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 pengkelasan semasa menyelesaikan masalah pengkelasan, 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 diminimakan. 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 ertikata 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.

Untuk menjalankan eksperiman, lima set data telah diambil. Mereka dikumpul daripada *UCI machine learning repositories*. Keputusan diukur dengan membandingkan output SIP/DRIP bersama penstrukturan masalah dan tanpanya. Sebagai tambahan, panjang reduksi yang dijanakan oleh kedua-dua pendekatan turut dibandingkan untuk mempastikan bahawa tiada penjejasan kualiti berlaku.

Hasil eksperimen menunjukkan bahawa penstrukturan masalah secara kasarnya 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 diutarakan 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

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SIP/DRIP	Standard Integer Programming/Decision Related Integer Programming
AM	American Mushroom
AUS	Australian Credit Card
BB	Branch and Bound
CLV	Cleveland Heart Desease
CNF	Conjunctive Normal Form
LYM	Lymphography
PR	Problem Restructuring
SCP	Set Covering Problem
SAT	Satisfiability
VR	Voting Record

CHAPTER 1

INTRODUCTION

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.