mechanical structure design problems.

sections mechanical structure design problems to solve.

several mechanical structure design problems have been purchased by

are important goals of NIP. A code cannot be purchased for the

computer while correct at the point in time of purchase. 0.

Silver [6] and the optimal velocity contour of a cographic

integrity of the data. 0.7. These have been practical applications to the

and generated performance of NIP has been evaluated on randomly

applying the computer to the best possible solution. 0.8. The

Numerical these: the program NIP has been tested.

6. Interference: the mechanical structure's interference

caused by some lifting supports called CON.

7. The CIP does not provide a lift to the system. In the test run conducted

sheet NIP could be implemented automatically and

which could be implemented automatically and

NIP: This test compares the mechanical structure's performance.

and some

repeated measurements of the mechanical structure's interference in

and some

repeated measurements of the mechanical structure's interference in

and some

repeated measurements of the mechanical structure's interference in

and some

repeated measurements of the mechanical structure's interference in

and some
10. **Availability:** For more information about the mathematical algorithm or the conditions for obtaining the program package on a magnetic tape, write to

Klaus Schittkowski  
Institut für Angewandte Mathematik und Statistik  
Universität Würzburg  
D-8700 Würzburg, Germany, F.R.

An order form is available to specify the desired magnetic tape format. The Institut für Angewandte Mathematik und Statistik requires a charge of DM 250.-- (approximately $100.--) for the handling costs.

**REFERENCES**


h) **Restart in error cases:** If requested by a user, NLPQL will perform automatic restarts in error situations, cf. k).

i) **Modular structure:** NLPQL possesses a modular structure. Besides the solution method for the subproblem, a user could replace any of the default subalgorithms, i.e. he could replace the merit function, the steplength algorithm, and the numerical differentiation routine. The usage of these modules is described in the documentation, cf.[8]. In particular he could use NLPQL as a frame to test one of the mentioned modules. This is facilitated by the inclusion of a complete test program and a set of 115 nonlinear programming test examples, cf.1).

j) **Output facilities:** A user is allowed to suppress all output. Alternatively a final convergence analysis or additional output for each iteration step can be required. Some further output from the program modules mentioned above can be printed on request.

k) **Error analysis:** A detailed error analysis shows the user the termination reason of the algorithm if the problem could not be solved successfully. Besides failures which could occur in a user provided module, the algorithm will stop in one of the following error cases: Upper bound of iteration number reached, uphill search direction, underflow in BFGS-update, upper bound for line search iteration number reached, length of a working array too short, false dimensions, etc.

l) **Test program:** The program package distributed by the author, contains a complete test program and a set of 115 test problems, cf.[7]. A user is able then to test the correct implementation of NLPQL and to compare his results with those obtained by the author.
The numerical performance of NPLP in this case, the numerical performance of NPLP in this case, the...
Consider a decision-maker who is in possession of several different methods or techniques for attacking a given class of decision problems. The alternative methods share a common objective but, being different, produce different results upon application to a particular problem. Given an arsenal of alternative methods a natural question is: Which one should be chosen to achieve the "best" results overall?

In the two Working Papers cited below, we provide a methodological procedure for choosing the best technique from a number of alternatives based on their performance over a battery of typical problems. The proposed expected utility approach was originally developed in connection with the comparison of heuristic procedures for combinatorial optimization problems.

Suppose that a decision-maker has available \( K \) alternative methods \( A_1, A_2, \ldots, A_K \) and that a set of \( n \) test problems \( P_1, P_2, \ldots, P_n \) are chosen on which to apply the \( K \) methods. We let \( X_{ij} \) denote the results of applying alternative \( A_j \) to the \( i \)-th test problem. Since there are serious drawbacks to analyzing the \( X_{ij} \) values directly, the results are modified to report the relative accuracy of each result in terms of a percentage deviation from a reference point known as the target value \( t_i \) for the \( i \)-th problem. For example, if the decision-maker is studying the performance of \( K \) heuristic procedures with respect to a set of \( n \) instances of a specific combinatorial minimization problem, then \( X_{ij} \) would represent a heuristic solution and the target value might be a known lower bound on the optimal solution to the problem. Thus, the percentage deviation \( P_{ij} \) (of the result obtained by method \( j \) on the \( i \)-th problem) from the target value \( t_i \) would be defined as

\[
P_{ij} = 100 \left( \frac{X_{ij}}{t_i} - 1 \right) \quad 1 \leq i \leq n; \ 1 \leq j \leq K.
\]

After the decision-maker has replaced the \( X_{ij} \) values with the corresponding set of \( P_{ij} \)'s, the expected utility approach would be used to provide a ranking of the alternative methods. The approach models the performance of each method probabilistically in terms of a gamma distribution and chooses a risk-averse a user adapts an available library routine or tries to implement one of the proposed algorithms, cf.[7]. The problem size is therefore limited by the core size and the capability of the quadratic programming subroutine to solve large problems. Note also that the accuracy and reliability of the quadratic programming algorithm could influence the overall numerical performance of NLPQL significantly. The nonlinear programming code has been tested successfully on small size, smooth, and deterministic problems, i.e. on continuously differentiable problems with \( m, n \leq 50 \), where all functions could be evaluated within the machine precision, cf.[9]. For this type of optimization problems, the algorithm seems to be quite robust with respect to numerical differentiation.

4. Capabilities of NLPQL: NLPQL possesses the following capabilities for facilitating the solution of a problem or to adapt the algorithm to a specific situation.

a) Different execution levels: The numerical solution of a nonlinear programming problem can be performed in three different ways. First it is possible to use a main program where only the most important data are read in, i.e. \( n, m, m, \) PRTIN, \( x_i, x_0, x_0 \). PRTIN controls the desired output and the array \( x_0 \) has to contain a starting point. Alternatively, an easy-to-use subroutine NLPQL may be executed, where the same data and some working arrays have to be declared in form of subroutine arguments. To alter default values or to solve the problem in one of the ways outlined subsequently, the flexible and problem adaptable subroutine NLPQL1 is offered.

b) Alternate subproblems: Instead of the quadratic programming subproblem, it is possible to formulate a linear least squares problem in each iteration. This allows the use of any available least squares program instead of a quadratic programming routine. In principle, any carefully implemented linear least squares algorithm will lead to the same
In the form of a complete ranking of all alternatives, the expected utility approach considers all available options. In step 2, a decision utility function is selected, which is either a two-parameter gamma distribution or a more complex function. In step 1, the decision maker needs to choose a target value for each problem and compute the expected utility values according to their expected utility function. The methods are then ranked to reflect the decision-maker's priorities.

1. Mathematical model: The program NCP has been designed to solve the generalized programming problem.

2. In step 1, a two-parameter gamma distribution or a more complex function is chosen. In step 2, a decision utility function is selected, which is either a two-parameter gamma distribution or a more complex function. In step 3, a decision utility function is selected, which is either a two-parameter gamma distribution or a more complex function. In step 4, the decision maker needs to choose a target value for each problem and compute the expected utility values according to their expected utility function. The methods are then ranked to reflect the decision-maker's priorities.

3. Special structure: It is to be emphasized by the user, can take advantage of the input to the program, which includes the solution method for the problem. Once the solution is found, the expected utility values are then calculated by applying the expected utility function to the solution.
References


Some implementation details for a nonlinear programming algorithm

- Klaus Schittkowski -
Institut für Angewandte Mathematik und Statistik
Universität Würzburg
8700 Würzburg
Germany, F.R.

The idea to implement the nonlinear programming algorithm considered here was conceived after finishing a comparative study of 26 optimization codes, cf. [3,7]. These FORTRAN programs were submitted in original form by their authors and have been tested extensively on about 500 randomly generated, hand selected, and real life test problems. The computational tests established the efficiency of the successive quadratic programming algorithms. To investigate the performance of these algorithms in more detail and to understand their outstanding behaviour, the author began to implement and test his own version, called NLPQL. This code was not designed to improve upon the efficiency of the best programs of the comparative study, e.g. Powell's program VPO2AD, since it uses the same mathematical strategy, but rather to implement an efficient, reliable algorithm in an user oriented, flexible, and modular way. This note describes the organization of the program, which is available now on request from the author.
not program it for further testing.

Presented to problem solvers, we presented the method in the volume but did

signification columns can lead to large expansions in large-value programs

exploiting problem structure. Even quite simple tests such as zero rows of

problems and the substantial reduction in problem size that can be made by

last method (sketching, proof, and graph) was mainly concerned with large-scale

programming. The first method (sketching, proof, and graph) was mainly concerned with large-scale

Thompson, G. Droop, and G. Droop, G. Droop.


developed by S. Doss and S. Doss, 1. Doss, 0. Doss, 0. Doss.

significance of reduced techniques for further investigation. These methods were

The survey resulted in selecting a set of the most promising

We began the study by performing a comprehensive survey of the literature

1. INTRODUCTION

This study is a brief summary of the results.

The key results are published in [2]. In this note we present a general description of

problems. The study's goal was to learn what was developed and developed a -

getting the simplest form of reduced techniques in integer programming (LP).

The authors of this paper conducted a study during 1986 and 1991. Invest-

redundancy in mathematical programming

and

State University of New York at Buffalo

Mark, L. Karmarkar, W. L. Karmarkar, and Z. L. Karmarkar

ACM TOMS/February 2023

Rudimentary Curriculum Amendment

NP MRPS FOR EXCELLENCE IN

CONTRIBUTION: MATHEMATICAL PROGRAMMING

- 23 -
2. PROGRAMMING THE METHODS

The FORTRAN language was used for our programming purposes because of its widespread use and because most of the participants were familiar with it. As a result, when we faced difficulties in interpretation, it was easier to consult with the participant to overcome the difficulties.

In programming the methods, we tried to minimize the effects of any programming bias. This was done by using common subroutines where possible (i.e., certain methods had one or more steps in common, e.g., the simplex pivot). Moreover, we utilized a relative zero \(10^{-8}\) in arithmetic operations to minimize the effects of round-off errors. Any number smaller than \(10^{-8}\) was set to zero. In this manner, disagreements among methods on the status of a given constraint (e.g., weakly redundant, strongly redundant and nonredundant) were eliminated. We also provided an initial feasible solution for all of the methods which needed one. The computational effort used to obtain such a solution was not included in evaluating the performance of each method.

3. THE EXPERIMENTAL DESIGN

In order to evaluate the relative efficiencies of the selected methods, two types of LP problems were used. First we solved thirty randomly generated problems by each method and collected relevant statistics. Then we used thirteen structured and real-world LP problems in order to verify our conclusions drawn from randomly generated problems.

These requirements are intended as guidelines to the screening committee but are not to be viewed as binding when work of exceptional merit comes close to satisfying them.

Frequency & Amount of the Award:

The award will be presented, triennially at the Annual Awards Ceremony to the International Symposium on Mathematical Programming Sponsored by the Operations Research Society. The prize for 1985 consists of $500.00 and a certificate.

Judgement Criteria:

Nominations will be judged on the following criteria:

1. Magnitude of the contribution to the advancement of computational and experimental mathematical programming.
2. Originality of ideas and methods.
3. Degree to which unification or simplification of existing methodologies is achieved.
4. Clarify and excellence of exposition.

The Awards Committee:

The screening process will be carried out by a committee of four to eight people appointed by the chairman of COAL at least one year prior to the awarding of the prize, with the review and approval of the chairman of MPS.

Each committee member will read all nominations and provide to the chairman of the committee their assessment based on the right to determine that there will be no prize given for that meeting.

Nominations:

Nominations must be in writing and include the title(s) of the paper(s) or book, the author(s), the place and date of publication and four copies of the materials. Supporting justification and any supplementary materials are welcome but not mandatory. All nominations must be received at least one year prior to the award date. The screening committee reserves the right to request further supporting materials from the nominees.
The structured problems were collected from various sources and ranged in size from 12-constructants to 96-constructants, 70-variables.

The structured problems were collected from various sources and ranged in size from 12-constructants to 96-constructants, 70-variables.

The performance of a given method on a given set of problems versus non-solvable sets of problems for their special characteristics. For example, we compared

The above design enabled us to make pairwise comparisons among different

**Excluding single variables.**

<table>
<thead>
<tr>
<th>Set</th>
<th>Constructants</th>
<th>Variables</th>
<th>Redundancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>None</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>None</td>
<td>None</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>None</td>
<td>None</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>None</td>
<td>None</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>None</td>
<td>None</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>None</td>
<td>None</td>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

The randomly generated problems were generated by using a modified version of O'Neill and Lam's method. The 50 randomly generated problems. By using a modified version of O'Neill and Lam's method. The 50 randomly generated problems were generated by using a modified version of O'Neill and Lam's method.
4. SUMMARY OF THE RESULTS

The methods of Zionts and Wallenius, Telgen, Gal, and Rubin are based on the same basic principle. That is, all of these methods classify constraints by minimizing the associated slack variable. In doing so, these methods perform various tests on the signs of the contracted-simplex tableau. We refer to these methods appropriately, as the sign test methods.

The main objective of the sign test methods is to identify redundant constraints. However, when applied to the dual problem, they may be used to identify nonbinding constraints and/or extraneous variables. We evaluated the sign test methods for their main objective. In general, these methods performed the same on smaller-sized problems. On larger-sized and structured problems Zionts and Wallenius' method generally performed best although there were some problems in which Gal's and Telgen's method did very well. The latter two methods' better performance was most evident on certain structured problems.

The random direction method of Boneh was rather disappointing. The coordinate direction option (suggested by Telgen and implemented and tested by Boneh) was much more efficient than the random direction option. However, when compared to the sign test methods it did not prove any better. Its strength lies in its ability to handle nonlinear problems while the sign test methods are only applicable to LP problems.

Examples of research which would not qualify for this prize:
1. Any proprietary software.
2. Unsubstantiated claims of performance of MP algorithms or software.
3. A collection of test problems without its application to some experimental or application effort.
4. The abstract design of an experiment without any examples of its usefulness to some important class of MP problems.
Testing of classes of software.

1. Development of methods for reducing the cost of experimental testing of classes of software. It also clearly reveals that a number of practical, real-world test problems do not lead to significant improvements in the efficiency of the regular simplex method. The nonadaptive contrastive method outperformed the algorithms and correlated strongly with Thompson and Wilkins' statistical contrastive method. The nonadaptive contrastive method was compared with Thompson's method. Thompson and Wilkins' methods showed an advantage in the nonadaptive contrastive method.

2. Development of algorithms which improve understanding of the computational characteristics of mathematical programming. The computational characteristics which contribute to the efficiency of the algorithms are not all the same. The innovation of some of the algorithms in the proposal are not all the same. The innovations of the algorithms in the proposal are not all the same. The innovations of some of the algorithms in the proposal are not all the same.

Examples of research which might be nominated:

In conclusion, the improvements of the proposals in the proposal are not all the same. The innovations of some of the algorithms in the proposal are not all the same. The innovations of some of the algorithms in the proposal are not all the same. The innovations of some of the algorithms in the proposal are not all the same.

Proposal for a Prize in...

- 27 -
improves with increase in size, we could not verify this. Thompson and Sethi have made improvements to their approach and also tested them on special problem structures since our study was completed. Their work looks quite promising.

5. OTHER DEVELOPMENTS

In studying the various size-reduction techniques, we discovered many possibilities for improvements and extensions. We selected some of the more promising ideas and tested them. Our first improvement, called the extended sign test method, was the result of combining and modifying the sign test methods. As we had anticipated, the extended sign test method outperformed the sign test methods consistently.

We then combined the extended sign test method with the coordinate direction method. This hybrid method outperformed the coordinate direction method. However, it performed the same as the extended sign test method.

Our last extension was to apply the extended sign test method to both the primal and the dual problems while solving the problem. This approach enabled us to identify redundant constraints as well as nonbinding constraints and extraneous variables. We eliminated such variables and constraints as soon as they were identified thereby reducing the size of the problem. This approach proved rather impressive in special cases. In particular, on larger problems

Besides these administrative activities, there were many well attended technical sessions covering topics related to computational and experimental mathematical programming. In all respects, the 11th MTS Symposium on Mathematical Programming helped advance the objectives of COAL.
In the course of study, several other ideas of improvements and extensions were developed. Many of these ideas warrant further study. The methods performed in the regular simplex method by a mixed margin, containing a high degree of redundancy and degeneracy, our extension.

REFERENCES


[4] 0.77, 0.8, and 0.97, respectively. The committee would be deprived.

[5] The committee would be deprived of the opportunity to participate in the meeting. It was agreed that the meeting should be held either directly before or directly after the conference. The committee would be deprived of the opportunity to participate in the meeting. It was agreed that the meeting should be held either directly before or directly after the conference. The committee would be deprived of the opportunity to participate in the meeting.

It was agreed at the meeting that:

1. The committee would be deprived of the opportunity to participate in the meeting. It was agreed that the meeting should be held either directly before or directly after the conference. The committee would be deprived of the opportunity to participate in the meeting. It was agreed that the meeting should be held either directly before or directly after the conference. The committee would be deprived of the opportunity to participate in the meeting.

2. In order to participate in the meeting, it was agreed that the meeting should be held either directly before or directly after the conference. The committee would be deprived of the opportunity to participate in the meeting. It was agreed that the meeting should be held either directly before or directly after the conference. The committee would be deprived of the opportunity to participate in the meeting.

3. The committee would be deprived of the opportunity to participate in the meeting. It was agreed that the meeting should be held either directly before or directly after the conference. The committee would be deprived of the opportunity to participate in the meeting. It was agreed that the meeting should be held either directly before or directly after the conference. The committee would be deprived of the opportunity to participate in the meeting.

4. The committee would be deprived of the opportunity to participate in the meeting. It was agreed that the meeting should be held either directly before or directly after the conference. The committee would be deprived of the opportunity to participate in the meeting. It was agreed that the meeting should be held either directly before or directly after the conference. The committee would be deprived of the opportunity to participate in the meeting.

5. After much discussion about the possibility of extending support, a budget of no more than $100,000 dollars per year was proposed by the committee. The committee agreed to extend additional financial support of the committee. The committee agreed to extend additional financial support of the committee. The committee agreed to extend additional financial support of the committee. The committee agreed to extend additional financial support of the committee.

6. The society agreed to undertake additional financial support of the committee. The society agreed to undertake additional financial support of the committee. The society agreed to undertake additional financial support of the committee. The society agreed to undertake additional financial support of the committee. The society agreed to undertake additional financial support of the committee.
Calendar of mathematical programming meetings
as of 15 December 1982

Maintained by the Mathematical Programming Society (MPS)

This Calendar lists 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 in the meeting. (These meetings are not necessarily "open".) Any one knowing of a forthcoming meeting not listed here is urged to inform Dr. Philip Wolfe, IBM Research 33-221, POB 216, 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. For further information address the Treasurer of the Society, Dr. A. C. Williams, Mobil Corporation, 150 East 42nd Street, New York, New York 10017, U.S.A.; Telephone 212-883-7678.

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.

1983

March 28-31: NETFLOW83, International Workshop on Network Flow Optimization Theory and Practice, Pisa, Italy. Contact: Dr. Claudio Sandi, IBM Scientific Center, Via S. Maria 67, 56100 Pisa, Italy; telephone 50-47383.

April 5-10: 15th Conference "Mathematical Optimization", Selin/Rügen, German Democratic Republic. Contact: Dr. R. Hansel, Sektion Mathematik, Humboldt-Universität zu Berlin, 1086 Berlin, German Democratic Republic; telephone 203 2239.

May 13-14: "Optimization Days", Campus de la Université de Montréal, Canada. Contact: Professor Michael Pólya, École Polytechnique de Montréal, C.P. 6079, Succ. "A", Montréal, Québec, Canada H3C 3A7; telephone 514-344-4884, Telex 05-24146 BIBPOLYTEC. Sponsored by ACFAS, CAMS, IEEE, and the MPS.


July 25-29: 11th IFIP Conference on System Modelling and Optimization, Copenhagen, Denmark. Deadline for abstracts, 31 December 1982. Contact: Professor P. Thoft-Christensen, Institute of Building Technology and Structural Engineering, Aalborg University Center, P.O. Box 159, DK-9100 Aalborg, Denmark.

We have also begun plans for a NATO-sponsored Summer School on Experimental Mathematical Programming to be held in the summer of 1984. Klaus Schittkowski is coordinator of this symposium. At this meeting, we hope to bring together both developers of MP algorithms, developers of MP software, and users of MP codes in order to expand the communication among these groups and to better assess likely avenues for future applications of MP software.

Finally, we hope to prepare and publish a catalog of test problems in mathematical programming.

These above-mentioned tasks represent the major efforts of COAL in the coming months. With your help, we will be successful in each of these.
MESSAGE FROM THE CHAIRMAN

I am delighted and surprised that in the past few years the number of persons receiving this newsletter has increased from an original mailing list of 60 to one now totalling over 900 names. This newsletter was always planned as an informal outlet for communication about research in computational mathematical programming and the response indicates to me that there is a broad community who is interested in the results of that research.

Beginning with this issue, COAL has a new newsletter editor. I would like to take this opportunity to ask you to help me keep the newsletter interesting and useful. I am sending him news of software and/or test problems and suggestions about what you would like to see in future issues of the newsletter.

Since this is my first message as chairman, I think it is an appropriate time for me to explain that the Committee on Algorithms is composed of only 14 people. Without the help of many "friends" of COAL, the achievements of the committee would not have been possible. I thank you for that assistance and hope that it will continue and even grow in the near future.

We are now in the process of reviewing our past objectives and considering what future directions the committee should take. Our activities in the near future will include sessions at upcoming ORSA/TIMS meetings and at other international society meetings. We will also be broadening our scope to include other computational aspects of mathematical programming. I welcome your views of this issue.

Included in this newsletter is a proposal for a prize to be awarded for research in computational mathematical programming. Originally, COAL was concerned solely with the testing of mathematical programming software. However, recently many of the members and "friends" of COAL have suggested that our objectives should be broadened to include other computational aspects of mathematical programming. I welcome your views of this issue.

If the Council of the Mathematical Programming Society endorses this idea, the first award will be presented at the 12th Mathematical Programming Symposium in 1985.

- 2 -

- 31 -
EDITOR'S COLUMN

It is quite common for newly appointed editors taking over a flourishing journal to state that they are facing an extremely difficult, if not impossible task. The evidence to support this statement consists of noting that:
- the leaving editor did such a great job and
- the journal is doing well.

These facts can be noted for the COAL Newsletter as well. While they provide me with an ideal assist to create an a priori excuse for my performance as editor of the COAL Newsletter, I want to use them to point out something else: Karla Hoffman's contribution to COAL and this Newsletter.
She had to bring the Newsletter into existence, and she did more than that. Under Karla's editorship the COAL Newsletter grew out to be a major factor in the very existence of the Committee on Algorithms. She did a marvellous job!

I'll try to put in as much creativeness, perseverance and labour as Karla did as an editor. And I am very glad I won't be alone. Starting with the next issue of the COAL Newsletter, Bob Meyer will be acting as co-editor for this Newsletter. Bob will be the contact for North American contributions to the Newsletter and he will have a special eye on contributions relating to Networks and NLP. I look forward to a fruitful cooperation.

Regarding the editorial policy for the Newsletter I think it is doing well considering the function it was meant for at its conception. Now, after 5 years of operation on this basis, it is appropriate to reconsider its function in view of new developments. These developments include the origination of OPTIMA (the other Newsletter from the Mathematical Programming Society) and the length of the Mailing List in combination with the increasing costs of mailing.
Since MPS is the only source of money for the COAL Newsletter and non-MPS members are receiving the Newsletter too (free of charge), it is only natural to discuss the role of the COAL Newsletter in the context of the other publications of MPS. Until this discussion is finalized the editorial policy for the COAL Newsletter will not be changed.
Mathematical Programming Society
Committee on Algorithms Newsletter

8 March 1983

Jan Telgen, Editor

Contents:

Editor's Column ............................................. 1
Message from the chairman ................................. 2
Coal-related activities at the 11th MPS symposium in Bonn .......................... 4
Proposal for a prize in computational mathematical programming .................. 6
An MPS prize for excellence in computational mathematical programming? .......... 10
J.K. Lenstra and A.H.G. Rinnooy Kan

Some implementation details for a nonlinear programming algorithm .................. 11
Klaus Schittkowski

An Expected Utility Approach for Ranking Heuristics and Other Alternatives ........ 10
B. Golden, A. Assad, F. Waeli

Redundancy in Mathematical programming ................................ 13
Mark H. Karwan, Vahid Lotfi, Stanley Zionts and Jan Telgen

Calendar of mathematical programming meetings ................................ 10