UNIVERSITI PUTRA MALAYSIA

PROBLEM RESTRUCTURING IN INTEGER PROGRAMMING
FOR REDUCT SEARCHING

UNGKU AZMI ISKANDAR UNGKU CHULAN

FSKTM 2003 2
PROBLEM RESTRUCTURING IN INTEGER PROGRAMMING
FOR REDUCT SEARCHING

UNGKU AZMI ISKANDAR UNGKU CHULAN

MASTER OF SCIENCE
UNIVERSITI PUTRA MALAYSIA
2003
PROBLEM RESTRUCTURING IN INTEGER PROGRAMMING
FOR REDUCT SEARCHING

By
UNGKU AZMI ISKANDAR UNGKU CHULAN

Thesis Submitted to the School of Graduate Studies Universiti Putra Malaysia
In Fulfillment of the Requirements for the Degree of Master of Science

May 2003
What's the best toy one man can find?

The mind, and its infinite possibilities.
Abstract of thesis presented to the Senate of Universiti Putra Malaysia in fulfillment of the requirements for the degree of Master of Science.

PROBLEM RESTRUCTURING IN INTEGER PROGRAMMING FOR REDUCT SEARCHING

By
UNGKU AZMI ISKANDAR UNGKU CHULAN

May 2003

Chairman : Associate Prof. Dr. Haji Md. Nasir Sulaiman
Faculty : Computer Science and Information Technology

Standard Integer Programming / Decision Related Integer Programming (SIP/DRIP) is a reduct searching system that finds the reducts in an information system. These reducts are the minimal attributes of the information system that are useful in classificatory task. They can describe the whole information system when implementing discernment. In effect, they are very useful in generating rules when solving the classification problem that is inherent in data mining.

The thesis emphasizes mainly on the improvement of the original SIP/DRIP algorithm in term of performance. By using problem restructuring, the searching time and memory are minimized. Simultaneously the approach adheres to an essential criterion of the original method. That is, to improve performance without sacrificing the quality of the reduct.
Problem restructuring changes the input of the SIP/DRIP without losing any of input's essential properties. In other words, no lost of knowledge occurs with problem restructuring. Only the structural order changes, with the contents kept intact. This hypothetically ensures that no adverse distortion transpired within SIP/DRIP.

Restructuring is done by speculating a promising structure for the input to SIP/DRIP based on the potentiality of the attributes in the information system. It uses a non-expensive approach to predict potentiality. Simply, based on the total covering of each attributes within the information system. Although this measurement is just an approximation, it can be proven to work.

To implement the experiment, five data sets were taken. They are gathered from the UCI machine learning repositories. Results are measured by comparing the performance of SIP/DRIP with and without problem restructuring. In addition, the length of reducts generated by both approaches are also compared to ensure that no quality deterioration occurred along the way.

Experimental results have shown that problem restructuring generally improves SIP/DRIP for all the data sets. This means that on average, it would enhance the performance of SIP/DRIP. The consumption of time and space were minimized quite significantly. Furthermore, the quality of the solutions was also successfully maintained. There was no increase in reduct length when using it.

The concept offered by the approach is an additional component to SIP/DRIP. It complements the process of searching done. By giving more consideration to the initial
problem space and not just the searching of the solution, the performance of SIP/DRIP has been humbly improved.
Abstrak tesis dikemukakan kepada Senat Universiti Putra Malaysia sebagai memenuhi keperluan untuk Master Sains.

PENSTRUKTURAN MASALAH DALAM PENGATURCARAAN INTEGER UNTUK PENCARIAN REDUKSI

Oleh
UNGKU AZMI ISKANDAR UNGKU CHULAN

Mei 2003

Pengerusi : Prof. Madya Dr. Haji Md. Nasir Sulaiman
Faculti : Sains Komputer dan Teknologi Maklumat

'Standard Integer Programming / Decision Related Integer Programming' (SIP/DRIP) adalah sejenis sistem pencarian reduksi yang mencari reduksi minimum dalam sistem maklumat. Reduksi –reduksi minimum ini merupakan atribut minimum yang berguna dalam aktiviti pengelasan. Mereka boleh menerangkan keseluruhan sistem maklumat semasa proses pembezaan. Oleh itu, mereka amat berguna untuk menghasilkan undang-undang pengkelasai semasa menyelesaikan masalah pengkelasai, yang sememangnya biasa dalam perlombongan data.

Tesis ini menekankan kepada peningkatan prestasi algoritma SIP/DRIP. Dengan menggunakan penstrukturan masalah, masa carian dan penggunaan memori dapat diminimkan. Pendekatan ini tertakluk kepada kriteria utama SIP/DRIP, iaitu peningkatan prestasi tanpa menjejaskan kualiti reduksi minimum.
Penstrukturan masalah menstruktur semula input SIP/DRIP tanpa menghilangkan mana-mana sifat input yang penting. Dalam erti kata lain, tiada kehilangan maklumat yang berlaku dan hanya struktur urutan input yang berubah. Secara hipotesis, ini menjamin bahawa tiada kesan buruk akan berlaku ketika penggunaan SIP/DRIP bersama penstrukturan masalah.

Penstrukturan dilakukan dengan menjana spekulasi mengenai struktur yang baik berpandukan kepada potensi atribut dalam sistem maklumat. Ia menggunakan kaedah yang tidak terlalu mahal dalam meramalkan potensi atribut. Ini dilakukan dengan mengira jumlah tutupan yang dirangkumi oleh atribut di dalam sistem maklumat. Walaupun pengiraan ini hanyalah suatu anggaran, kegunaannya boleh dibuktikan.


Hasil eksperimen menunjukkan bahawa penstrukturan masalah secara kasaranya meningkatkan prestasi SIP/DRIP bagi kesemua set data. Ini bermaksud bahawa secara purata, ia meningkatkan prestasi SIP/DRIP. Penggunaan masa dan ruang memori bagi proses carian telah dikurangkan dengan signifikannya. Tambahan pula, kualiti reduksi telah berjaya dikekalkan. Tiada penambahan panjang reduksi berlaku semasa menggunakan penstrukturan masalah.
Konsep yang diutarkan oleh pendekatan ini adalah sebagai komponen tambahan kepada SIP/DRIP. Ia membantu proses carian dengan mempertimbangkan ruang asal masalah dan bukan hanya kepada proses carian. Dengan rendah diri, prestasi SIP/DRIP telah berjaya ditingkatkan.
ACKNOWLEDGEMENTS

Alhamdulillah, all praises to Allah s.w.t, the only true God that rules the earth, and everything beyond it. Nothing can be accomplished without His consent. He is the all merciful, all giving. Never once has He left me in moments of despair. He is the all powerful, all knowing, the creator which bestowed me with the wisdom to take a pace in this odyssey of knowledge. I am forever indebted to His unconditional care.

I would like to express my greatest appreciation to the supervisor committee, lead by Assoc. Prof. Dr. Md. Nasir Sulaiman, Assoc. Prof. Dr. Ramlan Mahmood and Assoc. Prof. Mohd Hasan Selamat for their inspiring guidance and support in my quest for intellectual revelation. May Allah swt return their kindness and altruism. In addition, I would like to extend my gratitude to Dr. Azuraliza Abu Bakar and Pn. Norwati for their ongoing support and motivation that have helped me to overcome obstacles along the way. Finally, I thank my family and friends for their support, and everyone else that has helped and contributed to this research.

Ungku Azmi Iskandar Ungku Chulan

May 2003
I certify that an Examination Committee met on 9th of May 2003 to conduct the final examination of Ungku Azmi Iskandar Ungku Chulan on his Master of Science thesis entitled "Problem Restructuring in Integer Programming for Reduct Searching" in accordance with Universiti Pertanian Malaysia (Higher Degree) Act 1980 and Universiti Pertanian Malaysia (Higher Degree) Regulations 1981. The Committee recommends that the candidate be awarded the relevant degree. Members of the Examination Committee are as follows:

ALI MAMAT, Ph. D.
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Chairman)

MD. NASIR SULAIMAN, Ph.D.
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

RAMLAN MAHMOD, Ph.D.
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

HASAN SELAMAT
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

GULAM RUSUL RAHMAT ALI, Ph.D.
Professor/Deputy Dean
School of Graduate Studies,
Universiti Putra Malaysia
Date: 27 MAY 2003
This thesis is submitted to the Senate of University Putra Malaysia has been accepted as fulfillment of the requirements for the degree of Master of Science. The members of the Supervisory Committee are as follows:

MD. NASIR SULAIMAN, Ph.D.
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

RAMLAN MAHMOD, Ph.D.
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

HASAN SELAMAT
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

AINI IDERIS, Ph.D.
Professor/ Dean
School of Graduate Studies
Universiti Putra Malaysia

Date: 1 JUL 2003
DECLARATION

I hereby declare that the thesis is based on my original work except for quotations and citations, which are given the proper acknowledgement. I also declare that it has not been previously submitted to any other degree at UPM or other institutions.

(Ungku Azmi Iskandar Ungku Chulan)
Date: 27/5/03
# TABLE OF CONTENTS

DEDICATION                                                                                   ii  
ABSTRACT                                                                                     iii 
ABSTRAK                                                                                       vi 
ACKNOWLEDGEMENTS                                                                              ix  
APPROVAL SHEET                                                                                x   
DECLARATION FORM                                                                              xiii 
LIST OF TABLES                                                                                 xv  
LIST OF FIGURES                                                                                xvii 
LIST OF ABBREVIATIONS                                                                          xx 

1  INTRODUCTION 
1.1 Introduction                                                                               1  
1.2 Problem Statement                                                                          3  
1.3 Objective                                                                                  4  
1.4 Scope                                                                                    4  
1.5 Contribution                                                                              5  
1.6 Organization of Thesis                                                                     5  

2  LITERATURE REVIEW 
2.1 Introduction                                                                               8  
2.2 Data Mining                                                                                10  
  2.2.1 Data Mining Task                                                                       11  
  2.2.2 Modeling Classification                                                                14  
  2.2.3 Data Mining Challenges                                                                16  
2.3 Reduct Searching System                                                                   20  
  2.3.1 Genetic Algorithm                                                                      20  
  2.3.2 Holte 1R                                                                               23  
  2.3.3 Dynamic Reduct                                                                       25  
  2.3.4 SIP/DRIP                                                                              27  
2.4 Summary                                                                                  49  

3  REDUCT SEARCHING ADVANCEMENT 
3.1 Introduction                                                                               50  
3.2 Rough Sets and Classification                                                              51  
  3.2.1 Set Covering Problem                                                                  53  
  3.2.2 Propositional Satisfiability Problem                                                   55  
3.3 Problem Representation                                                                     56  
3.4 Reduct Searching System                                                                   58  
  3.4.1 SIP/DRIP                                                                              58  
  3.4.2 Branch and Bound                                                                       64  
  3.4.3 Input Processing                                                                     65  
                                                                                             70
3.5 Searching Process
3.5.1 Proposing Solution
3.5.2 Computing Solution
3.5.3 Conflict Analysis
3.5.4 Non-chronological Backtracking
3.5.5 Updating the Problem Space
3.5.6 Algorithm
3.6 SIP/DRIP Complexity
3.7 Benefits and Disadvantages
3.8 Problem Restructuring
3.8.1 Motivation
3.8.2 Understanding Problem Restructuring
3.8.3 Analyzing Problem Structure
3.8.4 Restructuring the Problem Structure
3.8.5 Space and Time Complexity
3.8.6 Semantics of Problem Structure
3.8.7 Illustration of Errors and Probable Solutions
3.8.8 Rationale of Non Deterministic Solution
3.8.9 Benefits and Risks of Problem Restructuring
3.8.10 Related Approach
3.9 Summary

4 EXPERIMENTATION AND RESULTS
4.1 Introduction
4.2 Implementation of Experiment
4.3 Experimentation Cases
4.3.1 Australian Credit Card (AUS)
4.3.2 Cleveland Heart Disease (CLV)
4.3.3 Lymphography (LYM)
4.3.4 American Mushroom (AM)
4.3.5 Voting Records (VR)
4.4 Discussion
4.5 Summary

5 CONCLUSION AND FUTURE WORK
5.1 Conclusion
5.2 Future Work

BIBLIOGRAPHY

APPENDICES
### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.0</td>
<td>Space Complexity</td>
</tr>
<tr>
<td>3.1</td>
<td>Static Variable Ordering and Problem Restructuring</td>
</tr>
<tr>
<td>3.2</td>
<td>CSP, SCP and SAT</td>
</tr>
<tr>
<td>4.1</td>
<td>Node (AUS)</td>
</tr>
<tr>
<td>4.2</td>
<td>Time (AUS)</td>
</tr>
<tr>
<td>4.3a</td>
<td>Node Severity and Improvement – Frequency and Average (AUS)</td>
</tr>
<tr>
<td>4.3b</td>
<td>Node Severity and Improvement - Maximum and Minimum (AUS)</td>
</tr>
<tr>
<td>4.4a</td>
<td>Time Severity and Improvement – Frequency and Average (AUS)</td>
</tr>
<tr>
<td>4.4b</td>
<td>Time Severity and Improvement - Maximum and Minimum (AUS)</td>
</tr>
<tr>
<td>4.5</td>
<td>Quality of Solution (AUS)</td>
</tr>
<tr>
<td>4.6</td>
<td>Node (CLV)</td>
</tr>
<tr>
<td>4.7</td>
<td>Time (CLV)</td>
</tr>
<tr>
<td>4.8a</td>
<td>Node Severity and Improvement - Frequency and Average (CLV)</td>
</tr>
<tr>
<td>4.8b</td>
<td>Node Severity and Improvement - Maximum and Minimum (CLV)</td>
</tr>
<tr>
<td>4.9a</td>
<td>Time Severity and Improvement - Frequency and Average (CLV)</td>
</tr>
<tr>
<td>4.9b</td>
<td>Time Severity and Improvement - Maximum and Minimum (CLV)</td>
</tr>
<tr>
<td>4.10</td>
<td>Quality of Solution (CLV)</td>
</tr>
<tr>
<td>4.11</td>
<td>Node (LYM)</td>
</tr>
<tr>
<td>4.12</td>
<td>Time (LYM)</td>
</tr>
<tr>
<td>4.13a</td>
<td>Node Severity and Improvement - Frequency and Average (LYM)</td>
</tr>
<tr>
<td>4.13b</td>
<td>Node Severity and Improvement - Maximum and Minimum (LYM)</td>
</tr>
</tbody>
</table>
4.14a  Time Severity and Improvement - Frequency and Average (LYM)  147
4.14b  Time Severity and Improvement - Maximum and Minimum (LYM)  147
4.15   Quality of Solution (LYM)  148
4.16   Node (AM)  149
4.17   Time (AM)  150
4.18a  Node Severity and Improvement - Frequency and Average (AM)  150
4.18b  Node Severity and Improvement - Maximum and Minimum (AM)  150
4.19a  Time Severity and Improvement - Frequency and Average (AM)  151
4.19b  Time Severity and Improvement - Maximum and Minimum (AM)  151
4.20   Quality of Solution (AM)  152
4.21   Node (VR)  153
4.22   Time (VR)  154
4.23a  Node Severity and Improvement - Frequency and Average (VR)  154
4.23b  Node Severity and Improvement - Maximum and Minimum (VR)  155
4.24a  Time Severity and Improvement - Frequency and Average (VR)  155
4.24b  Time Severity and Improvement - Maximum and Minimum (VR)  156
4.25   Quality of Solution (VR)  156
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0</td>
<td>Hybrid Algorithm</td>
<td>22</td>
</tr>
<tr>
<td>2.1</td>
<td>Holte 1R Algorithm</td>
<td>24</td>
</tr>
<tr>
<td>2.2</td>
<td>Dynamic Reduct Algorithm</td>
<td>26</td>
</tr>
<tr>
<td>2.3</td>
<td>Set Covering Problem</td>
<td>28</td>
</tr>
<tr>
<td>2.4</td>
<td>Matrix Reduction : Row and Column Elimination</td>
<td>29</td>
</tr>
<tr>
<td>2.5</td>
<td>Row Reduction Illustration</td>
<td>30</td>
</tr>
<tr>
<td>2.6</td>
<td>Row Reduction Illustration : Unit Row</td>
<td>31</td>
</tr>
<tr>
<td>2.7</td>
<td>Column Reduction</td>
<td>31</td>
</tr>
<tr>
<td>2.8</td>
<td>Random Greedy Algorithm</td>
<td>33</td>
</tr>
<tr>
<td>2.9</td>
<td>ITEG Algorithm</td>
<td>36</td>
</tr>
<tr>
<td>2.10</td>
<td>GIPP Algorithm</td>
<td>38</td>
</tr>
<tr>
<td>2.11</td>
<td>Davis Putnam Proof Procedure</td>
<td>39</td>
</tr>
<tr>
<td>2.12</td>
<td>Reduction and Partitioning Illustration</td>
<td>43</td>
</tr>
<tr>
<td>2.13</td>
<td>Hypergraph Illustration : Representing CNF</td>
<td>44</td>
</tr>
<tr>
<td>2.14</td>
<td>Hypergraph Illustration : Partitioning and Reordering</td>
<td>45</td>
</tr>
<tr>
<td>2.15</td>
<td>Local Search Algorithm</td>
<td>48</td>
</tr>
<tr>
<td>3.0</td>
<td>Problem Solving Process</td>
<td>51</td>
</tr>
<tr>
<td>3.1</td>
<td>Reduct Searching</td>
<td>51</td>
</tr>
<tr>
<td>3.2</td>
<td>Set Covering Problem Example</td>
<td>53</td>
</tr>
<tr>
<td>3.3</td>
<td>Set Covering Problem</td>
<td>55</td>
</tr>
<tr>
<td>3.4</td>
<td>Problem Representation</td>
<td>57</td>
</tr>
</tbody>
</table>
3.5 Integer Programming ModM Algorithm
62
3.6 Input and Output of Branch and Bound
65
3.7 Minimum Literal Length Theory Illustration
66
3.8 Clause Subsumption Example
68
3.9 Input Processing Algorithm
69
3.10 Clause Database
69
3.11 Simplified Searching Process in SIP/DRIP
70
3.12 Proposing Solutions
71
3.13 Lower Bound : Case I
72
3.14 Lower Bound Scenario
73
3.15 Lower Bound : Case II
73
3.16 Computing Lower Bound
73
3.17 Lower Bound Conflict
75
3.18 Logical Conflict
76
3.19 Upper Bound Conflict
76
3.20 Non-chronological Backtracking Algorithm
77
3.21 Non-chronological Backtracking Illustration
78
3.22 Updating the Problem Space Due to Conflict
80
3.23 Sample Problem Space
80
3.24 Updating Problem Space Illustration
81
3.25 SIP/DRIP Algorithm
82
3.26 Problem Restructuring and SIP/DRIP
85
3.27 Real Life Scenario
87
3.28 Best Case Scenario
87
3.29 Worst Case Scenario
88
3.30 Problem Restructuring
3.31 Simple Incremental Algorithm
3.32 Logic of Search
3.33 Original Column Ordering
3.34 New Column Ordering
3.35 Variable Arrangement
3.36 Column Arrangement
3.37 Dynamic Covering
3.38 Static and Dynamic Covering Illustration
3.39 Problem Restructuring Algorithm
3.40 Reconstructing the Problem Structure
3.41 Many Columns with Similar Column Summation
3.42 All Columns with Similar Column Summation
3.43 High Column Summation with Intersection
3.44 Essential Column with Low Column Summation
3.45 Effect of High Column Summation with Intersection
3.46 Effect of Essential Column with Low Column Summation
3.47 Probable Solution
4.0 Experimentation Data Set
4.1 Erroneous Problem Restructuring
4.2 Alternative Essential Variable
4.3 Original Scenario before Problem Restructuring
4.4 New Scenario after Problem Restructuring
**LIST OF ABBREVIATIONS**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIP/DRIP</td>
<td>Standard Integer Programming/Decision Related Integer Programming</td>
</tr>
<tr>
<td>AM</td>
<td>American Mushroom</td>
</tr>
<tr>
<td>AUS</td>
<td>Australian Credit Card</td>
</tr>
<tr>
<td>BB</td>
<td>Branch and Bound</td>
</tr>
<tr>
<td>CLV</td>
<td>Cleveland Heart Disease</td>
</tr>
<tr>
<td>CNF</td>
<td>Conjunctive Normal Form</td>
</tr>
<tr>
<td>LYM</td>
<td>Lymphography</td>
</tr>
<tr>
<td>PR</td>
<td>Problem Restructuring</td>
</tr>
<tr>
<td>SCP</td>
<td>Set Covering Problem</td>
</tr>
<tr>
<td>SAT</td>
<td>Satisfiability</td>
</tr>
<tr>
<td>VR</td>
<td>Voting Record</td>
</tr>
</tbody>
</table>
1.1 Introduction

In this era of technology, the production of data is growing tremendously even as we speak. To note, data is made and processed everywhere. From computers to credit cards, the mass production of digital data is likely inevitable.

In largely produced data, could there be important information hiding within it? If there is, how can we find it and use it? These are the questions that experts are trying to answer (Cabena et al. 1998). Having information means having power. So, there is no need to explain the motivation that lies behind this quest for information.

Data is usually stored and organized in a database. Figuratively, the attempt of discovering information within data can be related to mining gold. Hence, the phrase data mining refers to the effort of finding information in a vast amount of data.

Information is usually represented in the form of relationship between elements of data. That is, it answers the question of how things are related with one another. Knowing the underlying relationship within elements of data is very useful as we can then predict its effects. Consequently, the information can be put to use in real life situation, like in e-commerce applications (Buchner et al. 1998).
The importance of data mining has gained popularity in the business world (Srivastava et al. 2000). Collected data carries many hidden information on opportunities. By using data mining, better speculations on profits are made possible. Not just that, information gathered from data mining can also help in identifying the factors that can lead to business failure. This way, it allows proper actions to be taken before it is too late.

Due to the increasing demand for data mining, many researchers have devoted their time to studying it. Among its many uses, one of the most popular one is classification. It is simply the act of knowing the class and decision of an object given all its attributes. This trait is natural for human beings. For example, when the sky is dark, it might rain, so we decide to carry an umbrella along.

In business process however, the required classification is not as simple. To speculate the successfulness of a loan for instance, there are numerous factors that interact with one another simultaneously. Furthermore there might be thousands of loans to be processed in a short period of time. So, classification must be done intelligently to make all this doable.

Many classification systems have been developed. Most of them are based on statistical analysis. The problem with this method is the vast amount of processing that must be done. It also requires the help of an expert to give appropriate merits to certain values of the factors. To solve this problem, a more novel approach is proposed by using the rough sets theory.
The theory (Pawlak, 1982) proposes that within any knowledge, there exists a minimum amount of information needed to enable classification. If the minimum knowledge were found, processing of data into information would be more efficient. Plus, this approach does not require any additional information like the statistical approach. As a result, it is hypothetically more preferable.

From the eyes of the rough sets theory, this minimum knowledge is known as the reduct. Searching for it is called the reduct searching problem. The advent of rough sets theory is beginning to materialize the hope for a better way of doing classification. As such, it is the focus of this research.

1.2 Problem Statement

The classification problem is a well-known problem in data mining. The problem states that given a set of attributes for an object, the class of the object is to be found. There are numerous ways to solve the problem. The method of interest is the application of the rough sets theory in developing a solution. One of the requirements of applying the rough sets theory is the finding of reducts. The SIP/DRIP algorithm (Bakar, 2002; Bakar et al. 2000, 2001(a), 2001(b)) fulfills this requirement. It finds the reducts in an information system. Employing the branch and bound searching technology, the SIP/DRIP algorithm consumes exponential time and considerable space. It naturally assumes that the value of each attribute is the same. As such, no attention is given to its ordering. This assumption might be the cause of the inherent weakness in the algorithm.