functions are listed in Table 1:

- **Functional Analysis:** The functional analysis is used to identify the key functions of the system and their interdependencies. This involves breaking down the system into its basic components and analyzing how they interact with each other. The output of this step is a high-level description of the system's functionality.

- **Process Flow Diagram:** A process flow diagram is created to visualize the flow of data and control within the system. This helps in understanding the sequence of operations and identifying potential bottlenecks or inefficiencies.

- **Data Flow Diagram:** The data flow diagram illustrates the flow of data within the system, including data sources, data stores, and data sinks. It helps in identifying the data requirements and ensuring that the system is designed to handle the expected data volumes.

- **Object-Oriented Modeling:** Object-oriented modeling is used to represent the system as a collection of objects, each with its own characteristics and behaviors. This approach helps in understanding the system's behavior and making it easier to modify and extend the system in the future.

In conclusion, the systematic approach outlined in this paper provides a comprehensive framework for analyzing and designing complex systems. By following these steps, one can ensure that the resulting system is robust, efficient, and meets the needs of the users. The methodologies described here can be applied to a wide range of systems, from software applications to physical systems, and can be adapted to suit the specific requirements of each project.
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 distribution 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

$$
\Theta_k = m(R(x^*))/m(S), \quad k=1, \ldots, W,
$$

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’ 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 distribution 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

$$
\frac{w(N-1)}{N-w-1}
$$

This Bayesian analysis is quite an attractive one, the more so since it can be
shown that the smaller function value found corresponds to the global minimum.

The method did not find the global minimum.

<table>
<thead>
<tr>
<th>Iter</th>
<th>f(x)</th>
<th>h(x)</th>
<th>g(x)</th>
<th>f(x) - g(x)</th>
<th>h(x) - g(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>0.5</td>
<td>0.5</td>
<td>100</td>
<td>0.5</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>0.5</td>
<td>0.5</td>
<td>100</td>
<td>0.5</td>
</tr>
</tbody>
</table>

**Table 1**

**Table 2**

The method did not find the global minimum.

Introduction.

The neighborhoods

Economic theory

Alexander H. Kuo, University of California

Formal Optimization

**The Multi-Level Single Linear Method**

- 8 -
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


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 $O(n^{1.5})$ 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 ellipsoid or equivalently solution of a linear system of equations of the type $(Aa^T)x = b$, 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 successive rank-one modifications and proves a worst-case bound of $O(n^{1.5})$ 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 (1), but apparently the optimum was used as a first guess for the optimal objective function value (1).

We hope to report more on this algorithm in the next issue.
A NEW POLYNOMIAL-TIME ALGORITHM FOR LINEAR PROGRAMMING

The algorithm for linear programming is based on repeated application of a shorter problem to
shorter problems. In the worst-case complexity of any algorithm for linear pro-

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 FNET, 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. Similarly, 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 108 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 version of this problem too. For more information contact him directly.
The text on the page appears to be a description of research findings, possibly related to signal processing or data analysis. The text is not entirely clear due to the quality of the image, but it seems to discuss the effects of certain operations on data sets and the outcomes of various experiments or simulations.

The document appears to be a technical report or a research paper, given the context and the mathematical notation used. It mentions the use of matrices and possibly uses terms related to signal processing or data analysis.

Despite the partial obscurity of the text, it is evident that the document deals with quantitative data and is likely intended for an audience familiar with advanced mathematics or computer science.
bution. 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 optimize 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:

<table>
<thead>
<tr>
<th>D</th>
<th>TYPE</th>
<th>ROWS</th>
<th>COLUMNS</th>
<th>ITERATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>Primal</td>
<td>61</td>
<td>900</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>Dual</td>
<td>901</td>
<td>60</td>
<td>65</td>
</tr>
<tr>
<td>50</td>
<td>Primal</td>
<td>101</td>
<td>2500</td>
<td>102</td>
</tr>
<tr>
<td></td>
<td>Dual</td>
<td>2501</td>
<td>100</td>
<td>130</td>
</tr>
<tr>
<td>50</td>
<td>Primal</td>
<td>101</td>
<td>2500</td>
<td>120 Different Random Nos.</td>
</tr>
<tr>
<td></td>
<td>Dual</td>
<td>2501</td>
<td>100</td>
<td>143</td>
</tr>
<tr>
<td>90</td>
<td>Primal</td>
<td>181</td>
<td>8100</td>
<td>203</td>
</tr>
<tr>
<td></td>
<td>Dual</td>
<td>8101</td>
<td>180</td>
<td>256</td>
</tr>
</tbody>
</table>

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.

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 sciences (e.g. chemical equilibrium problems, crystallography 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
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: 23rd 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, Massachu-
setts, 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 an-
nouncement 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
The newsletter's primary objective is to serve as a forum for the exchange of ideas, sharing of experiences, and development of potential solutions. The committee also aims to promote the field of computational and mathematical sciences through various activities and initiatives.

**Goal Objectives**

- **Educational Objectives:**
  - Promote and support education in computational and mathematical sciences.
  - Foster a community of scholars and professionals in these fields.

- **Research Objectives:**
  - Encourage and support research in computational and mathematical sciences.
  - Foster collaboration among researchers and practitioners.

- **Technology Objectives:**
  - Develop and promote the use of computational tools and techniques.
  - Promote the integration of technology into teaching and learning.

- **Policy Objectives:**
  - Advocate for policies that support the growth and development of computational and mathematical sciences.
  - Promote equitable access to resources and opportunities.

- **Community Objectives:**
  - Foster a sense of community among members.
  - Encourage participation in professional and social activities.

**Committee Contact Information**

- **Chair:** Dr. Alan Johnson
- **Secretary:** Dr. Jane Doe
- **Treasurer:** Mr. John Smith

**Board of Directors**

- **President:** Dr. Robert Lee
- **Past President:** Dr. Emily Brown
- **Vice President:** Dr. Michael Davis
- **Secretary-Treasurer:** Dr. Jane Smith

**Committee Membership**

- **Individual Members:**
  - Dr. John Doe
  - Dr. Emily Brown
  - Mr. Michael Davis
- **Affiliated Organizations:**
  - National Science Foundation
  - American Mathematical Society
  - IEEE Computer Society

**Meeting Information**

- **Next Meeting:** December 2023
- **Location:** Virtual Event

**Contact Information**

- **Email:** info@computersociety.org
- **Phone:** 123-456-7890

**Sponsorship Opportunities**

- **Sponsorship Levels:**
  - Platinum: $10,000
  - Gold: $5,000
  - Silver: $2,500

**Supporting Members**

- **Corporate Members:**
  - Google
  - Microsoft
  - IBM

**Newsletter Subscriptions**

- **Subscription Information:**
  - Individual: $50/year
  - Institutional: $150/year

**Newsletter Distribution**

- **Distribution:** Global
  - Print: 10,000 copies
  - Digital: 15,000 subscribers
Contents:

Editorial column ........................................................................................................... 1

Robert R. Meyer and Jan Telgen

Matching lectures and speakers .................................................................................. 2

NATO ASI on Computational Mathematical Programming ........................................ 3

Klaus Schittkowski

Code Breaker Models .................................................................................................. 5

Larry Naverly

A new Polynomial-time Algorithm for Linear Programming ..................................... 6

The multilevel single linkage method for global optimization ................................... 8

Alexander H.J. Rinnooy Kan and Gerrit T. Timmer

The number of simplex iterations for a general LP code on network problems ........... 16

C.A. Haverly

MPS calendar .................................................................................................................. 20