South African
Computer
Journal
Number 18
December 1996

Suid-Afrikaanse
Rekenaar-
tydskrif
Nommer 18
Desember 1996

Computer Science and Information Systems

Special Edition: Computer Security

Rekenaarwetenskap en Inligtingstelsels

The South African Computer Journal
An official publication of the Computer Society of South Africa and the South African Institute of Computer Scientists

Die Suid-Afrikaanse Rekenaartydskrif
'n Amptelike publikasie van die Rekenaarverenging van Suid-Afrika en die Suid-Afrikaanse Instituut vir Rekenaarwetenskaplikes

Subscriptions

	Annual	Single copy
Southern Africa:	R50,00	R25,00
Elsewhere:	\$30,00	\$15,00

An additional \$15 per year is charged for airmail outside Southern Africa

to be sent to:
Computer Society of South Africa
Box 1714 Halfway House 1685

Phone: +27 (11) 315-1319 Fax: +27 (11) 315-2276

Drukkers Printers

Guest Contribution

A Pragmatic Approach to Development Information to Provide Service on a Wide Scale

Stephen S Mncube, PhD
Divisional Manager: Development Information Services, Centre for Policy and Information, Development Bank of Southern Africa

1 Introduction

The rapid technological growth in our time has produced an explosion of information. This, in turn, has spawned information systems based on the use of computers and automated systems. These mechanised devices with their seemingly infinite capacity to store and retrieve knowledge on command have myriad applications. But the use of computer and automated information devices pose serious problems to individuals, groups and societies on an international scale in disseminating the available information. This is even more true in information flow between regions with high information capabilities than in those that have little or none. The information flow between these regions has been varied and frequently haphazard whenever it has existed.

In this context the philosophy for development information speaks to interdependence and humanitarian concern in information flow to various regions of the earth. Information must be perceived as a universal entity. The initial failure of one region and the ability of another to acquire information should not dictate a permanent global demarcation into 'developed and less developed' categories.

2 Objectives

The objective of development information is for it to function as an interlinking mechanism between a complex industry of information resources and the users of information. In addition, development information should be a catalytic agent that aims at providing objective clarification in information needs between regions with high information capabilities and those with little.

3 Mission

The mission is to strive for a move in all regional, national and international agencies and organisations concerned with information to give full co-operation and assistance in setting criteria and standards, formulating policy, and assessing the information needs of a given region of the world. Development information is capable of realistically assessing information related to socio-economic development in the light of the unique requirements of world regions in need of information and information technology.

4 The Dynamics of Information Processing

The proliferation of information technology production has resulted in the growth and development of an ever expanding information packaging industry. This area has become so large in size and scope that it is necessary to discuss some of the important developments that are taking place in the area.

The newest phrases now being used in information packaging range from electronic archives, compact discs (CDs), computer tapes, microfiche, teletexts, video discs, magnetic tapes and interactive imaging systems (optical systems) to word processing and the use of laser technology. Developing countries will not escape this new wave of information packaging. It soon will be bombarded with vendors of these products, to a point where some adaptation will be inevitable.

Assuring the quality of technical processes and the accuracy of packaging information is becoming an increasingly difficult task. Rapid increases in the volume of information, the sophistication of information uses, and the complexity of material flows and processes are characteristic of most modern technical environments. As complexity increases, the risk of introducing significant errors into material processes increases. The very complexity of such systems makes the detection of error itself a complex task. With increasing frequency, public and private organisations are seeking help from corporations with experience in quality control and information validation to ensure that technical process and information packaging meet performance and accuracy standards. The problems association with quality control and validation can be minimised by following these guidelines:

- Establishing ways of aiding in planning, organisation and control of software purchasing and development

through
  - creating a directory of software suppliers
  - evaluating the quality of software supplied
  - keeping abreast of the state-of-the-art in software production
- Providing leadership in the innovative use of software materials and the utilisation of extensive market research on software before making a major purchase
- Establishing cost-efficient ways of packaging and designing your own software by learning how to design and evaluate software for your own use

Preparation for these new technologies for packaging information ought to be made in institutions of higher learning. Perhaps it would be timely to introduce some of these concepts in technical institutes in order that future demands imposed by the new information technology may be met.

The successful growth of developing countries information technology will ultimately depend upon the commitment of substantial resources, especially financial resources. The successful application of this information technology will require more than the mere receiving and storing of it. In addition to the tasks of acquiring and organising informational materials, channels must be established to analyse incoming information. Too much of the information technology that does get transferred out of the industrialised nations is never utilised because it is unsuited for the consumption of users in developing countries. A great deal more effort must be made to analyse, package and disseminate materials on existing and forthcoming information in all vital areas of work and study so that these technologies will be accessible to the developing countries' information-user communities.

5 The Need to Establish a Consortium

Information technology specialists need to establish a consortium of regional, national and international information networks and associations. The consortium could be an open structure inviting any institutions, organisations and agencies existing for the purpose of forming a network or documentation clearinghouse and of providing information technology not as an end-product but as a means for human change.

The element common to all membership is an interest in and dedication to providing useful and accurate information that can bring about humanistic change. Equally important is a commitment to the development of relevant information resources to meet the needs of regions with low information capacity.

The philosophical outlook of the consortium would therefore be to crystalise and emphasise broad knowledge, deep understanding, and imaginative efforts, including a dedication to great ideas in providing accurate solutions to the information needs of various regions on an international scale.

6 The Organisation of Services for Members

The ideas constituting a conceptual framework for a service-oriented consortium are as multitudinous as the Kalahari sands but in this instance the consortium could function to:

- support creative change within its membership
- facilitate and support new educational enterprises and programs addressed to meet the needs of previously disenfranchised persons
- develop and implement co-operative programs and projects among it members
- provide a meeting ground for a diversity of persons, institutions, and agencies with common values and purposes
- provide a forum for the exchange of ideas among its member associates
- encourage methods of solving social problems
- influence public policy to be consistent with its mission and purpose.

7 An Appraisal of Internet

While casual observers have the leisure to observe unobtrusively the growth and development in Internet to be a world-wide phenomenon in information sharing, they do so at no cost. On the other hand, information specialists have to judge and weigh the work of an ongoing Internet program and estimate its usefulness as a network or networks to their daily operations. Information specialists are, therefore, still more sceptical about the scope and magnitude of the Internet. They alone are facing challenges of adding another performance task of being evaluation researchers of Internet in order to provide objective clarification of incorporating Internet as an integral component of their information system. This can be a tedious undertaking because it entails not only knowing how to navigate the Internet network but also cognisance of the following key factors:

- how appropriate the Internet is to your information environments
- to what extent the databanks provided through Internet are relevant to the mission and objective of your environment
- what the relationship is between costs and benefits of having Internet at your disposal

All these factors need to be addressed to determine the effectiveness of Internet in any given information environment, be it in a government setting or in other work environments.

8 Training in Information Networks

A seminar for the network should be designed to launch the co-operative exchange of knowledge and experience with information accessibility and utility of the participant's respective information holdings. To succeed in this effort, the organisers will marshal appropriate interdisciplinary experts and technical resources from within the regions involved. The content of the seminar will consist of information related to formalising and establishing a plan for information networking. It is important that information ministries achieve a high level of knowledge and sensitivity to the information needs of their individual country, region, and ultimately the world, in order to assess, prescribe, design, manage and evaluate the most appropriate uses of information technology for enhancing the advancement of their world countries.

Although the organisers will determine who will be invited to participate in this seminar, special effort will be made to ensure that representatives come from a diversity of backgrounds, and have some knowledge or experience relating to information systems. Additionally, consideration will be given to the level of information technology which is currently utilised by the representative's country.

The content of the seminar will be tailored to meet the specific needs and issues designated by the participants through a pre-seminar survey and needs assessment, which will be administered by the organisers.

9 Summary

The above aims at presenting some possible scenario and does not pretend to be exhaustive. The issues, however vital to development information, are given cursory treatment here. It remains important for the luminaries in this area to expand on some of the thoughts contained above.

Bibliography

1. A D Burnett. *Technology for information in development: proceedings of the sixth conference of the International and Comparative Librarianship Group of the Library Association.* International and Comparative Librarianship Group of the Library Association, Brighton Polytechnic, Falmer, Sussex, August 21–23 1987.
2. M Cook. Guidelines on curriculum development in information technology for librarians, documentalists, and archivists. Paris Unesco, 1986.
3. O Fairfax, J Durham, and W W. Audio-visual materials: development of a national cataloguing and information service: Report of a joint feasability study for the british library and the council for educational technology. Distributed by Councils and Education Press, 1976.
4. M Fransman. *1948 – The market and beyond: Cooperation and competition in information technology development in the Japanese system.* Cambridge University Press, New York, 1990.
5. N K Hanna. The information technology revolution and economic development. World Bank Discussion Papers, World Bank Washington DC., 1991.
6. N Heaton and L W MacDonald. *Human factors in information technology product design and development.* Ellis Horwood, 1991.
7. T Koizumi and G E Lasker, eds. *Advances in education and human development: new trends in learning and teaching evolution of human systems and institutions, individual freedom and sociopolitical controls in the high technology environment of the advanced information society.* International Institute for Advanced Studies, Baden-Baden, West Germany, 1990.
8. V Sethi. The development of measures to assess the extent to which an information technology application provides competitive advantage. Ann Arbor: MI, University Microfilm International, University of Pittsburgh, 1988.

SACJ is produced with kind support from
Mosaic Software (Pty) Ltd.

An AI Search Algorithm to Obtain The Shortest Simplex Path

T G Scott J M Hattingh T Steyn
Department of Computer Science, Potchefstroom University for CHE, Potchefstroom 2520.
rkwtgs@pukrs8.puk.ac.za

Abstract

The simplex method is one way of solving a linear programming problem (LP-problem). An A search algorithm based on a certain evaluation function has been developed to obtain the shortest path to an optimal solution within the simplex rules. An evaluation function was used to guide the algorithm to obtain the shortest path through the convex polyhedron. Every extreme point in the polyhedron is presented by a node in the search tree. The evaluation function employs a lower bound of the real path length from node n to the optimal node. The algorithm has been applied successfully to some problems from the NETLIB test suite. Empirical work is presented that compares these shortest paths with some other known column selection heuristics.*

Keywords: *Linear programming, Artificial Intelligence, Simplex method, Search Algorithms, A*-Algorithm*
Computing Review Categories: *G.1.6, I.2.8.*

1 Introduction

Linear programming is a technique that can be applied to the problem of rationing limited facilities and resources among many alternative uses in such a way that their utilization is optimal. Linear programming problems mostly arise from modern technology in medicine, transportation, forestry, agriculture, fuels, networks, distribution of work loads and so on. As technology improves new applications arise. One way of solving a linear programming problem is to use the simplex method of Dantzig [2]. This method traverses along the edges of a convex polyhedron, from one extreme point to another, until it finds an optimal solution. It is not possible to draw a graph and find all the extreme points in a multi-dimensional space. The simplex method in general only visits a small subset of these extreme points although pathological examples have been constructed in such a manner that all extreme points are visited in the solution process [5].

2 The Shortest Simplex Path

Why find the shortest simplex path?

Since the development of the simplex method much research has been done to improve this method. One of the best ways of improving this method is to improve the column selection. It involves choosing an edge to traverse to the next extreme point; thus choosing a non- basic variable to become basic. Although choosing any edge with positive reduced cost is sufficient for the convergence of the simplex method, choosing a "good" edge can reduce the iteration count dramatically [1, 4, 8].

The more iterations the simplex method employs, the more roundoff errors can occur that can affect the process and the final result. Some heuristics have been developed to decide which edge is the "best" to choose. However, it is difficult to define the concept "best edge". None of the current heuristics can guarantee the shortest simplex path. When the shortest simplex path is known it can be used as a benchmark for current column selection heuristics. Another question arises: how much can current column selection heuristics be improved to get close to the shortest path? The simplex algorithm is a search algorithm that does not backtrack to a previous extreme point. The A* algorithm that we will consider here can backtrack and its use is considered for the identification of the shortest simplex path..

The shortest simplex path against the absolute shortest path

The simplex method is implemented in such a way that when it traverses from one extreme point to another, it will not allow a decrease in the value of the objective function. Thus it can happen that the shortest simplex path is not the absolute shortest path to an optimal solution in the sense that the absolute shortest path is the one that constructs a series of points that ignores feasibility of each step and does not look at monotonic behavior of the objective function. Consider Model 1 in Figure 1. Figure 2 is a graphical

$$
\begin{aligned}
\text{Maximize } & 2y - x \\
\text{subject to } 3y + 2x &\leq 18 \\
y - x &\leq 4 \\
3y + x &\geq 3 \\
3y - x &\geq -6 \\
x, y &\geq 0
\end{aligned}
$$

Figure 1. Model 1: First LP example.

presentation of Model 1. Starting at the point (6;0) there

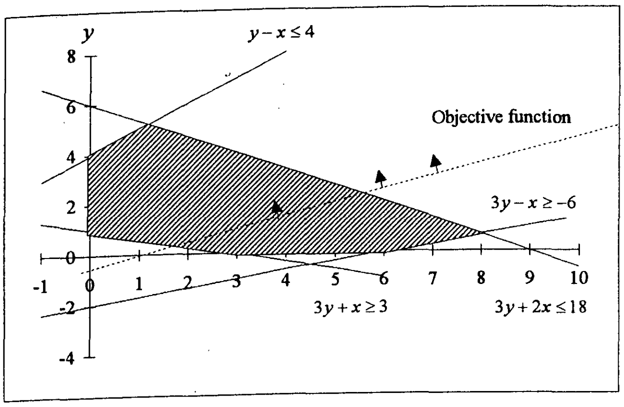

Figure 2. Graphical presentation of Model 1

are two edges to move along; the edge leading to the point (3;0) and the edge leading to (8;0.3). Moving to the point (8;0.3) will clearly lead us on the shortest path to the optimal solution. Unfortunately the simplex method will not consider this choice because it decreases the value of the objective function from -6 to -7.3. Thus the path that the simplex method takes is (6;0)-(3;0)-(0;1)-(0;4)-(1.2;5.2). The shorter path (6;0)-(8;0.3)-(1.2;5.2) ignores the simplex rules.

Finding a heuristic

In the column selection step of the simplex method, a choice may have to be made between two or more edges. Some heuristics can decide beforehand which edge seems to be promising or "best". Practice has shown that this kind of approach is not very effective. Other heuristics consider all the available edges and the associated moves to the next extreme point, and then decide which edge to choose. The heuristic then decides on one edge to traverse and continues along this path. This means that the heuristic cannot backtrack to a previous node. In our approach we will use an evaluation function to decide with which edge we wish to continue.

Another approach is based on the search algorithms of artificial intelligence [6]. It views the search process as a search in a graph with each node corresponding to an extreme point of the convex polyhedron and each arc between nodes corresponding to a simplex iteration. One such algorithm is called the breadth-first algorithm that starts with the root node, then investigates all nodes on the first level for the goal, then all nodes on the second level and so on. This algorithm is impractical for almost all nontrivial linear programs since the branching factor is usually prohibitive.

An algorithm that we have investigated here is the so-called A* algorithm that employs an evaluation function to guide the search process. Loosely speaking, the class of A* algorithms uses branch and bound principles with lower bounds for the remaining path lengths and incorporates dynamic programming principles. This algorithm requires the availability of a goal node which is defined as a specific optimal solution to the problem (determined in some way beforehand). This algorithm, depending on the heuristic used, reduces the work by investigating only a subset of the nodes. It still finds the optimal path. It uses an evaluation function $f(n) = g(n) + h(n)$ to decide which arc to explore in the search process. This evaluation function defines $g(n)$ as a measure of the path length from the root node to node n and $h(n)$ as a lower bound of the path length from any node n to the goal node. The root node is the first basic feasible solution where the process begins.

In this approach we define $g(n)$ as the number of simplex iterations performed to move from the root node to the node n. We calculate $h(n)$ as follows:

- Place those variables that are in the basis at the goal node, but not in the basis at node n in a list called G.
- Define $h(n)$ as the number of variables in list G.

This will ensure that $h(n)$ is a lower bound of the remaining path length (iterations). In the formal algorithm discussed below we will eliminate paths that lead to the same node thus reducing the search graph to a tree. A full description of the A* algorithm is given in Figure 3.

1. Initialize open-list by placing the root node as the only element in this list. Initialize another list, called closed-list which is initially empty.
2. Take the first node in open-list and explore its children. Place these children as the only elements in a list called children-list. If no children exist, then the first node of the open-list is the goal node; thus the algorithm can terminate by moving the node in the open-list to the closed-list and using the closed-list to identify the shortest path.
3. Compare children-list with the open-list. If there is a duplicate node remove either the node from the open-list or the node from the children-list according to a certain heuristic.
4. Compare children-list with the closed-list. If there is a duplicate node, remove a node according to a certain heuristic.
5. Sort the children-list using the evaluation function values with the smallest value in front.
6. Move the first node in the open-list to closed-list and note the pointer to its parent node in the closed-list. Merge every node in children-list into the open-list by using the evaluation function values.
7. Go to step two.

Figure 3. A algorithm*

Comments

1. Every node n has associated with it a triplet $(n, p(n), f(n))$ with n the identifier of the node, $p(n)$ its parent identifier and $f(n)$ the value of the evaluation function $f(n) = g(n) + h(n)$. This triplet enables the ranking of nodes and eventual determination of the optimal path.
2. When the first node in the open-list is the goal (i.e. it is the optimal solution to the LP and thus has no children) it also has at least as good a value of $f(n)$ as the other nodes in the open list. Since development of

nodes in open-list (to try to find other solution paths) cannot result in a smaller value of f ($h(n)$ is always a lower bound of remaining path length) anywhere down a path, the minimal path length has been found. The goal node has $g(n)$ equal to the optimal path length and $h(n) = 0$. If $h(n) \neq 0$ then the search process may also, if desired, be terminated with an alternative optimal solution giving a shorter path than the optimal path to the goal node. In such a case the first node in the open-list is simply another solution of the LP problem that is different from the goal node but also optimal.

It is possible to reduce the computational time for the A* algorithm by splitting the open-list into several open-lists and only doing duplicate removal when necessary. This is explained below.

Algorithmic details and simplifications

We can say that if node n_1 is a predecessor of node n_2 then $f(n_1) \leq f(n_2)$). All the nodes in the open-list have an evaluation function value which is greater than or equal to the evaluation function values of nodes in the closed-list.

Say we split the open-list into several open-lists in such a way that the nodes in each list has the same evaluation function values. Set the open list as $O = \{O_1, O_2, \ldots, O_n\}$ where $O_1, O_2, \ldots, O_n$ is each an open-list in such a way that $f(x) < f(y)$, with $x \in O_i$ and $y \in O_{i+1} \forall i = 1, , \ldots, n-1$. Also in our case $f(x) = f(y)$ with $x, y \in O_i \forall i = 1, , \ldots, n$ and $x \neq y$. A* will choose a node out of the open-list O_1 to explore next. The following can happen to a child of the explored node:

1. The evaluation function value can stay the same.
2. The evaluation function value can increase.

In case (1) the child node would be compared to the closed-list (step 4 of Figure 3) and the open-list O_1 (step 3 of Figure 3) to remove duplicate nodes. The child is not compared with the other open lists $O_2, \ldots, O_n$ because these lists would not have nodes with better (smaller) evaluation function values. If a child node is already present in the closed-list then the newly developed child node is discarded. If the child node is already present in the open-list O_1 then one of the nodes is removed according to a certain heuristic. In this case the node with smallest $g(n)$ was removed.

In case (2) we compare the child with the closed-list. If the child appears in the closed-list, then the child is discarded, otherwise the child is placed in one of the appropriate lists $O_2, \ldots, O_n$.

If none of the children is the goal node, we must explore another node (step 2 of Figure 3). We will choose this next node from the open-list O_1 and explore it. If we find that the open-list O_1 is empty, we then consider the open-list O_2. Before we can explore the nodes in this list, we need to remove duplicate nodes from this list.

To split the open-list into several lists has the advantage that there is no need to keep $O_2, \ldots, O_n$ in the computer's primary memory. This can be stored in secondary memory. If this information is needed by the computer, it can be

$$
\begin{array}{l}
\text{Min } z = -2x_1 - x_2 - \tfrac{2}{3}x_3 \\
\text{subject to } 0 \leq x_j \leq 2, j = 1, 2, 3 \\
\qquad x_1 - x_2 - x_3 \leq 1 \\
\qquad x_1 + x_2 - x_3 \leq 11\tfrac{1}{3} \\
\qquad 5x_1 + 3x_2 + x_3 \leq 17
\end{array}
$$

$$
\begin{array}{llllllllll}
\text{Min } z = -2x_1 - x_2 - \tfrac{2}{3}x_3 & & & & & & & & & \\
\text{subject to } x_1 - & x_2 - 2x_3 + x_4 & & & & & & & & = 1 \\
x_1 + & x_2 - & x_3 & & +x_5 & & & & & = 11\tfrac{1}{3} \\
5x_1 + 3x_2 - & x_3 & & & + & +x_6 & & & & = 17 \\
x_1 & & & & & & x_7 & & & = 2 \\
 & x_2 & & & & & & +x_8 & & = 2 \\
 & & x_3 & & & & & & +x_9 & = 2
\end{array}
$$

Figure 4. Model 2: Second LP example

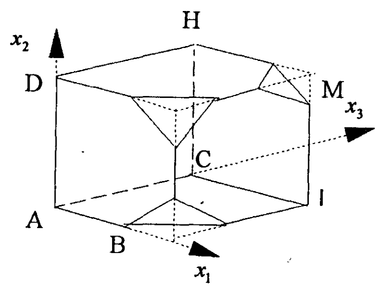

Figure 5. Graphical presentation of Model 2

retrieved to the primary memory. Another advantage is that only duplicate nodes are checked in the open-list O_1 and the closed-list. The other list's $O_2, \ldots, O_n$ are only checked when they are needed, if they are ever needed.

3 Illustration

Consider the LP-problem defined in Model 2 (Figure 4) and presented graphically in Figure 5. Some of the extreme points (or nodes) are marked. Our starting point is at node A, that is presented by the basis $(x_7, x_8, x_9, x_5, x_6, x_4)$. Our

Table 1. Initial open-list

Node	A
Parent	-
$f(n)$	3

evaluation function calculates $g(A) = 0$ because we are at the root node. The goal node has basis $(x_1, x_8, x_3, x_5, x, x_4)$ as previously calculated. At node A we still need to pivot at least three variables (that is x_1, x_C, and x_3) into the basis to get to the goal node; thus $h(A) = 3$. At node A we can traverse along three edges as in indicated in Figure 6. We therefore obtain nodes B, C and D. The basis at B is $(x_7, x_8, x_9, x_5, x_6, x_1)$. This means that $g(B) = 1$ and $h(B) = 3$. At node C the basis is $(x_7, x_8, x_3, x_5, x_6, x_4)$, thus $g(C) = 1$ and $h(C) = 2$. At node D the basis is

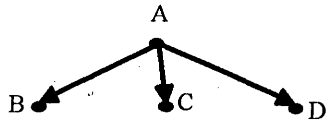

Figure 6. Node A

$(x_7, x_2, x_9, x_5, x_6, x_4)$, thus $g(D) = 1$ and $h(D) = 3$. A summary of these values is given in Table 2. We choose the

Table 2. Open list after node A is explored

Node	C	D	B
Parent	A	A	A
$f(n)$	3	4	4

Table 3. Closed list after node A is explored

Node	A
Parent	-
$f(n)$	3

node with the smallest $f(n)$ value. Thus we choose node C. Node C has two children: nodes I and H. The basis at node

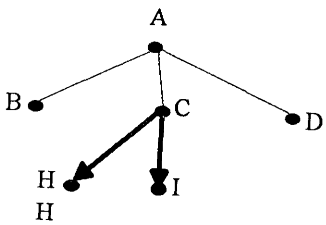

Figure 7. Node C

H is $(x_7, x_2, x_3, x_5, x_6, x_4)$, thus giving us $g(H) = 2$ and $h(H) = 2$. The basis at node I is $(x_1, x_8, x_3, x_5, x_6, x_4)$ thus giving us $g(I) = 2$ and $h(I) = 1$. Table 4 shows a summary of this. From Table 4 we can see that node I is the node to explore. Node I gives us one child: Node M. The basis is $(x_1, x_8, x_3, x_5, x_2, x_4)$ with $g(M) = 3$ and $h(M) = 0$ as shown in Table 6. We continue with node M. Because node M is the optimal node, the algorithm terminates with A-C-I-M (the closed-list of Table 8) as the optimal path.

4 Empirical Experiments

This A* algorithm has been tested on several LP problems from the NETLIB test suite on an IBM RS/6000 Model 40P workstation. These test problems can also be obtained via anonymous ftp from "ftp://ftp.puk.ac.za/pub/local/lpdata/". In order to obtain a shortest simplex path with A*, we need to know a goal node. A goal node can be obtained by simply solving the problem. In order to solve the problem, one should supply the LP solver with a column heuristic for the simplex method. We used three column heuristics:

1. The method of greatest reduced cost (also known as the Dantzig rule [2])

Table 4. Open list after node C is explored

Node	I	H	D	B
Parent	C	C	A	A
$f(n)$	3	4	4	4

Table 5. Closed list after node C is explored

Node	C	A
Parent	A	-
$f(n)$	3	3

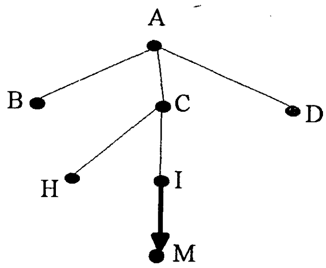

Figure 8. Node I

2. The method of greatest change in the objective function value (we will call it GVD, [8])

3. The full-norm method (explained in [1]).

In order for the simplex method to solve a LP problem, the simplex method needs an initial solution. This solution is obtained in phase 1 of the simplex method. Table 9 is a summary of the experimental results. The first row in this table indicates the name of the problem. The second row indicates the ratio between the number of iterations taken by norm and the number of iterations needed for the shortest simplex path. The third row is ratio of the time (in seconds) needed by norm and shortest simplex path. The next four rows are similar to rows two and three, except that the Dantzig rule is used for rows four and five, while the greatest change in the objective function value (GVD) method is used in rows six and seven. The third last row indicates the number of rows the problem has, and the second last row the number of columns. The last row is the iteration count of the shortest simplex path

Table 6. Open list after node I is explored

Node	M	H	D	B
Parent	I	C	A	A
$f(n)$	3	4	4	4

Table 7. Closed list after node I is explored

Node	I	C	A
Parent	C	A	-
$f(n)$	3	3	3

Table 9. Results of empirical experiments

	adlittle	afiro	beaconfd	sc105	s50a	sc50b	stocfor1	share1b	Average
norm ratio	1.88	1.10	1.00	1.00	1.00	1.00	1.18*	2.41	1.32
timex ratio	2.12	0.92	1.20	1.29	2.13	1.04	0.01	0.44	1.14
Dan. ratio	2.94	1.25*	1.00	1.54*	1.13*	1.00	1.67*	4.63	1.90
timex ratio	0.51	0.71	0.21	0.68	0.68	0.82	0.001	0.08	0.46
GVD ratio	1.53*	1.13*	1.03	1.32*	1.06	1.00	1.35	1.66	1.26
timex ratio	0.14	1.1	1.13	1.12	1.06	0.86	0.03	0.24	0.71
Rows	57	28	174	106	51	51	118	118	
Columns	97	32	262	103	48	48	111	225	
Sh. Path	32	8	30	28	16	12	17	32	

*Alternative LP solutions were found in these cases.

Table 8. Closed list after node M is explored

Node	M	I	C	A
Parent	I	C	A	-
$f(n)$	3	3	3	3

5 Conclusions

For large problems the A* algorithm investigated here were found to be computationally intractable and further algorithmic refinements can be investigated. One possible further approach would be to investigate the application of parallel processing [7]. The experiments suggest that the NORM method finds the shortest path in some cases. Overall the GVD method comes closest to the shortest path on average. The Dantzig rule does almost twice the number of pivots compared to the shortest path on these problems. If we keep in mind that for large problems iteration count can be reduced by large factors if a form of steepest edge is employed [3], these experiments indicate that existing algorithms generally fails to produce shortest paths, and that improved algorithms almost certainly exist.

References

1. H Crowder and J M Hattingh. 'Partially normalized pivot selection in linear programming'. *Mathematical Programming Study*, **4**, (1975).
2. G B Dantzig. *Linear programming and extensions.* Princeton University Press, New Jersey, 1963.
3. J J H Forrest and D Goldfard. 'Steepest-edge simplex algorithms for linear programming'. *Mathematical Programming Study*, **57**, (1992).
4. P M J Harris. 'Pivot selection methods of the devex LP code'. *Mathematical Programming Study*, **4**, (1975).
5. V Klee and G J Minty. 'How good is the simplex algorithm'. In *Inequalities-III*. Academic Press, New York, (1972).
6. G F Luger and W A Stubblefield. *Artificial Intelligence structures and strategies for complex problem solving.* The Benjamin/Cummings Publishing Company, Inc., California, 3rd edition, 1993.
7. J Luo. *Parallel Processing for Linear Programming.* PhD thesis, Delft University, Delft, 1992.
8. P Wolfe and L Cutler. 'Experiments in linear programming'. *Journal of the Society for Industrial Applied Mathematics*, **11**(2), (1963).

Received: 12/95, Accepted: 10/96, Final copy: 10/96.

The prime purpose of the journal is to publish original research papers in the fields of Computer Science and Information Systems, as well as shorter technical research notes. However, non-refereed review and exploratory articles of interest to the journal's readers will be considered for publication under sections marked as Communications or Viewpoints. While English is the preferred language of the journal, papers in Afrikaans will also be accepted. Typed manuscripts for review should be submitted in triplicate to the editor.

Form of Manuscript

Manuscripts for *review* should be prepared according to the following guidelines.

- Use wide margins and $1\frac{1}{2}$ or double spacing.
- The first page should include:
  - title (as brief as possible);
  - author's initials and surname;
  - author's affiliation and address;
  - an abstract of less than 200 words;
  - an appropriate keyword list;
  - a list of relevant Computing Review Categories.
- Tables and figures should be numbered and titled.
- References should be listed at the end of the text in alphabetic order of the (first) author's surname, and should be cited in the text in square brackets [1–3]. References should take the form shown at the end of these notes.

Manuscripts accepted for publication should comply with the above guidelines (except for the spacing requirements), and may be provided in one of the following formats (listed in order of preference):

1. As (a) LaTeX file(s), either on a diskette, or via e-mail/ftp – a LaTeX style file is available from the production editor;
2. As an ASCII file accompanied by a hard-copy showing formatting intentions:
   - Tables and figures should be original line drawings/printouts, (not photocopies) on separate sheets of paper, clearly numbered on the back and ready for cutting and pasting. Figure titles should appear in the text where the figures are to be placed.
   - Mathematical and other symbols may be either handwritten or typed. Greek letters and unusual symbols should be identified in the margin, if they are not clear in the text.

   Contact the production editor for markup instructions.
3. In exceptional cases camera-ready format may be accepted – a detailed page specification is available from the production editor;

Authors of accepted papers will be required to sign a copyright transfer form.

Charges

Charges per final page will be levied on papers accepted for publication. They will be scaled to reflect typesetting, reproduction and other costs. Currently, the minimum rate is R30-00 per final page for LaTeX or camera-ready contributions that require no further attention. The maximum is R120-00 per page (charges include VAT).

These charges may be waived upon request of the author and at the discretion of the editor.

Proofs

Proofs of accepted papers in category 2 above may be sent to the author to ensure that typesetting is correct, and not for addition of new material or major amendments to the text. Corrected proofs should be returned to the production editor within three days.

Note that, in the case of camera-ready submissions, it is the author's responsibility to ensure that such submissions are error-free. Camera-ready submissions will only be accepted if they are in strict accordance with the detailed guidelines.

Letters and Communications

Letters to the editor are welcomed. They should be signed, and should be limited to less than about 500 words.

Announcements and communications of interest to the readership will be considered for publication in a separate section of the journal. Communications may also reflect minor research contributions. However, such communications will not be refereed and will not be deemed as fully-fledged publications for state subsidy purposes.

Book reviews

Contributions in this regard will be welcomed. Views and opinions expressed in such reviews should, however, be regarded as those of the reviewer alone.

Advertisement

Placement of advertisements at R1000-00 per full page per issue and R500-00 per half page per issue will be considered. These charges exclude specialized production costs which will be borne by the advertiser. Enquiries should be directed to the editor.

References

1. E Ashcroft and Z Manna. 'The translation of 'goto' programs to 'while' programs'. In *Proceedings of IFIP Congress 71*, pp. 250–255, Amsterdam, (1972). North-Holland.
2. C Bohm and G Jacopini. 'Flow diagrams, turing machines and languages with only two formation rules'. *Communications of the ACM*, **9**:366–371, (1966).
3. S Ginsburg. *Mathematical theory of context free languages*. McGraw Hill, New York, 1966.

South African Computer Journal

Number 18, December 1996
ISSN 1015-7999

Suid-Afrikaanse Rekenaar-tydskrif

Nommer 18, Desember 1996
ISSN 1015-7999

Contents

GUEST CONTRIBUTION

RESEARCH CONTRIBUTIONS

COMMUNICATIONS AND VIEWPOINTS