

**STAFF ASSIGNMENT PROBLEM**

**(CASE STUDY AT MAMPONG-AKUAPEM PRESBY SENIOR HIGH SCHOOL)**

**KNUST**  
BY

**KORSORKU SIMON**

**(PG4067210)**

**THESIS SUBMITTED TO THE DEPARTMENT OF MATHEMATICS  
KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR  
THE DEGREE OF  
MASTER OF SCIENCE IN INDUSTRIAL MATHEMATICS  
INSTITUTE OF DISTANCE LEARNING**

**JUNE 2012**

## DECLARATION

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

# KNUST

KORSORKU SIMON, PG4067210

.....

.....

Student's Name & ID

Signature

Date

Certified By

Dr. S. K. Amponsah

.....

.....

Supervisor's Name

Signature

Date

Certified By

Dr. F. K. Darkwa

.....

.....

Head of Department's Name

Signature

Date



## ABSTRACT

Matching highly skilled people to available positions is a high-stakes task that requires careful consideration by experienced resource managers. A wrong decision may result in significant loss of value due to understaffing, under-qualification or over-qualification of assigned personnel, and high turnover of poorly matched workers. While the importance of quality matching is clear, dealing with pools of hundreds of jobs and resources in a dynamic market generates a significant amount of pressure to make decisions rapidly. We present a novel solution designed to bridge the gap between the need for high-quality matches and the need for timeliness. By applying mathematical programming, we are able to deal successfully with the complex constraints encountered in the field and reach near-optimal assignments that take into account all resources and positions in the pool. The considerations include constraints on job role, skill level, geographical location, language, potential retraining, and many more. Constraints are applied at both the individual and team levels.

This thesis models staff subject assignment problem as an assignment problem. The model developed could be adopted for any problem that can be modelled as an assignment problem.

## ACKNOWLEDGEMENT

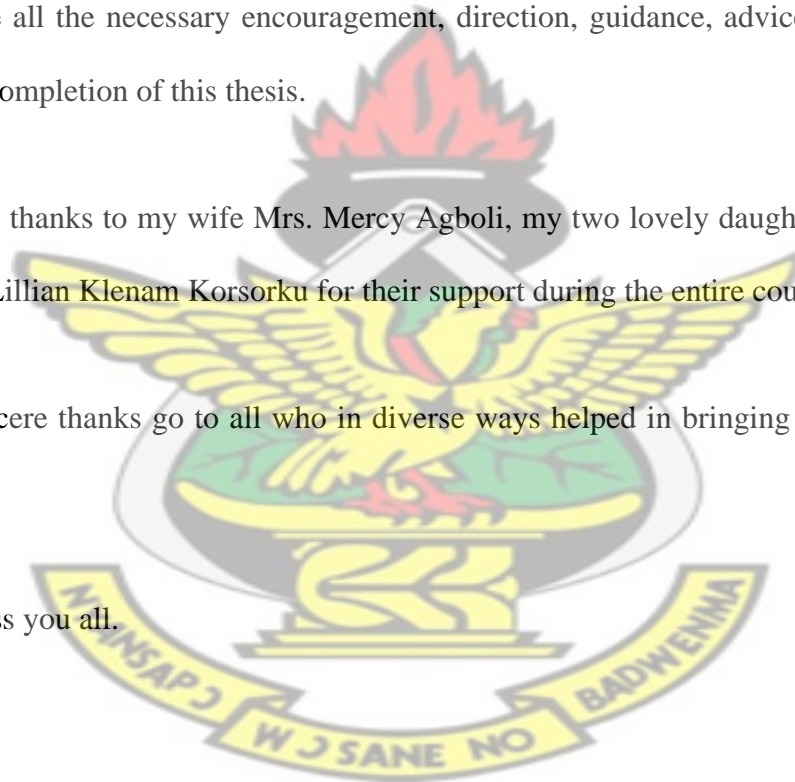
I would like to give thanks to the Almighty God for granting me the strength and knowledge for understanding this course and the completion of this write-up.

I am very grateful to my supervisor, Dr. Samuel K. Amponsah of the Department of mathematics, who painstakingly read through every line of the text and offered through his rich experience all the necessary encouragement, direction, guidance, advice and correction for the timely completion of this thesis.

I will also give thanks to my wife Mrs. Mercy Agboli, my two lovely daughters Pearl Yayra Korsorku and Lillian Klenam Korsorku for their support 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.



# TABLE OF CONTENTS

DECLARATION	i
ABSTRACT	ii
ACKNOWLEDGEMENT	iii
TABLE OF CONTENTS	iv
LIST OF TABLES	vi
CHAPTER ONE	1
1.0 INTRODUCTION	1
1.1 Background of Study	3
1.2 Problem Statement	5
1.3 Objectives	6
1.4 Methodology	6
1.5 Justification	7
1.6 Limitations	9
1.7 Organization of the Thesis	9
1.8 Summary	9
CHAPTER TWO	11
LITERATURE REVIEW	11

CHAPTER THREE	35
METHODOLOGY	35
3.0 Introduction	35
3.1 Graph Theory	36
3.1.1 Graphs for the Assignment Problem	37
3.1.2 Verification of the Assignment Algorithm	38
3.1.3 Maximal Matching's Using the Hungarian Algorithm	41
3.1.4 Flow Chart for Assignment Problem	47
3.1.5 Hungarian Method: Algorithm	48
CHAPTER FOUR	49
DATA COLLECTION AND ANALYSIS	49
4.0 Introduction	49
4.1 Data Collection and Analysis	49
CHAPTER FIVE	57
CONCLUSIONS AND RECOMMENDATIONS	57
5.0 Introduction	57
5.1 Conclusions	57
5.2 Recommendations	57
REFERENCES	58
APPENDIX_1–FORTRAN 90 CODES FOR THE HUNGARIAN ALGORITHM	

## LIST OF TABLES

Table 4.1	50
Table 4.2	51
Table 4.3	53
Table 4.4	53
Table 4.5	54
Table 4.6	54
Table 4.7	55

KNUST





# CHAPTER ONE

## 1.0 INTRODUCTION

Employees are the most important asset of any technology-based company. This statement is not a mere slogan, but a genuine business reality that requires careful consideration at all management levels of a company. While this reality has been recognized for a long time, only recently have rigorous processes, backed by automation, become central in reaching workforce-related decisions.

One of the main reasons for this is the fact that professional workers, being humans, are complex entities. They have individual skills, interests, expectations, and limitations. They may live in a particular area, have family-related constraints, prefer working solo, or function best as team players. They may be more or less susceptible to pressure, easy or difficult to retrain, and motivated by completely diverse factors. Most significantly, it is perceived that human professionals cannot possibly be described as a mere set of attributes, no matter how large the set. For example, most resumes—formal documents designed to best describe the aspects of people relevant to their hiring—contain lengthy textual descriptions rather than a structured list of attributes and values. Summarized eloquently, it is often maintained that “people are not parts.” While it is true that people are not parts, the situation still exists in which a large number of professionals must be matched and assigned to a similarly large number of demanding jobs. In fact, this problem lies at the heart of the execution phase of the workforce management (WM) cycle (Cerulli et al., 1992). The problem applies to many different business cases in the technology industry, including assigning service professionals to short-term maintenance



tasks (Lesaint et al., 2004), team-building for contracted projects (Kliem and Anderson, 1996), maintaining staff with multiple skills (Eitzen et al., 2004), and more.

In all of these cases, the consequences of failing to find the best assignments for the jobs are extremely severe. Less-than-optimal assignments can be manifested in three general forms: underqualified professionals assigned to highly demanding jobs, overqualified professionals assigned to less-demanding jobs, and a total number of assignments smaller than the maximum achievable. An underqualified assignment may result in the need to reperform the job without compensation, costly onsite training, and customer dissatisfaction with the job, eventual loss of this customer, and loss of referrals from the customer. In addition, qualification may refer to various attributes, not necessarily the professional level of the worker. For example, an underqualification may be a travel distance that is too long, with direct travel costs being incurred by the provider. The costs of an overqualified assignment may relate directly to the unrecovered high salary of the professional or indirectly to the loss of a more profitable job assignment for the employee, employee dissatisfaction, and eventual employee attrition. A less-than-optimal number of assignments may result in loss of revenue from unassigned jobs, increased costs from subcontracting external providers for the unassigned jobs, and the general dissatisfaction of the customers ordering the unassigned jobs.

In this chapter, an overview of assignment problem would be given; a brief description of the problem statement of the thesis is also presented together with the objectives, the methodology, the justification and the organization of the thesis.

## 1.1 BACKGROUND OF STUDY

The assignment problem is a special type of linear programming problem where assignees are being assigned to perform tasks. For example, the assignees might be employees who need to be given work assignments. Assigning people to jobs is a common application of the assignment problem. However, the assignees need not be people. They also could be machines, or vehicles, or plants, or even time slots to be assigned tasks. To fit the definition of an assignment problem, these kinds of applications need to be formulated in a way that satisfies the following assumptions.

- (i) The number of assignees and the number of tasks are the same. (This number is denoted by  $n$ ).
- (ii) Each assignee is to be assigned to exactly one task.
- (iii) Each task is to be performed by exactly one assignee.
- (iv) There is a cost  $c_{ij}$  associated with assignee  $i$  ( $i = 1, 2, \dots, n$ ) performing task  $j$  ( $j = 1, 2, \dots, n$ ).
- (v) The objective is to determine how all  $n$  assignments should be made to minimize the total cost.

Any problem satisfying all these assumptions can be solved extremely efficiently by algorithms designed specifically for assignment problems.

The first three assumptions are fairly restrictive. Many potential applications do not quite satisfy these assumptions. However, it often is possible to reformulate the problem to make it fit. For example, dummy assignees or dummy tasks frequently can be used for this purpose.

Assignment model comes under the class of linear programming model, which looks alike with transportation model with an objective function of minimizing the time or cost of manufacturing the products by allocating one job to one machine or one machine to one job or one destination to one origin or one origin to one destination only. Basically assignment model is a minimization model. If we want to maximize the objective function, then there are two methods. One is to subtract all the elements of the matrix from the highest element in the matrix or to multiply the entire matrix by -1 and continue with the procedure. For solving the assignment problem we use Assignment technique or Hungarian method or Flood's technique. All are one and the same. Above, it is mentioned that one origin is to be assigned to one destination. This feature implies the existence of two specific characteristics in linear programming problems, which when present, give rise to an assignment problem. The first one being the pay off matrix for a given problem is a square matrix and the second is the optimum solution (or any solution with given constraints) for the problem is such that there can be one and only one assignment in a given row or column of the given payoff matrix.

The basic objective of an assignment problem is to assign  $n$  number of resources to  $n$  number of activities so as to minimize the total cost or to maximize the total profit of allocation in such a way that the measure of effectiveness is optimized. The problem of assignment arises because available resources such as men, machines, etc., have varying degree of efficiency for performing different activities such as job. Therefore cost, profit or time for performing the different activities is different. Hence the problem is how should the assignment be made so as to optimize (maximize or minimize) the given objective. The assignment model can be applied in many decision-making processes like

determining optimum processing time in machine operators and jobs, effectiveness of teachers and subjects, designing of good plant input, etc. This technique is found suitable for routing travelling salesman to minimize the total travelling cost, or to maximize the sales.

The assignment problem is a special case of the transportation problem where all sources and demand are equal to 1. The assignment problems are of two types: (1) balanced and (2) unbalanced. If the number of row is equal to the number of columns or if the given problem is a square matrix, the problem is termed as a balanced assignment problem. If the given problem is not a square matrix, the problem is termed as an unbalanced assignment problem. If the problem is an unbalanced one, we add dummy rows/columns as required so that the matrix becomes a square matrix or a balanced one. The cost or time values for the dummy cells are assumed as zero.

## 1.2 PROBLEM STATEMENT

The specific form of problem that this thesis seeks to solve is to mathematically model a company's staff job placement problem as an assignment problem and solve the problem.

Assignment problem is a special type of transportation problem which is also a resource allocation problem. Here we have  $n$  jobs to perform with  $n$  persons and the problem is how to distribute the job to the different persons involved. Depending on the intrinsic capacity or merit or potential of the individual, he will be able to accomplish the task in different times. Then the objective function in assigning the different jobs to different persons is to find the optimal assignment that will minimize the total time taken to finish

all jobs by the individuals. For example, we have four different building activities say, construction of a hotel, a theatre, a hospital and a multi-storied building and there are four contractors competing for these jobs. Each contractor has to be assigned only one job. The allocation should aim to minimize the total time taken to complete the construction of all four activities after assigning only one job to one individual. In fact, there are four permutations possible for allocating four jobs to four contractors. We have twenty-four possible ways and it is tiresome to list all the possible ways and find the best one. If we have more jobs to be allocated, it is even difficult to list out the different permutations of allocations, than to speak of choosing the best combinations.

Assignment problems are widely used in financial decision making, with examples being assignment of employees to machines, assignment of operators to jobs, allocation of machines for optimum utilization of space, assigning salesmen to different sales areas, and assigning clerks to various counters. In all the cases, the objective is to minimize the total time and cost or otherwise maximize the sales and returns.

### **1.3 OBJECTIVES**

The goal of this research is to mathematically model the staff job placement problem of a company (service industry) as an assignment problem and solve the problem.

### **1.4 METHODOLOGY**

In our methodology, we shall use the Hungarian approach in solving our problem. First, the algorithm is presented along with relevant examples.



## 1.5 JUSTIFICATION

The usual way of solving the general assignment problem presented above by most institutions is to manually examine the full list of jobs in some predefined order and for each job find a corresponding shortlist of best-fitting candidates, and then assign one of those candidates to the job. (An equivalent option is to look at the full list of professionals in a predefined order, find a shortlist of best-fitting jobs for each professional, and then assign the professional to one of those jobs.) This procedure is simple and can be accomplished by a human Resource Deployment Professional (RDP), because at any one time the actual fitting procedure looks only at a single job and a shortlist of professionals. As part of this procedure, the RDP may use search tools to search for an employee with characteristics required by the job, provided that relevant data for all professionals is stored in some database. However, the procedure has the following significant drawbacks:

- ( i ) It is tedious, repetitive, and time-consuming.
- ( ii ) Since the shortlist of matches is not prioritized within itself, it requires further manual work to rank-order the individuals in the shortlist and is thus likely to result in a suboptimal choice, even for the single job currently considered.
- ( iii ) The first job considered will likely be assigned the best-found professional for the job (a greedy policy), even though that professional may be better suited to other jobs that have not yet been considered. This may lead to fewer assignments to jobs because the other jobs may not find another match, while alternative professionals may exist for the current job. It can also lead to possible over qualification of the professional for the assigned job.



( iv) Competition among jobs considered (or owned) by different RDPs is even less likely to be resolved fairly, because each RDP sees and applies only his or her own criteria, and there is no mechanism for finding a fair and optimal assignment among all RDPs.

When the number of available jobs and professionals is large, say a few dozen or more, it becomes impossible to find the best matches manually. This is true even when the matching criteria are stated correctly and the RDPs are motivated to seek a global best solution. The reason for this is that the optimization problem is known mathematically to be NP-hard, which means that beyond a certain number of jobs, an exponentially large number of comparisons between different candidates must be done in order to reach the optimal assignment.

(vi) Only the most simplistic types of matching rules, or constraints, can be considered by human RDPs. One example of a simple rule may be exact matches on several searched attributes, such as skills, availability, and pay rates. However, even a simple matching rule that requires, e.g., a short travel distance between work and the person's location is difficult to enforce manually, because this distance must be recalculated for any job-candidate pair. Finding a good solution that complies with rules that are inherently complex (for example, team-building rules) is far beyond the capacity of a human RDP.

(vii) In searching for candidates who possess a number of desired attributes, all attributes are viewed as having the same importance. When some attributes are of higher importance than others, finding the best matches must be achieved manually by first searching for candidates with the most important attribute, then reducing the list to those also having the next important attribute, and so on. In addition to the slowness of this

procedure, it will likely miss a professional with many of the less-important required attributes who lacks only one of the more-important attributes.

Given the above drawbacks, the potential for large amounts of data, and the need for a short response time, an automated procedure to optimize the set of assignments could offer a significant benefit, hence the reason for solving the assignment problem.

## **1.6 LIMITATIONS OF THE STUDY**

Due to limited availability of funds and time constraints, this study only focuses on the Mampong-Akuapem Presby Senior High School.

## **1.7 ORGANIZATION OF THE THESIS**

The thesis is organized as follows:

Chapter one, we presents the background study of mathematical programming model.

Chapter two is devoted for related works in the field of assignment problem.

In chapter three, the heuristic algorithm by Amponsah and Darkwah (2009) will be introduced and explained.

Chapter four presents data collection and analysis.

Chapter five, the final chapter provides conclusion and recommendations of the study.

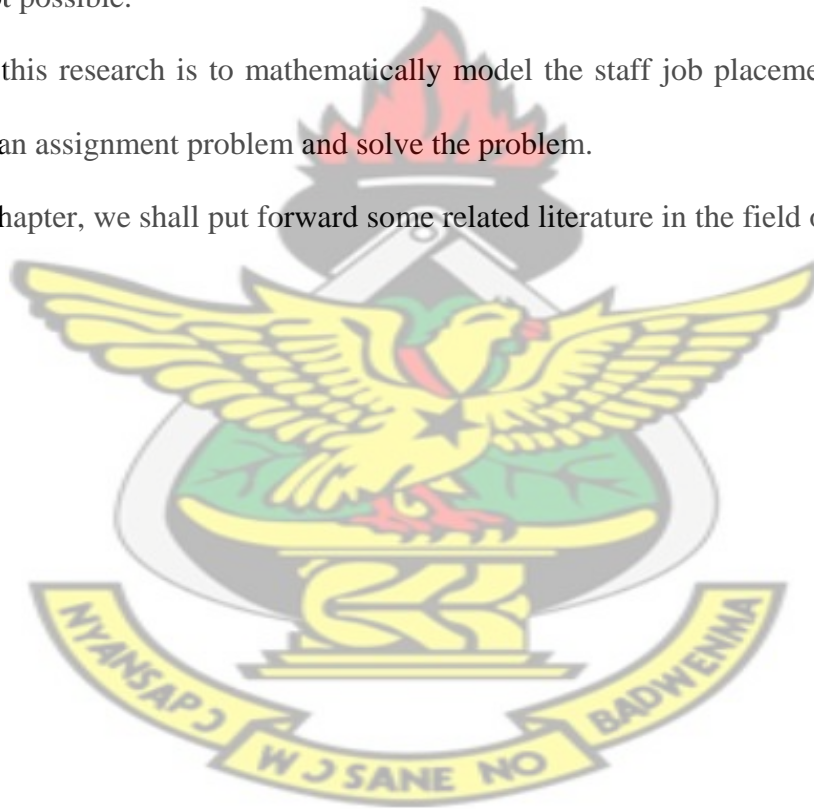
## **1.7 SUMMARY**

The assignment problem is a special case of the transportation problem, where the objective is to minimize the cost or time of completing a number of jobs by a number of persons and Maximize efficiently Revenue, sales etc In other words, when the problem

involves the allocation of  $n$  different facilities to  $n$  different tasks, it is often termed as an assignment problem. This model is mostly used for planning. The assignment model is also useful in solving problems such as, assignment of machines to jobs, assignment of salesman to sales territories, travelling salesman problem etc. It may be noted that with  $n$  facilities and  $n$  jobs, there are  $n!$  possible assignments. One way of finding an optimal assignment is to write all the  $n!$  possible arrangement, evaluate their total cost and select the assignment with minimum cost. But because of many computational procedures this method is not possible.

The goal of this research is to mathematically model the staff job placement problem a company as an assignment problem and solve the problem.

In the next chapter, we shall put forward some related literature in the field of assignment problems.



## CHAPTER TWO

### LITERATURE REVIEW

Matching highly skilled people to available positions is a high-stakes task that requires careful consideration by experienced resource managers. A wrong decision may result in significant loss of value due to understaffing, under qualification or over qualification of assigned personnel, and high turnover of poorly matched workers. While the importance of quality matching is clear, dealing with pools of hundreds of jobs and resources in a dynamic market generates a significant amount of pressure to make decisions rapidly. Naveh et al., (2007) presented a novel solution designed to bridge the gap between the need for high-quality matches and the need for timeliness. By applying constraint programming, a subfield of artificial intelligence, we are able to deal successfully with the complex constraints encountered in the field and reach near-optimal assignments that take into account all resources and positions in the pool. The considerations include constraints on job role, skill level, geographical location, language, potential retraining, and many more. Constraints are applied at both the individual and team levels. The authors introduced a technology and then describe its use by IBM Global Services, where large numbers of service and consulting employees are considered when forming teams assigned to customer projects.

Katta and Jay (2005) presented the problem of allocating a set of indivisible objects to agents in a fair and efficient manner. In a recent paper, Bogomolnaia and Moulin (2001) considered the case in which all agents have strict preferences, and propose the probabilistic serial (PS) mechanism; the authors defined a new notion of efficiency,

called ordinal efficiency, and prove that the probabilistic serial mechanism finds an envy-free ordinally efficient assignment. However, the restrictive assumption of strict preferences is critical to their algorithm. Our main contribution is an analogous algorithm for the full preference domain in which agents are allowed to be indifferent between objects. Our algorithm is based on a reinterpretation of the PS mechanism as an iterative algorithm to compute a flow in an associated network. In addition we show that on the full preference domain it is impossible for even a weak strategy proof mechanism to find a random assignment that is both ordinally efficient and envy-free.

The assignment problem constitutes one of the fundamental problems in the context of linear programming. Besides its theoretical significance, its frequent appearance in the areas of distributed control and facility allocation, where the problems' size and the cost for global computation and information can be highly prohibitive, gives rise to the need for local solutions that dynamically assign distinct agents to distinct tasks, while maximizing the total assignment benefit.

Michael et al., (2008) considered the linear assignment problem in the context of networked systems, where the main challenge is dealing with the lack of global information due to the limited communication capabilities of the agents. The authors addressed this challenge by means of a distributed auction algorithm, where the agents are able to bid for the task to which they wish to be assigned. The desired assignment relies on an appropriate selection of bids that determine the prices of the tasks and render them more or less attractive for the agents to bid for. Up to date pricing information,



necessary for accurate bidding, can be obtained in a multi-hop fashion by means of local communication between adjacent agents. Our algorithm is an extension to the parallel auction algorithm proposed by Bertsekas et al to the case where only local information is available and it is shown to always converge to an assignment that maximizes the total assignment benefit within a linear approximation of the optimal one.

Hui and Jonathan (2002) studied a multi-period assignment problem that arises as part of a weekly planning problem at mail processing and distribution centers. These facilities contain a wide variety of automation equipment that is used to cancel, sort, and sequence the mail. The input to the problem is an equipment schedule that indicates the number of machines required for each job or operation during the day. This result is then post-processed by solving a multi-period assignment problem to determine the sequence of operations for each machine. Two criteria are used for this purpose. The first is to minimize the number of startups, and the second is to minimize the number of machines used per operation. The problem is modeled as a 0–1 integer program that can be solved in polynomial time when only the first criterion is considered. To find solutions in general, a two-stage heuristic is developed that always obtains the minimum number of startups, but not necessarily the minimum number of machines per operation. In a comparative study, high quality solutions were routinely provided by the heuristic in negligible time when compared to a commercial branch-and-bound code (Xpress). For most hard instances, the branch-and-bound code was not able to even find continuous solutions within acceptable time limits.



In a Processing and Distribution Centers (P&DC), the equipment is grouped into clusters of identical machines. As such, the scheduling model only provides the number of machines that should be running for each operation during the day, but does not specify the sequence of operations for each machine. To resolve this issue, the schedule must be post-processed by solving a multi-period assignment problem.

Aronson (1986) considered a version of this problem designed to minimize (i) the cost of assigning a person to an activity, and (ii) the cost of transferring a person from one job to another. The latter was assumed to be sequence dependent. The author presented an integer multi-commodity network flow model and developed a specialized branch and bound algorithm to find solutions. Instead of solving the linear programming relaxation, his idea was to solve a set of shortest path subproblems and branch on jobs that were assigned to more than one machine.

Maxon and Bhadury (2001) studied a multi-period assignment problem with repetitive tasks and tried to introduce a human element into the analysis. Their objective was to minimize a combination of the assignment cost and the “boredom” that results when workers are required to repeat the same task in consecutive periods. A mathematical model was proposed and a simple iterative heuristic was developed and implemented in Excel.

The Airline Crew Assignment Problem (ACAP) consists of assigning lines of work to a set of crew members such that a set of activities is partitioned and the costs for that

assignment are minimized. Especially for European airline companies, complex constraints defining the feasibility of a line of work have to be respected.

Meinolf et al., (2002) developed two different algorithms to tackle the large scale optimization problem of Airline Crew Assignment. The first is an application of the Constraint Programming (CP) based Column Generation Framework. The second approach performs a CP based heuristic tree search. The authors presented how both algorithms can be coupled to overcome their inherent weaknesses by integrating methods from Constraint Programming and Operations Research. Numerical results show the superiority of the hybrid algorithm in comparison to CP based tree search and column generation alone.

“Traffic assignment is the process of allocating a set of present or future trip interchanges, known as Origin-Destination (OD) demands, to a specified transportation network” (Easa, 1991). The results of traffic assignment model contribute in many transportation planning and design decisions such as evaluation of what if scenarios for different improvements, environmental and transportation impact analysis, highway design. Traffic assignment models evolved from system level approach to subarea-level approach involving the same elements but with different implementation details. In general the application of traffic assignment models consist of five basic elements including preparing the network, establishing the OD demands, identifying a traffic assignment technique, calibrating and validating a model, and forecasting (Easa, 1991).

If any model concentrates on capacity restraint as a generator of a spread of trips on a network, one should consider a different set of models which usually attempt, with different degrees of success, to approximate to the equilibrium conditions as described by Wardrop (1952): “Under equilibrium conditions traffic arranges itself in congested networks in such a way that no individual trip maker can reduce his path costs by switching routes”. This principal focuses on the behavior of individual drivers trying to minimize their own trip costs. Wardrop (1952) proposed an alternative way of assigning traffic onto a network and this is usually referred to as his second principle: “Under social equilibrium conditions traffic should be arranged in congested networks in such a way that the average (or total) travel cost is minimized”. This principle is aimed to achieve an optimum social equilibrium which helps transport planners and engineers trying to manage traffic to minimize travel costs.

Smith and Brennan (1980) investigated the performance of the heuristic assignment techniques currently available to transportation planners in the United States in terms of accuracy for small and medium-sized cities in order to assess the potential for future applications of equilibrium assignment techniques. The study revealed that using the percentage of root mean squared error as the primary accuracy measure/percentage of the accuracy of the assignments in the order of increasing accuracy was all-or-nothing, multipath, and capacity-restrained, and the accuracy of the capacity-restrained assignments appeared to be more sensitive to the assumptions made in computing the peak-hour assigned volumes and capacities than to differences in the capacity restraint techniques.

Meneguzzo (1998) presented the Stochastic User Equilibrium (SUE) assignment problem for a signal-controlled network in which intersection control is flow-responsive and the problem is addressed within a Combined Traffic Assignment and Control (CTAC) modeling framework, in which the calculation of user equilibrium link flows is integrated with the calculation of consistent signal settings. In this model, network users have limited information of travel times and signal control is traffic-actuated. This study solved SUE- based CTAC model algorithmically by means of the so- called Iterative Optimization and Assignment (IOA) procedure and define a methodological framework for the evaluation of the performance of various traffic-responsive signal control strategies in interaction with different levels of user information, as represented by the spread parameter of the perceived travel time distribution assumed in the SUE assignment sub model.

Shafahi and Ramezani (2007) tried to provide more flexibility to model driver characteristics which affect drivers' route choice by introducing a new fuzzy assignment model. The obtained result of this method is the same as UE results when there are risk-neutral motorists and/or deterministic travel time. They also derived mathematical fuzzy user equilibrium condition.

Theoretically, microscopic simulation models can be used to evaluate traffic management strategies in real time but this might not be computationally feasible for large-scale networks and complex simulation models. Chowdhury et al (2006) presented two

Artificial Intelligence (AI) paradigms—Support Vector Regression (SVR) and Case-Based Reasoning (CBR) as alternatives to the simulation models as a decision support tool. They developed two prototype decision support tools to evaluate the likely impacts of implementing diversion strategies in response to incidents on a highway network in Anderson, South Carolina. Then VISSIM, a microscopic simulation model is used to evaluate the performances of the two prototypes by comparing their predictions of traffic conditions.

Haphuoc et al., (2002) proposed an integrated model of modal split and Traffic Assignment, in which the interaction between transit vehicles and the general traffic is modeled explicitly. In this model they applied fuzzy reasoning instead of Logit model for traffic choice behaviour because fuzzy model can describe more precisely the traffic choice behavior compared to Logic model.

Sadek et al., (1998) proposed an architecture for a routing decision support system (DSS) based on two emerging artificial intelligence paradigms: case-based reasoning and stochastic search algorithms which is expected to allow the routing DSS to (a) process information in real time, (b) learn from experience, (c) handle the uncertainty associated with predicting traffic conditions and driver behavior, (d) balance the trade-off between accuracy and efficiency, and (e) deal with missing and incomplete data problems. However, the motivation of this study is to overcome the limitations of real-time traffic flow management.



Hu et al., (2008) aimed at developing simulation-based algorithm for dynamic traffic assignment problems under mixed traffic flow considerations of car, bus, motorcycle, and truck which consists of an inner loop that incorporates a direction finding mechanism for the search process for System Optimization (SO) and User Equilibrium (UE) classes based on the simulation results of the current iteration, including experienced vehicular trip times and marginal trip times. They conducted a survey in order to understand trip maker acceptance toward route guidance.

Moreover, they conducted numerical experiments in a test network to illustrate the capabilities of the proposed algorithm.

Henry et al., (2001) addresses the fact that the traffic network itself is probabilistic and uncertain and that different classes of travelers respond differently within this uncertain environment given different levels of traffic information considered in their proposed model by capturing the travelers' decision making among discrete choices in a probabilistic and uncertain environment in which both probabilistic travel times and random perception errors that are specific to individual travelers are considered. Travelers' route choices are assumed to be made with the objective of minimizing perceived disutility at each time which depends on the distribution of the variable route travel times, the distribution of individual perception errors and the individual traveler's risk-taking nature at each time instant.

Chiu and Mahmassani (2001) proposed a hybrid framework for Dynamic Traffic Assignment (DTA) problems which is based on Stackelberg Game, in which the



Centralized DTA (CDTA) model is considered as the game leader and the Decentralized DTA (DDTA) model is the follower.

Varia and Dhingra (2004) presented the development of dynamic user equilibrium (DUE) traffic assignment model for the congested urban road network with signalized intersections which adopted a simulation-based approach for the case of multiple-origin multiple-destination traffic flows and developed a modified method of successive averages (MSA) to arrive at the user equilibrium condition.

An effective storage location assignment policy, in addition to its potential for optimal usage of warehouse space, reduces travel times related to storage, retrieval and order-picking activities. Moreover, helps to reduce congestions also enhances the balance among different warehouse activities. While previous research works exist regarding warehouse space allocation problem, considering modern concepts of logistics systems and also specific limitations for each case, further research in this respect is needed.

Sanei et al., (2010) undertook studies with the problem of space assignment for the products in a warehouse considering various operational constraints. These constraints are mainly set to prevent decentralization of products in storage locations considering more explicit and more exquisite inventory control. A linear integer programming model and a heuristic algorithm based on the branch and bound method is proposed to solve the problem. Further, a software has been developed based on proposed algorithm for

industrial usage. An experimental study, based on real data from an auto-industry shows the efficiency of the proposed algorithm achieving reasonable solutions.

Ashayeri and Golders (1985) worked on warehouse assignment design and offered two ways to optimize design of the storage. The authors also presented a step design algorithm and design for warehouse. Also in literature a step method for the design of the warehouse and several solved examples, assumption for warehouse design, hierarchical design method were presented.

Muppant (2007) presented important factors for assigning storage and proposed a Meta-heuristic slow freezing algorithm that COI has been considered as a selected criterion for selecting the locations. The author proposed an algorithm based on branch and bound in order to assign places for storage.

Semih et al., (2008) considered a distribution-type warehouse assignment problem that various type of products were collected from different suppliers for storing in the warehouse for a determined period and for delivery to different customers. The aim of their study was to design a multiple-level warehouse shelf configuration which minimized the annual carrying costs. Since proposed mathematical model was shown to be NP-hard, a particle swarm optimization algorithm (PSO) as a novel heuristic was developed for determining the optimal layout.

Jinxiang et al., (2010) presented a detailed survey of the research on warehouse assignment design, performance evaluation, practical case studies, and computational support tools. The authors presented an extensive review on warehouse operation planning problems. The problems were classified according to the basic warehouse functions, i.e., receiving, storage, order picking, and shipping. Their purpose was to provide a bridge between academic researchers and warehouse practitioners, explaining what planning models and methods were currently available for warehouse operations, and what were the future research opportunities.

Peter and Marco (2009) explored the current literature on the overall methodology of warehouse assignment design, together with the literature on tools and techniques used for specific areas of analysis. The general results from the literature had then been validated and refined with reference to warehouse design companies.

Liong et al., (2000) solved a special Quadratic Assignment Problem (QAP) which consists of deciding the assignment of customers to loading positions as well as fulfilling their demands, i.e. a double-assignment problem. Brief introductions about QAP and its applications are also given. In this work, the main questions were (i) where a customer should be assigned in a list of possible loading positions, (ii) from which storage areas should the customer be served, and (iii) how many lifting truck should be assigned to each loading position in order to minimize cost and residence time. We have applied the Greedy Algorithm to get a good initial solution, and then a modified Genetic Algorithm to find the best solution to the problem. We explored the nearest neighbour using

recombination procedure and maintaining the elements with the lower cost. The best solution found is always saved. Comparison with previous work and suggestions for further work has also been included.

Ahmad and Nima (2002) studied a problem of assigning a set of applicants to the service stations, which can be state as follows: A set of geographically scattered applicants must be served from a set of service stations so that the total cost of services is minimized. The authors considered two capacity for each service station, i.e. usual capacity and extra capacity. The set of applicants partitioned in two sets, special and ordinary applicants. A Mixed Integer Programming (MIP) formulation is given and a genetic algorithm (GA) proposed for solving the problem. The authors solved some randomly generated instances of introduced problem with the GA.

Dimitri et al., (1992) proposed auction algorithms for solving several types of assignment problems with inequality constraints. Included are asymmetric problems with different numbers of persons and objects, and multi-assignment problems, where persons may be assigned to several objects and reversely. A central new idea in all these algorithms is to combine regular auction, where persons bid for objects by raising their prices, with reverse auction, where objects compete for persons by essentially offering discounts. Reverse auction can also be used to accelerate substantially (and sometimes dramatically) the convergence of regular auction for symmetric assignment problems.

It is with the aim of solving scheduling problems with irregular cost functions that Sourd (2004) studied the continuous assignment problem. It consists in partitioning a d

dimensional region into subregions of prescribed volumes so that the total cost is minimized. The dual problem of the continuous assignment problem is an unconstrained maximisation of a non-smooth concave function. The preemptive variant of the scheduling problem with irregular cost functions corresponds to the one-dimensional continuous assignment problem and a lower bound for the non-preemptive variant can be derived. It is computationally tested in a branch-and-bound algorithm.

The formulation of Facility Layout Problems (FLPs) as Quadratic Assignment Problems (QAPs) has gained substantial attention from researchers. The main reason is that, QAPs provide possibilities to solve FLPs computationally. To date, there are two common approaches used to solve FLPs formulated as QAPs, that is, exact methods and approximate methods (also known as heuristics). In recent years, there is an increasing interest in solving QAPs using the general extension of heuristic methods called meta-heuristics. Ant Colony Optimisation (ACO) has currently emerged as a new and promising meta-heuristic. Phen et al., (2008) presented a model aimed to provide a comprehensive review of the concepts of ACO and its application in solving QAPs. In addition, the various ACO algorithms or variants developed to solve them are critically analysed and discussed. It is shown that these existing algorithms still possess many limitations and weaknesses. Finally, useful strategies and research directions are provided to improve these weaknesses.

Assignment problems are defined with two sets of inputs, i.e. set of resources and set of demands. Assignment of each resource to each demand has its own cost. Exactly one



resource has to be assigned to each of the demands in such way, that maximal cost of the assignment is minimal when comparing to other assignments.

Hungarian algorithm (also known as Kuhn-Munkres algorithm) is able to find an optimal solution of assignment problems in polynomial time, but is only able to solve assignment problems with precisely defined demands and resources. This presents a major problem in many real-life scenarios while the nature of these problems is such that inputs are commonly defined only vaguely (i.e. fuzzily). In order to solve them, their precise formalization is needed. Formalization of their properties is normally far from being a straightforward procedure and can present large costs in the meaning of time and money. Fuzzy logic on the other hand successfully copes with the processing of imprecise data.

Miha (2009) presented an extension of the Hungarian algorithm with the introduction of fuzzy logic methods – fuzzy Hungarian algorithm. Vaguely defined resources and demands can be easily described with fuzzy values which present an input to fuzzy Hungarian algorithm.

The extended version of the algorithm is therefore able to cope with vaguely defined assignment problems, can be used more efficiently (i.e. with no further formalization of vaguely defined terms) and in a wider scope of assignment problems than the basic approach. Basic version of the Hungarian algorithm which was firstly presented by Kuhn (2001) is presented in this article. Its extension with fuzzy logic methods is described and its usage on an example of vaguely defined assignment problem is demonstrated. Its benefits were also justified by the comparison of the results between the basic version of Hungarian algorithm and the fuzzy version of Hungarian algorithm on the same problem.



The channel-assignment problem is important in mobile telephone communication. Since the usable range of the frequency spectrum is limited, the optimal channel-assignment problem has become increasingly important. Omid (2010) presented a model and the goal of this is to find a channel assignment to requested calls with the minimum number of channels subject to interference constraints between channels. This algorithm consists of: 1) the fixed channel assignment stage; 2) the neural network stage. In the first stage, the calls in a cell determining the lower bound on the total number of channels are assigned channels at regular intervals, then the calls in adjacent six cells are assigned channels by a cluster heuristic method sequentially. In the second stage, the calls in the remaining cells are assigned channels by a binary neural network. The performance is verified through solving well-known benchmark problems. Especially for Sivarajan's benchmark problems, my algorithm first achieves the lower bound solutions in all of the 12 instances.

Mingfang et al., (2010) studied the Weapon-Target Assignment (WTA) problem, which has wide applications in the area of defense-related operations research. This problem calls for finding a proper assignment of weapons to targets such that the total expected damaged value of the targets to be maximized. The WTA problem can be formulated as a nonlinear integer programming problem which is known to be NP-complete. There does not exist any exact method for the WTA problem even small size problems, although several heuristic methods have been proposed. In this paper, Lagrange relaxation method is proposed for the WTA problem. The method is an iterative approach which is to

decompose the Lagrange relaxation into two subproblems, and each subproblem can be easy to solve to optimality based on its specific features. Then, we use the optimal solutions of the two subproblems to update Lagrange multipliers and solve the Lagrange relaxation problem iteratively. Our computational efforts signify that the proposed method is very effective and can find high quality solutions for the WTA problem in reasonable amount of time.

KNUST

Assignment problem (AP) is a well known topic and is used very often in solving problems of engineering and management science. In this problem  $a_{ij}$  denotes the cost for assigning the  $j^{\text{th}}$  job to the  $i^{\text{th}}$  person. The cost is usually deterministic in nature. Nagarajan and Solairaju (2010) presented studies which  $a_{ij}$  was considered to be trapezoidal and triangular numbers denoted by  $a_{ij}$  which are more realistic and general in nature. Robust's ranking method has been used for ranking the fuzzy numbers. The fuzzy assignment problem has been transformed into crisp assignment problem in the linear programming problem form and solved by using Hungarian method; Numerical examples show that the fuzzy ranking method offers an effective tool for handling the fuzzy assignment problem.

The assignment problem (AP) and bottleneck assignment problem (BAP) are well studied in operational research literature. Abraham and Aneja (1993) considered two related problems which simultaneously generalize both AP and BAP. Unlike AP and BAP, these generalizations are strongly NP-complete. The authors propose two heuristics to solve these generalized problems: one based on a greedy principle and the other based on tabu

search. Computational results are presented which show that these heuristics, when used together, produce good quality solutions. Our adaptation of tabu search also gives some new insight into the application of tabu search on permutation problems.

Wang and Liu (2010) presented a new algorithm on a special assignment problem in which the real assigned jobs are less than or equal to both the total persons and the total jobs. To this special assignment problem the authors posed the concept of reserve point, discussed the character of reserve point and accessed to relevant conclusion a new method to solve this special assignment problem is given through increasing reserve points finally.

One-sided assignment problems combine important features of two well-known matching models. First, as in roommate problems, any two agents can be matched and second, as in two-sided assignment problems, the payoffs of a matching can be divided between the agents. Bettina and Alexandru (2009) presented a similar approach to one-sided assignment problems as Sasaki (1995) for two-sided assignment problems and we analyze various desirable properties of solutions including consistency and weak pairwise-monotonicity. The authors showed that for the class of solvable one-sided assignment problems (i.e., the subset of one-sided assignment problems with a non-empty core), if a subsolution of the core satisfies (indifference with respect to dummy agents, continuity, and consistency) or (Pareto indifference and consistency), then it coincides with the core. However, the authors also prove that on the class of all one-sided assignment problems (solvable or not), no solution satisfies consistency and coincides with the core whenever the core is non-empty. Finally, the authors commented on the

difficulty in obtaining further positive results for the class of solvable one-sided assignment problems in line with Sasaki's (1995) characterizations of the core for two-sided assignment problems.

Eriksson and Karlander (2001) and Sotomayor (2005) modelled and analyzed one-sided assignment problems. A one-sided assignment problem consists of a set of agents and a value function that specifies the worth of trade gain or the payoff working together for each pair of agents. A feasible outcome for a one-sided assignment problem is a matching that partitions the set of agents in pairs and singletons and a payoff vector that divides the total value of the matching between the agents. A solution assigns to any one-sided assignment problem a non-empty subset of feasible outcomes. As in many other economies, a concept of special interest is the core. Eriksson and Karlander gave a characterization of the core by a forbidden minor's criterion while Sotomayor showed that there are one-sided assignment problems with an empty core and identify necessary and sufficient conditions for the non-emptiness of the core. Hence, strictly speaking, the core is not a solution for the class of all one-sided assignment problems.

Anshuman and Rudrajit (2006) solved the generalized "Assignment problem" through genetic algorithm and simulated annealing. The generalized assignment problem is basically the "N men- N jobs" problem where a single job can be assigned to only one person in such a way that the overall cost of assignment is minimized. While solving this problem through Genetic Algorithm (GA), a unique encoding scheme is used together with Partially Matched Crossover (PMC). The population size can also be varied in each

iteration. In Simulated Annealing (SA) method, an exponential cooling schedule based on Newtonian cooling process is employed and experimentation is done on choosing the number of iterations (m) at each step. The source codes for the above have been developed in C language and compiled in GCC. Several test cases have been taken and the results obtained from both the methods have been tabulated and compared against the results obtained by coding in AMPL.

Solving the state assignment problem means finding the optimum assignment for each state within a sequential digital circuit. These optimum assignments will result in decreasing the hardware realization cost and increasing the reliability of the digital circuit. Unfortunately, the state assignment problem belongs to the class of nondeterministic polynomial time problems (NP complete) which requires heavy computations. Different attempts have been made towards solving the problem with reasonable recourses. Walid (2009) presented a methodology for solving the state assignment problem, the methodology conducted a neighborhood search while using a heuristic to determine the fitness of solution. To avoid being trapped at a local optimum solution, a metaheuristic (simulated annealing) was utilized for deciding whether a new solution should be accepted. A case study was included to demonstrate the proposed procedure efficiency. The proposed approach finds the optimum assignment for the case study. The authors explored the usage of a stochastic search technique inspired by simulated annealing to solve the problem of the state assignment problem. This proved the efficiency of the methodology.



Wan (2001) studied the component assignment problem in PCB assembly, where assigning components to appropriate machines, in order to get a minimum assembly time for the assembly line, can be formulated as an integer linear programming model. In order to obtain the optimal solution to the component assignment problem, the branch-and-bound method can be applied. However, it is not efficient. The author proposed the tabu search heuristic approach to the component assignment problem. The procedure of the tabu search to the problem is presented, and a numerical example is provided. Finally, the performance of the tabu search is analyzed with the example.

Dritanet et al., (1988) studied the route and level flight assignment problem aiming at global flight plan optimization, which has already become a key issue owing to the growth of air traffic. Better coordination of all existing flights for all airlines is becoming an increasingly desirable goal. A number of related problems appear in the operations research literature, notably vehicle routing, scheduling and other transportation problems. Several studies have been especially devoted to the problem of aircraft scheduling and routing. Aircraft routing requires the generation of non-colliding, time-dependent routes through a specified airspace that we call the airspace network. The problem considered here can be modeled as a specific flow problem in a given space-time network. This study aims at estimating the effects of routing capabilities at a quantitative level (the congestion level, i.e. the number of potential en-route conflicts), and at a qualitative level (traffic smoothing). The authors presented a deterministic model based on a Linear Programming approach for optimizing the level route assignment in a trajectory-based Air Traffic Management (ATM) environment. This problem can be seen as a multi-period (dynamic) problem where the time dimension is an essential ingredient to consider

when constructing flight plans. This dynamic problem can be transformed into a static one by using standard technique of time-expanding the underlying network. We propose here a model to consider the airspace congestion in a finer way: we consider the number of aircraft involved in potential en-route conflicts rather than the number of aircraft in a sector, sometimes implicitly understood as en-route capacities in ATM.

Odior et al., (2010) addressed the problem of effectiveness of feasible solutions of a multi-criteria assignment problem and this was done in two steps. In the first step, the authors determine whether or not a given feasible solution of a multicriteria assignment problem is a real efficient one. In the second step, if the feasible solution is not real efficient, the authors provided a real efficient solution that dominates that not real efficient solution, using their proposed method which consists of transforming the original problem into an assignment problem.

The Generalized Assignment Problem consists in assigning a set of tasks to a set of agents with minimum cost. Each agent has a limited amount of a single resource and each task must be assigned to one and only one agent, requiring a certain amount of the resource of the agent. Helina and Daniel (1998) presented new meta-heuristics for the generalized assignment problem based on hybrid approaches. One meta-heuristic is a MAX-MIN Ant System (MMAS), an improved version of the Ant System, which was recently proposed by Stutzle and Hoos to combinatorial optimization problems, and it can be seen has an adaptive sampling algorithm that takes in consideration the experience gathered in earlier iterations of the algorithm. Moreover, the latter heuristic is combined

with local search and tabu search heuristics to improve the search. A greedy randomized adaptive search heuristic (GRASP) is also proposed. Several neighbourhoods are studied, including one based on ejection chains that produce good moves without increasing the computational effort. The authors presented computational results of the comparative performance, followed by concluding remarks and ideas on future research in generalized assignment related problems.

KNUST

Assignment problems are used throughout many research disciplines. Most assignment problems in the literature have focused on solving a single objective. Mark and Garry (2008) studied assignment problems that have multiple objectives that need to be satisfied. In particular, this chapter looks at how multi-objective evolutionary algorithms have been used to solve some of these problems. Additionally, the authors examined many of the operators that have been utilized to solve assignment problems and discuss some of the advantages and disadvantages of using specific operators.

The extended usage of Distributed Computing Systems (DCS) has made the task assignment strategies more attractive. Various types of algorithms have been developed for the Task Assignment Problem (TAP) in distributed computing systems along the different definitions of cost function. The final goal of task assignment algorithms is the assignment of some cooperative tasks to a set of interconnected processors. This assignment must minimize the total system cost and obtain a reasonable amount of load balancing. Abbas and Nasrollah (1992) studied a fair cost functions for task assignment problem in distributed computing systems are defined in order to satisfy some system

requirements appropriately. Then, by the employment of the linear programming (LP) concepts, a polynomial approximation algorithm for task assignment problem is designed and the validity of the proposed algorithm is proven by theoretical analysis. Finally, the results of the execution of this algorithm on several problem instances are provided.

The classical generalized assignment problem (GAP) may be stated as finding a minimum-cost assignment of tasks to agents such that each task is assigned to exactly one agent and such that each agent's resource capacity is honored. This NP-hard problem has applications that include job scheduling, routing, loading for flexible manufacturing systems, and facility location. Due to the difficulty in solving "hard" GAPs to optimality, most recent papers either describe heuristic methods for generating "good" solutions or, in the case of optimizing methods, computational results are limited to 500 to 1,000 binary variables. Nuass (2003) described a special purpose branch-and-bound algorithm that utilizes linear programming cuts, feasible-solution generators, Lagrangean relaxation, and subgradient optimization. The author presented computational results for solving "hard" problems with up to 3,000 binary variables. An unanticipated benefit of the algorithm is its ability to generate good feasible solutions early in the process whose solution quality generally dominates the solutions generated by two recently published heuristics. Furthermore, the computation time required is often less than the time taken by the heuristics. Thus, we have an optimizing algorithm that can be used quite effectively as a heuristic when proof of optimality is not an absolute requirement.

## CHAPTER THREE

### METHODOLOGY

#### 2.0 INTRODUCTION

This chapter provides discussions of the methods for solving assignment problems.

In order to understand the Hungarian method in solving linear assignment problems, it is necessary to have a good understanding of some of the background graph theory for combinatorial optimization problems.

Assignment problem is a special type of transportation problem which is also a resource allocation problem. Here, we have  $n$  jobs to perform with  $n$  persons and the problem is how to distribute the job to the different persons involved. Depending on the intrinsic capacity or merit or potential of the individual, he will be able to accomplish the task in different times. Then the objective function in assigning the different jobs to different persons is to find the optimal assignment that will minimize the total time taken to finish all jobs by the individuals.

The problem may be stated formally as follows: Given an  $n \times n$  array of real numbers representing the individual return associated with assigning one item to one person. We have to find the best assignment so that the total return is optimal.

The general problem is modeled as;

$$\text{Maximize } \sum_{i=1}^n \sum_{j=1}^n C_{ij} X_{ij}$$

$$\text{Subject to } \sum_{j=1}^n X_{ij} = 1$$

$$\sum_{i=1}^n X_{ij} = 1$$

$$X_{ij} \in \{0, 1\}.$$



Solving assignment problem can be difficult and time consuming. A great deal of research has been performed to improve the solving times and ease of assignment problem. One of the major areas which research has been done in is the bipartite matching and graph theory.

### 3.1 GRAPH THEORY.

DEFINITION: A graph is an ordered pair  $G = (V, E)$  consisting of a finite set and a subset  $E$  of elements of the form  $(x, y)$  where  $x$  and  $y$  are in  $V$ . The set  $V$  are called the vertices of the graph and the set  $E$  are called the edges

DEFINITION: A graph  $G$  is said to be a bipartite (or bicolored) graph if the vertices can be partitioned into two mutually disjoint sets  $X$  and  $Y$  so that there is no edge of the form  $(x, x')$  with  $x$  and  $x'$  in  $X$  or of the form  $(y, y')$  with  $y$  and  $y'$  in  $Y$ . A bipartite graph will be denoted by  $G = (\{X, Y\}, E)$ .

NOTATION: The cardinality of a set  $X$  will be denoted by  $|X|$ .

Bipartite graphs  $G = (\{X, Y\}, E)$  are represented by matrices. The  $X$  vertices are for example used for row indices and the  $Y$  vertices are used as column indices. Generally, the existence of an edge  $(x, y)$  is indicated by a 1 in the  $x, y$  cell of the  $|X| \times |Y|$  matrix; no edge is indicated by 0. For the assignment problem, we are representing an edge by 0 and no edge by a nonzero number.

DEFINITION: A matching for a bipartite graph  $G = (\{X, Y\}, E)$  is a subset  $M$  of  $E$  such that no two elements of  $M$  have a common vertex.

DEFINITION: If  $G = (\{X, Y\}, E)$  is a bipartite graph, set

$$\rho(G) = \max \{|M| \mid M \text{ is a matching of } G\}.$$

A matching  $M$  such that  $|M| = \rho(G)$  will be called a maximal matching.

DEFINITION: A set of vertices  $V'$  is said to be a cover of a set of edges  $E'$  if every edge in  $E'$  is incident on one or more of the vertices of  $V'$ . A set of vertices  $S$  will be called a cover of the bipartite graph  $G = (\{X, Y\}, E)$  if every edge of  $G$  is incident on one or more of the vertices of  $S$ .

DEFINITION: If  $G = (\{X, Y\}, E)$  is a bipartite graph, set

$$c(G) = \min \{|S| \mid S \text{ is a cover of } G\}.$$

A cover  $S$  such that  $|S| = c(G)$  will be called a minimal cover of  $G$ .

THEOREM: If  $G = (\{X, Y\}, E)$  is a bipartite graph, then  $\rho(G) \leq c(G)$ .

PROOF: Let  $S$  be a cover with  $|S| = c(G)$ . Let  $M$  be a matching. Then each  $e$  in  $M$  has at least one of its vertices in  $S$ . If  $|M| > |S|$ , then by the pigeonhole principle, two edges  $e_1$  and  $e_2$  meet the same vertex  $v$  in  $S$ . This contradicts the definition of a matching. So we have that  $|M| \leq |S| = c(G)$ .

### 3.1.1 GRAPHS FOR THE ASSIGNMENT PROBLEM

The assignment problem corresponds to a bipartite graph. The vertices  $V$  may be partitioned into two sets: (1) the assigned tasks and (2) the assignees. The edges consist of (unordered) pairs connected the assigned tasks and the assignees. Since we allow the theoretic possibility of assigning any task to any assignee, the graph for the assignment problem consists of vertices  $V = (X, Y)$  where  $X = \{x_1, \dots, x_m\}$  and  $Y = \{y_1, \dots, y_m\}$  and edges consisting of all combinations  $E = \{(x_i, y_j) \mid 1 \leq i \leq m, 1 \leq j \leq m\}$ . We can denote the incidence matrix of this graph as an  $m \times m$  matrix consisting entirely of 1's

and we can use the matrix as a substitute for the graph. This has some disadvantage in that an index, say 1, can denote  $x_1$  or  $y_1$  and so the row and column indices should be kept separate.

Now consider sub graphs of the assignment problem. For example, let  $(a_{ij})$  be the  $m \times m$  cost matrix of the assignment problem and let  $G'$  be the sub graph whose edges are all  $(i, j)$  with  $a_{ij} = 0$  and all vertices incident on any of the edges. If we are considering  $i$  as row indices and  $j$  as column indices, the vertices will consist of all rows of  $A$  which have a 0 entry and all columns of  $A$  which have a zero entry.

### 3.1.2 VERIFICATION OF THE ASSIGNMENT ALGORITHM

We verify the assignment algorithm terminates at some stage with a cover of size  $m$ . The proof is obtained by noting that sum of the current cost matrices is strictly decreasing at each stage provided there is a minimal cover of size less than  $m$ . We use the viewpoint of the previous section.

**THEOREM:** Let be  $A = (a_{ij})$  be an  $m \times m$  matrix of positive entries. Suppose that there is a cover  $S$  of the set  $E'$  of edges  $(i, j)$  with  $a_{ij} = 0$  of size less than or equal to  $m - 1$ .

Let  $D = \{ (i, j) \in E' \mid \text{both } i \text{ and } j \text{ are in } S \}$ . Let  $a_0$  be the minimum of  $\{a_{ij} \mid i \notin S, j \notin S\}$ .

Suppose that

$$\begin{cases} a_{ij} + a_0 & (i, j) \in D \\ a_{ij} - a_0 & (i, j) \notin E' \\ a_{ij} & \text{otherwise} \end{cases}$$

Then  $B = (b_{ij})$  is a positive matrix with  $\sum a_{ij} > \sum b_{ij}$ .

**REMARK:** We can paraphrase the theorem as follows. Suppose that 0's of A are crossed by crossing out the rows and columns containing a 0. Suppose that the minimal number of crossed out rows and columns necessary to cross out all the 0's of A is less than  $m - 1$ . Suppose that the minimal entry in the entries not crossed out by the minimal number of crossed out rows and columns is added to all doubly crossed out entries of A and subtracted from all non crossed out entries to give a matrix  $B = (b_{ij})$ . Then B is a positive matrix and  $\sum a_{ij} > \sum b_{ij}$ .

**PROOF:** Let  $s$  and  $t$  be any integers with  $0 \leq s + t \leq m - 1$  and suppose that  $s$  rows and  $t$  columns are crossed out, i.e., the cover  $C$  consists of  $s$  vertices of  $X$  and  $t$  vertices of  $Y$ . Then there are  $st$  doubly crossed out elements,  $m(s + t) - st$  crossed out elements, and  $m(s + t)$  singly crossed out elements. So there are

$$m^2 - m(s + t) + st$$

entries that are not crossed out. Now suppose that the minimal (nonzero) entry in the uncrossed out part of A is  $r$ .

Since all the zero entries are crossed out, we get that  $r > 0$ . So we get

$$\sum a_{ij} - \sum b_{ij} = -rst + r(m^2 - m(s + t) + st) = rm(m - (s + t)) > 0$$

since  $m > s + t$ . Also note that all  $b_{ij} \geq 0$  since the minimal entry is subtracted.

The process of adding to double crossed out entries and subtracting from non crossed out entries is a combination of operations that does not change the optimal solution of the assignment problem.

**THEOREM:** Suppose that the first three steps in the assignment algorithm is implemented. Then the optimal solution of the assignment problem with the new cost matrix does not change.

**PROOF:** For first step note that the optimal solution  $x^*$  does not change if a constant is subtracted from any row or column. To see this suppose that  $c$  is subtracted from every element in the first row of the cost coefficient matrix.

Then the problem becomes

$$\text{Minimize } \sum_i (c_{1j} - c)x_{1j} + \sum_{i \geq 2, j} c_{ij} x_{ij}$$

Subject to the constraints

$$\sum_j x_{ij} = 1 \quad (1 \leq i \leq m)$$

$$\sum_i x_{ij} = 1 \quad (1 \leq j \leq n)$$

$$0 \leq x_{ij} \leq 1.$$

But we note that

$$\sum_j (c_{1j} - c)x_{1j} + \sum_{i \geq 2, j} c_{ij} x_{ij} = \sum_{i,j} c_{ij} x_{ij} - \sum_j c x_{1j} = \sum_{i,j} c_{ij} x_{ij} - c$$

So the optimal solution for the original problem is exactly the optimal solution for the perturbed problem. The same holds for all other rows and columns. So the first step of the algorithm does not change the optimal solution.

Now suppose  $c$  is added to each cost of a doubly covered entry and  $c$  is subtracted from each cost of an uncovered entry. This is equivalent to adding  $c$  to each covered column and subtracting  $c$  from each uncovered row. To see this suppose  $c$  is added to the covered



column 1 and subtracted from the uncovered row 1. Suppose column 2 is uncovered and row 2 is covered. Then we get the northwest  $2 \times 2$  corner is

	<i>cov</i> (+c)	<i>uncov</i> (no action)
<i>uncov</i> (-c)	$c - c = 0$	$-c$
<i>cov</i> (no action)	$+c$	0

which covers all 4 cases. So the action in the third step is a series of action in the first step and the optimal solution does not change.

### 3.1.3 MAXIMAL MATCHINGS USING THE HUNGARIAN ALGORITHM

Let  $G = (\{X, Y\}, E)$  be a bipartite graph. Let  $M$  be a matching for  $G$ . The Hungarian Algorithm either shows that  $M$  is a maximal matching for  $G$  or finds a matching  $M'$  for  $G$  with  $|M'| = |M| + 1$ .

- I. Label all vertices in  $X$  with (\*) when the vertices do not meet an edge of  $M$  and call all vertices untested.
- II. If in the previous step no new labels have been given to a vertex of  $X$ , then STOP. Otherwise, go to III.
- III. While there is a labeled but untested vertex  $x_i$  of  $X$ , label with  $x_i$  all vertices  $y_j$  of  $Y$  that have not yet been labeled and that can be connected to  $x_i$  with an edge NOT IN  $M$ . The vertex  $x_i$  is now tested (even if no edge is added) and the vertices  $y_j$  are now labeled.
- IV. If no new label has been given in III, then STOP. Otherwise, find an untested but labeled vertex  $y_j$  of  $Y$  and label with  $y_j$  any unlabeled vertex  $x_k$  of  $X$  which is joined to  $y_j$  by an edge IN  $M$ . The vertex  $y_j$  is now tested (even if no edge is added) and vertex  $x_k$

is now labeled. If  $y_j$  cannot be connected to an unlabeled vertex in  $X$ , then STOP and an Augmenting Tree has been found.

V. Return to II.

The algorithm stops after a finite number of iterations since each vertex is receiving at most one label and each vertex is tested at most once. There are two possibilities:

I. (Augmenting Tree) There is a labeled vertex of  $Y$  that does not meet an edge of  $M$ . This has the following diagram.



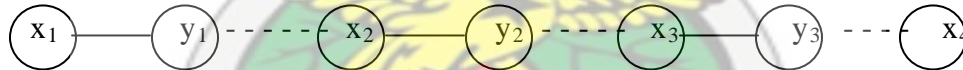
where the dashed lines represent edges in  $M$  and the solid lines represent edges not in  $M$ .

In this case the size of the matching **can be increased by** switching edges to look like



where an additional vertex  $x_1$  has been matched to  $y_1$ .

II. (Hungarian Tree) A diagram of the form



where no increase in the matching to include  $x_1$  is possible.

The termination is obtained when all unmatched elements in  $X$  have been matched or produce Hungarian trees. Elements that produce Hungarian trees are called Hungarian acorns.

**THEOREM:** Suppose that  $G = (\{X, Y\}, E)$  be a bipartite graph and that  $M$  is a matching for  $G$ . Suppose that a Augmenting Tree  $\{x_1, y_1, \dots, x_n, y_n\}$  has been found using the Hungarian algorithm. Let  $S$  be the set of edges given by

$$S = \{(y_1, x_2), (y_2, x_3), \dots, (y_{n-1}, x_n)\}$$

and let  $M_1$  be the set of edges

$$M_1 = \{(x_1, y_2), (x_2, y_2), \dots, (x_n, y_n)\}.$$

Then the set of edges  $M'$  given by

$$M' = S \cup (M - M_1)$$

is a matching for  $G$  with  $|M'| = |M| + 1$ .

**PROOF:** First note that  $y_n$  cannot be attached to a labeled vertex  $x$  by an edge  $e = (y_n, x)$  in  $M$ ; otherwise, the vertex  $y_n$  would have to be a labeled and an unlabeled vertex at the stage when  $\{x_1, y_1, \dots, x_n\}$  is formed. To see that  $y_n$  is labeled at this stage, we see that the vertex  $x$  cannot be a  $*$  vertex (since  $*$  vertices are incident on no edges of  $M$ ), and therefore,  $x$  must have been labeled by some  $y$  using an edge  $(y, x)$  in  $M$ . This would mean that  $y = y_n$  since both  $(y_n, x)$  and  $(y, x)$  are in the matching  $M$  and have a vertex in common. So  $x$  must have been labeled by  $y_n$  and this implies that  $y_n$  must have been labeled. To see  $y_n$  is unlabeled, we see that the vertex  $y_n$  is labeled by  $x_n$  in the algorithm only if  $y_n$  is unlabeled.

Now we show that  $M'$  is a matching by showing each vertex  $z$  is incident on at most one edge in  $M'$ . If  $z$  is in  $\{y_1, x_2, \dots, y_{n-1}, x_n\}$ , then  $z$  is incident on an edge  $(y_i, x_{i+1})$  in  $M_1$  and no other edge in  $M$ . So  $z$  is incident on one edge in  $M$ . If  $z = x_1$ , then  $z$  is  $*$ -vertex and  $z$  is incident on no edge in  $M$  and incident only on the edge  $(x_1, y_1)$  in  $M'$ . If  $z = y_n$ , then  $z$  is incident on one edge  $(x_n, y_n)$  since  $y_n$  is incident on no edge of  $M$ ; otherwise,  $(y_n, x)$  would be in  $M$  but  $x$  cannot be labeled by the preceding paragraph or unlabeled by the fact that the Hungarian Algorithm has stopped on  $y_n$ . So we get that the Augmenting Tree produces a matching  $M'$ .

We note that the matching  $M'$  has one more edge than  $M$ .

We see that the Hungarian algorithm terminates after a finite number of iterations. On each level the points are tested one after the other and after all the points are tested or perhaps before the algorithm ends. As a whole the algorithm either terminates when the only  $X$  elements left are Hungarian acorns.

**THEOREM:** Suppose  $M$  is a matching of the bipartite graph  $G = (\{X, Y\}, E)$  and suppose the Hungarian algorithm produces only Hungarian trees for all unmatched  $X$ . Let  $X^{\text{un}}$  be all unlabeled  $X$  vertices and  $Y^{\text{lab}}$  be all labeled  $Y$  vertices. Then

- 1)  $S = X^{\text{un}} \cup Y^{\text{lab}}$  is a minimal cover of  $G$ , and
- 2)  $|M| = |S|$  and  $M$  is a maximal matching of  $G$ .

**PROOF:** First we show  $S$  is a cover. We argue by contradiction. Suppose  $(x, y) \in E$  and assume  $x \in X^{\text{lab}}$  and  $y \in Y^{\text{un}}$ . We show that such an edge does not exist. First assume  $(x, y) \in M$ . Since  $x$  is labelled and  $(x, y)$  is in  $M$ , the vertex cannot be the first vertex in a path labelled with a (\*) due to I of the algorithm. So  $x$  must be part of chain of the form

$$x_1 \text{-----} y_1 \text{-----} x_2 \text{-----} y_2 \text{-----} \dots \text{-----} x_k \text{-----} y_k \text{-----} x_{k+1} = x.$$

where the solid lines are not in  $M$  and the dashed lines are in  $M$ . But there is at most one edge in  $M$  incident of  $x$  and this edge is  $(x, y)$ . So we must have that  $(x, y) = (x, y_k)$  and  $y = y_k$ . The vertex  $y_k$  is labelled and this contradicts the assumption that  $y$  is not labelled. Now suppose the  $(x, y) \notin M$ . Since  $x$  is labeled, it follows that  $y$  must be labelled which is a contradiction. So we have a contradiction in all cases. This means that  $S$  is a cover.

Now we show that  $|M| = |S|$ . This shows that  $M$  is a maximal matching and  $S$  is a minimal cover. We find a one-one function  $f$  of  $S$  onto  $M$ . If  $y \in Y^{\text{lab}}$ , then  $y$  cannot be a termination of some tree starting at a (\*); otherwise, there would be an augmenting step.

So  $y$  meets an edge  $\{y, x\}$  of  $M$ . Since  $M$  is a matching,  $\{y, x\}$  is the unique edge of  $M$  that  $y$  meets. Note that  $x$  gets a label from  $y$  and so  $x$  is not in  $X^{\text{un}}$ . So we set

$f(y) = (y, x) \in M$ . Now let  $x' \in X^{\text{un}}$ . Note that  $x'$  meets an edge  $(x', y')$  of  $M$ ; otherwise, the vertex  $x'$  would have the label  $(*)$ . As before the edge  $(x', y')$  is the only edge in  $M$  that  $x'$  meets. Finally, the element  $y'$  is unlabeled. Indeed, if  $y'$  were labeled, then there would be a labeled  $x''$  with an edge  $(x'', y')$  not in  $M$  and with  $(y', x')$  in  $M$  and so there would be an augmenting tree through  $y'$ . We let  $f(x') = (x', y')$ .

Now we see that the function  $f$  is one-one. In fact,  $f(y) = f(y')$  implies that  $y = y'$  since  $y$  is the unique  $y$  element on which  $f(y)$  is incident. Also  $f(x) = f(x')$  implies  $x = x'$  and finally  $f(x) = f(y)$  is not possible as we have already shown. So  $f$  is one-one.

Since  $f$  is one-one, we have that  $|S| \leq |M|$ . But we have already seen that  $|M| \geq |S|$ . So we get that  $|S| = |M|$ .

**COROLLARY:** Let  $G = (\{X, Y\}, E)$  be a bipartite graph. Then  $\rho(G) = c(G)$ .

**PROOF:** We have that  $\rho(G) \leq c(G)$ . But let  $M$  be a maximal matching for  $G$ , i.e., a matching with  $|M| = \rho(G)$ . Then the Hungarian algorithm terminates on  $M$  without a breakthrough and the preceding theorem implies that there is a cover  $S$  with  $|S| = |M|$ . So we get that  $c(G) = |S| = |M| \leq \rho(G)$ . So we get that  $\rho(G) = c(G)$ .

**COROLLARY:** Let  $G = (\{X, Y\}, E)$  be a bipartite graph with  $|X| = |Y|$ . Suppose that  $G$  has a minimal cover  $S$  with  $|S| = |X|$ . Then there is a matching  $M$  such that  $M$  is incident on all vertices of  $G$ .

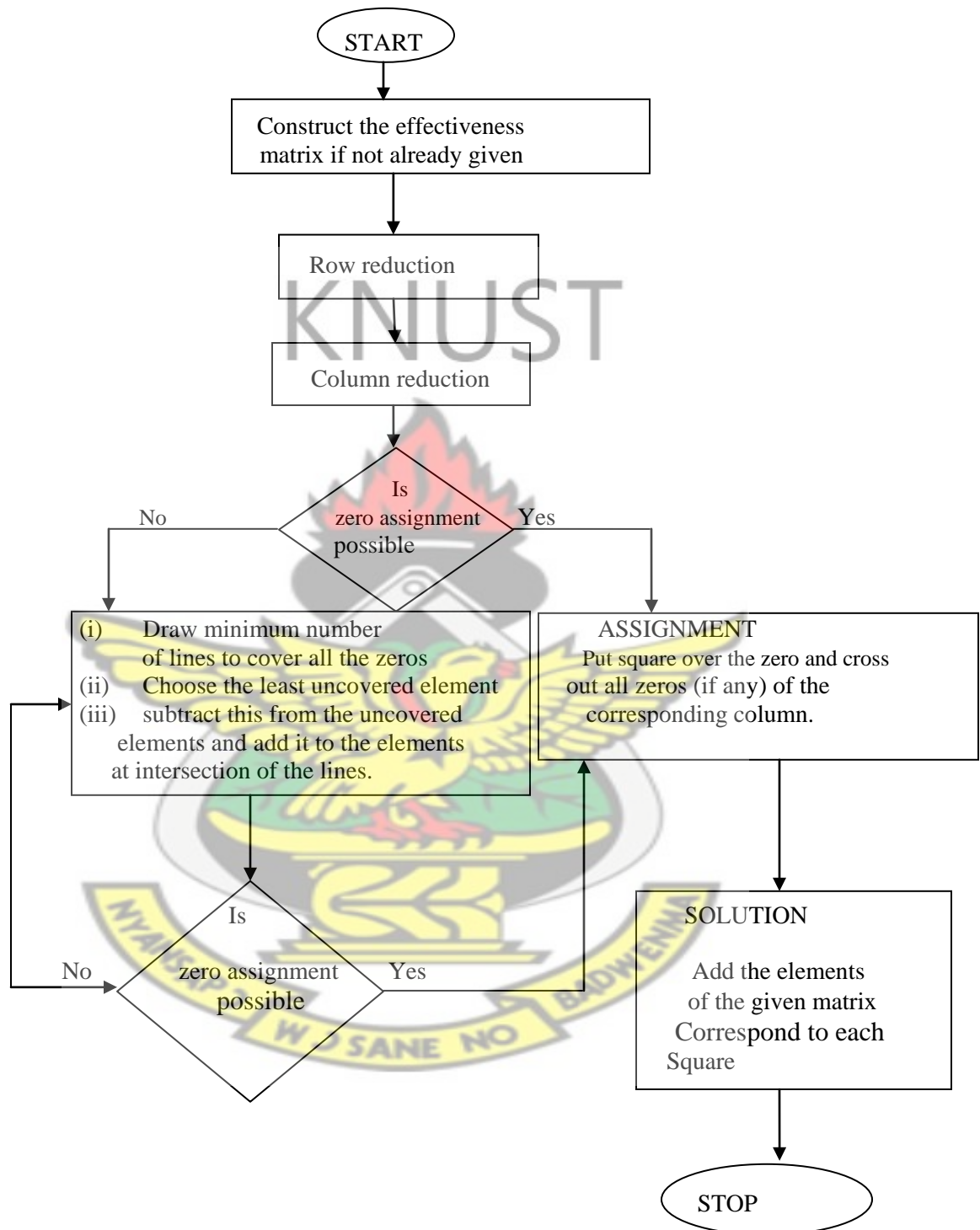
**PROOF:** Since  $S$  is a minimal cover, there is a matching  $M$  with  $\rho(M) = |S| = |X|$ . Since no vertex of  $G$  is incident on more than one edge of  $M$ , we must have that  $M$  is incident on  $2|X|$  vertices which must mean the  $M$  is incident on all vertices of  $G$ .



Now back to the algorithm for the assignment problem running on a cost matrix of dimension  $m \times m$ . The first part of the algorithm runs until a minimal cover of size  $M$  is for the zeroes in the cost matrix. The first part of the algorithm goes to termination since the sum of all the costs decreases at every stage when the minimal cover has less than  $m$  vertices. When the minimal cover is reached with  $m$  vertices, we run the Hungarian algorithm on the bipartite graph defined by the zeroes in the final cost matrix. A matching of size  $m$  is obtained to give the minimal cost assignment.



### Flow chart for Assignment problem



### 3.1.4 HUNGARIAN METHOD: ALGORITHM

**Step 1:** Prepare Row ruled Matrix by selecting the minimum values for each row and subtract it from other elements of the row

**Step 2:** Prepare column reduced Matrix by subtracting minimum value of the column from the other values of that column

**Step 3:** First row-wise assign a zero by if there is only one zero in the row and cross (X) other zeros in that column.

**Step 4:** Now assign column wise if there is only one zero in that column and cross other zeros in that row.

Repeat Step 3 and 4 till all zeros are either assigned or crossed. If the number of assignments made is equal to number of rows present, then it is the optimal solution otherwise proceed as follows.

**Step 5:** Mark (P) the row which is not assigned. Look for crossed zero in that row. Mark the column containing the crossed zero. Look for assigned zero in that column. Mark the row containing assigned zero. Repeat this process till all makings are over.

**Step 6:** Draw straight line through unmarked rows and marked column. The number of straight line drawn will be equal to number of assignments made.

**Step 7:** Examine the uncovered elements. Select the minimum.

- a. Subtract it from uncovered elements.
- b. Add it at the point of intersection of lines.
- c. Leave the rest as it is.

Prepare a New Table.

**Step 8:** Repeat Steps 3 to 7 till number of allocations = Number of rows.

## CHAPTER FOUR

### DATA COLLECTION AND ANALYSIS

#### 4.0 INTRODUCTION

In this chapter, we shall use the Hungarian algorithm to solve a Senior High School staff assignment problem.

The choice of the staff assignment model is a real life problem in the service industry. The aim is to determine the best assignment policy in the institution so that the institution gets the best results of student's performance from the various staff assign to the various subjects. The general practice is that most establishments do not have a well structured plan on how to assign subject teachers to the various subjects. Teachers are assigned by the discretion of Assistant Headmaster Administration or Departmental Heads. These methods are faulted, and are basically inefficient.

#### 4.1 Data Collection and Analysis

The Assistant Headmaster Administration or departmental head has the problem of providing teachers for the six subjects offered by his department at the highest possible level of educational 'quality'. He has a posting for six graduate teachers from Ghana Education Service (GES) who can handle at least one of the six subjects. After appropriate introspection and evaluation he has arrived at the following relative ratings (100 = basic rating) regarding the ability of each instructor to teach the six subjects, respectively, as shown in Table 4.1.

Table 4.1: Relative ratings of staff to various subjects

TEACHER	SUBJECTS					
	SOCIAL	SCIENCE	ENGLISH	MATHS	BIOLOGY	PHYSICS
A	87	78	81	79	84	80
B	82	83	76	82	78	73
C	80	78	77	76	83	69
D	86	81	87	70	77	78
E	79	86	83	75	85	77
F	83	77	82	80	84	76

The problem now is how the head should assign his staff to the courses so as to maximize the educational quality in his department.

Since the assignment problem deals with the minimization problem, the above maximization problem is reduced to a minimization problem by finding the regrets matrix, as shown in Table 4.2



Table 4.2: Regrets Matrix

TEACHER	SUBJECTS					
	SOCIAL	SCIENCE	ENGLISH	MATHS	BIOLOGY	PHYSICS
A	13	22	19	21	16	20
B	18	17	24	18	22	27
C	20	22	23	24	17	31
D	14	19	13	30	23	22
E	21	14	17	25	15	23
F	17	23	18	20	16	24

The problem can be modeled as:

$$\text{Minimize } \sum_{i=1}^n \sum_{j=1}^n C_{ij} X_{ij}$$

$$\text{Subject to } \sum_{j=1}^n X_{ij} = 1$$

$$\sum_{i=1}^n X_{ij} = 1$$

Where

$X_{ij}$  = the assignment of teacher  $i$  to subject  $j$

$C_{ij}$  = the regret cost or time of assigning teacher  $i$  to subject  $j$

Hence the problem becomes:

$$\text{Minimize } 13x_{11} + 22 x_{12} + 19 x_{13} + 21x_{14} + 16 x_{15} + 20x_{16}$$

$$18x_{21} + 17 x_{22} + 24 x_{23} + 18 x_{24} + 22 x_{25} + 27 x_{26}$$

$$20x_{31} + 22 x_{32} + 23 x_{33} + 24 x_{34} + 17 x_{35} + 31 x_{36}$$

$$14x_{41} + 19x_{42} + 13 x_{43} + 30 x_{44} + 23 x_{45} + 22x_{46}$$

$$21x_{51} + 14x_{52} + 17x_{53} + 25x_{54} + 15x_{55} + 23x_{56}$$

$$17x_{61} + 23x_{62} + 18x_{63} + 20x_{64} + 16x_{65} + 24x_{66}$$

Subject to:

$$x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} = 1$$

$$x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} = 1$$

$$x_{31} + x_{32} + x_{33} + x_{34} + x_{35} + x_{36} = 1$$

$$x_{41} + x_{42} + x_{43} + x_{44} + x_{45} + x_{46} = 1$$

$$x_{51} + x_{52} + x_{53} + x_{54} + x_{55} + x_{56} = 1$$

$$x_{61} + x_{62} + x_{63} + x_{64} + x_{65} + x_{66} = 1$$

$$x_{11} + x_{21} + x_{31} + x_{41} + x_{51} = 1$$

$$x_{12} + x_{22} + x_{32} + x_{42} + x_{52} + x_{62} = 1$$

$$x_{13} + x_{23} + x_{33} + x_{43} + x_{53} + x_{63} = 1$$

$$x_{14} + x_{24} + x_{34} + x_{44} + x_{54} + x_{64} = 1$$

$$x_{15} + x_{25} + x_{35} + x_{45} + x_{55} + x_{65} = 1$$

$$x_{16} + x_{26} + x_{36} + x_{46} + x_{56} + x_{66} = 1$$

A walk through the Hungarian algorithm with the above model gives the following values. A FORTRAN 90 code for the implementation of this is given in Appendix1.

At the end of the first iteration, the following values were obtained.

Table 4.3: Regrets Matrix

TEACHER	SUBJECTS					
	SOCIAL	SCIENCE	ENGLISH	MATHS	BIOLOGY	PHYSICS
A	0	9	6	8	3	7
B	1	0	7	1	5	10
C	3	5	6	7	0	14
D	1	6	0	17	10	9
E	7	0	3	11	1	9
F	1	7	2	4	0	8

This result is not optimal, so we can not make assignment.

The algorithm will then compute the next iterative stage, with the table values shown in table 4.4

Table 4.4: Regrets Matrix

TEACHER	SUBJECTS					
	SOCIAL	SCIENCE	ENGLISH	MATHS	BIOLOGY	PHYSICS
A	0	10	7	8	4	7
B	0	0	7	0	5	9
C	2	5	6	6	0	13
D	0	6	0	16	10	8
E	6	0	3	10	1	8
F	0	7	2	3	0	7

This result is not optimal, so we can not make assignment. The algorithm will then compute the next iterative value, with the values shown in Table 4.5

Table 4.5: Regrets Matrix

TEACHER	SUBJECTS					
	SOCIAL	SCIENCE	ENGLISH	MATHS	BIOLOGY	PHYSICS
A	0	10	7	5	4	4
B	3	3	10	0	8	9
C	2	5	6	3	0	10
D	0	6	0	13	10	5
E	6	0	3	7	1	5
F	0	7	2	0	0	4

This result is not optimal, so we can not make assignment. The algorithm will then compute the next iterative values, with the table values as shown in Table 4.6

Table 4.6: Regrets Matrix

TEACHER	SUBJECTS					
	SOCIAL	SCIENCE	ENGLISH	MATHS	BIOLOGY	PHYSICS
A	0	7	7	5	4	1
B	3	0	10	0	8	6
C	2	2	6	3	0	7
D	0	3	0	13	10	2
E	9	0	6	10	4	5
F	0	4	2	0	0	1

This result is not optimal, so we can not make assignment. The algorithm will then compute the next iterative values as shown in Table 4.7.

Table 4.7: Regrets Matrix

TEACHER	SUBJECTS					
	SOCIAL	SCIENCE	ENGLISH	MATHS	BIOLOGY	PHYSICS
A	0	7	6	5	4	0
B	3	0	9	0	8	5
C	2	2	5	3	0	6
D	1	4	0	14	11	2
E	9	0	5	10	4	4
F	0	4	1	0	0	0

This result is optimal, so we make our assignment.

Teacher A → Social Studies

Teacher B → Mathematics

Teacher C → Biology

Teacher D → English Language

Teacher E → Integrated Science

Teacher F → Physics

The total minimum regret that will maximize total educational quality is given as:



$$\text{Total} = 13 + 18 + 17 + 13 + 14 + 24$$

$$\text{Total} = 99$$

$$\text{Minimize } Z = 13(1) + 22(0) + 19(0) + 21(0) + 16(0) + 20(0)$$

$$18(0) + 17(0) + 24(0) + 18(1) + 22(0) + 27(0)$$

$$20(0) + 22(0) + 23(0) + 24(0) + 17(1) + 31(0)$$

$$14(0) + 19(0) + 13(1) + 30(0) + 23(0) + 22(0)$$

$$21(0) + 14(1) + 17(0) + 25(0) + 15(0) + 23(0)$$

$$17(0) + 23(0) + 18(0) + 20(0) + 16(0) + 24(1)$$

$$Z = 99$$

By applying their criteria of assignment to this data, the assignment below were obtained:

Teacher A → Physics

Teacher B → Mathematics

Teacher C → Social Studies

Teacher D → English Language

Teacher E → Integrated Science

Teacher F → Biology

The total minimum regret that will maximize total educational quality is given as:

$$\text{Total} = 16 + 14 + 13 + 20 + 18 + 20$$

$$\text{Total} = 101$$

## **CHAPTER FIVE**

### **CONCLUSIONS AND RECOMMENDATION**

#### **5.0 INTRODUCTION**

The staff placement and selection problem of Mampong-Akuapem Presby Senior High School as an assignment problem have been addressed. The Hungarian algorithm was used to solve the staff placement and selection problem. Our research focused on the use of the assignment problem for placement and selection of staff to a given subject in order to obtain the best quality results from a teacher. It can however be applied to any situation that can be modeled as an assignment problem.

#### **5.1 CONCLUSIONS**

This thesis seeks to solve a real-life problem of a Ghana Education Service (GES) staff placement problem using the Hungarian assignment algorithm. It was observed that the solution that gave maximum achievable results from a teacher or the minimum regret for assigning a teacher to a subject was 99.

For the data used for our analysis, the school using their criteria arrived at a total regret of 101.

#### **5.2 RECOMMENDATION**

The use of a scientific approach gives a systematic and transparent solution as compared with a haphazard method. Using the more scientific assignment problem model for the placement and selection of the schools staff to various subjects gives a better result.

Management may benefit from the proposed approach for placement and selection of staff to guarantee optimal results from staff. We therefore recommend that the assignment problem model should be adopted by the school for staff placement

## REFERENCES

1. A. Abdulkadiroglu and T.Sonmez (1998). Random serial dictatorship and the core from random endowments in house allocation problems", *Econometrica*, 66, 689.
2. A. Abdulkadiroglu and T.Sonmez (2003). Ordinal efficiency and dominated sets of assignments. *Journal of Economic Theory*, 112, 157-172.
3. A. Bogomolnaia and H. Moulin (2002). A Simple Random Assignment Problem with a Unique Solution", *Economic Theory*, 19, 623 - 636.
4. A. Bogomolnaia and H. Moulin (2004). Random Matching under Dichotomous preferences. *Econometrica*, 72, 257.
5. A. Bogomolnaia, R. Deb and L. Ehlers (2001). Strategy-proof assignment on the full preference domain. *Journal of Economic Theory*, forthcoming.
6. A. Bogomolnaia and H. Moulin (2001). A new solution to the Random Assignment problem. *Journal of Economic Theory*, 100, 295.
7. A. Hylland and R. Zeckhauser (1979), The efficient allocation of individuals to Positions", *J. Polit. Econ.*, 91, 293.
8. A.V. Goldberg and R.E.Tarjan (1998). A new approach to the maximum flow problem. 18<sup>th</sup> Annual ACM Symposium on Theory of Computing, 1986, 136.
9. Abbas Mehrabi1, and Nasrollah Moghaddam Charkari (2010). A LP-Based Polynomial Approximation Algorithm for Task Assignment Problem in Distributed Computing Systems. *International Journal of Computational Science*.

10. Abraham P. Punnen and Y. P. Aneja (1993). Categorized Assignment Scheduling. *Journal of the Operational Research Society* (1993) 44, 673–679.
11. Ahuja R.K., T.L. Magnanti and B. Orlin, (1993), *Network Flows: Theory, Algorithms and Applications*, Prentice Hall.
12. Ahuja, Magnanti and Orlin (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall.
13. Ahuja, R.K., Magnanti, T.L., and Orlin, J.B. (1989). Network flows. In: Nemhauser, G.L., Rinnooy Kan, A.H.G., Todd, M.J. (Eds.). *Handbooks in Operations Research and Management Science, Optimization*, vol. 1. North-Holland, Amsterdam, pp. 211–369.
14. Akshay-Kumar Katta and Jay Sethuraman (2004). A solution to the random assignment problem on the full preference domain. Master degree Thesis, Department of Industrial Engineering and Operations Research, Columbia University, New York, NY.
15. Amponsah S.K and Darkwa F.K, (2009), *Optimization Techniques*. pp. 164-176
16. Aronson, J.E. (1986). The multiperiod assignment problem: A multicommodity network flow model and specialized branch and bound algorithm. *European Journal of Operational Research* 23 (3), 367–381.
17. Bard, J.F. (1988). A heuristic for minimizing the number of tool switches on a flexible machine. *IIE Transactions* 20 (4), 382–391.
18. Barnier N. and P. Brisset, (2002). Graph Coloring for Air Traffic Flow Management, *Proceedings CPAIOR'02*, pp. 1-15.

19. Bertsimas D. and A. Odoni, (2000). The traffic flow management rerouting problem in Air Traffic Control: A dynamic Network Flow Approach, Transportation Science 2000 INFORMS, Vol. 34 No. 3, pp. 239–255.
20. Bertsimas D.J. and A.R. Odoni, (1998). The air traffic flow management problem with en-route capacities, Operations Research, Vol. 46, pp. 406- 422.
21. Bettina Klaus and Alexandru Nichifor (2009). Consistency and Monotonicity in One-Sided Assignment Problems. JEL classification: C71, C78, D63
22. D. Gale (1987). College course assignments and optimal lotteries, mimeo, University of California, Berkeley.
23. D.Gusfeld (1987). Computing the strength of a network in  $O(jV j3jEj)$  times.
24. Dash Optimization (2002). Xpress-Mosel: User Guide, Englewood Cliffs, NJ. 18<sup>th</sup> Annual ACM Symposium on theory of Computing,(1986),136.
25. Delahaye D. and A. Odoni, (1997). Airspace Congestion Smoothing by Stochastic Optimization, Evolutionary Programming VI, pp. 163--176.
26. Demange, G. and Gale, D. (1985). The strategy structure of two-sided markets. Econometrica, 53(4): 873 888.
27. Duong V., G. Gawinowski, J.P. Nicolaon and D. Smith, (2001). Sector-Less Air Traffic Management, The 4th ATM R and D Seminar, Santa Fe, USA.
28. Eriksson, K. and Karlander, J. (2001). Stable Outcomes of the Roommate Game with Transferable Utility. International Journal of Game Theory, 29: 555569.



29. Ford, L. R., and D. R. Fulkerson., (1958). Constructing maximal dynamic flows from static flows. *Operations Research*, 6:419-433.
30. Francis Sourd (2004). The continuous assignment problem and its application to preemptive and non-preemptive scheduling with irregular cost functions. *Inform Journal on Computing* 16(2). pp.198-208.
31. Gale, D. and Shapley, L. S. (1962). College Admissions and the Stability of Marriage. *American Mathematical Monthly*, 69: 915.
32. Giorgio Gallo, Michael D Grigoriadis and Robert E Tarjan (1989). A Fast Parametric Maximum Flow Algorithm and Applications. *SIAM J. Comput.* , 18, 30.
33. Gupta, B., (1995). A parallel multiperiod assignment algorithm. Ph.D. Dissertation, University of Georgia, Athens, GA.
34. H. Cres and H. Moulin (2001). Scheduling with Opting Out: Improving upon Random Priority. *Operations Research*, 49(4):565.
35. Helena Ramalhinho-Lourenço and Daniel Serra (1998). Adaptive approach heuristics for the generalized assignment problem.  
<http://www.econ.upf.edu/docs/papers/downloads>.
36. J.R.Brown (1979). The sharing problem. *Operation. Research*. 27, 324.

37. Kurz, M.E., Askin, R.G. (2001). Heuristic scheduling of parallel machines with sequence-dependent set-up times. *International Journal of Production Research* 39 (16), 3747–3769.
38. Letrouit V. (1998). *Optimisation du Réseau des Routes Aériennes en Europe*, PhD Disertation, INPG.
39. Maxon, S.L., Bhadury, J. (2001). An ms-excel implementation of a multi-period assignment problem with repetitive tasks. In: *Proceedings of the 13th Annual CSU-POM Conference*, California State University San Bernardino, pp. 39–48.
40. Mingfang Ni, Zhanke Yu, Feng Ma, and Xinrong Wu (2010). A Lagrange Relaxation Method for Solving Weapon-Target Assignment Problem. Thesis, Institute of Communication Engineering, PLA University of Science and Technology, NanJing 210007, China
41. Mohamad Ali Duki (2008). facility layout optimisation; metaheuristics; QAP; quadratic assignment problems.
42. Nace D. (2002). A linear programming based approach for computing optimal splitable fair routing, *ISCC'02*, Taormina-Giardini Naxos, Italy.
43. Nace D., J. Carlier, N-L. Doan and V. Duong, (2002). Using Dynamic Flow Network Modeling for Aircraft Route Assignment, the 21st Digital Avionics Systems Conference (DASC), Irvine, California, USA, 27-31 October 2002.
44. Nemhauser, G.L., Wolsey, L.A. (1988). *Integer and Combinatorial Optimization*. John Wiley and Sons, NY.

45. Nilim A, L El Ghaoui, V. Duong, and M. Hansen (2007). Trajectory-based Air Traffic Management (TB-ATM) under Weather Uncertainty, the 4th ATM R&D Seminar, Santa Fe, USA.
46. Odior, A.O.; Charles, Owaba, O.E. & Oyawale (2010). Determining Feasible Solutions of a Multicriteria Assignment Problem. Journal of Applied Sciences and Environmental Management, Vol. 14, No. 1, 2010, pp. 35-38
47. Omid Moradi (2010). Fixed Channel Assignment and Neural Network Algorithm for Channel Assignment Problem in Cellular Radio Networks. Mathematical Problems in Engineering Volume 2011 (2011), Article ID 873292, 10 pages.
48. Phen, Chiak See and Wong, Kuan Yew, (2008). Application of ant colony optimisation algorithms in solving facility layout problems formulated as quadratic assignment problems: a review. International Journal of Industrial and Systems Engineering , 3 (6). pp. 644-672. ISSN 1748-5037.
49. Rex Cheung (2011). The Geometry of the Simplex Method and Applications to the Assignment Problems. SENIOR THESIS COLLEGE OF LETTERS AND SCIENCE of the university of California.
50. Robert M. Nauss (2003). Solving the generalized assignment problem: an optimizing and heuristic approach. INFORMS Institute for Operations Research and the Management Sciences (INFORMS).
51. Roth, A. E. and Sotomayor, M. A. O. (1990). Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis. Cambridge University Press, Cambridge.

52. Sasaki, H. (1995). Consistency and Monotonicity in Assignment Problems." International Journal of Game Theory, 24: 373-397.
53. Shapley, L. and Shubik, M. (1972). The assignment game I: The core. International Journal of Game Theory, 1: 111-130.
54. Sherali H.D., J.C. Smith and A. Trani, (2000). An Airspace Planning Model for Selecting Flight-Plans under Workload, Safety and Equity Considerations, The Third International Air Traffic Management R and D Seminar ATM - 2000, Napoli, Italy.
55. Sherry J. E., C.G. Ball, S.M. Zobell, (2003). Traffic Flow Management (TFM) weather Rerouting, The 4th ATM R & D Seminar, Santa Fe, USA.
56. Solymosi, T. (1999). On the Bargaining Set, Kernel and Core of Superadditive Games. International Journal of Game Theory, 28: 229.
57. Sotomayor, M. (2003). Some further remark on the core structure of the assignment game. Mathematical Social Sciences, 46(3): 261.
58. Sotomayor, M. (2005). On the Core of the One-Sided Assignment Game.
59. T. Ichimori, H. Ishii and T. Nishida (1982). Optimal sharing. Math. Programming, 23, 341.
60. Technical report CSE-87-2, Department of Electrical and Computer Engineering, University of California, Davis, CA.
61. Thomson, W. (2009). Consistent Allocation Rules. Book manuscript.

62. Toda, M. (2005). Axiomatization of the Core of Assignment Games. Games and Economic Behavior, 53: 248.
63. Tuytens, D., Teghem, J., Fortemps, P., Nieuwenhuyze, K.V. (2000). Performance of MOSA method for the bicriteria assignment problem. Journal of Heuristics 6, 295–310.
64. Walid M. Aly, (2009). Solving the State Assignment Problem Using Stochastic Search Aided with Simulated Annealing. American Journal of Engineering and Applied Sciences.
65. WANG Li-Zhu, and LIU Yang (2010) .Reverse Point Algorithm of Assignment Problem on Assignment Less Than Jobs and Persons. Operations Research TRANSACTIONS, 2011,V15(3): 124-128.
66. Xinhui Zhang and Jonathan F. Bard (2004). A multi-period machine assignment problem. European Journal of Operational Research 170 (2006) 398–415.
67. Y.F. Wan, P. Ji, (2001). A tabu search heuristic for the component assignment problem in PCB assembly. Assembly Automation, Vol. 21 Iss: 3, pp.236 – 240.
68. Zhang X., and Bard, J.F. (2005). Equipment scheduling at USPS processing and distribution centers. IIE Transactions on Scheduling and Logistics.



## APPENDIX\_A

```
PROGRAM APPOINTMENT
parameter (NMAX= 10)
REAL C (0: NMAX, 0:NMAX)
INTEGER MP (0: NMAX, 0: NMAX)
INTEGER NP
    CALL DATA1 (NP, C)
    CALL SUBMAIN (NP, C, MP)
    CALL RESULTS (NP, MP)
END
SUBROUTINE DATA1 (NP, C)
parameter (NMAX = 10)
REAL C (0: NMAX, 0: NMAX)
    PRINT *, ' '
    PRINT *, ' LINEAR PROGRAMMING'
    PRINT *, ' '
    PRINT *, ' APPOINTMENT METHOD'
    PRINT *, ' '
    WRITE(*, 10, advance = 'no'); read *, NP
    PRINT *, ' '
    PRINT *, ' INPUT APPOINTMENT COSTS/ REGRETS OF APPLICANTS:'
DO I = 1, NP
    PRINT *, ' '
    WRITE (*, 20) I
    DO J = 1, NP
        WRITE (*, 30, advance = 'no') J
        READ *, C (I, J)
    END DO
END DO
    PRINT *, ' '
    PRINT *, ' APPOINTMENTS:'
    PRINT *, ' '
    RETURN
10 FORMAT (' NUMBER OF JOBS ? ')
20 FORMAT (' APPLICANT #', I1, ':')
30 FORMAT ('          JOB #', I1, ' ? ')
END
SUBROUTINE SUBMAIN (NP,C,MP)
parameter (NMAX = 10)
REAL C(0: NMAX, 0: NMAX)
INTEGER MP (0: NMAX, 0: NMAX)
10 CALL ZEROES (NP,C)
    CALL APPOINT (NP, C,MP, IF1)
    CALL MARK (NP, C, MP)
    CALL SUBADD (NP, C)
```



```

    CALL APPOINT (NP, C,MP, IF1)
    IF (IF1.NE.NP) GOTO 10
RETURN
END
SUBROUTINE ZEROES (NP,C)
parameter (NMAX = 10)
REAL C (0:NMAX, 0:NMAX)
    IZ = 0
    DO I = 1, NP
        XMIN = C(I, 1)
        DO J = 1, NP
            IF (C(I, J).EQ.0.) IZ = 1
            IF (C(I, J) < XMIN) XMIN = C(I, J)
        END DO
        IF (IZ .EQ.1) THEN
            IZ = 0; GO TO 100
        END IF
        DO J = 1, NP
            C(I, J) = C(I, J) - XMIN
        END DO
100 END DO
        DO J = 1, NP
            XMIN = C(1, J)
            DO I = 1, NP
                IF (C(I, J).EQ.0.) IZ = 1
                IF (C(I, J) < XMIN) XMIN = C(I, J)
            END DO
            IF (IZ.EQ.1) THEN
                IZ = 0; GOTO 200
            END IF
            DO I = 1, NP
                C(I, J) = C(I, J) - XMIN
            END DO
200 END DO
    RETURN
END
SUBROUTINE APPOINT (NP, C,MP,IF1)
parameter (NMAX = 10)
REAL C(0:NMAX, 0:NMAX)
INTEGER MP (0:NMAX, 0:NMAX)
INTEGER ZI, ZJ
    DO I = 1, NP
        DO J = 1, NP
            MP (I, J) = 0; C(0, J) = 0.
        END DO
        C(I, 0) = 0.

```

```

END DO
DO I = 1, NP
  XCASE = 999999.
  DO J = 1, NP
    IF (C(I, J).NE.0 .OR.MP(I, J).NE.0) GOTO 10
    NZ = 0
    DO K = 1, NP
      IF (C(K, J) .EQ.0.) NZ = NZ + 1
    END DO
    IF (1.0*NZ < XCASE) THEN
      XCASE = 1.0*NZ; ZI = I; ZJ = J
    END IF
10  END DO
    MP (ZI, ZJ) = 1
    DO K = 1, NP
      IF (C(K, ZJ) .EQ.0. .AND.MP(K, ZJ) .EQ.0) MP(K, ZJ) = - 1
    END DO
    DO K = 1, NP
      IF (C(ZI, K) .EQ.0 .AND.MP (ZI, K) .EQ.0) MP(ZI, K) = -1
    END DO
  END DO
  IF1 = 0
  DO I = 1, NP
    DO J = 1, NP
      IF (MP(I, J) .EQ.1) IF1 = IF1 + 1
    END DO
  END DO
  RETURN
END
SUBROUTINE MARK (NP, C, MP)
parameter (NMAX = 0)
REAL C(0:NMAX, 0:NMAX)
10  DO I = 1, NP
    N = 0
    DO J = 1, NP
      IF (MP(I, J) .EQ.1) N = 1)
    END DO
    IF (N.EQ.0 .AND. C(I, 0) .EQ.0.) THEN
      C(I, 0) = 1.; M = 1
    END IF
  END DO
  DO J = 1, NP
    DO I = 1, NP
      IF (MP(I, J) .EQ. -1 .AND. C(I, 0) .EQ.1. .AND.C(0, J) .EQ.0.) THEN
        C(0, J) = 1.; M = 1
      END IF
    END DO
  END DO

```

```

        END DO
    END DO
    DO I = 1, NP
        DO J = 1, NP
            IF (MP(I, J) .EQ. 1 .AND. C(0, J) .EQ. 1 .AND. C(I, 0) .EQ. 0) THEN
                C(I, 0) = 1.; M = 1
            END IF
        END DO
    END DO
    IF (M.EQ.1) THEN
        M = 0; GOTO 10
    END IF
    RETURN
END
SUBROUTINE SUBADD (NP, C)
parameter (NMAX = 10)
REAL C(0:NMAX, 0:NMAX)
    XMIN = 999999.
    DO I = 1, NP
        DO J = 1, NP
            A = C(I, 0); B = C(0, J)
            IF (A.EQ.1. .AND.B.EQ.0. .AND.C(I, J)< XMIN) XMIN = C(I, J)
        END DO
    END DO
    DO J = 1, NP
        A = C(I, 0); B = C(0, J)
        IF (A.EQ.1. .AND.B.EQ.0.) C(I, J) = C(I, J) - XMIN
        IF (A.EQ.0. .AND.B.EQ.1.) C(I, J) = C(I, J) + 2. * XMIN
    END DO
END DO
RETURN
END
SUBROUTINE RESULTS (NP, MP)
parameter (NMAX=10)
INTEGER MP (0:NMAX, 0:NMAX)
    DO I = 1, NP
        DO J = 1, NP
            IF (MP(I, J) .NE.1) GOTO 10
            WRITE (*,20) I, J
10      END DO
        END DO
        PRINT *, ' '
        RETURN
20  FORMAT ('      APPLICANT #', I2, ' → JOB # ', I2)
END
END OF APPOINT

```

# KNUST

