slack basis.

Results

The transportation model has a size of S+D+1 rows and SD structural columns; in an assignment model, S=D. A 20x20 assignment problem would result in a model of 41 rows and 400 structural columns. I expected it would take between 41 and 61 iterations to solve. Eleven varied models of this size actually took between 38 and 52 iterations to reach optimal. This is substantially less than the 121 predicted by the formula for PNET (1).

No significant difference in the number of iterations was found between assignment and transportation models of the same row size.

No significant difference in the number of iterations was found for different random selection of objective row and RHS values from a uniform distribution.

Since general LP codes allow decimal values, the number of significant places in the values is normally 7.5. However, special cases were run where the random values were reduced to two integer places and then to a single integer value. No significant difference in the number of iterations was found.

Number of columns

The effect of the number of columns was checked by using transportation models of 40x60 (which gave a matrix of 101 rows x 2400 columns) and 10x90 (which gave a matrix of 101 rows x 900 columns). These gave iteration counts of 110 and 104 respectively. This small difference for models of the same row size appears to be too small to attribute to differences in the number of columns which varied by a factor of 2.7. The full 101 x 2400 model was run with a different random distribution.Then a random selection of columns was made to get 101 x 740 and 101 x 361 models. The three had iteration counts to optimize of 144, 152 and 138, respectively. In all tests the number of iterations to optmize transportation models seemed related to the number of rows and not to the number of columns.

Dual

A test was made to see what would happen when a primal model with few rows and many columns was changed to its dual equivalent of many rows and few columns. An assignment problem (D=30) was run as an LP model of 61 rows x 900 structural columns and the dual of 901 rows x 60 structural columns. Assignment models for D=50 and 90 were also run. The results were:

D	TYPE	ROWS	COLUMNS	ITERATIONS	
30	Primal	61	900	70	
	Dual	901	60	65	
50	Primal	101	2500	102	
	Dual	2501	100	130	
50	Primal	101	2500	120	Different Random Nos.
	Dual	2501	100	143	
90	Primal	181	8100	203	
	Dual	8101	180	256	

Conclusions

The number of iterations seems to relate to either the number of columns or number of rows - whichever is less. It is concluded that for these model structures iterations can be estimated by approximately (1.2 +/- 3) times the minimum of the number of rows or columns.

NATO ADVANCED STUDY INSTITUTE

ON

COMPUTATIONAL MATHEMATICAL PROGRAMMING

K. Schittkowski
Institut für Informatik
Universität Stuttgart
D-7000 Stuttgart 1, Germany F.R.

The Advanced Study Institute (ASI) on Computational Mathematical Programming was held in Bad Windsheim, Germany F.R., from July 23 to August 2, 1984, under the sponsorship of NATO. The ASI was organized by the Committee on Algorithms (COAL) of the Mathematical Programming Society. Co-directors were Karla Hoffman (National Bureau of Standards, Washington, U.S.A.) and Jan Telgen (Rabobank Nederland,Zeist, The Netherlands). Ninety participants coming from about 20 different countries have been admitted by the organizers and contributed their efforts to achieve a highly interesting and stimulating meeting.

The basic idea of optimization theory is to minimize (or maximize) a function of several variables subject to certain restrictions. This general mathematical concept covers a broad class of possible practical applications arising in mechanical, electrical, or chemical engineering, physics, economics, medicine, biology, etc. There are both industrial applications (e.g. design of mechanical structures, production plans) and applications in the natural, engineering, and social scienes (e.g. chemical equilibrium problems, cristallography problems).

In most cases, a theoretical foundation which predicts the numerical performance of a mathematical programming algorithm, does not exist. Nor can one at the present time prove theoretically that one algorithm is best for a given application. As a

MATCHING LECTURES AND SPEAKERS

The Committe on Algorithms has initiated a system to assist (1) travellers who can give lectures on various topics of Mathematical Programming and Operations Research and (2) institutions that are interested in inviting speakers. The system covers the entire world. Ashok Idnani has volunteered to maintain the system. Interested parties fill in the appropriate form below and mail it to: Ashok Idnani

Computer Science Department
Pace University
Pleasantville, NY 10570
U.S.A.

He will match the forms and inform both parties if there is a match. Then the parties themselves should follow up on that information and make arrangements. There is no charge for using the system.

TRAVELLERS FORM:

Name:
Address + tel.no.: Country:
Period of travel:
Geographical area:
Person to contact in the area (include tel.no.):
Topics of lectures:
Additional information:

INSTITUTIONS FORM:

Name:
Address: Country:
Person to contact (+ tel.no.):
Period:
Additional information:

In all tests, except two, the F factor was between .93 and 1.42. This included some models which had no feasible solutions. In my experience these often have taken more iterations to resolve. One exception was a 996 row x 14,700 columns transportation model (980 sources, 15 destinations) in which the total availability exceeded total demand by about 60%. F was .71. This is an example of F less than 1 as discussed above. The other exception was F= 1.61 for a degenerate 40x60 transportation model in which all supply came from a single source.

It appears that a reasonable expectation for assignment and transportation models in the size ranges tested, with close balance of total supply and demand, and using a general LP code is:

$$\text{Simplex Iterations} = 1.2(+/-\ .3) * \min(R,C)$$

References

(1) Snjolfun Olofsson and P.O. Lindberg: "On the Lengths of Simplex Paths". COAL Newsletter No. 10 (1984), p.23+.

(2) Robert Fourer: "Technical Correspondence". Communications of ACM, (1983), p. 382+.

(3) Edward H. McCall: "Performance Results of the Simplex Algorithm for a Set of Real-world LP Models". Communications of ACM, (1982), pp. 207-212.

Calendar of mathematical programming meetings as of 1 November 1984

Maintained by the Mathematical Programming Society (MPS)

This Calendar lists noncommercial meetings specializing in mathematical programming or one of its subfields in the general area of optimization and applications, whether or not the Society is involved. (The meetings are not necessarily 'open'.) Any one knowing of a meeting that should be listed here is urged to inform Dr. Philip Wolfe, IBM Research 33-2, POB 218, Yorktown Heights, NY 10598, U.S.A; Telephone 914-945-1642, Telex 137456.

Some of these meetings are sponsored by the Society as part of its world-wide support of activity in mathematical programming. Under certain guidelines the Society can offer publicity, mailing lists and labels, and the loan of money to the organizers of a qualified meeting.

Substantial portions of meetings of other societies such as SIAM, TIMS, and the many national OR societies are devoted to mathematical programming, and their schedules should be consulted.

1984

December 12-14: 23d IEEE Conference on Decision and Control, Las Vegas, Nevada, U.S.A. Contact: Abraham Haddad, School of Electrical Engineering, Georgia Institute of Technology, Atlanta, GA 30332, U.S.A. Telephone 404-894-3930.

1985

June 11-14: 5th IFAC Workshop on Control Applications of Nonlinear Programming and Optimization, Capri, Italy. Contact: Professor G. Di Pillo, Dipartimento di Informatica e Sistemistica, Università degli Studi di Roma 'La Sapienza', Via Eudossiana 18, 00184 Roma, Italy. Telephone (39) 6-484441.

August 5-9: Twelfth International Symposium on Mathematical Programming in Cambridge, Massachusetts, U.S.A. Contact: Professor Jeremy Shapiro, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A. Telephone 617-253-7165. Official triennial meeting of the MPS.

EDITORIAL COLUMN

As announced in our last editorial column, this is the last COAL Newsletter that is sent to "friends of COAL" free of charge.

Members of the Mathematical Programming Society will continue to receive the COAL Newsletter without any charge. But "friends" who do not want to join the Society will have to pay a fee of US $ 5.- a year to continue receiving the Newsletter. All "friends" will be contacted regarding this matter in due time.

The current issue contains two types of material. First, we have some COAL related business, including a report on the Bad Windsheim meeting, a service for traveling lecturers and an announcement on a code breaking LP model.

Of particular interest to many readers may be a very short description of the recent LP code of Narendra Karmakar. The excitement about this result is similar to the one caused by the announcement of Khachian's ellipsoidal method in 1979.

Second, there are some papers reporting on recent computational testing in the areas of networks (Larry Haverly adds his results to a paper by Jeff Kennington et al in COAL Newsletter no. 10) and global optimization (Gerrit Timmer and Alexander Rinnooy Kan present results with their state of the art code).

We hope this issue provides interesting news and good reading.

Robert R. Meyer
Jan Telgen

COMMITTEE ON ALGORITHMS of the MATHEMATICAL PROGRAMMING SOCIETY

CHAIRMAN

Karla Hoffman
Center for Applied Mathematics
National Bureau of Standards
Washington, D.C. 20234
USA

EDITORS OF THE NEWSLETTER

Jan Telgen
RABOBANK NEDERLAND
Laan van Eikenstein 9
3705 AR ZEIST
The Netherlands

Robert R. Meyer
University of Wisconsin
Computer Sciences Dept.
1210 W. Dayton St.
Madison, WI 53706

MEMBERS

Harlan P. Crowder
IBM Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Richard H.F. Jackson
Center for Applied Mathematics
National Bureau of Standards
Washington, D.C. 20234

Leon S. Lasdon
School of Business Administration
University of Texas
Austin, TX 78712

Claude LeMarechal
INRIA Domaine DeVolvceau
Requen Ct. BP 105
78150 Le Chesnay
France

Richard P. O'Neill
EI 622, rm 4447
Division of Oil and Gas Analysis
Dept. of Energy
Washington, D.C. 20461

Susan S. Powell
The London School of Economics
and Political Science
London WC2A 2AE
United Kingdom

Ken M. Ragsdell
School of Mechanical Engineering
Purdue University
West Lafayette, IN 47907

Ronald Rardin
School of Ind. & Systems Engineering
Georgia Inst. of Technology
Atlanta, GA 30332

Patsy Saunders
Center for Applied Mathematics
National Bureau of Standards
Washington, D.C. 20234

Klaus Schittkowski
Institut für Informatik
Universität Stuttgart
Azenbergstrasse 12
D-7000 Stuttgart 1
Germany, F.R.

Robert B. Schnabel
University of Colorado
Dept. of Computer Science
Boulder, CO 80309

EX OFFICIO MEMBERS

Michael Held, Chairman, Excom MPS
IBM Academic Information Systems
360 Hamilton Avenue
White Plains, NY 10601

Alex Orden, Chairman, MPS
Graduate School of Business
University of Chicago
Chicago, IL 60637

COAL OBJECTIVES

The Committee on Algorithms is involved in computational developments in mathematical programming. There are three major goals: (1) ensuring a suitable basis for comparing algorithms, (2) acting as a focal point for computer programs that are available for general calculations and for test problems, and (3) encouraging those who distribute programs to meet certain standards of portability, testing, ease of use and documentation.

NEWSLETTER OBJECTIVES

The newsletter's primary objective is to serve as a forum for the Friends of COAL. Through an informal exchange of opinions, members have an opportunity to share their experiences. To date, our profession has not developed a clear understanding on the issues of how computational tests should be carried out, how the results of these tests should be presented in the literature, or how mathematical programming algorithms should be properly evaluated and compared. These issues will be addressed in the newsletter.

Dr. JAN TELGEN
RABOBANK NEDERLAND
LAAN VAN EIKENSTEIN 9
3705 AR ZEIST
THE NETHERLANDS

Mr. T. Steihaug,
STATOIL,
Forus, P. O. Box 300,
STAVANGER, 4001,
Norway.

Res.

Mathematical Programming Society
Committee on Algorithms Newsletter

No. 11
November 1984

JAN TELGEN
ROBERT R. MEYER
EDITORS

Contents: