**KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY, KUMASI**



# OPTIMAL PORTFOLIO SELECTION (A CASE STUDY OF GHANA STOCK EXCHANGE)

By

HERMAN YIRBECHAA TAGYANG

(B. A. ECONOMICS AND MATHEMATICS)

A THESIS SUBMITTED TO THE DEPARTMENT OF MATHEMATICS, KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY IN PARTIAL FUFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MSc. INDUSTRIAL MATHEMATICS

MAY, 2014

# DECLARATION

I hereby declare that this submission is my own work towards the award of the MSc. degree and that, to the best of my knowledge, it contains no material previously published by another person nor material which had been accepted for the award of any other degree of the university, except where due acknowledgement had been made in the text.



| | | |
|---|---|---|
| Tagyang Yirbechaa Herman | ..................... | .................. |
| Student | Signature | Date |

Certified by:

| | | |
|---|---|---|
| Mr. Kwaku Darkwa | ..................... | .................. |
| Supervisor | Signature | Date |

Certified by:

| | | |
|---|---|---|
| Prof. S. K. Amponsah | ..................... | .................. |
| Head of Department | Signature | Date |

# DEDICATION

I wish to dedicate this work to the Glory of God Almighty who has been our help in ages past, our fortress now and is still our hope for many years to come. Also to my lovely wife, Irene Kumah and my parents, Mr and Mrs Tagyang.

# Abstract

The knapsack model is employed in many fields of study including Business, Engineering and Economics to solve problems related to resource constraints. The knapsack problem is a form of integer programming problem that has only one constraint and can be used to strengthen cutting planes for general integer programs. These facts make the studies of the knapsack problems and their variants extremely important area of research in the field of operations research. This thesis seeks to apply the branch-and-bound algorithm to construct an optimal portfolio from the listed shares on the Ghana Stock Exchange (GSE) in the model of the 0-1 knapsack problem. This paper will among other things seek to contribute to making the financial market efficient with particular reference to accessing information of listed companies. The paper will also consider the factors to note when forming a portfolio and its capitalization. The model developed could be adopted for decision making in choosing shares for the optimal portfolio. In the end, we form a portfolio from the listed shares on the Ghana Stock Exchange using the concept of the 0-1 knapsack model to see if we will obtain a good return on our investment. All listed shares of the GSE were considered and it proved out that AADS, ACI, AYRTN, CAL, CMLT, CPC, ETI, GCB and GOIL shares should be selected to obtain an optimum output and recommend that Knapsack problem model should be adopted by fund managers and other financial market players.

# ACKNOWLEDGMENT

I would like thanks the Almighty God for granting me the strength, travelling mercies and knowledge to complete this course successfully. I am very grateful to my supervisor, Mr. F.K. Darkwa of the Department of Mathematics, KNUST, who was always ready to guide me through my work, may God richly bless him. I also wish to express my profound gratitude to all the lecturers at the mathematics Department who contributed in one way or the other for the successful completion of this project. I also give thanks to my wife Irene Kumah for her support and the sleepless nights you endured waiting for me to get back home during the entire course. Finally my sincere thanks go to all who in diverse ways helped in bringing this project to a successful end. God richly bless you all.

# Contents

# List of Tables

# Chapter 1

# INTRODUCTION

## 1.1 INTRODUCTION TO PORTFOLIO

### 1.1.1 PORTFOLIO CONSTRUCTION

Portfolio construction is an investment strategy that aims to balance risks and rewards by apportioning a portfolio assets' according to an individual's goals, risks, tolerance and investment horizon. The three main asset classes- equities, fixed income, and cash and equivalents-have different levels of risks and returns, so each will behave differently over time. There is no simple formula that can find the right asset allocation for every individual. However, the consensus among most financial professionals is that asset allocation is one of the most important decisions that investors make. In other words, your selection of individual securities is secondary to the way you allocate your investment in stocks, bonds, and cash and equivalents, which will be the principal determinants of your investment results.

### 1.1.2 INVESTING

Investors should be willing to assume some risk in order to achieve a higher return on their money than a saver gets. The key to success for investors is in allowing their money to grow over a long period of time. Generally speaking, the longer an investment is held, the greater the chance of a higher returns. People in their 30s and 40s should be willing to take on a fair amount of risk with the money they are saving for retirement. An investor may have a long, medium or short term horizon and all these come with specific portfolio construction strategies.

### 1.1.3 INVESTORS WITH SHORT TERM HORIZON

The short term investor invest exclusively in short term government and corporate attractive yields whilst maintaining significant liquidity and preserving capital. The short-term income portfolio comprises short-term treasury bills, money market instruments and high quality corporate commercial paper and secured notes. A high level of liquidity is maintained through fixed cash deposits. It guarantees a secure, short-term haven for their liquid assets, higher yields than current and savings accounts and instant liquidity.

### 1.1.4 THE MEDIUM TERM INVESTOR

The medium term investor seeks a portfolio that strikes a balance between growth and income. This style seeks to provide medium-term, conservative capital growth whilst generating current income, by investing primarily in companies and sovereign papers. This strategy offers a highly diversified portfolio aimed at securing high medium term returns while preserving invested capital. The investor leverages in-depth research to build a diversified portfolio of core holdings of opportunistic stocks with additional investment in income-oriented securities. The investor reduces investment risk by diversifying across asset classes. The purpose is to construct a conservative growth portfolio, with an added balance of income-oriented securities, a balanced mix of growth and income and a core holding that looks for opportunity in both up and down markets while keeping an eye on risk.

### 1.1.5 THE LONG TERM INVESTOR

The long-term investor's goal is to seek long-term growth of capital via a diversified portfolio that invests primarily in the listed shares of companies that are selected for their long-term growth potential. This style provides instant diversification by adopting a bottom-up stock selection style, backed by fundamental research. The investor may occasionally accumulate new positions, phase out and replace existing positions, or respond to exceptional market conditions. It suits investors with a higher-than-average risk and volatility tolerance seeking long-term capital appreciation, a well-diversified portfolio, exposure to both stocks and bonds in a professionally built and dynamically rebalanced portfolio. The portfolio manager in this regard utilizes a fairly aggressive growth stance and, with its high exposure to equities, is only suitable for investors who are able to tolerate severe year-to-year fluctuations in value, in exchange for higher long-term return potential.



### 1.1.6 SECURITIES IN GHANA

The investment landscape Ghana has been changing rapidly. Well before the establishment of the Ghana Stock Exchange, securities were being traded in Ghana. On the short-term market, Treasury Bills and Bank of Ghana Bills were being issued by the Bank of Ghana. While Treasury Bills are issued to finance the government deficit, Bank of Ghana bills were being issued for open market operations. The Bank of Ghana would, therefore sell Bank of Ghana Bills as needed to mop up excess liquidity and buy Bank of Ghana Bills to inject liquidity in the system. Equity securities were being traded over the counter. In fact, the equity issues of companies such as the Mobil and Shell were being traded by National Trust Holding Company (NTHC) well before the Ghana Stock Exchange was established.

### 1.1.7 THE GHANA STOCK EXCHANGE

The establishment of the Ghana Stock Exchange in 1990 represented a revolutionary change in securities markets in Ghana from 9 initial listings in 1990; the GSE currently has 35 fully listed companies and one provisional listing. Today a portfolio manager in Ghana can select from the following universe of financial instruments:

- Money Market Instruments
- Listed shares
- Unlisted Shares
- Bonds
- Unit Trusts and Mutual Funds

### 1.1.8 MONEY MARKET INSTRUMENTS

Money market instruments are instruments that at issues have a maturity of one-year or less. The most actively traded money market instruments are Treasury Bills which are government issues with maturities of 91-and 182days. The bank of Ghana holds auctions of Treasury Bills every week on Thursdays. A group of dealers consisting of banks, securities firms and discount houses collectively known as "primary dealers" participate in th weekly auction. All buyers must place their orders through one of the primary dealers. Between auctions, Treasury Bills are bought and sold among the primary dealers. Primary dealers also provide facilities for holders to sell their holdings before maturity. Other money market instruments exist on the market. For example, large credit worthy companies issue promissory notes (also called Commercial Paper) that are sold to the discount houses. Large institutional investors often participate the commercial paper market.

### 1.1.9 LISTED SHARES

The Ghana Stock Exchange currently lists the shares of 35 companies shown in the table below.

.

Table 1.1: listed Shares

| SHARE CODE | OFFICIAL NAME | NAME | WEBSITE |
|---|---|---|---|
| ABL | Accra Brewery Company Ltd | Accra Brewery Company | |
| ACI | African Champion Industries Limited | African Champion Industries | |
| ALW | ALUWORKS LTD | Aluworks | www.aluworks.com |
| AADS | AngloGold Ashanti Limited | AngloGold Ashanti | www.anglogold ashanti.com |
| AGA | AngloGold Ashanti Limited | AngloGold Ashanti GoldFields | www.anglogold ashanti.com |
| AYRTN | Ayrton Drugs Manufacturing Company LTD | Ayrton Drugs Manufacturing Company | www.ayrtondrugs.com |
| BOPP | Benso Oil Palm Plantation Limited | Benso Oil Palm Plantation | |
| CAL | CAL Bank Limited | CAL Bank | www.calbank.net |
| CMLT | Camelot Ghana Ltd. | Camelot Ghana | www.camelotprint.com |
| CFAO | CFAO (Ghana) Ltd. | CFAO | |
| CLYD | Clydestone (Ghana) Limited | Clydestone | www.clydestone.com |

| SHARE CODE | OFFICIAL NAME | NAME | WEBSITE |
|---|---|---|---|
| CPC | Cocoa Processing Company | Cocoa Processing Company | www.golden treeghana.com |
| EBG | Ecobank Ghana Limited | Ecobank Ghana | www.ecobank.com |
| ETI | Ecobank Transnational Incorporation | Ecobank Transnational Incorporation | www.ecobank.com |
| ETI | Ecobank Transnational Incorporation | Ecobank Transnational Incorporation | www.ecobank.com |
| EGL | Enterprise Group Limited | Enterprise Group | www.eicghana.com |
| EIC | Enterprise Insurance Company Ltd. | Enterprise Insurance Company | www.eicghana.com |
| FML | Fan Milk Ltd | Fan Milk | www.fanmilk-gh.net |
| GCB | Ghana Commercial Bank Ltd. | Ghana Commercial Bank | www.gcb.com.gh |
| GOIL | Ghana Oil Company Limited | Ghana Oil Company | www.goilonline.com |
| GSR | Golden Star Resources Ltd | Golden Star Resources | www.gsr.com |
| GWEB | Golden Web Ltd. | Golden Web | |
| GGBL | Guiness Ghana Breweries Ltd. | Guiness Ghana Breweries | |
| HFC | HFC Bank Ltd | HFC Bank | www.hfcbankgh.com |
| MLC | Mechanical Lloyd Company Ltd | Mechanical Lloyd Company | www.mechlloyd.com |

| SHARE CODE | OFFICIAL NAME | NAME | WEBSITE |
|---|---|---|---|
| PKL | Pioneer Kitchenware Ltd. | Pioneer Kitchenware | www.pioneer kitchenwareltd.com |
| PBC | Produce Buying Company Ltd. | Produce Buying Company | |
| PZC | PZ Cussons (Ghana) Ltd | PZ Cussons | www.pzcussons.com |
| SWL | Sam Wood Ltd. | Sam Wood | www.samwoode.com |
| SG-SSB | SG-SSB Ltd. | SG-SSB | www.sg-ssb.com.gh |
| SIC | SIC Insurance Company Limited | SIC Insurance Company | www.sic-gh.com |
| SCB PREF | Standard Chartered Bank (Ghana) Ltd. | Standard Chartered Bank | www.standard chartered.com/gh |
| SPL | Starwin Products Limited | Starwin Products | |
| TOTAL | Total Petroleum Ghana Ltd | Total Petroleum Ghana | |
| TRANSOL | Transol Solutions (Ghana) Limited | Transol Solutions | www.transolghana.com |
| TBL | Trust Bank Ltd | Trust Bank | trustbank.gm |
| UNIL | Unilever (Ghana) Ltd. | Unilever | www.unileverghana.com |

Because the shares on the Ghana Stock exchange are traded regularly, listed equities provide more liquidity than unlisted equities.

### Unlisted equities

A number of companies have chosen to trade their shares traded informally at brokerage houses. NTHC is active in the trading of such unlisted shares.

Astute portfolio managers will also recognise that there is also growing market for privately placed equities. Many companies in Ghana looking for equity capital choose to place their equities privately. A number of international financial institutions such as the International Finance Company (IFC) and CDC actively participate in a private placement market, taking equity positions in private Ghana companies. Also active in the private placement market is the Social Security and National Insurance Trust (SSNIT) and venture capital companies such as Ghana venture Company Limited. Naturally, it is more difficult to dispose of unlisted shares than listed companies.



### 1.1.10 BONDS

Bonds are interest-bearing instruments with maturies that at issue exceed one year. Unlike shares which pay dividends as and when declared by the Board of Directors, issuers of bond at maturity. In Ghana, bonds are issued by Ghana Government and by companies. The instrument that is currently being issued by the Government of Ghana are:

- 91-days Treasury Bills
- 182- days Treasury Bills
- 1 year Note
- 2- year fixed Rate Bond
- 3- year Fixed Rate Bond
- 5- year Fixed rate Bond
- 5- year Golden Jubilee Bond

Companies may also issue bonds. However, the corporate market in Ghana is relatively underdeveloped. Only two companies. HFC Bank and Standard Chartered Bank, have issued bonds publicly. As at end of Febuary 2010, only one bond issued by HFC Bank remains outstanding and its denominated in foreign currency (US Dollas)

### 1.1.11 UNIT TRUSTS AND MUTUAL FUNDS

Unit trusts and mutual funds are called "collective Investment schemes". Although the legal constitution of unit trusts and mutual funds differ, their characteristics as investments vehicles are identical. A unit trust consists of a collection of securities, which are held by a trustee on behalf of beneficial owners who holds units of the trust. There are two main unit trust in Ghana all operated by HFC. The HFC unit trust is designed to hold a wide variety of securities including money market instruments, bonds and shares. The HFC Real Estate Investment Trust. (REIT) hold real estate investments. Mutual funds are corporate entities with shareholders except that the corporate entity only invests in securities of other companies. Shareholders of the mutual fund hold shares, which are equivalent to units in a unit trust. Both unit trusts and mutual funds are recognised in the securities industry law (P.N.D.C.L 333), 1993. Managed by Databank Asset Management Services Ltd. Databank's Epack used to be the only mutual fund that was fully operational in the early years of the GSE. Today, a number of mutual funds and unit trust are available for investors to choose from, and the numbers keep increasing. This is a good sign for the investment climates in Ghana.

## 1.2 BACKGROUND STUDY

The Ghana Stock Exchange as a public company limited by guarantee has no owners or shareholders as such, but members are either corporate bodies or individuals. There are three categories of members, namely Licensed Dealing Members, Associate Members and Government Securities Dealers (PDs). An LDM is a corporate body licensed by the Exchange to deal in all securities. An

Associate member is an individual or corporate body which has satisfied the Exchange's membership requirements but is not licensed to deal in securities. A PD is a corporate body, which is approved by the Bank of Ghana and registered by the Exchange to deal only in government securities.

### 1.2.1 Regulatory Framework:

GSE operates within a set of Rules, including membership, listing, trading, clearing & settlement and depository. These are collectively referred to as the GSE Rule Book.

Membership rules deals with the criteria for membership of the GSE, code of conduct or ethics for members, among others whiles listing rules prescribe among others, criteria for listing securities (local and external), continued obligations of the listed companies as well as Take-over and merger procedures.

GSE Automated Trading (GATS) rules govern electronic trading done by the brokers whether on the Floor, from Dealers offices or through the secured internet daily and within a given time. Pre-opening period for trading is 9:30 to 10:00 hrs (GMT) whiles Market opens for continuous trading between 10.00 hrs to 15.00 hrs (GMT).

Settlement of trades is done electronically using a web based application. Settlement occurs three business days (T+3) after the trade date. The System allows for mutual settlement of trade on T+0 or T+1 basis. On settlement dates shares are moved automatically to client's accounts in the depository system and the brokers settlement account debited.

The GSE has set up a wholly -owned subsidiary called GSE Securities Depository Company Limited. The key objective is to offer depository services to complement the Exchange's automated trading, clearing and settlement systems. As a result investors on the Exchange must open securities account with the Depository through their Stockbrokers. Under the automated trading and settlement system, an investor cannot sell nor buy securities on the market if he or she has no securities account.

The Securities and Exchange Commission (SEC) carries out regular inspection of Licensed Dealing Members' operations and books. Brokers are also required to submit returns to GSE. GSE has two categories of listing. These are 1st and 2nd. The second list is essentially aimed at small and medium sized enterprises (SMEs)

### 1.2.2 Types of Securities that can be listed

- Shares (preference or equities);
- Debt in the form of corporate bonds (and notes), municipal bonds (and notes), & government bonds (and notes); and
- Close-end unit trusts and mutual funds

Starting from January 4, 2011, the GSE publishes two indices, namely the GSE Composite Index (GSE-CI) and the GSE Financial Stocks Index (GSE-FSI)

### 1.2.3 GSE Composite Index (GSE-CI)

The calculation of the GSE Composite Index (GSE-CI) is based on the volume weighted average closing price of all listed stocks. All ordinary shares listed on GSE are included in the GSE-CI at total market capitalization, with the exception of those of listed companies which have shares listed on other markets. The GSE-CI is a market capitalization weighted index, i.e. each constituent is given weight according to its market capitalization. The base date for the GSE-CI is December 31, 2010 and the base index value is 1000.

### 1.2.4 GSE Financial Stocks Index (GSE-FSI)

This index have its constituents as listed stocks from the financial sector including banking and insurance sector stocks. All ordinary shares of the financial stocks listed on GSE are included in the GSE-FSI at total market capitalization, except for those of stocks which are listed on other markets. The base date of GSE-FSI is also December 31, 2010 and the base index value is 1000.

### 1.2.5 Procedures For Non-Residents Investing Through The Stock Exchange

There are Sixteen licensed stock broking firms which have set up systems for serving non-residents. Custodial services for non-resident investors are provided by: Merchant Bank Ghana Limited, Ecobank Custody Services, Stanbic Bank Ghana Ltd. - (SBL), Standard Chartered Bank Gh. Ltd, Cal Bank Limited, Fidelity Bank Ghana Limited, Societe Generale Ghana Limited, HFC Bank Ghana Limited, Prudential Bank Limited, Zenith Bank Ghana Limited.

### 1.2.6 Investor Protection Provisions

The Exchange has various provisions in its rules which have been designed to protect the investor in addition to what the securities regulator (SEC) provides. Under the SECURITIES INDUSTRY LAW PNDCL 333 (1993), as amended, the apex regulatory body in the securities market is the Securities and Exchange Commission and is functions include: maintaining surveillance over the securities business to ensure orderly, fair and equitable dealing in securities:

- Registering, licensing, authorizing, a Stock Exchange, investment advisors, securities dealers, etc.
- Protecting the integrity of the securities markets against any abuses arising from the practice of insider trading.

### 1.2.7 PROBLEM STATEMENT

It is very worrying when an investment underperforms or does not yield its expected return. Portfolio underperformance can linked to a number of reasons.

Some are within the scope of the fund manager whiles others are not. When key macroeconomic variables such as inflation and interest rates go up, they tend to dampen the profitability of companies and adversely affect the returns of stocks. The capitalization of any portfolio has a direct correlation with its return. For reasons best known to the fund manager, he may choose to put more money in one stock than the others. This stock with the greater weight in the portfolio becomes the driving force. In the case where the lead stock falls flat at the end of the investment cycle, regardless of the performances of the other stocks, the portfolio will certainly underperform. Investing in stocks which are over-priced will certainly yield a low return. It is the duty of the fund manager to be able identify such shares with all the necessary information on the market.

## 1.3 AIMS AND OBJECTIVES

- To model the selection of shares as a 0-1 knapsack problem.
- To determine the optimal portfolio using the branch and bound method.

## 1.4 METHODOLOGY

This thesis seeks to apply the branch-and-bound algorithm for solving our proposed knapsack problem. The algorithm is presented along with relevant examples. A computational study is performed and a code in MATLAB programming language will be employed to implement the algorithm.

## 1.5 JUSTIFICATION

Investment decisions are very critical issues that confront us either as an individuals or as an institution. Once such decisions can be modelled as a Knapsack problem it becomes very interesting from the perspective of computer science because; there is a pseudo polynomial time algorithm using dynamic programming, there is a fully polynomial-time approximation scheme, which uses the pseudo-polynomial time algorithm as a subroutine, the problem is NP-complete to solve exactly, thus it is expected that no known algorithm can be both correct and fast (polynomial-time) on all cases, and many cases that arise in practice, and "random instances" from some distributions, can nonetheless be solved exactly. This makes the study of knapsack problems and their algorithms an area of much interest in the contribution to academic knowledge.

## 1.6 LIMITATIONS OF THE STUDY



The limitation of the study is the fact that historical data is used to forecast for expected profits and cost per share. Using the past as a guide in forecasting for future values is not the best since each day comes with its own issues.

## 1.7 ORGANIZATION OF THE THESIS

The study is organized in five chapters. In chapter one, we presented a background study, the problem statement, the objectives, methodology, justification and limitation of the study. In chapter two, we will review some literature in the field Knapsack problem. Chapter three deals with the branch-and-bound algorithm. Chapter four is data collection and analysis

Chapter five talks about conclusion and recommendations of the study.

# Chapter 2

# LITERATURE REVIEW

The portfolio selection problem has found a first mathematical formulation in the pioneering paper of Markowitz thanks to which the investments diversification has been translated into computational terms. In the last 40 years we have been witness to a great evolution with respect to the traditional Mean-Variance (MV) scheme, introduced by Markowitz. Some of the main drawbacks recognized to MV model are its high computational complexity and the input problem of estimating parameters (expected returns, variances and covariances), which made the model a milestone in finance theory, but a scarcely used tool in practice. This situation justified the several attempts in literature to linearize the quadratic objective function. Nowadays MV models consisting of more than a few thousand assets have been solved changing dramatically the practical role of MV approach for constructing large scale portfolios. Real time solutions are obtainable through the use of interior point algorithm for quadratic programming problem, or by using compact factorizations and piecewise linear approximations. The first linear model for portfolio selection is due to Konno and Yamazaki. The linear form of the model is made possible by the use of a risk function different from the classical portfolio variance, namely the portfolio absolute deviation. A relevant feature of the model is that no probabilistic assumptions are made on the securities rates of return, while in the case the rates of return are multivariate normally distributed the model is shown to be equivalent to Markowitz's one. The Konno and Yamazaki's model, the so-called Mean Absolute Deviation (MAD), has been applied by Zenios and Kang to a mortgage-backed securities portfolio optimization in which the rates of return distribution is asymmetric. Speranza introduced a more general model with a weighted risk function. The author showed how a suitable choice for the coefficients in the linear combination gives rise to a model equivalent to Konno and Yamazaki's but halving its number of constraints. A similar result has been independently obtained by Feinstein and Thapa. The largest part of the portfolio selection models which have been proposed in the literature are based on the assumption of a perfect fractionability of the investments in such a way that the portfolio fraction for each security could be represented by a real variable. In the real world, securities are negotiated as multiples of a minimum transaction lot (the so called rounds). As a consequence of considering rounds, solving a portfolio selection problem requires finding the solution of a mixed integer programming model. When applied to real problems, the tractability of the integer model is subject to the availability of algorithms able to find a good, even if not optimal, integer solution in a reasonable amount of time. A general mixed integer model including real characteristics of the problem has been presented in Speranza, where a simple heuristic is proposed and tested for the case when minimum transaction lots are considered. The problem with fixed transaction costs with and without minimum transaction lots has been recently studied in Mansini and Speranza. Moreover, new algorithms are proposed for the solution of the model with rounds. As the number of securities selected by a standard (quadratic or linear) portfolio optimization model is observed to be almost always smaller than 20, the heuristics proposed herein are based upon the idea of constructing and solving mixed integer sub-problems which consider subsets of the investment choices available. The subsets are generated by exploiting the information obtained from the relaxed problem (selected securities and reduced costs). The heuristics have the relevant advantage of being general. Different mixed integer models can be of interest in portfolio selection if, for instance, transaction costs are considered. The knapsack problem is one of the most intensively studied discrete programming problems. The reason for such interest basically derives from three facts: (a) it can be viewed as the Simplest Integer Linear Programming problem; (b) it appears as a sub-problem in the more complex problems; (c) it may represent a great many practical situations. Recently, it has been used for generating minimal cover induced constraints (see, e.g., Crowder, Johnson and Padberg, (1983) and in several coefficient reduction procedures for strengthening LP bounds in general integer programming (see, e.g., Dietrich and Escudero, (1989a, 1989b). During the last few decades, KP has been studied through different approaches, according to the theoretical development of Combinatorial Optimization. In the fifties, Bellman's dynamic programming theory produced the first algorithms to exactly solve the 0-1 knapsack problem. In 1957 Dantzig gave an elegant and efficient method to determine the solution to the continuous relaxation of the problem, and hence an upper bound on z which was used in the following twenty years in almost all studies on KP. In the sixties, the dynamic programming approach to the KP and other knapsack-type problems was deeply investigated by Gilmore and Gomory. In 1967 Kolesar experimented with the first branch-and-bound algorithm for the problem. In the seventies, the branch-and-bound approach was further developed, proving to be the only method capable of solving problems with a high number of variables. The most well-known algorithm of this period is due to Horowitz and Sahni. In 1973 Ingargiola and Korsh presented the first reduction procedure, a preprocessing algorithm which significantly reduces the number of variables. In 1974 Johnson gave the first polynomial-time approximation scheme for the subset-sum problem; the result was extended by Sahni to the 0-1 knapsack problem. The first fully polynomial-time approximation scheme was obtained by Ibarra and Kim in 1975. In 1977 Martello and Toth proposed the first upper bound dominating the value of the continuous relaxation. The main results of the eighties concern the solution of large-size problems, for which sorting of the variables (required by all the most effective algorithms) takes a very high percentage of the running time. In 1980 Balas and Zemel presented a new approach to solve the problem by sorting, in many cases, only a small subset of the variables (the core problems). Oppong (2009) presented the application of classical 0-1 knapsack problem with a single constraint to selection of television advertisements at critical periods such as Prime time News, news adjacencies, Break in News and peak times. The Television (TV) stations have to schedule programmes interspersed with adverts or commercials which are the main sources of income of broadcasting stations. The goal in scheduling commercials is to achieve wider audience satisfaction and making maximum income from the commercials or adverts. The author approach is flexible and can incorporate the use of the knapsack for Profit maximization in the TV adverts selection problem, and focused on using a simple heuristic scheme (Simple flip) for the solution of knapsack problems.

The collapsing knapsack problem is a generalization of the ordinary knapsack problem, where the knapsack capacity is a non-increasing function of the number of items included. Whereas previous methods on the topic have applied quite involved techniques,

Ulrich et al., (1995) presented and analyze two rather simple approaches: One approach that was based on the reduction to a standard knapsack problem, and another approach that was based on a simple dynamic programming recursion. Both algorithms have pseudo-polynomial solution times, guaranteeing reasonable solution times for moderate coefficient sizes. Computational experiments are provided to expose the efficiency of the two approaches compared to previous algorithms.

Kosuch and Lisser (2009) studied a particular version of the stochastic knapsack problem with normally distributed weights: the two-stage stochastic knapsack problem. Contrary to the single- stage knapsack problem, items can be added to or removed from the knapsack at the moment the actual weights become known (second stage). In addition, a chance-constraint is introduced in the first stage in order to restrict the percentage of cases where the items chosen lead to an overload in the second stage. According to the authors, there is no method known to exactly evaluate the objective function for a given first-stage solution, and therefore proposed methods to calculate the upper and lower bounds. These bounds are used in a branch-and-bound framework in order to search the first-stage solution space. Special interest was given to the case where the items have similar weight means. Numerical results are presented and analyzed.

Stefanie (2010) presented an Ant Colony Optimization algorithm for the Two-Stage Knapsack problem with discretely distributed weights and capacity, using a meta-heuristic approach. Two heuristic utility measures were proposed and compared. Moreover, the author introduced the novel idea of non-utility measures in order to obtain a criterion for the construction termination. The author argued why for the proposed measures, it is more efficient to place pheromone on arcs instead of vertices or edges of the complete search graph. Numerical tests show that the author's algorithm is able to produce, in much shorter computing time, solutions of similar quality than CPLEX after two hour. Moreover, with increasing number of scenarios the percentage of runs where his algorithm is able to produce better solutions than CPLEX (after 2h) increases.

Mattfeld and Kopfer (2003) described terminal operations for the vehicle transshipment hub in Bremerhaven as a knapsack and have derived an integral decision model for manpower planning and inventory control. The authors proposed a hierarchical separation of the integral model into sub models and can develop integer programming algorithm to solve the arising sub problems. In bus transit operations planning process, the important components are network route design, setting timetables, scheduling vehicles, assignment of drivers, and maintenance scheduling.

Haghani and Shafahi (2002) presented integer programming model to design daily inspection and maintenance schedules for the buses that are due for inspection so as to minimize the interruptions in the daily bus operating schedule, and maximize the utilization of the maintenance facilities. The setting of timetables and bus routing or scheduling are essential to an intercity bus carrier's profitability, its level of service, and its competitive capacity in the market. Yan and Chen (2002) developed a model that help Taiwanese intercity bus carriers in timetable settings and bus routing or scheduling. The model employs multiple time-space networks that can formulate bus movements and passenger flows and manage the interrelationships between passenger trip demands and bus trip suppliers to produce the best timetables and bus routes or schedules.

Higgins et al., (1996) described the development and use of integer programming model to optimize train schedules on single-line rail corridors. The model has been developed with two major applications in mind: as a decision support tool for train dispatchers to schedule trains in real time in an optimal way and as a planning tool to evaluate the impact of timetable changes, as well as railroad infrastructure changes. The model was developed based on a real-life problem.

Ghoseiri et al., (2004) developed an optimization model for the passenger train-scheduling problem on a railroad network, which includes single, and multiple tracks, as well as multiple platforms with different train capacities.

Claessens et al., (1998) considered the problem of cost optimal railway line allocation for passenger trains for the Dutch railway system. A mathematical programming model was developed, which minimized the operating costs subject to service constraints and capacity requirements. The model optimized on lines, line types, routes, frequencies, and train lengths. First, the line allocation model was formulated as an integer nonlinear programming model. The model was then transformed into an integer linear programming model with binary decision variables. The model was solved and applied to a sub network of the Dutch railway system for which it showed a substantial cost reduction.

The deterministic knapsack problem is a well-known and well-studied NP-hard combinatorial optimization problem. It consists in filling a knapsack with items out of a given set such that the weight capacity of the knapsack is respected and the total reward maximized. In the deterministic problem, all parameters (item weights, rewards, knapsack capacity) are known (deterministic). In the stochastic counterpart, some (or all) of these parameters are assumed to be random, i.e. not known at the moment the decision has to be made.

Stefanie et al., (2010) studied the stochastic knapsack problem with expectation constraint. The item weights are assumed to be independently normally distributed. The authors solved the relaxed version of this problem using a stochastic gradient algorithm in order to provide upper bounds for a branch-and-bound framework. Two approaches to estimate the needed gradients are applied, one based on Integration by Parts and one using Finite Differences. Finite Differences is a robust and simple approach with efficient results despite the fact that the estimated gradients are biased; meanwhile Integration by Parts is based upon a more theoretical analysis and permits to enlarge the field of applications.



Stefanie et al., (2009) proposed a mixed integer bi-level problem having a probabilistic knapsack constraint in the first level. The problem formulation is mainly motivated by practical pricing and service provision problems as it can be interpreted as a model for the interaction between a service provider and clients. The authors assumed the probability space to be discrete which allows us to reformulate the problem as a deterministic equivalent bi-level problem. Via a re-formulation as linear bi-level problem, we obtain a quadratic optimization problem, the so called Global Linear Complementarity Problem. Based on this quadratic problem, the authors finally proposed a procedure to compute upper bounds on the initial problem by using a Lagrangian relaxation and an iterative linear min-max scheme. The knapsack problem (KP) and its multidimensional version (MKP) are basic problems in combinatorial optimization.

Eleni and Nicos (2010) presented a new exact tree-search procedure for solving two-dimensional knapsack problems in which a number of small rectangular pieces, each of a given size and value, are required to be cut from a large rectangular stock plate. The objective is to maximize the value of pieces cut or minimize the wastage. The authors considered the case where there are a maximum number of times that a piece may be used in a cutting pattern. The algorithm limits the size of the tree search by using a bound derived from a Langrangean relaxation of a 0–1 integer programming formulation of the problem. Sub-gradient optimization is used to optimize this bound. Reduction tests derived from both the original problem and the Lagrangean relaxation produce substantial computational gains. The computational performance of the algorithm indicates that it is an effective procedure capable of solving optimally practical two- dimensional cutting problems of medium size.

Lawler (1997) presented fully polynomial approximation algorithms for knapsack problems are presented. These algorithms are based on ideas of Ibarra and Kim, with modifications which yield better time and space bounds, and also tend to improve the practicality of the procedures. Among the principal improvements are the introduction of a more efficient method of scaling and the use of a median-finding routine to eliminate sorting. The 0-1 knapsack problem, for n items and accuracy $\epsilon > 0$, is solved in $(nlog(1/\epsilon) + 1/\epsilon 4)$ time and $0(n + 1/\epsilon 3)$ space. The time bound is reduced to $0(n + 1/\epsilon 3)$ for the "unbounded" knapsack problem. For the "subset-sum" problem, $0(n + 1/\epsilon 3)$ times and $0(n + 1/\epsilon 2)$ spaces, or $0(n+1/\epsilon 2log(1/\epsilon))$ time and space, are achieved. The "multiple choice" problem, with m equivalence classes, is solved in $0(nm2/\epsilon)$ time and space.

Balasubramanian and Sanjiv (1988) considered the equality-constraint knapsack problem, which has received relatively little attention. The authors described a branch-and-bound algorithm for this problem, and present computational experience with up to 10,000 variables. An important feature of this algorithm is a least-lower-bound discipline for candidate problem selection.

Esther et al., (1993) studied a variety of geometric versions of the classical knapsack problem. In particular, the authors considered the following fence enclosure problem: given a set S of n points in the plane with values vi $>$ 0, we wish to enclose a subset of the points with a fence (a simple closed curve) in order to maximize the value of the enclosure. The value of the enclosure is defined to be the sum of the values of the enclosed points minus the cost of the fence. They also considered various versions of the problem, such as allowing S to consist of points and/or simple polygons. Other versions of the problems are obtained by restricting the total amount of fence available and also allowing the enclosure to consist of at most M connected components.

When there is an upper bound on the length of fence available, we show that the problem is NP-complete. We also provide polynomial-time algorithms for many versions of the fence problem when an unrestricted amount of fence is available.

Volgenant and Zoon (1990) presented a multidimensional 0-1 knapsack problem using heuristic, based on Lagrange multipliers, that also enables the determination of an upper bound to the optimal criterion value. This heuristic is extended in two ways: (1) in each step, not one, but more multiplier values are computed simultaneously, and (2) at the end the upper bound is sharpened by changing some multiplier values. From a comparison using a large series of different test problems, the extensions appear to yield an improvement, on average, at the cost of only a modest amount of extra computing time.

The binary knapsack problem is a combinatorial optimization problem in which a subset of a given set of elements needs to be chosen in order to maximize profit, given a budget constraint. Das and Ghosh (2003) studied a stochastic version of the problem in which the budget is random. The authors proposed two different formulations of this problem, based on different ways of handling infeasibility, and propose an exact algorithm and a local search-based heuristic to solve the problems represented by these formulations. The authors also presented the results from some computational experiment.

Goyal and Ravi (2009) presented a stochastic knapsack problem where each item has a known profit but a random size. The goal is to select a profit maximizing set of items such that the probability of the total size of selected items exceeding the knapsack size is at most a given threshold. The authors presented a parametric linear programming (LP) formulation and showed that it is a good approximation of the chance-constrained stochastic knapsack problem. The knapsack problem is known to be a typical NP-complete problem, which has 2n possible solutions to search over. Thus a task for solving the knapsack problem can be accomplished in 2n trials if an exhaustive search is applied. In the past decade, much effort has been devoted in order to reduce the computation time of this problem instead of exhaustive search.

In 1984, Karnin proposed a brilliant parallel algorithm, which needs $O(2^{n/6})$ processors to solve the knapsack problem in $O(2^{n/2})$ time; that is, the cost of Karnin's parallel algorithm is $O(2^{2n/3})$. Der-Chyuan Lou and Chin-Chen Chang (1997) proposed a fast search technique to improve Karnin's parallel algorithm by reducing the search time complexity of Karnin's parallel algorithm to be $O(2^{n/3})$ under the same $O(2^{n/6})$ processors available. Thus, the cost of the proposed parallel algorithm is $O(2^{n/2})$. Furthermore, the authors extended their technique to the case that the number of available processors is $P = O(2^x)$, where $x \geq 1$. From the analytical results, they saw that their search technique is indeed superior to the previously proposed methods. They do believe their proposed parallel algorithm is pragmatically feasible at the moment when multiprocessor systems become more and more popular.

Knapsack problem is a typical NP complete problem. During last few decades, Knapsack problem has been studied through different approaches, according to the theoretical development of combinatorial optimization. Garg and Sunanda (2009) put forward the evolutionary algorithm for 0/1 knapsack problem. A new objective function evaluation operator was proposed which employed adaptive repair function named as repair and elitism operator to achieve optimal results in place of problem specific knowledge or domain specific operator like penalty operator (which are still being used). Additional features had also been incorporated which allowed the algorithm to perform more consistently on a larger set of problem instances. Their study also focused on the change in behavior of outputs generated on varying the crossover and mutation rates. New algorithm exhibited a significant reduction in number of function evaluations required for problems investigated.

Srisuwannapa and Charnsethikul (2007) presented a variant of the unbounded knapsack problem (UKP) into which the processing time of each item is also put and considered, referred as MMPTUKP. The MMPTUKP is a decision problem of allocating amount of n items, such that the maximum processing time of the selected items is minimized and the total profit is gained as at least as determined without exceeding capacity of knapsack. In this study, we proposed a new exact algorithm for this problem, called MMPTUKP algorithm. This pseudo polynomial time algorithm solves the bounded knapsack problem (BKP) sequentially with the updated bounds until reaching an optimal solution. The authors presented computational experience with various data instances randomly generated to validate their ideas and demonstrate the efficiency of the proposed algorithm.

Ronghua et al., (2006) presented a new multiobjective optimization (MO) algorithm to solve 0/1 knapsack problems using the immune Clonal principle. This algorithm is termed Immune Clonal MO Algorithm (ICMOA). In ICMOA, the antibody population is split into the population of the non-dominated antibodies and that of the dominated anti-bodied. Meanwhile, the non-dominated antibodies are allowed to survive and to clone. A metric of Coverage of Two Sets are adopted for the problems. This quantitative metric is used for testing the convergence to the Pareto-optimal front. Simulation results on the 0/1 knapsack problems show that ICMOA, in most problems, is able to find much better spread of solutions and better convergence near the true Pareto-optimal front compared with SPEA, NSGA, NPGA and VEGA.

Deniz et al., (2010) studied maximization of revenue in the dynamic and stochastic knapsack problem where a given capacity needs to be allocated by a given deadline to sequentially arriving agents. Each agent is described by a two-dimensional type that reflects his capacity requirement and his willingness to pay per unit of capacity. Types are private information. The authors first characterize implementable policies. Then they solved the revenue maximization problem for the special case where there is private information about per-unit values, but capacity needs are observable. After that they derived two sets of additional conditions on the joint distribution of values and weights under which the revenue maximizing policy for the case with observable weights is implementable, and thus optimal also for the case with two- dimensional private information. In particular, they investigated the role of concave continuation revenues for implementation. We also construct a simple policy for which per- unit prices vary with requested weight but not with time, and prove that it is asymptotically revenue maximizing when available capacity/ time to the deadline both go to infinity. This highlights the importance of nonlinear as opposed to dynamic pricing.

Computational grids are distributed systems composed of heterogeneous computing resources which are distributed geographically and administratively. These highly scalable systems are designed to meet the large computational demands of many users from scientific and business orientations. However, there are problems related to the allocation of the computing resources which compose of a grid.

Van dester et al., (2008) studied the design of a Pan-Canadian grid. The design exploits the maturing stability of grid deployment toolkits, and introduces novel services for efficiently allocating the grid resources. The changes faced by this grid deployment motivate further exploration in optimizing grid resource allocations. By applying this model to the grid allocation option, it is possible to quantify the relative merits of the various possible scheduling decisions. Using this model, the allocation problem was formulated as a knapsack problem. Formulation in this manner allows for rapid solution times and results in nearly optimal allocations. Last few years have seen exponential growth in the area of web applications, especially, e- commerce and web-services. One of the most important qualities of service metric for web applications is the response time for the user. Web application normally has a multi-tier architecture and a request might have to traverse through all the tiers before finishing its processing. Therefore, a request's total response time is the sum of response time at all the tiers. Since the expected response time at any tier depends upon the number of servers allocated to this tier, many different configurations (number of servers allocated to each tier) can give the same quality of service guarantee in terms of total response time. Naturally, one would like to find the configuration which minimizes the total system cost and satisfies the total response time guarantee. Zhang et al., (2004) modelled this problem as integer optimization problem.

The strike-force asset allocation problem consists of grouping strike force assets into packages and assigning these packages to targets and defensive assets in a way that maximizes the strike force potential. Chi-Wei, et al., (2001) modeled this problem as integer programming formulation, and proposed a branch and bound algorithm to solve it.

Sung-Ho (1998) presented a techniques for obtaining strategies to allocate rooms to customers belonging to various market segments, considering time dependent demand forecasts and a fixed hotel capacity. This technique explicitly accounts for group and multi-night reservation requests in an efficient and effective manner. This is accomplished by combining an optimal discrete-dynamic model for handling single-night reservation requests, bases on a static integer programming model, developed to handle multi-night reservation requests.

Allocation of resources under uncertainty is a very common problem in many real-life scenarios. Employers have to decide whether or not to hire candidates, not knowing whether future candidates will be stronger or more desirable. Machines need to decide whether to accept jobs without knowledge of the importance or profitability of future jobs. Consulting companies must decide which jobs to take on, not knowing the revenue and resources associated with potential future requests. More recently, online auctions have proved to be a very important resource allocation problem. Advertising auctions in particular provide the main source of monetization for a variety of internet services including search engines, blogs, and social networking sites. Additionally, they are the main source of customer acquisition for a wide array of small online business, of the networked world. In bidding for the right to appear on a web page (such as a search engine), advertisers have to trade-off between large numbers of parameters, including keywords and viewer attributes. In this scenario, an advertiser may be able to estimate accurately the bid required to win a particular auction, and benefit either in direct revenue or name recognition to be gained, but may not know about the trade-off for future auctions.

All of these problems involve an online scenario, where an algorithm has to make decisions on whether to accept an offer, based solely on the required resource investment (or weight) and projected value of the current offer, with the total weight of all selected offer not exceeding a given budget. When the weights are uniform and equal to the weight constraint, the problems above reduces to the famous secretary problem which was first introduced by (Dynkin, 1963). Moshe et al., (2008), studied this model as a knapsack problem.

Kleinberg (2009) presented a model for the multiple-choice secretary problem in which k elements need to be selected and the goal is to maximize the combined value (sum) of the selected elements.

Rajeev and Ramesh (1992) presented a new greedy heuristic for the integer knapsack problem. The proposed heuristic selects items in non-increasing order of their maximum possible contribution to the solution value given the available knapsack capacity at each step. The lower bound on the performance ratio for this “total-value” greedy heuristic is shown to dominate the corresponding lower bound for the density-ordered greedy heuristic.

George (1995) proposed the average-case behavior of the Zero–One Knapsack problem, as well as an on-line version. The authors allowed the capacity of the knapsack to grow proportionally to the number of items, so that the optimum solution tends to be $\Theta(n)$. Under fairly general conditions on the distribution, they obtained a description of the expected value of the optimum offline solution which is accurate up to terms which are o (1). The authors then considered a simple greedy method for the on-line problem, which is called Online Greedy and is allowed to use knowledge of the distribution, and shown that the solution obtained by this algorithm differs from the true optimum by an average of $\Theta(\log n)$; in fact, and can determine the multiplicative constant hidden by the $\Theta$-notation. Thus on average the cost of being forced to give answers on-line is quite small compared to the optimum solution.

The constrained compartmentalized knapsack problem is an extension of the classical integer constrained knapsack problem which can be stated as the following hypothetical situation: a climber must load his/her knapsack with a number of items. For each item a weight, a utility value and an upper bound are given. However, the items are of different classes (food, medicine, utensils, etc.) and they have to be loaded in separate compartments inside the knapsack (each compartment is itself a knapsack to be loaded by items from the same class). The compartments have flexible capacities which are lower and upper bounded. Each compartment has a fixed cost to be included inside the knapsack that depends on the class of items chosen to load it and, in addition, each new compartment introduces a fixed loss of capacity of the original knapsack.

The constrained compartmentalized knapsack problem consists of determining suitable capacities of each compartment and how these compartments should be loaded, such that the total items inside all compartments does not exceed the upper bound given. The objective is to maximize the total utility value minus the cost of the compartments. This kind of problem arises in practice, such as in the cutting of steel or paper reels. Doprado and Nereu (2007) modeled the problem as an integer non-linear optimization problem for which some heuristic methods are designed. Finally, computational experiments were given to analyze the methods.

The Multiple Knapsack Problem (MKP) is a NP-hard combinatorial optimization problem in many real-word applications. An algorithm with the behaviors of preying, following and swarming of artificial fish for searching optimal solution was proposed by Ma Xuan (2009). With regard to the problem that infeasible solutions are largely produced in the process of initializing individuals and implementing the behaviors of artificial fish due to the multiple constraints, which undermines the algorithm performance, an adjusting operator based on heuristic rule was designed to ensure all the individuals in the feasible solution areas. Computational results show that the algorithm can quickly find optimal solution. The proposed algorithm can also be applied to other constrained combinatorial optimization problems. The above literature shows that knapsack is a very key tool which has helped in many fields.

# Chapter 3

# METHODOLOGY

## 3.1 Introduction

The knapsack problem appears in many forms in major fields such as Economics, Engineering and Business. However it can be employed anywhere one must allocate a scare resource among many contenders to get the optimal return or output.



## 3.2 Some Examples of the Knapsack Problem

There are various types of the Knapsack Problems. This is dependent on the distribution of the items and knapsacks. Mention can be made of the $0-1$ Knapsack Problem in which each item can be chosen at most once. For the Bounded Knapsack problem we have a bounded amount of each item type. The Multiple-choice knapsack Problem arises when the items should be chosen from disjoint classes and, if several Knapsacks are to be filled simultaneously, we get the Multiple Knapsack problem.

## 3.3 Single Knapsack Problem

Problems that fall under this type of knapsack have one container (or knapsack) that must be filled with optimal subset of items. If we choose to denote the capacity of the container by c, we can examine some problems that fall under this group:

- 0-1 knapsack problem

- Bounded knapsack problem

### 3.3.1 The Single 0-1 Knapsack Problem

Considering the classical $0-1$ knapsack problem (KP) where a subset of $n$ given items has to be packed in a knapsack of capacity $c$. Each item has a profit $P_j$ and a weight $w_j$ and the problem is to select a subset of the items whose total weight does not exceed $c$ and whose total profit is a maximum,

If we adopt a basic assumption that all input data are positive integers and also add a binary decision variable $x_j$ with $x_j = 1$ if item $j$ is selected and $x_j = 0$ otherwise, we obtain the integer linear programming (ILP) model:

$$\text{Maximize} \qquad z = \sum_{j=1}^{n} P_j x_j \tag{3.1}$$

$$\text{Subject to} \qquad \sum_{j=1}^{n} w_j x_j \leq c \tag{3.2}$$

$$\forall x_j \in \{0, j\} \text{ where } \{1, \cdots, n\} \tag{3.3}$$

Where all data are positive integers

Equation 3.1 represents the objective function to be maximized. Equation 3.2 is the constraint of the problem whiles equation 3.3 shows the items selected to be selected.

### 3.3.2 The Bounded Knapsack Problem

Let us consider the following definition for the various inputs of the bounded knapsack problem (BKP) :

Given $n$ item types and a knapsack, with

$p_j$ = profit of an item of type $j$;

$w_j$ = weight of an item of type $j$;

$b_j$ = upper bound on the availability of an items of type $j$;

$c$ = capacity of the knapsack,

$x_j (j = 1, \cdots, n)$ = items to be selected

Again we assume that all input data are positive integers and given a binary decision variable $x_j = 1$ if item $j$ is selected and $x_j = 0$ otherwise, we get the integer linear programming (ILP) model:

$$\text{Maximize} \qquad z = \sum_{j=1}^{n} P_j x_j \tag{3.4}$$

$$\text{Subject to} \qquad \sum_{j=1}^{n} w_j x_j \leq c \tag{3.5}$$



$$\forall x_j \in \{0, b_j\} \text{ where } j \in N = 1, \cdots, n \tag{3.6}$$

Where $x_j$ is bounded non negative number.

Equation 3.4 represents the objective function to be maximized. Equation 3.5 is the constraint of the problem whiles equation 3.6 shows the items selected to be selected.

### 3.3.3 Multiple Knapsack Problem

The key factor for this kind of problem is the fact that more than one container is available and must be filled with optimal subset of items. We will consider to the 0-1 Multiple Knapsack problem.

Given a set of $n$ items and a set of $m$ knapsacks $(m \leq n)$ , with

$P_j$ = profit of item j;

$w_j$ = weight of item j;

$c_j$ = capacity of knapsack i,

Select in disjoint subsets of items so that the total profit of the selected items is a maximum and each subset can be assigned to a different knapsack whose capacity is not less than the total weight of items in the subject. Hence

$$\text{Maximize} \qquad z = \sum_{i=1}^{m} \sum_{j=1}^{n} P_j x_{ij} \tag{3.7}$$

$$\text{Subject to} \qquad \sum_{j=1}^{n} w_j x_{ij} \leq c_i \tag{3.8}$$

$$\sum_{j=1}^{n} x_{ij} \leq 1 \text{where } j \in N = \{1, \cdots, n\},\ i \in M = \{1, \cdots, m\} \tag{3.9}$$



$$x_{ij} \in \{0,1\} \text{where } x_{ij} = \begin{cases} 1 & \text{if item j is assigned to knapsack i} \\ 0 & \text{if otherwise} \end{cases} \tag{3.10}$$

Equation 3.7 represents the objective function to be maximized. Equation 3.8 is the constraint of the problem whiles equation 3.9 and 3.10 shows the items selected to be selected.

## 3.4 Formulation of the Knapsack Problem (KS)

Given a set of n times; each item $i(i = 1, \cdots, n)$ has a value $c_i > 0$ and a weight $a_i > 0$. We have a knapsack with (weight) capacity b has to be filled with items so as to maximize the total value of the items included in the knapsack. Without loss of generality, we assume that all weights, $a_i$ and values $c_i$ are integral; hence b is integral due to the integrality of the weights.

We can now formulate the knapsack problem (KS) by using the binary variables $x_i(i = 1, \cdots, n)$ , where the outcome $x_i = 1$ signals that item $i$ must be included in the knapsack and $x_i = 0$ signals that item $i$ has been be left at home.

$$(KS) \qquad \max \sum_{i=1}^{n} c_i x_i \tag{3.11}$$

$$\text{Subject to} \qquad \sum_{i=1}^{n} a_i x_i \leq b \tag{3.12}$$

$$\forall x_i \in \{0,1\} \text{ where } i = \{1, \cdots n\} \tag{3.13}$$

We suppose that $\sum_{i=1}^{n} > b$ and $a_i \leq b(i = 1, \cdots, n)$ to avoid trivialities.
Equation 3.11 is the objective function in which we seek to maximize the values placed on the items to be selected. Equation 3.12 represents the constraint. It reminds us of the limiting factor which is the capacity as we make the choice of which item to pick. Equation 3.13 shows us the items to choose with 1 representing a chosen item and 0 otherwise.



## 3.5 The Continuous Knapsack Problem

If we no longer require that a solution of the knapsack problem to be integral but allow for fractional values, then we get the continuous knapsack problem (CKS); this is a relaxation of the knapsack problem in the sense that the set of feasible solutions for the continuous knapsack problem contains the solution set of knapsack as a subset.

$$(CKS) \qquad \max \sum_{i=1}^{n} c_i x_i \tag{3.14}$$

$$\text{Subject to} \qquad \sum_{i=1}^{n} a_i x_i \leq b \tag{3.15}$$

$$\forall x_i \in \{0,1\} \text{ where } i = \{1, \cdots n\} \tag{3.16}$$

Equation 3.14 represents the objective function to be maximized. Equation 3.15 is the constraint of the problem whiles equation 3.16 shows the items selected and that selected items can be fractions as well.
Consider any instance $I_{KS}$ of the knapsack problem and the corresponding instance of the continuous knapsack problem $I_{CKS}$, $I_{CKS}$ is called the linear

programming relaxation, since the integrally constraints of KS are relaxed to linear constraints. Clearly, for the optimal values of $I_{KS}$, and $I_{CKS}$ which are denoted by $Z(I_{KS})$, and $Z(I_{CKS})$ we have that $Z(I_{KS}) \leq Z(I_{CKS})$. Hence $Z(I_{CKS})$ forms an upper bound on $Z(I_{KS})$.

Although the set of feasible solutions of $I_{CKS}$ extends the set of feasible solution of $I_{KS}$, there is a straightforward greedy algorithm that loves CKS. Renumber the items according to non-increasing $\frac{c_i}{a_i}(i = 1, \cdots, n)$ that is, $\frac{c_1}{a_1} \geq \frac{c_2}{a_2} \geq \cdots \geq \frac{c_n}{a_n}$. For this sequence we have hat the most interesting items, that is, those with the highest value per unit of weight are numbered lowest. The greedy algorithm raises the values of the variables in the order $x_1, x_2, \cdots, x_n$



## 3.6 Method of Solving Knapsack Problems

The $0-1$ knapsack problems can be solved by the general techniques of the branch and bound method.

## 3.7 Branch and Bound Algorithm for Knapsack

The first branch-and-bound approach to the exact solution of KP was presented by Kolesar (1967). His algorithm consists of a highest-first binary branching scheme. The large computer memory and time requirements of the Kolesar algorithm were greatly reduced by the Greenberg and Hegerich (1970) approach, differing in two main respects:

(a) At each mode, the continuous relaxation of the indicated sub problem is solved and the corresponding critical items is selected to generate the two descendent nodes (by imposing $X_s = 0$; on $(X_s = 1)$

(b) The search continues from the node associated with the exclusion of item $s$ ( condition $X_t = 0$)

When the continuous relaxation has an all-integer solution the search is resumed form the last node degenerated by imposing $X_s = 1$ i.e. the algorithm is of depth

– first type. Horowitz and Sahni (1997) (and independently, Ahrens and Finke (1975)) derived from the previous scheme a depth-first algorithm in which;

(a) Selection of the branching variable $X_j$ is the same as Koleaser;

(b) The search continues from the node associated with the insertion of item j (Condition $X_j = 1$), i.e. following a greedy strategy.

The Horowitz – Sahni algorithm is the most effective, structured and easy to implement and has constituted the basis for several improvements, including that of Martello – Toth algorithm (Martello and Toth, 1977), which is generally considered highly effective.



## 3.8 Branch and Bound Method

Branch and bound method employs a strategy in which the feasible region is segmented into smaller sub-regions or nodes. Each node is examined for integer feasibility. This is done by relaxing the integrality requirement of the knapsack problem as shown in the following steps:

Step 1:

We relax the integrality requirements of the knapsack problem. The Linear Programming (LP) problem is referred to as Node 1. The optimal value of the objective function is the initial upper bound (UB) for the objective function value. If this relaxed LP is found infeasible the Integer Programming problem is infeasible, STOP.

Step 2:

Compare the UB values for any currently defined nodes. If the solution at the node with the highest UB value satisfies the integrality requirements STOP, that solution is the optimal. If the highest UB value is -$\infty$ STOP, the problem is (integer) infeasible.

Step 3:

Branch at the node with the highest UB value, by imposing two mutually exclusive constraints on the value of variable $x_k$ whose present value ($x_k^*$) violates the integrality requirements. The constraints:

$$x_k \leq [x_k^*] \text{ and } x_k \leq [x_k^*] + 1$$

are added to the existing constraints defining the feasible region thus partitioning it in two.

Solve the LP problems corresponding to the two newly defined nodes thus establishing the UB values for those nodes. If any node is found to be infeasible the corresponding UB value is put up equal to -$\infty$. Go to step 2.

## 3.9 Linear Programming

Linear programming adopts mathematical models to describe any problem of concern. By its first name, ie Linear, there is an implicit requirement that mathematical functions in the models should be linear. Programming is means putting in place a series of activities to come out with an optimal result.

## 3.10 Integer Linear Programming

Problems that come under this category are basically Linear Programming problems with a key conditionality with respect to integrality of all or some variables. When the integrality requirements are relaxed, the relaxed Linear Programming problem could be a bound on the best possible value for the optimal objective function value of the Integer Programming problem.

## 3.11 The Horowitz – Sahni Algorithm

One key property of this algorithm is that the items are sorted in a decreasing profit to weight ratio. A forward move consists of inserting the largest set of new consecutive items into the current solution. Whenever a forward move is exhausted, the upper bound $U_1$ corresponding to the current solution is computed and compared with the best solution so far, in order to check whether further forward moves could lead to a better one; if so, a new forward move is performed, otherwise a backtracking follows. When the last item has been considered, the current solution is complete and possible updating of the best solution so far occurs. The algorithm stops when no further backtracking can be performed. In the following description of the algorithm we use these notations.

$n$ = number of items

$\hat{X}_j$ = current solution;

$P_j$ = profit of item j;

$w_j$ = weight of itme j;

$C$ = capacity of the knapsack;

$\hat{X}$ = current solution value $\left(= \sum_{j=1}^{n} p_j \hat{x}_j\right)$

$\hat{C}$ = current residual capacity $\left(= \sum_{j=1}^{n} w_j \hat{x}_j\right)$

$\hat{x}_j$ = best solution so far;

$Z$ = value of the best solution so far $\left(= \sum_{j=1}^{n} p_j \hat{x}_j\right)$

**The Harowiz - Sahni Algorithm**

Input: $n, C, (P_j), (w_j)$;

Output: $Z; (x_j)$;

Begin

1 [Initialize]

$Z := 0$;

$\hat{Z} := 0$;

$\hat{C} := C$;

$p_{n+1} := 0$;

$w_{n+1} := +\infty$;

$j := 1$

2 [Compute upper bound $U_1$]

Find $r = \min\left\{ i : \sum_{k=1}^{i} w_k > \hat{C}; \right\}$

$U : \sum_{k=j}^{r-1} p_k + \left[ \left( \hat{C} - \sum_{k=j}^{r-1} w_k \right) \frac{p_r}{w_k} \right]$

If $Z \geq \hat{Z} + U$ then go to 5;

3 [Perform a forward step]

While $w_j \leq \hat{C}$ do

begin

$\hat{C} := C - w_j;$

$\hat{Z} := \hat{Z} + P_j$

$\hat{x}_j := 1$

$j = j + 1$

end

if $j \leq n$ then

begin

$\hat{x}_j = 0$

$j = j + 1$

end

if $j < n$ then go to 3:

if $j = n$ then go to 2;

4 [Update the best solution so far]

if $\hat{Z} > Z$ then

begin

$Z := \hat{Z};$

for $k := 1$ to $n$ do $x_k := \hat{x}$

end

$j := n;$

if $\hat{x} = 1$ then

begin $\hat{C} := \hat{C} + w_i$ :

$\hat{Z} := \hat{Z} - p_i;$

$\hat{x}_i := 0;$

$j := i + 1;$

go to 2

end

5 [Backtracking]

find $i = \max\{k < j : \hat{x}_k = 1\};$

if no such i then return to 4;

$\hat{C} := \hat{C} + w_i :$

$\hat{Z} := \hat{Z} - p_i;$

$\hat{x}_i := 0;$

$j := i + 1;$

go to 2

**Example 3.11.1** *Consider the instance of KP defined by $n = 7$*

$$\begin{aligned}(p_j) &= (70, 20, 39, 37, 7, 5, 10)\\ (w_j) &= (31, 10, 20, 19, 4, 3, 6)\\ C &= 50\end{aligned} \tag{3.17}$$

To obtain a good appreciation of the Harowiz-Sahni algorithm, we will apply the steps involved to example 3.17 above in order to get the first solution set before coming out with the decision tree.

Solution:

Input: $n, C, (P_j), (w_j);$

n=7, $(p_j) = (70, 20, 39, 37, 7, 5, 10), (w_j) = (31, 10, 20, 19, 4, 3, 6), C = 50$

Output: $Z : (x_j)$

r=3

$$U = 90 + \left[(50 - 41)\frac{39}{20}\right] = 107$$

for j=1, $31 \leq 50$,

$\hat{C} = 30 - 31 = 19,\ \hat{Z} = 0 + 70 = 70,\ \hat{x_1} = 1$ ie chosen and j= 1+2=3

j=3, $20 > 9$

$\hat{x_3} = 0$ ie chosen and j= 3+1=4

j=4, $19 > 9$

$\hat{x_4} = 0$ ie chosen and j= 4+1=5

for j=5, $4 \leq 9$

$\hat{C} = 9 - 4 = 5,\ \hat{Z} = 90 + 7 = 97,\ \hat{x_5} = 1$ ie chosen and j= 5+1=6

for j=6, $3 \leq 5$

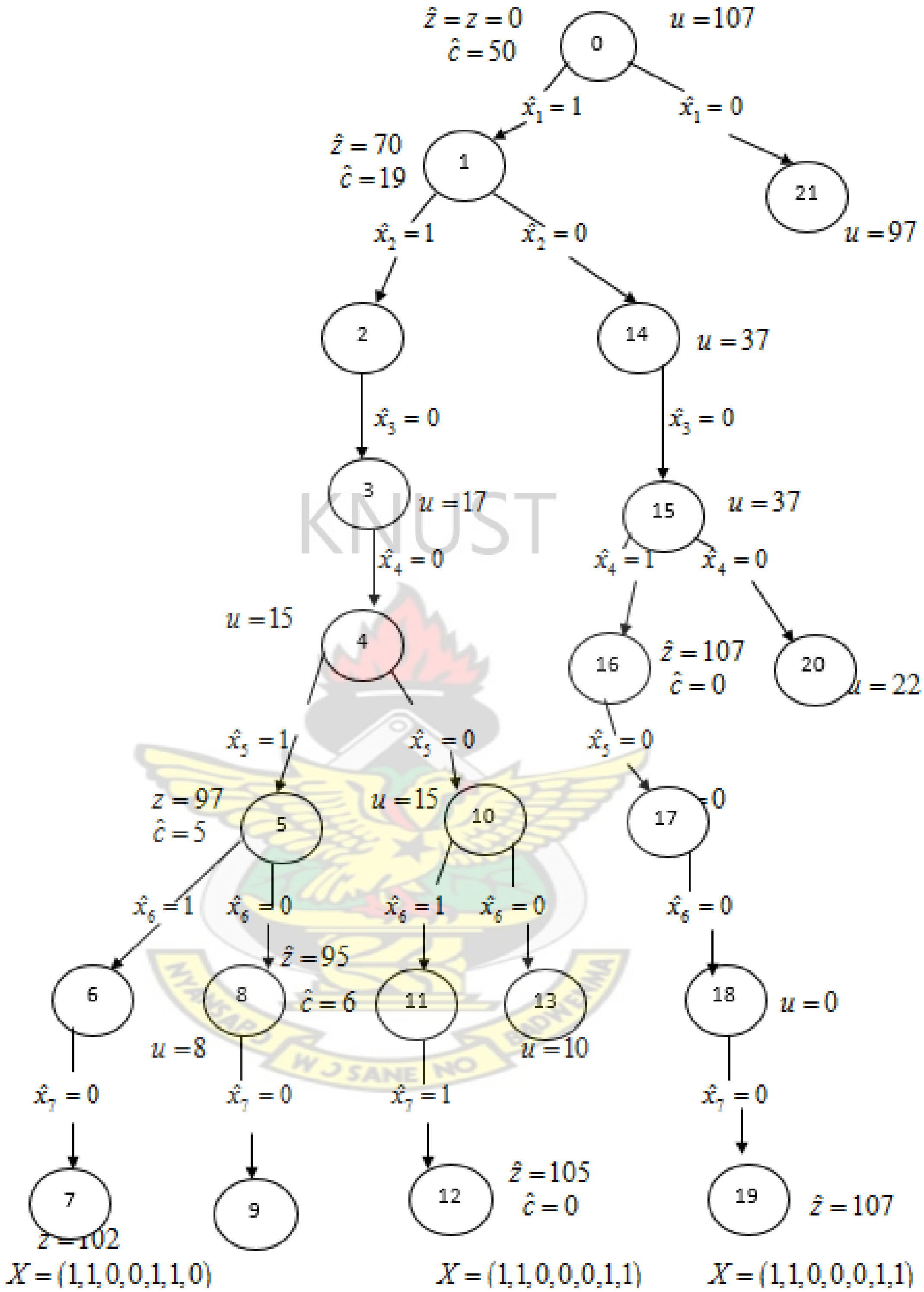
$\hat{C} = 5 - 3 = 2,\ \hat{Z} = 97 + 5 = 102,\ \hat{x_6} = 1$ ie chosen and j= 6+1=7

for j=7, $6 > 2$

$\hat{x_7} = 0$ ie not chosen and j=n so go to 2

So we have (1,1,0,0,1,1,0) as the first solution set with a value of 102.

Below is the decision tree of example 3.1 using the Harowiz-Sahni algorithm




By applying the above algorithm, we would have the decision tree of above. The optimal solution of this example from the decision tree of Horowitz and Sahni algorithm is $X = (1, 0, 0, 1, 0, 0, 0)$

## 3.12 The Martello – Toth algorithm

Their method differs from that of Horowitz and Sahni (1974) in the following main respect (we use the notations introduced in the previous method)

(a) Upper bond $U_2$ is used instead of $U_1$

(b) The forward move associated with the selection of the $j^{th}$ item is spilt into two phases: building of a new current solution and saving the current solution. In the first phase, the largest $N_j$ of consecutive items which can be inserted into the current solution staring from the $j^{th}$ item is defined, and the upper bound corresponding to the inserting of the $j^{th}$ item is computed. If this bound is less than or equal to the value of the best solution so far, a backtracking move immediately follows. If it is greater, the second phase, that is, insertion of the items of set $N_j$ into the current solution is performed only if the value of such new solution does not represent the maximum which can be obtained by inserting the $j^{th}$ item. Otherwise, the best solution so far is changed, but the current solute is not updated, so that unnecessary backtracking on the items in $N_t$ are avoided.

(c) A particular forward procedure, based on dominance criteria, is performed whenever, before a backtracking move on the $i^{th}$ item, the residual capacity $\hat{C}$ does not allow insertion into the current solution of any item following the $i^{th}$. The procedure is based on the following consideration;

The current solution could be improved only if the $i^{th}$ item is replaced by an item having greater profit and a weight small enough to allow its insertion, or by at least two items having global weighty not greater than $W_i + \hat{C}$. By this approach it is generally possible to eliminate most of the unnecessary nodes generated at the lowest levels of the decision – tree.

(d) The upper bounds associated with the nodes of the decision – tree are computed through a parametric technique based on the storing of information related to the current solution. Supposing the current solution has been built by inserting all the items from the $j^{th}$ to the $r^{th}$ then, when performing a backtracking on one of these items (say the $i^{th}, j \leq i < r$, if no insertion occurred for the items preceding the $j^{th}$, it is possible to insert at least items.

**The Martello – Toth algorithm**

Input: $n, C, (P_j), (w_j)$;

Output: $Z; (x_j)$;

Begin

1 [Initialize]

$Z := 0$;

$Z := 0$;

$C := C$;

$p_{n+1} := 0$;

$w_{n+1} := +\infty$;

for $k := 1$ to $n$ do $x_k := 0$;

Compute the upper bound $U = U_2$ in the optimal solution value;

$\bar{w}_1 := 0$;

$\bar{p}_1 := 0$;

$\bar{r}_1 := 0$;

$\bar{r} := n$;

for $k := n$ to 1 do compute $m_k = \min\{w_i; i > k\}$;

j:=1

2 [build a current solution]

While $w_j > C$ do

If $Z \geq Z + [CP_{j+2}/W_{j+1}]$ then go to 5 else $j := j + 1$;

find $r = \min\left\{i : \bar{w}_j + \sum_{k=\bar{r}}^{i} w_k > C\right\}$

$p := \bar{p}_j + \sum_{k=\bar{r}_j}^{r-1} p_k$;

$w := \bar{w}_j + \sum_{k=\bar{r}_j}^{r-1} w_k;$

If $r \leq n$ then $U := \max\left(\left[(\hat{C} - w')\frac{P_{r+1}}{w_{r+1}}\right], \left[p_r - \left[w_r - (\hat{C} - w')\right]\frac{P_{r-1}}{w_{r-1}}\right]\right)$

else $U := 0;$

if $Z \leq Z + P + U$ then go to 5;

if $U = 0$, then go to 4

3 [save the current solution]

$\hat{C} := \hat{C} - w';$

$Z \geq Z + P$

for $k := j$ to $r - 1$ do $x_k := 1$

$\bar{w}_j := w'$

$\bar{p}_j := P';$

$\bar{r}_j := r;$

for k=j+1 to r-1 do

begin

$\bar{w}_k := \bar{w}_{k-1} = w_{k-1}$

$\bar{p}_k := \bar{p}_{k-1} = p_{k-1}$

$\bar{r}_k := r$

end

for $k := r$ to $\bar{r}$ do

begin

$\bar{w}_k := 0;$

$\bar{p}_k := 0;$

$\bar{r}_k := k;$

end

$\bar{r} := r - 1;$

$j := r + 1;$

If $C \geq m_{j-1}$ then go to 2;

if $Z \geq Z$ then go to 5;

$P' := 0$

4 [Update the best solution so far]

$Z \geq Z + P'$

for $k := 1$ to $j - 1$ do $x_k := x_k$

for $k := 1$ to $r - 1$ do $x_k := 1$

for $k := r$ to $n$ do $x_k := 0$

if $Z = U$ then return;

5 [Backtracking]

find $i = \max\{k < j : x_k = 1\}$;

if no such i then return to 4

$\hat{C} := \hat{C} + w_i$;

$Z = Z - p_i$;

$x_i := 0$;

$j := i + 1$;

if $C - w_i \geq m_i$ then go to 2;

j:=i;

h:=i;

6 [try to replace item i with item h]

h:= h+1;

if $Z \geq Z + \left[C\dfrac{P_h}{w_h}\right]$ then go to 5;

if $w_h = w_i$ then go to 6

if $w_h > w_i$ then

begin

if $w_h > \hat{C}$ or $Z \geq \hat{Z} + p_h$ then go to 6

Z:=$\hat{Z} + p_h$;

for $k := 1$ to $n$ do $x_k := x_k$;

$x_h = 1$;

if $Z = U$ then return;

i:=h;

go to 6

else

begin

if $C - w_h < m_h$ then go to 6;

$C := C - w_h$;

$Z := Z + p_h$;

$x_h := 1$;

$j := h + 1$;

$\bar{w}_h := w_h$;

$\bar{p}_h := p_h$;

$\bar{r}_h := h + 1$‘;

for k:=h+1 to $\bar{r}$ do

begin

$\bar{w}_k := 0$;

$\bar{p}_k := 0$;

$\bar{r}_k := k$;

end

$\bar{r} := h$;

go to 2

end

end

Again, we apply the Marthelo and Toth algorithm to example 3.17 to obtain the first solution set.

Solution to Example 3.17

Input: $n, C, (P_j), (w_j)$;

n=7, $(p_j) = (70, 20, 39, 37, 7, 5, 10), (w_j) = (31, 10, 20, 19, 4, 3, 6), C = 50$

Output: $Z : (x_j)$

Let r=3

P'= 0+90=90,

W'=0+41=41,

Since r=3<7,

$$U = \max\left[(50-41)\frac{37}{19}, 39-(20-(50-41)\frac{20}{10})\right]$$

$U = \max(17, 37)$

$U = 37$

For k=j to r-1, $\hat{x}_k = 1$ i.e $\hat{x_1}$ and $\hat{x}_2$ have been chosen.



For r=3-4, $\hat{C} < W_j$,

For r=6

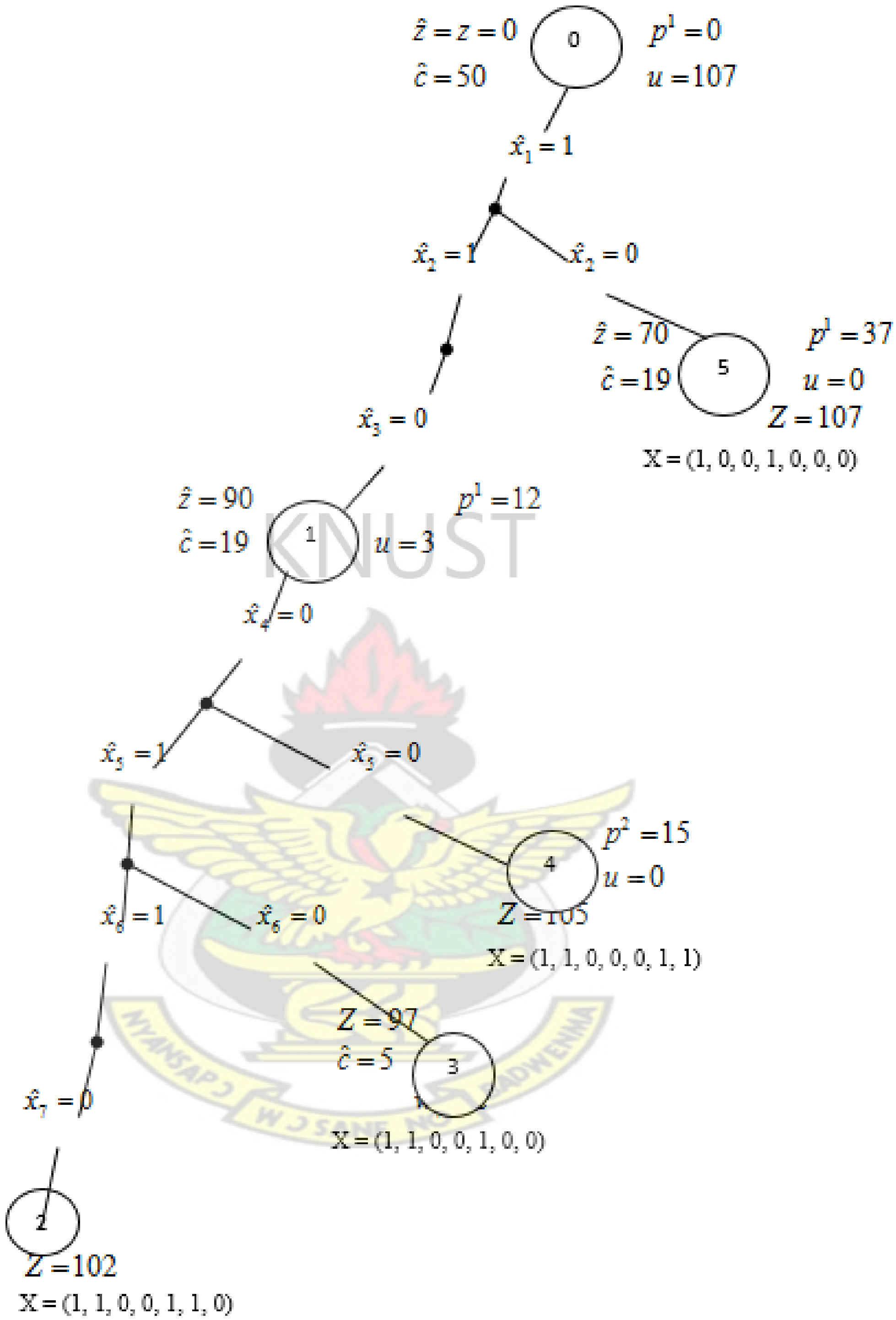
P'= 90+12=102

W'= 41+7= 48

U=0

For k=5 to r-1, $x_k = 1$;

So the chosen set consist of (1,1,0,0,1,1,0) with a value of 102 Below is the decision tree obtained by applying the Martello and Toth algorithm to example 3.17




Applying the Martello and Toth algorithm to example 3.1 gives the decision – tree above. The optimal solution of this example from the decision tree of Martello and Toth algorithm is X = (1, 0, 0, 1, 0, 0, 0) with a value of 107.

# Chapter 4

# DATA ANALYSIS AND RESULTS

## 4.1 Data Collection

The original data obtained from Ghana Stock exchange (GSE), (2013) involves Daily Stock of fifty- four (54) listed companies. However, data from only thirty-three companies were used for the analysis as some of the stocks had no price values. Others had more than 50% cases of missing values and for which reason could also not be included in the analysis. In all, there were 1,211 observations of Daily Stock Prices (DSP) spanning from September 1, 2008 to August 31, 2013. Below is a table showing of DSP. The first row captures the names of the various listed shares while the first column indicates dates from September 2008 to August 2013. The entry for each cell indicates the share price for that particular share on the corresponding date.

Daily Growth Rates (DGR) will now be computed from the Daily Stock prices spanning from 1st September, 2008 to 31st August, 2013. The formula for computing DGR is given as,

$$DGR_i = \frac{DSP_i - DSP_{i-1}}{DSP_{i-1}}$$

where,

$DGR_i$ = Daily Growth Rate for day i,

$DSP_i$ = Daily Stock Price for day i,

$DSP_{(i-1)}$ = Daily Stock Price for the day before day i (i.e. day i-1)

The table 4.2 below shows the computed Daily Growth Rate for the various shares. The first column shows the names of the various shares whiles the first

Table 4.1: Daily share Prices of Listed Shares

| Date | AADS | ACI | AGA | ALW | | TBL | TOTAL | TRANSOL | UNIL |
|---|---|---|---|---|---|---|---|---|---|
| 9/1/2008 | 0.35 | 0.08 | 30 | 0.62 | ...... | 1.33 | 7.2 | 0.11 | 3.51 |
| 9/2/2008 | 0.35 | 0.09 | 30 | 0.62 | | 1.33 | 7.2 | 0.11 | 3.55 |
| 9/3/2008 | 0.35 | 0.09 | 30 | 0.62 | | 1.33 | 7.2 | 0.11 | 4 |
| 9/4/2008 | 0.35 | 0.09 | 30 | 0.62 | | 1.33 | 7.2 | 0.11 | 4 |
| 9/5/2008 | 0.35 | 0.09 | 30 | 0.62 | | 1.33 | 7.2 | 0.11 | 4.01 |
| 9/8/2008 | 0.35 | 0.09 | 30 | 0.62 | | 1.33 | 7.21 | 0.11 | 4.01 |
| 9/9/2008 | 0.35 | 0.09 | 30 | 0.62 | | 1.33 | 7.21 | 0.11 | 4.02 |
| ... | | | | | | | | | |
| .... | | | | | | | | | |
| 22/08/2013 | 0.52 | 0.06 | 37 | 0.05 | | 0.35 | 5.24 | 0.04 | 15.15 |
| 23/08/2013 | 0.52 | 0.06 | 37 | 0.05 | | 0.35 | 5.24 | 0.04 | 15.17 |
| 26/08/2013 | 0.52 | 0.06 | 37 | 0.05 | | 0.35 | 5.24 | 0.04 | 15.17 |
| 27/08/2013 | 0.52 | 0.06 | 37 | 0.05 | | 0.35 | 5.24 | 0.04 | 15.17 |
| 28/08/2013 | 0.52 | 0.06 | 37 | 0.05 | | 0.35 | 5.22 | 0.04 | 15.17 |
| 29/08/2013 | 0.52 | 0.06 | 37 | 0.05 | | 0.35 | 5.22 | 0.04 | 15.17 |
| 30/08/2013 | 0.52 | 0.06 | 37 | 0.05 | ... | 0.35 | 5.22 | 0.04 | 15.18 |

column indicates the period. The entry in each cell is the growth rate of that particular share and for that period.

Table 4.2: Daily Growth Rates for various shares

| Period | AADS | ACI | AGA | ALW | ··· | TBL | TOTAL | TRANSOL | UNIL |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0.125 | 0 | 0 | | 0 | 0 | 0 | 0.011396 |
| 2 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0.126761 |
| 3 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0.0025 |
| 5 | 0 | 0 | 0 | 0 | | 0 | 0.001389 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0.002494 |
| 7 | 0 | 0 | 0 | 0 | | 0 | 0.001387 | 0 | 0.002488 |
| 8 | 0 | 0 | 0 | 0 | | 0 | 0.052632 | 0 | 0.01737 |
| ⋮ | | | | | | | | ⋮ | |
| 1203 | 0 | 0 | 0 | 0 | | 0 | -0.0019 | 0 | 0 |
| 1204 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |
| 1205 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0.00132 |
| 1206 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |
| 1207 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |
| 1208 | 0 | 0 | 0 | 0 | | 0 | -0.00382 | 0 | 0 |
| 1209 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |
| 1210 | 0 | 0 | 0 | 0 | ··· | 0 | 0 | 0 | 0.000659 |

Just as the Daily Stock Prices, there were 1,210 observations on the Daily Growth

Rates for each Stock. As a result Monthly Averages for each month were computed to abridge the data for convenient data analysis instead of Daily Growth Rates. Below is table 4.3 showing the computed Monthly Averages for each month. The first row shows the various shares while the first column indicates the various months. Each entry reflects the monthly average of that particular share for that particular month.

Table 4.3: Monthly Averages for the various shares

| Year | **Period (Month)** | **AADS** | **ACI** | **AGA** | **ALW** | … | **TBL** | **TOTAL** | **TRANSOL** | **UNIL** |
|---|---|---|---|---|---|---|---|---|---|---|
| 2008 | SEPT | 0 | 0.00625 | 0 | 0 | | 0 | 0.00277 | 0 | 0.012878 |
| | OCT | 0 | 0.005051 | 0 | 0 | | 0 | 0 | 0 | 0.000807 |
| | NOV | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | -0.00198 |
| | DEC | 0 | 0 | 0 | -0.00085 | | 0 | 0 | 0 | -0.00478 |
| 2009 | JAN | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |
| | FEB | -0.00733 | 0 | 0 | 0 | | 0 | 0 | 0 | -0.005 |
| | MAR | 0 | 0 | 0 | -0.00475 | | 0 | 0 | 0 | -0.00132 |
| ⋮ | | | | | | | | | | ⋮ |
| 2013 | JAN | 0 | -0.00649 | 0 | 0.009091 | | 0 | -0.00152 | 0 | 0.007494 |
| | FEB | 0 | 0 | 0 | 0.0154476 | | 0 | -0.00152 | 0 | 0.003025 |
| | MAR | 0 | 0 | 0 | -0.00658 | | 0 | 0.007155 | 0 | 0.007577 |
| | APR | 0 | 0 | 0 | -0.0068 | | 0.00595 | 0.001043 | 0 | 0.00498 |
| | MAY | 0 | 0 | 0 | 0.003968 | | 0 | -0.000278 | 0 | 0.003506 |
| | JUN | 0 | 0 | 0 | 0 | | 0 | 0.021527 | 0 | 0.002146 |
| | JUL | 0 | 0 | 0 | -0.00758 | | 0 | 0.000294 | 0 | 4.17E-07 |
| | AUG | 0 | 0 | 0 | 0 | | 0 | -0.04188 | 0 | 0.000252 |

The Monthly Averages will now represent our data points. In order to obtain an estimate for expected return on our investments or expected profits, we will solve for the mean of our data points which are the Monthly Averages. We will also calculate for the Average share price for all individual shares and use the answer as the cost per share of the various shares. Table 4.4 shows the Mean (Expected Profits) of the Monthly Averages for the various shares.

As shown in table 4.4, stocks with * by their names have positive figures as their mean whiles those without the * have negative mean. A share with a positive

Table 4.4: Mean of the monthly averages and cost per share of listed shares

| STOCKS | MEAN (EXPECTED PROFITS) | COST PER SHARE | | STOCKS | MEAN (EXPECTED PROFITS) | COST PER SHARE |
|---|---|---|---|---|---|---|
| AADS * | 0.000378232 | 0.44 | | GWEB | -0.0024441 | 0.04 |
| ACI * | 0.162880206 | 0.91 | | HFC | -0.002208 | 0.65 |
| AGA * | 0.000174669 | 33.5 | | MLC | -0.0016782 | 0.29 |
| ALW | -0.00161724 | 0.34 | | PBC | -0.0022632 | 0.2 |
| AYRTN * | 0.000190122 | 0.17 | | PKL | -0.0024892 | 0.06 |
| BOPP * | 0.001225053 | 3.85 | | PZC | -0.0028834 | 0.57 |
| CAL * | 0.000433537 | 1.15 | | SCB | -0.0023136 | 14.1 |
| CLYD | -0.000531857 | 0.06 | | SCB_PREF | -0.002153 | 0.52 |
| CMLT * | 0.00007979 | 0.16 | | SG_SSB | -0.0031598 | 0.55 |
| CPC * | 0.005633994 | 0.04 | | SIC | -0.0051935 | 0.38 |
| EBG | -0.000364815 | 4.6 | | SPL | -0.001167 | 0.05 |
| ETI * | 0.000118879 | 0.33 | | SWL | -0.0026335 | 0.02 |
| FML * | 0.000503319 | 5.42 | | TBL | -0.0030351 | 0.35 |
| GCB * | 0.001687682 | 3.34 | | TOTAL | -0.0015584 | 5.22 |
| GGBL * | 0.000626731 | 3.63 | | TRANSOL | -0.0031612 | 0.04 |
| GOIL * | 0.001051179 | 0.63 | | UNIL | -0.0010587 | 15.18 |
| GSR | -0.000041582 | 2.93 | | | | |



mean indicates that share is expected to yield positive returns (profits) while those with negative means those shares will decline in value over a period.

With the main objective of this work being to maximize the return on our investments with the help of the knapsack concept, we will now consider stocks with positive expected returns. These profitable stocks are shown in the table below with their respective unit cost as well as expected rate of returns per share.

Table 4.5 captures profitable shares as well as their respective prices. For example, an ADDS share, costing GHS0.44 is expected to earn GHS0.000378232 in value if one invest in that particular share. The various expected profits for the individual

Table 4.5: Profitable Shares with their respective share prices

| Stocks | Price Per Share (GHS) | Expected Profit (GHS) |
|---|---|---|
| AADS | 0.44 | 0.000378232 |
| ACI | 0.91 | 0.162880206 |
| AGA | 33.5 | 0.00017467 |
| AYRTN | 0.17 | 0.000190122 |
| BOPP | 3.85 | 0.001225053 |
| CAL | 1.15 | 0.000433537 |
| CMLT | 0.16 | 0.00007979 |
| CPC | 0.04 | 0.005633994 |
| ETI | 0.33 | 0.000118879 |
| FML | 5.42 | 0.000503319 |
| GCB | 3.34 | 0.001687682 |
| GGBL | 3.63 | 0.000626731 |
| GOIL | 0.63 | 0.001051179 |

shares will form the coefficients of the objective function of our problem whiles their corresponding share prices will be the coefficients of the constraint.

## 4.2 Formulation of Problem

There are millions of shares of these stocks which are traded daily on the stock exchange. The amount of money you need to invest on the GSE depends on the prices of shares you select. Shares are usually traded in batches or round lots of 100. Where the price of a particular stock is high, an investor can contact a broker to buy fewer than 100 shares or what is commonly referred to as odd lots. Depending on the brokerage firm, the minimum lump sum investment one can make is GHS 10.00 and these funds are invested mainly on listed shares of the GSE. For this paper, we shall adopt the GHS10.00 as the amount to be invested.

## 4.3 Objective Function

The objective function which seeks to maximize the Return of Investment (R), will be equated to the summation of the expected returns of the various individual shares. The coefficients of the objective function are derived from the expected profits indicated in the table.

This is given by:

$R = 0.000378S_1 + 0.162880S_2 + 0.000175S_3 + 0.000190S_4 + 0.001225S_5 + 0.000434S_6 + 0.000080S_7 + 0.005634S_8 + 0.000119S_9 + 0.000503S_{10} + 0.001688S_{11} + 0.000627S_{12} + 0.001051S_{13}$



## 4.4 Constraint

The constraint consist of the summation of the individual share price which is bounded by the minimum amount one can start an investment with on the GSE. These coefficients are indicated in the table as the price per share

$0.44S_1 + 0.91S_2 + 33.50S_3 + 0.17S_4 + 3.85S_5 + 1.15S_6 + 0.16S_7 + 0.04S_8 + 0.33S_9 + 5.42S_{10} + 3.34S_{11} + 3.63S_{12} + 0.63S_{13} \leq 10.$

## 4.5 Computational Procedure

A matlab code was written using the Harowitz-Sahni method. The code was run on a Toshiba Windows 7 PC rated 3.8 per the Windows Experience Index. It has a processor of Intel(R) Core(TM) i3 CPU M380 @ 2.53GHz 2.53GHz with a RAM of 32 – bit Operating System.

## 4.6 Results

The optimal solution obtained for the objective function is GHS 0.1725 and the portfolio consists of (1,1,0,1,0,1,1,1,1,0,1,0,1). This implies every investment of

GHS 10.00 will yield a return of 1.725%.

## 4.7 Discussion

The solution set $(S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9, S_{10}, S_{11}, S_{12}, S13)$ with the optimal value is (1,1,0,1,0,1,1,1,1,0,1,0,1). All entries with 1 indicate that particular stock is part of the optimal solution. All entries with 0, indicates that particular stock was not part of the optimal portfolio. The optimal portfolio consists of the following shares AADS, ACI, AYRTN, CAL, CMLT, CPC, ETI, GCB and GOIL. The selected shares contributed 0.000378, 0.162880, 0.000190, 0.000434, 0.000078, 0.005634, 0.000119, 0.001688 and 0.001051 respectively to obtain the optimal value. Other stocks like AGA could not be included in the optimal solution because of its low profit to cost ratio.

# Chapter 5

# CONCLUSIONS AND RECOMMENDATIONS

This thesis has modelled the selection of stocks in a portfolio as a 0-1 knapsack programming problem. We made use of the branch-and-bound algorithm of The Horowitz–Shani to solve our problem. This research paper focused on the use of the Knapsack problem in selecting a portfolio from the Ghana Stock Exchange given GH ¢10.00 to invest. This idea can be used to solve any real life problem which can be formulated into a 0 – 1 knapsack problem

## 5.1 CONCLUSIONS

This thesis seeks to maximize the return on a GH ¢10.00 investment on the Ghana Stock Exchange which has been modelled as a knapsack problem. The solution was obtained using the branch-and-bound algorithm of The Horowitz-Sahni. It was observed that the solution that gave maximum achievable value $(S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9, S_{10}, S_{11}, S_{12}\ and\ S_{13})$ was (1,1,0,1,0,1,1,1,1,0,1,0,1). The optimal portfolio consists of the following shares AADS, ACI, AYRTN, CAL, CMLT, CPC, ETI, GCB and GOIL. This means that an investor could choose a portfolio costing GHS 10.00 and make a return of 1.725%.

## 5.2 RECOMMENDATIONS.

The use of a scientific approach in computation gives a systematic and transparent solution as compared with an arbitrary method. Making use of the Knapsack model for portfolio selection with the aim of maximizing returns gives a better result as compared with the choosing a portfolio using just earnings per share. Investors and Fund Managers will benefit from the proposed approach for portfolio construction. I hereby recommend that the Knapsack problem model should be adopted by Investors, Fund Managers, Brokerage Firms and other Financial Market Players.

# REFERENCE


[1] H. Markowitz, Portfolio selection, Journal of Finance 7 (1952) 77-91.

[2] W.F. Sharpe, A linear programming approximation for a mutual fund portfolio selection, Management Science 13 (1967) 499-510.

[3] W.F. Sharpe, A linear programming approximation for the general portfolio analysis problem. Journal of Financial and Quantitative Analysis, December (1971) 1263-1275.

[4] B.K. Stone, A linear programming formulation of the general portfolio selection problem, Journal of Financial and Quantitative Analysis, September (1973) 621-636.

[5] H. Takehara, An application of the interior point algorithm for large scale optimization in Ⓡnance, Proceedings Third RAMP Symposium, Journal of the Operational Research Society of Japan, 1991, pp. 43-52.

[6] H. Markowitz, P. Todd, G. Xu, Y. Yamane, Computation of mean-semivariance efficient sets by the Critical Line Algorithm, Annals of Operations Research 45 (1993) 307-317.

[7] H. Konno, K. Suzuki, A fast algorithm of solving large scale mean-variance models by compact factorization of covariance matrices, Journal of the Operational Research Society of Japan 33 (1992) 93-104.

[8] H. Konno, H. Yamazaki, Mean-absolute deviation portfolio optimization model and its application to Tokyo Stock Market, Management Science 37 (5) (1991) 519-531.

[9] S.A. Zenios, P. Kang, Mean-absolute deviation portfolio optimization for mortgage-backed securities, Annals of Operations Research 45 (1993) 433-450.

[10] M.G. Speranza, Linear programming models for portfolio optimization, Finance 14 (1993) 107-123.

[11] C.D. Feinstein, M.N. Thapa, Notes: A reformulation of a mean-absolute

deviation portfolio optimization model, Management Science 39 (12) (1993) 1552-1553.

[12] M.G. Speranza, A heuristic algorithm for a portfolio optimization model applied to the Milan Stock Market, Computers and Operations Research 23 (1996) 433-441.

[13] R. Mansini, M.G. Speranza, On selecting a portfolio with fixed costs and minimum transaction lots, Report no. 134, Dip. Metodi Quantitativi, University of Brescia, 1997.

[14] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-completeness. Freeman, San Francisco, 1979.

[15] R. Mansini, Mixed Integer Linear Programming Models for Financial Problems: Analysis, Algorithms and Com- putational Results, Ph.D. Thesis, University of Bergamo, 1997.

[16] Arkin, E. M., Khuller, S. and Mitchell, J. S. B. (1993). Geometric knapsack problems. http://www.springerlink.com/content/g007w81p153h3326

[17] Geir, D. (1997). An introduction to convexity, polyhedral theory and combinatorial optimization. University of Oslo, Department of Informatics

[18] Lueker, G. S. (1995). Average-Case Analysis of Off-Line and On-Line Knapsack Problems. Journal of Algorithms Volume 29, Issue 2, Pages 277-305

[19] Ghoseiri K., Szidaroyszky, F. and Asgharpour, M. J. (2004). A multi-objective train scheduling model and solution. Transportation research part B: Methodological 38, 927.

[20] Granmo O. C., Oommen, B. J. Myrer, S. A. and Olsen, M. G. (2007). Learning automated-based solutions to the nonlinear fractional knapsack problem with applications to optimal resource allocation. IEE Transactions on systems, man and cybernetics, part B (cybernetics), 37 n1, 166-175.

[21] Gutierrez and Maria Talia (2007). Lifting general integer programs. Kansas State University Masters thesis

[22] Haghani, A. and Shafali, Y. (2002). Bus maintenance systems and scheduling:

model formulations and solutions. Transportation research part A: Policy and Practice, 36, 453.

[23] Higgins, A., Kozan, E. and Ferreira, L. (1996). Optimal scheduling of train on a single line track. Transportation research part B: Methodological, 38, 927.

[24] Hillier, F. S. and Lieberman, G. J. (2001). Introduction to Operations research. McGraw- Hill, New York 576-581

[25] Horowi, E. and Sahni (1974). Computing partitions with applications to knapsack problems. Journal of ACM21, 277-292

[26] Bryce, H. (2007). Finding adjacent facet-defining inequalities. Kansas State University Masters thesis.

[27] Tauhidul, I. M. (2009). Approximation algorithms for minimum knapsack problem Master's degree Thesis, UNIVERSITY OF LETHBRIDGE

[28] Kalai, R. and Vanderpooten, D. (2006). Lexicographic $\alpha$-Robust Knapsack Problem http://ieeexplore.ieee.org/xpl/freeabs Michel, S., Perrot, N. and Vanderbeck, F. (2009). Knapsack problems with setups http://ieeexplore.ieee.org/xpl/freeabs

[29] Karp, R. M. (1972). Reducibility among combinatorial problems. Complexity of computer computations; Plenum Press New York 85-103

[30] Zhanhong, L. S., Wang, Y. Y. and Wei, L. (2010). Streaming Media Caching Model Based on Knapsack Problem. Journal of Networks, Vol 6, No 9 (2011), 1379-1386.

[31] Xin, L. and Denggu, F. (2004). Quantum algorithm analysis of knapsack problem. JOURNAL OF BEIJING UNIVERSITY OF AERONAUTICS AND A, V 30(11)

[32] Garg, L. M. and Gupta, S. (2009). An Improved Genetic Algorithm Based on Adaptive Repair Operator for Solving the Knapsack Problem. Journal of Computer Science, volume 5, issue 8, page 544-547

[33] Mattfeld, D. C. and Kopfer, H. (2003). Terminal operations management in vehicle transshipment. Transportation research part A: Policy and Practice, 37,

435.

[34] Xuan, M. A. (2009). Artificial fish swarm algorithm for multiple knapsack problem.

Journal of Computer Applications 2010, 30(2) 469-471

[35] Hristakeva, M. and Shrestha, D. (2011). Solving the 0-1 Knapsack Problem with Genetic Algorithms. http://freetechebooks.com/file-2011/knapsack-problem

[36] Hristakeva, M. and Shrestha, D. (2005). Different Approaches to Solve the 0/1 Knapsack Problem. http://micsymposium.org/mics 2005/papers/paper102.

[37] Michel, S., Perrot, N. and Vanderbeck, F. (2009). Knapsack problems with setups http://ieeexplore.ieee.org/xpl/freeabs

[38] Nemhauser, G. L. and Wolsey, L. A. (1998). Integer and Combinatorial Optimization. John Wiley and Sons, New York.

[39] On Stochastic Bilevel Programming Problem with Knapsack Constraints. http://www.kosuch.eu/stefanie/veroeffentlichungen

[40] Ofori, O. E. (2009). Optimal resource Allocation Using Knapsack Problems: A case Study of Television Advertisements at GTV. Master's degree thesis, KNUST

[41] Kohli, R. and Krishnamurti, R. (1992). A total-value greedy heuristic for the integer knapsack problem. Operations Research Letters Volume 12, Issue 2

[42] Mansini, R. and Speranza, M. G. (2009. An Exact Algorithm for the Multidimensional Knapsack Problem http://ideas.repec.org/a/eee/ejores/v196y2009i3p909-918

[43] Shang, R., Ma, W. and Zhang, W (2006). Immune Clonal MO Algorithm for 0/1 Knapsack Problems. Lecture Notes in Computer Science, 2006, Volume 4221/2006, 870- 878.

[44] Kosuch, S. and Lisser, A. (2009). On two-stage stochastic knapsack problems. Discrete Applied Mathematics Volume 159, Issue 16

[45] Khan, S., LI, K. F., Manning, E. G. and Akbar, M. D. M. (2002).SOLVING

THE KNAPSACK PROBLEM FOR ADAPTIVE MULTIMEDIA SYSTEMS. http://studia.complexica.net/Art/RI020108

[46] Srisuwannapa, C. and Charnsethikul, P. ( 2007). An Exact Algorithm for the Unbounded Knapsack Problem with Minimizing Maximum Processing Time. Journal of Computer Science, 3: 138-143.

[47] Kosuch, S.(2010). An Ant Colony Optimization Algorithm for theTwo-Stage Knapsack Problem. http://www.kosuch.eu/stefanie

[48] Kosuch, S., Letournel, M. and Lisser, A. (2009). On a Stochastic Knapsack Problem. Laboratoire de recherche en Informatique, Universite Paris Sud 91405 Orsay Cedex.

[49] Stefanie Kosuch, S., Le Bodic, P., Leung, J. and Lisser, A. (2009).

[50] Tomastik, R. N. (1993). The facet ascending algorithm for integer programming problems. Proceedings on the 32nd IEEE conference on decision and control, 3, 2880-2884.

[51] Boryczka, U. (2006). The ihfluence of Trial representation in ACO for good results in MKP. From Proceeding (505) Advances in Computer Science and Technology

[52] Pferschy, U., Pisinger, D. and Woeginger, G. J. (1995). Simple but efficient approaches for the collapsing knapsack problem. Journals of Operations Research.

[53] Ravi, V. G. R. (2003). Chance Constrained Knapsack Problem with Random Item Sizes. http://www.columbia.edu/ vg2277/stoch knapsack

[54] Yan, S. and Chen, H. L. (2002). A scheduling model and a solution algorithm for inter- city bus carriers. Transportation research part A: Policy & Practice, 36, 805.

[55] Zhou, Y. and Naroditskiy, V. (2008). Algorithm for Stochastic MultipleChoice Knapsack Problem and Application to Keywords Bidding. http://research.yahoo.com /workshops/troa-2008/papers/submission

[56]. Volgenant, A. and Zoon, J. A. (1990). An Improved Heuristic for

Multidimensional 0-1 Knapsack Problems. Journal of the Operational Research Society 41, 963–970.

[57]. Aggarwal and Hartline (2006). Knapsack Auctions. www.research.microsoft.com

[58]. Caprara, A., Pisinger, D. and Toth, P.(2007). Exact Solution of the Quadratic Knapsack Problem. Journals of Operations Research 55:1001-1021

[59]. Ram, B. and Sarin, S. (1988). An Algorithm for the 0-1 Equality Knapsack Problem. Journal of the Operational Research Society 39, 1045–1049.

[60]. Bertsimas D., Darnell, C. and Soucy, R. (1999). Portfolio construction through mixed- integer programming at Grantham, Mayo, Van Otterloo and Company. Interfaces 29, n1, Jan. – Feb. 1999, 49-66.

[61]. Sung-Ho, C. (1998). Tactical-Level Resource allocation procedure for the hotel industry. Journals of Texas A & M Industrial and Systems Engineering.

[62]. Christopher, C. A. (1998). Solving Geometric Knapsack Problems using Tabu Search Heuristics. http://oai.dtic.mil/oai/oai?verb=getRecord metadataPrefix=html identifier

[63] Claessens T., Van Dijk, N. and Zwaneveld P.J. (1998). Coast optimal allocation of rail passenger lines. European Journal of Operational Research 110, 474

[64] Das, S. and Ghosh, D. (2003). Binary knapsack problems with random budgets . Journal of the Operational Research Society

[65] Dizdar, D., Gershkov, A. and Moldovanu, B. (2010). Revenue maximization in the dynamic knapsack problem. Theoretical Economics 6 (2011), 157–184

[66] Der-Chyuan, L. and Chin-Chen, C. (1997). International Journal of High Speed Computing (IJHSC). Change in behaviour of outputs generated on varying the crossover and mutation rates.

[67] Doprado M. F. and Marcos, N. A.(2007).The constrained compartmentalised knapsack problem. Journals of Computers and Operations research

[68] Easton, K., Nemhauser, G., and Trick, M. (2003). Solving the

travelling tournament problem, a combined integer programming and constraint programming approach (practice and theory of automated timetabling IV). 4th International conference, PATAT 2002, Lecture notes in computer science vol. 2740, pp. 100-109

[69] Lawler, E. L. (1977). Fast approximation algorithms for knapsack problems. cs, pp.206-213 18th Annual Symposium on Foundations of Computer Science.

[70] Hadjiconstantinou, E. and Christofides, N. (2010). An exact algorithm for general, orthogonal, two-dimensional knapsack problems. European Journal of Operational Research

[71] Essandoh, K (2012). Modelling site development for garbage disposal as a 0-1 knapsack problem. Thesis.

[72] Essumah, J. (2012). Modelling the maximum revenue from the repairs of cars of an auto workshop in Ghana. Thesis.