AN INVESTIGATION INTO SUPPLIER SELECTION AND CONTINGENCY FREIGHT CONSOLIDATION FOR LESS-THAN-TRUCKLOAD LOGISTICS

Rukiye Kaya

Supervisors:
Prof Said Salhi, Dr Virginia Spiegler and Prof Shaomin Wu

Word Count: 33333

June 2022

Synopsis

This study deals with the supplier selection problem in which truck companies are considered as supplier for the transportation service and freight consolidation scheduling problems for Third Party Logistic (3PL) companies.

We present two novel investigations for the supplier selection problem. In the first one, we make some analyses on the commonly used methods for supplier selection problems which are the Multi-Criteria Decision Making (MCDM) methods, namely Analytic Hierarchy Process (AHP), Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) and VIekriterijumsko KOmpromisno Rangiranje (VIKOR). Then we evaluate the results of each method based on the other two methods and conduct some tests for relying on a single method by the regret based measure approaches that we developed. In this way, we offer two effective approaches for combining the results of the individual MCDM approaches. Note that we do not propose an integration of the approaches, but a combination of the results of different MCDM methods in a systematic way instead of relying on a result of an individual method. In the second study of supplier selection, we handle the issue of missing expert knowledge. When data is not available, researchers rely on expert knowledge. Therefore, there is a tendency to use MCDM methods for supplier selection problem due to working ability of MCDM methods with expert knowledge. However, experts do not always have full knowledge of all evaluation criteria. We offer a reliable solution for this problem. We integrate MCDM methods and Bayesian Network (BN) in a novel way that they can compensate each others' limitations with their strengths. We mainly rank the alternative suppliers with TOPSIS which has two inputs: weights of the decision criteria and initial decision matrix. We obtain the weights of the criteria from AHP and elicit the initial decision matrix from BN. Causal graphical structure and parameterization of BN is done by Decision Making Trial and Evaluation Laboratory (DEMATEL). Here, experts submit their knowledge about the decision criteria linguistically. Ranked Nodes tool of BN provides the experts to submit their knowledge with verbal expressions in an ordinal scale as low, medium, high. If the experts do not have full knowledge about some of the criteria BN estimates the missing value of criteria based on the available knowledge of the experts and causal relationship between the criteria probabilistically. According to the obtained new knowledge(evidence) BN updates the values of the network and provides updated information to decision makers dynamically. Finally, we conducted sensitivity analyses for the value of knowledge followed by a case study.

In the second part of this research, we investigate the freight consolidation scheduling problem. We address the problem in a particular way due to the preference of a 3PL company that operates in the UK. We consolidate the orders up to 3. First we investigate the possible consolidation configurations of orders as singleton(one), pair and triplets. We compute all the savings obtained by consolidation among non-consolidation case. Then we use these configurations and their saving values as input in our exact approaches like the 0-1 Integer Linear Programming (ILP) and the set partitioning formulation which we developed. We also presented some tightening constraints into the set partitioning formulation and tested them for different size of the data sets. On the other hand we also tackled the problem using metaheuristics to overcome the computational time for larger instances. We offered Variable Neighbourhood Search (VNS) algorithm using six neighbourhood structures and two local searches: one performs within the route and the other one performs between the routes. The proposed neighbourhood structures are compatible with the purpose of the improvement of the consolidated shipment configurations up to three requests. On the other hand to perturb the solution and improve it with the repair mechanism we offered Large Neighbourhood Search (LNS) algorithm. In LNS algorithm, one of the removal operators performs effectively in a guided way by destroying the consolidation configurations which have negative effect on savings. We also propose to hybridize the VNS/LNS algorithms. Lastly we discuss about the computational results in terms of deviation from the optimal results and computational time effectiveness. We finalize the study with a summary of the research, limitations and suggestions for further work.

The thesis is made up of eight chapters. In the first chapter, the problem definition, a brief of the study and contributions are presented. In chapter 2, the literature review for supplier selection and order consolidation scheduling problems are given. Chapter 3 propose a deterministic rule for the combination of the results of different methods for supplier selection problem while Chapter 4 deals with the case of lack of complete expert knowledge for the supplier selection problem and proposes a novel MCDM-BN integration for this purpose. Chapter 5 discusses the order consolidation scheduling problem and defines all the possible configurations with their respective savings. In chapter 6, exact approaches for the order consolidation scheduling problem are provided, namely, a 0-1 ILP and a set partitioning formulation enhanced by valid inequalities. Chapter 7 treats the same scheduling problem by designing and implementing two metaheuristics approaches, namely, variable neighbourhood search (VNS) and large neighbourhood search (LNS) as well as their hybridisation. Chapter 8 is the final chapter, it covers a summary of our findings and present limitations of the study and outlines suggestions as future work.

Acknowledgement

I would like to express my sincere gratitude to my main supervisor, Professor Said Salhi for his continuous support, patience, encouragement, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

I would like to thank my second supervisor, Dr Virginia Spiegler for her support and insightful comments and suggestions which improved the quality and clarity of my work. My sincere thanks also for my third supervisor, Professor Shaomin Wu for his support and contributions to my work.

I am extremely grateful to my family who are always there to support and help me through difficult times.

My gratitude also goes towards the Republic of Turkiye Ministry of National Education for their sponsorship. Special thanks to Kent Business School (KBS) and the support provided by the administrative staff at the school.

I have been blessed with a friendly and enthusiastic group of fellow PhD researchers. Thanks to all my friends especially Sheri then Eirini, Shashitha, Arwa, Rania, Marina, Katie and Giorgia. You helped this adventure to be a more enjoyable one.

Finally, my thanks go to all the people who have direct or indirectly contributed to the completion of this work.

Contents

List of Tables

List of Figures

Acronyms

3PL Third Party Logistic

AC Ant Colony

AHP Analytic Hierarchy Process

ANN Artificial Neural Networks

AQM Alternative Queuing Method

BN Bayesian Network

BWM Best-Worst Method

CbR Case-based Reasoning

CI Consistency Index

DEA Data Envelopment Analysis

DEMATEL Decision Making Trial and Evaluation Laboratory

DT Decision Trees

GA Genetic Algorithm

GP Goal Programming

GRASP Greedy Randomized Adaptive Search Procedure

GST Grey Systems Theory

ILP Integer Linear Programming

ILS Iterated local search

ILSP Iterative Logistics Solution Planner

IVIUL interval valued intuitionistic uncertain linguistic

LNS Large Neighbourhood Search

LTL Less-Than-TruckLoad

MABAC Multi-Attribute Border Approximation Comparison

MCDM Multi-Criteria Decision Making

PS Particle Swarm

PSO Particle Swarm Optimization

QFD Quality Function Deployment

RBDR Rank-Based Deterministic Rule

RBRA The Rank-Based Regret Algorithm

RI Random Consistency Index

RST Rough Set Theory

SBDR Score-Based Deterministic Rule

SBDRP The incorporation of the positions of the alternatives in SBDR

SBDRPP The incorporation of the position of the alternatives and the preferences for the MCDM methods in SBDR

SBRA The Score-Based Regret Algorithm

SMART Simple Multi-Attribute Rating Technique

SWARA Step-wise Weight Assessment Ratio Analysis

TCO Total Cost of Ownership

TL TruckLoad

TOPSIS Technique for Order of Preference by Similarity to Ideal Solution

VIKOR VIekriterijumsko KOmpromisno Rangiranje

VNS Variable Neighbourhood Search

WASPAS Weighted Aggregated Sum-Product Assesment

Chapter 1

INTRODUCTION

1.1 Introduction

In this introductory chapter we first define the problem, give the methods to be used, outline the aim and contributions of the study and finally present the organisation of the thesis.

1.2 Problem Definition

We aim to investigate the decision problem in contingency logistics operations. This is based on the operations of a third party logistics company which performs in contingency logistics. We address this challenging issue by studying the supplier selection problem, which in our case the suppliers represent the freight companies (providers), and the order request consolidation scheduling problem. For the first part, we propose a selection tool based on multi criteria decision analysis that considers the selection of the transportation suppliers (the freight companies) whereas in the second part, we solve the order consolidation scheduling of the customer requests which also incorporates the presence of the transshipment points both optimally and heuristically.

The focus of the work is those third party logistic (3PL) companies that do not have their own fleet. Usually the company is faced with two interrelated challenges, namely, the selection of their suppliers known as truck carriers (providers) throughout the world while organising in an efficient way the delivery/collection schedules of the customers that require the good to be arrived urgently, known as requests. These customers are usually companies, some of them are rather big like Jaguar, that require special parts that suddenly failed and can be fatal in their production system. However, the 3PL company has to bid for the job, namely, the customer order request, as will be briefly explained below. This latter situation provides the 3PL company with the challenge of not only finding the right freight providers and also producing the right order consolidation (up to 2 or 3 requests at most) that could lead to cost saving, added advantage of gaining the bids and a by-product of reducing the level CO2 emission as a consequence of using less vehicles on the roads. It is also important that the selection of the most suitable truck companies (providers) is carefully performed as customer service is paramount for the 3PL company. This type of logistics company fall into the category of third party logistic companies that perform in the operational contingency logistic area. In other words, they are asked to collect and deliver these parts in strict due date, say next day, as any unnecessary delays will affect the production that generates massive financial losses and also may have a negative effect on their clients.

The current study was inspired from a real case study of a 3PL logistic company in the Kent region (UK) where an initial study was recently carried out by Salhi et al. (2020). The authors investigated order consolidation strategies using pairs of requests to be consolidated while also briefly examining the inclusion of the transshipment points. The aim was to maximize the profit by reducing unnecessary driving costs and also enhancing or at least retaining the customer satisfaction at their current level for this 3PL company. More details will be provided in subsequent sections and chapters. However, in this study, the selection of the truck suppliers (providers) though investigated to produce a list of chosen promising providers by addressing the supplier selection problem, it was not integrated with the order consolidation scheduling part. This is due to the assumption that there will be enough chosen providers around the collection points and hence the additional cost will be either minimal or constant throughout the search and hence will have no bearing in the scheduling decision. This linkage can be easily incorporated as will be highlighted in the limitations and suggestions in the final chapter of this thesis.

The process is briefly given as follows. The third party logistic company bids for a customer order (request) with the view that if they gain the bid they will make a profit. If the bid is too high it is likely the bid will be rejected but if it is too low, the company may lose money. In other words, the 3PL company has the challenge of offering the right quote to the customers (mother companies) with the aim of gaining the bid and yielding profit. There

Determining the price the 3PL company offers to the chosen truck company (provider) for the transportation service and delivering the product on time to the customer is a major challenge. The 3PL company if they manage effectively on how to consolidate orders can reduce the transportation cost, decrease CO2 emission and enhance customer service while increasing their chances of gaining future bids. To maintain customer service and its reputation, the 3PL company which historically delivers each customer its request on a direct route via a van or a truck, decided to explore order consolidation but with the strict condition that the maximum number of requests to be examined for consolidation in any route is three at most. To respond to this challenge, we need to study the selection of these suppliers (providers) in terms of quality, reliability etc and then address the decision problem of order consolidation scheduling accordingly. Our first task is therefore to address the selection problem of the truck company for transportation service as a supplier selection problem. These selected suppliers could then be chosen as part of the second task, namely, the order consolidation scheduling where we assume that there will be enough providers near the collection points to choose from. Additionally, in this study we also consider the selection of transshipment points and we extend the work to triplets in addition to pairs as initially attempted by Salhi et al. (2020). The methodology adopted here is also very different as will be shown in the subsequent chapters.

In summary, we handle supplier selection problem and order consolidation scheduling problems for third party logistic companies.We aim to select the best freight company as the transportation service supplier for third party logistics companies. We evaluate and rank all the suppliers based on the region. Then we aim to schedule the orders as they can be consolidated up to 3 in most effective way by consideration of transshipment points.

1.3 Methods adopted in this study and how they are implemented

The following approaches are used in this work. These include (i) methods based on selection rules that are used in multi criteria decision analysis, (ii) exact methods that are based on mathematical programming with a focus on 0-1 programming, and (iii) metaheuristics.

In (i) methods such as AHP, TOPSIS and DEMATEL will be incorporated into our new methodology as well as Bayesian Network(BN) used with Ranked Nodes tool in AgenaRisk 10 Academic Desktop software to tackle lack of knowledge. In (ii) methods based on Branch & Bound, Cutting Planes, Branch/Cut/Price including the new advances in mathematical programming form the basis of the optimisation software IBM ILOG CPLEX studio 12.9 that will be used here. In (iii) two powerful and known metaheuristics such as Variable Neighbourhood Search and Large Neighbourhood Search will be explored. A brief review of the above techniques is given in the next chapter.

The algorithms in this study, the multi criteria decision making(MCDM) methods used and regret based evaluation algorithms we developed in the supplier selection chapters, the use of CPLEX in solving the mathematical models that we developed and the metaheuristics that we constructed, are all coded in Visual Studio C++ and executed on an Intel Core i5-7300U CPU @ 2.60GHz 64-bit operating system with 8 GB RAM. The data generation in both parts, namely, the supplier selection and the order consolidation scheduling are also coded and executed in the same way.

1.4 Aims and Contributions of the study

In this study, we aim to explore the time critical scheduling problem where the company operator needs to provide a quick response to the customers on how much the cost is with the objective of gaining the bid as the customer may or may not choose this particular third party logistic provider. In other words if the price is too high there may be a risk of losing the customer whereas if it is too low the company may lose money. To respond to this need we aim to investigate the selection of those suppliers (truck providers) while producing efficient schedules that could reduce travel time wastage and be environmentally attractive due to resulting relatively cleaner air.

In the first part of the study, we investigated the supplier selection problem as truck companies are the transportation service suppliers for third party logistics companies. We conducted two studies for supplier selection problem. In the first one, we made some analyses on relying on an individual method with common used MCDM methods for supplier selection problem, namely AHP, TOPSIS and VIKOR. We developed a regret based measurement approach for evaluating the results of each method based on the other two methods. We found out that in different tests, different methods outperform the other methods and relying on an individual method is not reliable. Then we developed some deterministic rules for the combinations of the results of the individual methods. We also tested the results of these rules against the results of individual ones by regret based measurement approach we developed and found out that the proposed deterministic rules which combine the results of the individual methods outperform the individual methods. Note that despite the common practice of integration of methods, we offer to combine the results of the methods. In the second study for the supplier selection problem, we tackle incomplete expert knowledge problem. As data is not always available, researchers use expert knowledge for the decision making problems. Therefore there is a tendency to use MCDM methods due to their ability to work with expert knowledge. However, full expert knowledge is also not available for all the criteria. We offer a novel approach for decision making based on incomplete expert knowledge. In the proposed approach, we rank the suppliers with TOPSIS. TOPSIS needs weights of the decision criteria and initial decision matrix as inputs to perform. We obtain the weights of the decision criteria from AHP and elicited the initial decision matrix by BN. We use DEMATEL for building causal graphical structure and parameterization of BN as proposed by (Kaya & Yet 2019). In supplier selection problems the experts prefer to submit their knowledge with linguistic expressions instead of dealing with complicated mathematical expressions. Therefore there is a tendency to use fuzzy approach for the elicitation of expert knowledge with linguistic expressions which is still a bit complicated for experts. We offer to use Ranked Nodes tool of BN which allows experts to submit their knowledge with linguistic expressions in an ordinal scale as very low, low, medium, high and very high. In addition to this, if the experts have some knowledge/belief about some criteria and do not have about some other criteria, BN estimates the missing values of the other criteria based on the cause-effect relationship between the criteria probabilistically in a systematic way. Therefore using BN with Ranked Nodes tool expert friendly alternative approach to fuzzy approach with additional aspects of consideration of causal relationship between the criteria and estimation of missing knowledge. When there is a new knowledge(evidence) about any criterion, BN updates all the network based on the obtained evidence dynamically and propose updated results. The proposed approach can be adapted to any multi-criteria decision making problem.

In the second part of the study, we examine the order consolidation scheduling problem. As we tackle the particular case of a 3PL company which prefers to consolidate orders up to 3, we first define all possible consolidation configurations as pair and triplet consolidation with and without transshipment case. Then compute all the savings gained among the non-consolidation case for all configurations. These computations are carried out beforehand and given to the exact approaches. In the common practice of formulations all these consolidation possibilities are computed inside the formulation and for each iteration. However we make it out of the formulation once and only. As exact approaches, we offer set partitioning formulation with valid inequalities(tightenings). We found interesting computational results. For the computational time concerns for large sizes, we offer metaheuristic approaches, VNS, LNS and VNS/LNS hybridization. We offer six neighbourhoods which are designed as they systematically extend from first neighbourhood to sixth neighbourhood compatible with the problem structure and two local searches:one works within the routes(consolidated orders) and find the best consolidation configuration for the candidate orders to be consolidated and the second local search works between the routes to find the best routes. We use first local search inside the second local search as well. Computation of saving of the consolidation configurations is done only for the candidate route(consolidated orders) not for all possible routes. This brings significant computational time saving among the exact approaches. On the other hand to perturb the solution and repair the ineffective parts of the solution we propose LNS algorithm which works in guided way and eliminates the routes which bring negative savings. Lastly we hybridize the VNS/LNS as to perturb and improve the solution of VNS with LNS by running LNS just after VNS. We present computational results for different size of instances. Finally we discuss the limitations of the study and propose suggestions as future avenue.

The contributions of the study can be summarised under the following items

(i) A novel integrated multi criteria decision making (MCDM) method is developed and analysed.

(ii) As the presence of full expert knowledge may be lacking in practice, an efficient integration of probabilistic BN and deterministic MCDM methods is designed and computationally tested.

(iii) Data set generators are constructed. For (i), a data set generator that reflects several criteria is constructed to provide a set of instances for empirical testing. For(ii), a survey for the supplier selection process of a forging company is conducted. Another generator designed for the scheduling problem is also developed to construct data sets with 50 to 200 requests with a step size of 50. Ten instances for each class are generated which are used in testing our exact and heuristic approaches that are outlined below.

(iv) For the scheduling problem, all the configurations for pairs and triplets with and without transshipment points with their respective costs are identified. These computations are done at the outset and once only and given to exact approaches as input despite the general approaches in the literature carry out all these computations in each iteration.

(v) A set partitioning formulation for the problem is first developed which is then tightened by creating new valid inequalities. Comparison with classical 0-1 ILP using CPLEX is also performed.

(vi) Meta-heuristics such as Variable Neighbourhood Search (VNS) and Large Neighbourhood Search (LNS) are also developed for the same scheduling problem to provide another way of looking at the problem. Here, appropriate neighbourhood structures, local searches, removal and repair operators are constructed. Various variants are also examined.

(vii) Finally, some limitations as well as potential research avenues that we believe to be worth examining are outlined.

1.5 Structure of the thesis

The rest of the thesis is organised into seven chapters. Chapter 2 deals with the review for both supplier selection and order consolidation scheduling. Chapter 3 develops a new integrated MCDM approach based on individual approaches whereas chapter 4 treats the case when there is lack of knowledge using Bayesian network. Chapter 5 discusses the order consolidation scheduling problem and defines all the possible configurations with their respective costs. This is followed by chapter 6 that provides a mathematical formulation based on standard integer programming with a focus on set partitioning and valid inequalities. Chapter 7 treats the same scheduling problem by designing and implementing two metaheuristics approaches, namely, variable neighbourhood search (VNS) and large neighbourhood search (LNS) as well as their hybridisation. Chapter 8 is the final chapter and covers a summary of our findings, highlights some the limitations of the study while outlining some potential suggestions that we believe to be useful to investigate in the future.

1.6 Summary

In this short introductory chapter, we define the problem, briefly outline some of the techniques that will be used, present the aims and the potential contributions and provides the structure of the thesis. The next chapter will present the literature review.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

This chapter covers the literature review. We first provide those studies on the supplier selection related to logistics, followed by the work done in the order consolidation scheduling problem. A brief of the methods adopted in this study will also be presented at the end of this chapter.

2.2 Supplier Selection

We classified the reviewed supplier selection studies under three different categories; individual approaches for supplier selection, integrated approaches for supplier selection and sustainability in supplier selection.

We first examined the common used individual methods in supplier selection mainly based on the insights of two very detailed literature review studies in supplier selection (Chai et al. 2013, Ho et al. 2010). Though these are relatively not recent, their classification of the methods and contribution still reflects the new advances in this area as will be shown in this section.

Chai et al. (2013) reviewed supplier selection studies between 2008 and 2012. They review the methods used for supplier selection effectively. They pointed out the uncertainty of decision environment in the supplier selection problem and the dominance of fuzzy hybridization among the approaches. They classified methods used for supplier selection under three categories: Multi Criteria Decision Making (MCDM), Mathematical Programming (MP) and Artificial Intelligence (AI) methods. According to this review, mostly used methods for the supplier selection problem are MCDM methods, which include AHP, ANP (Analytic Network), TOPSIS, ELECTRE (Elimination and Choice Expressing Reality), DEMATEL, VIKOR (ViseKriterijumska

Optimizacija I Kompromisno) and Simple Multi-Attribute Rating Technique (SMART). MP techniques are also common in supplier selection problems and include Data Envelopment Analysis (DEA), Linear Programming and Multi-objective Programming. Most of MP methods need data as Integer Programming, Data Envelopment Analysis. However, in most of supplier selection problems, data is not always available or it can be even sparse. In addition to this, the preferences of decision makers are important for supplier selection decision. MCDM methods allow using preferences of decision makers and having the ability to work without data. According to Chai et al. (2013), mostly used AI methods for the supplier selection problem are Genetic Algorithm (GA), Grey Systems Theory (GST), Artificial Neural Networks (ANN), Rough Set Theory (RST), Bayesian Networks (BN), Decision Trees (DT), Case-based Reasoning (CbR) and Particle Swarm Optimization (PSO).

Another earlier literature review was conducted by Ho et al. (2010). The authors reviewed multi-criteria decision making approaches for supplier evaluation and selection between 2000-2008. Their aim is to find out the common approaches and criteria, and limitations of the approaches by offering future research areas. They reviewed 78 articles and analysed the studies under two main categories: individual approaches and integrated approaches. According to their review, individual approaches (58.97%) are preferred more than the integrated approaches (41.03%). Most common individual approaches are DEA, Mathematical Programming, AHP, CbR, fuzzy set theory, SMART and GA. The authors explained the popularity of DEA was due to its robustness. On the other hand, the authors presented that the original DEA was able to analyse quantitative data. It was then revised, as it is also able to analyse qualitative data for the supplier selection problem.

On the other hand, the authors found the common approach in integrated approaches is AHP. According to the authors, the motivation to prefer AHP is its simplicity, flexibility, ease of use and consistency check mechanism. They emphasized the consistency check and feedback mechanism as supplier selection decision requires judgements of decision makers. This is important as an elicitation of consistent judgements and a revision of the judgements are essential for reliable decision-making. Additionally, they present that the AHP- Goal Programming (GP) integration is a popular integration. AHP is not able to evaluate the resource constraints as the capacity of the suppliers and integration of GP considers these sort of criteria. The authors explain the usage of integrated approaches, as it is useful to take advantage of the strength of different approaches.

Another aim of this review is to find the most popular decision criteria in supplier selection decision. It is found that three most popular criteria are quality, delivery and cost/price. The authors remarked that the cost/price criterion is not the most popular criterion unlike the earlier observation given in the past studies.

Lastly, the authors present the limitations of the most common individual approaches: DEA and integrated AHP-GP approach. The first limitation of DEA is the potential confusion of the decision-makers about discrimination of input/output criteria. The second limitation is subjective judgement of decision-makers and a lack of a consistency check mechanism. The authors also criticized the integrated AHP-GP approach due to its inefficiency in terms of time consumption. AHP needs to evaluate alternatives under each criterion and sub-criterion, and in case of consistency, evaluations must be revised. Due to these constraints, it requires long time computation. It was also highlighted that the stakeholders are not considered as part of the model. One interesting avenue is to integrate the AHP-Quality Function Deployment (QFD) approach as a future research.

In this study, we reviewed the articles in the supplier selection under three categories: individual supplier selection approaches, integrated supplier selection approaches and sustainability in supplier selection. We added the last aspect due to the importance of sustainability in supplier selection in recent years.

Individual Approaches for Supplier Selection Problem

According to Chai et al. (2013), the most popular MCDM method for the supplier selection problem is AHP. AHP works based on the pairwise comparison matrix and subjective judgement of experts (Saaty 2008). Experts submit their relative preferences between alternatives based on evaluation criteria. Priority values of criteria are computed based on the pairwise comparison of these criteria. On the other hand, priority values of alternatives are calculated according to pairwise comparison of alternatives based on the criteria. As it works with subjective judgements it checks consistency of decision makers with consistency ratios. If the consistency of the judgements of decision makers is not sufficient, the pairwise decision matrix is revised. Since AHP considers both quantitative and qualitative criteria, it is advantageous for the supplier selection problem (Kilincci & Onal 2011). It is effective to deal with uncertainty in the supplier selection problem. Moreover, it allows to use judgement of multiple experts. However, it has disadvantages: It is not always possible to keep the judgements consistent; another disadvantage is its inability to consider the relationship between the criteria. AHP has a hierarchical structure and evaluates alternatives based on each level. In case there is a need to add a new criterion, all pairwise comparisons in every level must be revised. It is therefore not an effective method for the problems which have dynamic nature. This approach will be considered in our methodology in the next chapter.

Integration of AHP with fuzzy logic is a very common hybridization approach. By integrating AHP with fuzzy logic, decision makers assign the relative preferences between criteria and alternatives by linguistic values as recently and successfully carried out by Ecer (2022), Ho et al. (2021) who used fuzzy AHP for supplier selection.

ANP is a variant of AHP where the hierarchical structure works in opposite way (Aguezzoul 2014). The authors used ANP for selection of logistic service providers (3PLs). According to the authors, building pairwise comparison matrices is complex and time-consuming. In case of inconsistency, it is even more challenging.

TOPSIS is another common MCDM approach for the selection problem. It works based on similarity to obtain ideal solutions (Rodrigues et al. (2014)). It finds out the ideal and negative-ideal alternatives for each criterion and calculates the distance to ideal and negative-ideal alternatives to rank the alternatives. It considers benefit and cost criteria in their clusters. The ideal alternative for the benefit criteria as quality criterion is the alternative with the highest score and the negative ideal is the alternative with the lowest score. On the other hand, the ideal alternative for cost criteria as a price criterion is the alternative with the lowest score and negative-ideal alternative is the alternative with the highest score. When data is available, data is used as input of initial decision matrix. However, in case of data sparsity, TOPSIS is able to work with expert judgement. Lei et al. (2020) used TOPSIS for supplier selection with linguistic information. Fuzzy integration is also common for TOPSIS as other fuzzy hybrid MCDM methods with using expert knowledge. For instance, an interesting study by Memari, Dargi, Akbari Jokar, Ahmad & Abdul Rahim (2019) used fuzzy TOPSIS for sustainable supplier selection where they opted to use intuitionistic fuzzy numbers to handle uncertainty and provide experts to submit their knowledge by linguistic expressions.

Both AHP and TOPSIS are two common fuzzy hybridized MCDM methods in the supplier selection literature. Rodrigues et al. (2014) compares TOPSIS and AHP for supplier selection decision on the basis of adequacy to changes of alternatives or criteria, agility in the decision process, computational complexity, adequacy to support group decision making, the number of alternative suppliers and criteria and modelling of uncertainty. The methods were applied on the automotive manufacturer company. They have five alternative suppliers and five decision criteria. It is expected that the method gives consistent results after the addition or deduction of alternatives. Adequacy to changes of alternatives were tested with five different scenarios. In each scenario, a new supplier alternative was added with the equal rate of the current alternatives. In Fuzzy AHP, when the new alternative has the same rating with the best current alternative, the worst alternative became the best alternative. The same scenarios were tested with Fuzzy TOPSIS, contrary to Fuzzy AHP, ranking of alternatives do not show significant changes. This demonstrates TOPSIS is adequate to changes of alternatives. In addition to adequacy test to changes of alternatives, changes of criteria tested with addition and exclusion of new criteria with equal weight to each criteria in five scenarios. Similar results with the adequacy to changes of alternatives were obtained. In Fuzzy AHP, inclusion and exclusion of new criteria cause significant changes in results. Fuzzy TOPSIS shows consistent results with those obtained prior to the addition of new criteria. Due to the agility in the decision process, the technique used for supplier selection decision is desired to require the minimum amount of judgement. Fuzzy TOPSIS needs less judgement than the Fuzzy AHP. Depending on the technique, computation time of the algorithm changes and the technique with the less computation time is preferred. TOPSIS algorithm run in less time compared to AHP. In some cases with the increase of alternatives, AHP performs better than TOPSIS. The method enables to consider judgements of multiple decision makers is more preferable. Fuzzy AHP and Fuzzy TOPSIS consider judgements of different decision makers. However, as the increase of the number of decision makers, TOPSIS becomes advantageous compared to AHP depending on the increase of computation time. The number of criteria and alternative suppliers is another important factor for supplier selection decision. While TOPSIS has no restriction for the number of criteria, AHP is used for only the case with limited criteria. However, TOPSIS does not allow evaluating sub-criteria. TOPSIS will also be adopted in our study.

VIKOR is another MCDM approach for the supplier selection problem. Recently, Wu et al. (2019) used VIKOR under fuzzy environment for green supplier selection. It works based on the compromise of the decision makers. It aims to have a minimum regret and a maximum utility. It works similarly to TOPSIS. Chai et al. (2013) categorize TOPSIS and VIKOR as compromise solution approaches under MCDM category. Opricovic (2004) compares and presents differences between VIKOR and TOPSIS based on evaluation metrics and a normalization method. VIKOR ranks the alternatives based on the compromise function, which depends on regret and utility measures. On the other hand, TOPSIS ranks the alternatives based on the distance to the ideal solution. VIKOR uses a linear normalization, whereas TOPSIS uses a vector normalization. VIKOR is also used in our study.

ELECTRE and PROMETHEE methods are other used approaches for supplier selection problem. They work based on the dominance of the alternatives (Qu et al. 2020, Tong et al. 2022).

DEMATEL is one of the MCDM methods, which is used to determine the cause effect relationship between the criteria (Li et al. 2020). However, it is not able to rank the alternatives. Hence, in supplier selection problems, it is integrated with other approaches to rank the alternative suppliers ((Zhang et al. 2021)).

Metaheuristics and AI methods are also used for supplier selection problem. Che et al. (2021) very recently used the multi-objective genetic algorithm for supplier selection. Another effective AI technique for supplier selection is Bayesian Networks. It is able to deal with uncertainty probabilistically. It visualises causal relations between factors and the effects of them on each other dynamically (Fenton and Neil, 2012). For example, when a buyer has no historical data about the supplier but she has a belief about the supplier, BN allows using expert knowledge. Ferreira and Borenstein (2012) integrated fuzzy logic with BNs to provide updated performances of the suppliers. The authors constructed a large number of simulation modules, which include Purchasing Strategy Module, Decision Network Module, Database Module, Enterprise Database, Fuzzy Module and Supply Chain Simulator. This approach was used in a case study carried out in a biodiesel production plant in Brazil. Sener et al. (2021) used Bayesian belief networks to evaluate the supplier performance in aircraft manufacturing supply chain in US. In this research we will also extend an approach based on BNs.

As using individual approaches bring some limitations, researchers tend to integrate different methods. Integrated approaches are presented in the next section.

Integrated Approaches for Supplier Selection Problem

In recent years, there is a trend to use integrated approaches for the supplier selection problem. We first reviewed the integrated MCDM approaches due to prevalence of them in the literature and also as we will put forward a novel approach that fits in this category.

Wang et al. (2009) used fuzzy hierarchical TOPSIS for supplier selection. They offered an alternative way to fuzzy TOPSIS, elicit fuzzy weights from fuzzy AHP and integrates with fuzzy linguistic scores into TOPSIS and rank the alternatives. They compared their proposed approach with AHP, TOPSIS and fuzzy TOPSIS and concluded that the proposed approach outperforms these methods. Recently, Jain et al. (2018) integrated fuzzy AHP and TOPSIS for the supplier selection problem in an Indian automotive industry. Abdel-Basset et al. (2019) used integrated neutrosphic ANP and VIKOR for sustainable supplier selection. They use triangular neutrosphic numbers (TriNs) for submission of expert's opinion. Moreover, the neutrosphic approach helps deal with incomplete information and imprecision. They use ANP for the calculation of the weights of the criteria and VIKOR for ranking the alternatives. Ortiz-Barrios et al. (2020) integrated fuzzy AHP, fuzzy TOPSIS and fuzzy DEMATEL for forklift supplier selection whereas Mohammed, Harris & Dukyil (2019) integrated DEMATEL, ELECTRE and TOPSIS is adopted for vendor selection.

According to (Chai et al. (2013)), MCDM methods are effective to work with expert knowledge when data is not available or limited. However, they are not able to give dynamic and updated results. MP techniques are effective methods as they give exact and objective solutions. AI techniques are effective techniques for dealing with uncertainty and giving updated and dynamic solutions. In summary, the integration of MCDM, MP and AI methods provide more effective methods in which the limitations of each method were compensated by each other.

For example, Mohammed, Harris, Soroka & Nujoom (2019) integrates fuzzy MCDM and fuzzy multi-objective programming approach for supply chain design while very recently Dohale et al. (2021) integrates Delphi-MCDM and BNs for a production system selection problem.

Dogan & Aydin (2011*a*) integrated the BN and Total Cost of Ownership (TCO) method for the supplier selection problem. A model based on BN was built including supplier selection criteria and factors related to criteria and lastly cost items connected with factors. It provides an evaluation of the suppliers based on both qualitative and quantitative criteria and their causal relations between the cost items. In this approach, TCO evaluates the supplier selection performance in terms of the total cost and also the other costs arising from the other criteria, unlike the tendency of evaluation based on only unit price in previous TCO studies. Both financial data and domain knowledge were used in this approach. The integrated approach was applied for a tier-1 supplier automotive sector. Kaya & Yet (2019) integrated DEMATEL and BN for the supplier selection problem. The causal graph of the BN and parameterization of the graph is done by using DEMATEL. Then suppliers are evaluated based on the selection criteria based on the causal relationship between them probabilistically by BN.

In the next section, we discuss the sustainability concept in supplier selection problem.

Sustainability in Supplier Selection

In recent years, due to legal obligations and increasing environmental awareness, most of the supplier selection studies turned into sustainable supplier selection and green supplier studies. Govindan et al. (2015) reviewed multi criteria decision making approaches for green supplier evaluation and selection literature from 1997 to 2011. They investigated which selection approaches and criteria are common and checked the limitations of the used approaches. Thirty-three articles were reviewed where, most of the studies are fuzzy based single approaches. They classified the papers based on methodology and selection criteria. They classified the papers as individual approaches and integrated approaches. The authors found that researchers who use individual approaches integrate their approaches by fuzzy logic to deal with uncertainty from human judgement. On the other hand, 22.2% of the papers use integrated approaches as the researchers and experts find more complicated integrated approaches than the individual approaches. In criteria based classification of the review, the authors found that the 30.55% of the papers used environmental management systems. This shows most common criteria for green supplier evaluation and selection is environmental management. The authors emphasize that as this criterion and its subcriteria are qualitative criteria, they need subjective evaluation. One of the most remarkable findings in this review is the prevalence of the fuzzy logic that is used in this area, which can be due to the uncertainty and ambiguity of the supplier selection decision analysis. Another important finding of the authors is the tendency of the researchers to use individual approaches as shown by 25 of the papers (77.7%). One of the reasons is that researchers and experts find it complicated to use integrated approaches and they want to focus on individual approaches. 8 of the reviewed papers focus on individual approaches. In this thesis, we will explore the different individual approaches where 8 of the reviewed papers (22.2%) use integrated approaches and offer more realistic approaches. According to the authors, an integrated approach must be adopted by researchers and explained to the experts clearly. Therefore, integrated approaches are expected to be expert friendly particularly in need of human judgement for decision. In this study, we also propose, in the fourth chapter, an approach that uses the integration of MCDM with BNs based on Ranked Nodes. According to this review, the most popular individual multi-criteria decision making approach is AHP including fuzzy integration papers with 16.6%. According to the authors, AHP is the most common approach due to the following reasons: Firstly, it considers the quantitative and qualitative criteria in the evaluation process, secondly, it is able to handle uncertainty and subjective judgement of the decision makers and it is easy to understand mathematically by the experts and provides flexible and robust solutions. Moreover, its consistency check mechanism provides experts with the opportunity to submit consistent judgements. However, in most of the studies that use AHP, the fuzzy approach is integrated in the elicitation of the expert knowledge. The authors criticize this philosophy since the AHP approaches which were integrated by fuzzy logic do not give different results than in the case of the utilization of AHP approach without fuzzy logic. Therefore, the authors claim that the integration of fuzzy logic does not make contribution apart from causing extra complexity. They offer this claim as a potential future research avenue. Another future research insight of the authors is the categorization of the criteria by using a methodological approach as an exploratory factor analysis. The authors criticize that most of the studies do not give insight why some suppliers are best and others are worst and also how some suppliers could improve. In the fourth chapter of this thesis, we will investigate this issue by incorporating probabilistic evaluation of the alternatives.

In recent years, particularly in 2019, researchers have been concentrated on the selection of sustainable suppliers (Yu et al. 2019, Memari, Dargi, Akbari Jokar, Ahmad & Abdul Rahim 2019, Wu et al. 2021). Although, the used approaches are the same with the approaches that are used for the general supplier selection problem, the sustainable supplier selection problem emphasizes the need of using decision makers' judgements as the sustainable supplier selection problem consists of more social and environmental criteria. For instance, Liu, Quan, Li & Wang (2019) propose an MCDM model composed of the Best-Worst Method (BWM) and Alternative Queuing Method (AQM) within the interval valued intuitionistic uncertain linguistic (IVIUL) setting for sustainable supplier selection problem. In this model, IVIUL-BWM is used to calculate the weights of the criteria and IVIUL is used to rank the alternative sustainable suppliers. The authors emphasize the tendency of decision-makers to evaluate suppliers with their fuzzy knowledge by linguistic expressions. They also offer to handle the uncertainty and vagueness of the decision-makers by using interval valued intuitionistic uncertain linguistic sets. Interesting limitations of the proposed approach are also presented; one is that the final decision is highly dependent on the subjective judgement of the decision makers. One way to overcome this issue is to increase the number of decision makers for this limitation. Amin et al. (2019) used the fuzzy grey cognitive map to analyse the interdependencies between the criteria for the grey system theory based green supplier selection model for uncertain environments. They also use the best-worst analysis method to determine the weights of the criteria. Memari, Dargi, Reza, Jokar, Ahmad, Rahman & Rahim (2019) use fuzzy TOPSIS for the selection of sustainable supplier selection as decision maker's knowledge is imprecise and vague.

Gupta et al. (2019) used the fuzzy approach with AHP, TOPSIS, Multi-Attribute Border Approximation Comparison (MABAC) and Weighted Aggregated Sum-Product Assesment (WASPAS). Pishchulov et al. (2019) propose to use a revised Voting AHP approach for the sustainable supplier selection problem. The Voting AHP approach was originally developed by Liu & Hai (2005) for the supplier selection problem. It is composed of AHP and DEA approaches. In AHP, the weights of the criteria are determined based on the pairwise comparison of the criteria. However, this procedure is challenging when there are many criteria. The voting AHP approach utilizes DEA for this purpose where DEA derives the weights of the criteria based on the ordinal preferences of the decision makers instead of the complicated pair-wise comparison procedure. However, Pishchulov et al. (2019) adjusted this approach by using a game-theoretic approach for the derivation of criteria weights to handle subjectivity and arbitrariness of rank discrimination.

The Positioning of Our Proposed Supplier Selection Methods

As can be seen from the literature review on the supplier selection, different approaches under different categories perform differently. Hence, we first offer to combine the results of different approaches to construct an effective deterministic rule that will be presented in chapter 3. Govindan et al. (2015) state that researchers might be biased by one approach. In our proposed deterministic approach, we offer to combine the results of different approaches with a deterministic approach to prevent the biasedness that might come from using one approach. The authors also propose to investigate the acceptance level of the approaches by the decision makers and building experimental designs. We also consider in chapter 3 the preference level of the decision-makers for each individual method that is integrated with the other methods.

There is much tendency to fuzzy hybridization because of the uncertainty and vagueness of the expert judgement. However, fuzzy logic is still complicated to understand by decision-makers in practice. We offer an alternative and more expert-friendly approach to fuzzy, Ranked Nodes in BNs. Ranked Nodes allows experts to submit their judgement in linguistic ordinal scale as low, medium and high. Moreover, BNs consider causal relationship between the criteria and compute the performances of suppliers based on criteria probabilistically. It has the ability to work with data and expert knowledge, even missing knowledge. When experts have missing knowledge about some of the criteria, BN estimates the probabilities of the missing ones based on the causal relationship between the criteria. By obtaining any new evidence, it updates the network and shows the updated performances of suppliers based on each criterion. However, it does not rank the alternatives. For this purpose, we offer to integrate with TOPSIS to rank the alternatives. TOPSIS needs weights of the criteria, we offer to elicit the weights of the criteria from AHP. The causal graph of BN is also built by DEMATEL systematically. As a result, we propose an effective integrated MCDM and BN approach for supplier selection decision. Most important distinction of this approach is Ranked Nodes as being an effective alternative to the fuzzy approach. The proposed approach will be presented in chapter 4.

Govindan et al. (2015) also emphasize the importance of expert friendly approaches in need of human judgement. The BN approach based on ranked nodes makes it easy to elicit the expert knowledge. Experts are only needed to submit their opinion about the criteria and alternatives with linguistic variables as low, medium and high. After that, BN computes the probability of alternatives based on the criteria mathematically. We offer to use AgenaRisk software as researchers and experts do not have to deal with the mathematical background of the approach and they can see the updated results immediately. In addition to this, the proposed approach considers the causal relationship between the criteria. Due to these causal relationships, BN is able to estimate even missing criteria of the alternatives. The authors propose to investigate the advantages and disadvantages of using multi-methods as a future research. The authors also criticize that most studies do not give insight why some suppliers are best and worst. BN visualises the evaluation of each supplier based on each criterion probabilistically. The probability values of the suppliers are updated with the new evidences by the time. This also visualises the improvement opportunities of the suppliers. Another important finding of the authors is the importance of sensitivity analysis. The software AgenaRisk which we use in our approach is able to make effective sensitivity analyses as will be shown in chapter 4.

2.3 Order Consolidation Scheduling

Freight consolidation has attracted recently research attention due to the financial benefits of maximization of the profit as well as the reduction of CO2 emissions due to a reduction in unnecessary distance travelled. We mainly conducted our review based on freight consolidation and some related vehicle routing studies such as the pickup and delivery routing problem, the capacitated vehicle routing problem, the vehicle routing problem with transshipment, and the vehicle routing problem with time windows. Some of these variants have some similarities with our study. On the other hand, we mainly focussed on the approaches that are commonly used in these studies such as metaheuristics.

First we start our research with freight consolidation especially for Less-Than-TruckLoad (LTL) operations. Mesa-Arango & Ukkusuri (2013) search the benefits of in-vehicle consolidation less than truckload freight transportation operations. They found out that in auctions for the procurement of freight transportation services, most likely, consolidated bids, by LTL operations, from the carriers take the job by discarding the carriers which offer non-consolidated bids as these prefer to use full TL with direct transportation resulting in high cost profile. The authors also found out that, the customers who prefer the TruckLoad (TL) companies prefer them because of the delivery time concerns as these tend to guarantee customer service by arriving on time. However, mostly LTL operations offer longer delivery times. The authors investigate the benefits of the consolidation in competitiveness and the challenges that could arise in less than truck load transportation. They define this problem as multi-commodity one-to-one pick-up and delivery vehicle routing problem and offer to solve by branch-and-price algorithm. They obtain promising results that required large CPU time. They offer to speed up the algorithm to solve larger instances as future research by using hybrid metaheuristics. On the other hand, they use a deterministic demand though they offer the idea of the incorporation of the stochastic demand for consideration of the uncertainty. Simoni et al. (2018) believe that effective freight consolidations can help to prevent the environmental side effects of freight shipments. In this study, they investigate the consolidation solutions for parcel delivery considering location, vehicle fleet and route choice for a mail delivery service in Austin, Texas. They define the problem as multi-depot vehicle routing problem with heterogeneous vehicle fleet for urban consolidation centers. They propose a mixed-integer linear program (MILP) model which considers the total cost and environmental impact of alternative urban consolidation configurations and policy scenarios. To solve the large-scale real-world cases they offer a metaheuristic based on a genetic algorithm. Qiu & Huang (2013) evaluates the value of freight consolidation in aspect of using supply hub in industrial park(SHIP). They offer two mathematical models to compare the cases with and without SHIP. They solve these models with genetic algorithm (GA). The computational results show that consolidation of shipment by using SHIP brings benefits as the increase of the size of the supply chain decreases the total cost.

Ülkü & Bookbinder (2012) investigate the effect of different pricing schemes for third party logistics (3PL) providers. According to the authors, on-time delivery guarantee and offer of the competitive quote are the key success factors for 3PL providers. They offer four temporal pricing scheme based on the shipment consolidation with the objective of the maximization of profit for 3PL providers. This observation is also highly stressed by the 3PL company for which our study is based upon. Hu et al. (2018) propose a two-echelon mathematical model which combines the inventory routing and freight consolidation for perishable goods. They follow an iterative framework, first using a decomposition and then a local search. Freight consolidation problem is solved in the decomposition part whereas the assignment type mixed integer programming model is adopted to solve the inventory routing problem based on the freight consolidation decision. In the local search part, the incumbent solution cluster is improved gradually with each iteration.

Very recently, Anand et al. (2021) emphasize the importance of the freight consolidation in urban goods in terms of the reduction of carbon emission. This support the work by Ülkü (2012) who also emphasized the contribution of shipment consolidation in terms of environmental aspects.

Lin & Lee (2018) handles a hub network design problem for time-definite LTL freight transportation. In this problem, carriers guarantee to deliver the shipments on time while maximizing profit with split deliveries. They formulate the problem as a mixed integer program with a pricing subproblem. They found out that effective design of the hub-network design makes significant effect on the time-definite LTL freight transportation problem in aspect of profit maximization. The use of hubs is also considered in our research by introducing and locating promising transshipmemt points which also act as hubs. These are defined in chapters 5 under the best configuration consolidation of pairs or triplets of requests which may or may not use transshipment points depending on the obtained solution.

Cunha & Silva (2007) studied a similar problem. They handle the hub-and-spoke network problem for one of the top trucking companies in LTL operations in Brazil. They aim to determine consolidation locations(hubs) and assign the spokes to the hubs in a way which minimize the total cost. They offer a metaheuristic approach based on a genetic algorithm and obtained effective improvements in the operations for the trucking company. Computational results show that using a heuristic provide effective real life solutions for LTL optimization problem. The authors indicate that using spoke-hub configuration restrict the consolidation figures, therefore, the case which does not consider specific consolidation nodes can be considered as a future research. This case can provide more cost and time savings. In this study, we also design metaheuristics to tackle our order consolidation type problem.

Moon et al. (2012) extended vehicle routing problem with time windows with overtime working of drivers and outsourcing of the vehicles for third party logistics companies. They offer a decision support system based on GA which gives opportunity to the company reschedule the orders in real life.

Recently, Lu & Yang (2019) Iterative Logistics Solution Planner (ILSP) which first generates an initial solution and then improves it based on expert knowledge using ant colony algorithm designed for for logistics with pickup and delivery problem.

Hybridization of the metaheuristics is common in vehicle routing problem.

For instance, Allahyari et al. (2015) hybridized Greedy Randomized Adaptive Search Procedure (GRASP), Iterated local search (ILS) and simulated annealing for a multi-depot covering tour vehicle routing problem. Also, Sze et al. (2017) hybridized adaptive variable neighbourhood search and large neighbourhood search for the cumulative capacitated vehicle routing problem.

Similarly, Alinaghian & Shokouhi (2018) hybridized the adaptive large neighbourhood search and variable neighbourhood search algorithm for multi-depot multi compartment vehicle routing problem. They try to minimize the total number of vehicles used and total distance taken under constraints of split delivery allowance of requests and assignment of the specific products on specific compartments of the vehicles.

Baños et al. (2013) propose simulated annealing based parallel multi-objective approach for a vehicle routing problem with time windows. They try to minimize the distance and the imbalance of the routes which is defined by the imbalance in distance travelled by the vehicles and also the imbalance in the loads of the vehicles.

Kuo & Wang (2012) developed a VNS algorithm for the multi-depot vehicle routing problem with loading cost. The authors point out the gap in the literature by stressing the consideration of loading cost in multi depot vehicle routing problem. They use a stochastic method for generating initial solution for VNS, then they choose randomly between four operators and finally the solution is accepted based on the criterion which is similar to simulated annealing. They tested the approach with 23 cases and obtained 23.7% improvement on the transportation cost.

On the other hand, Rais et al. (2014) proposed an effective mixed integer programming model for the pickup and delivery problem with transshipment. One of the important results of the study is the obtained benefits of transshipment in the networks.

According to the above studies, the order consolidation is a hot practical logistic problem that can bring a lot of benefits not only financial but also environmental. Our chapters, namely, chapters 5, 6 and 7, aim to respond to these needs.

2.4 A Brief of the methods adopted

Literature review on the above transportation problem variants shows that there are mainly two solution approaches: exact and heuristic approaches. Exact methods are based on mathematical programming or dynamic programming and aim to guarantee optimality. These are mainly branching and exhaustive performed approaches.They always guarantee optimal solution but may require an excessive computation burden in terms of CPU. Rais et al. (2014) used exact approach, MIP for pickup and delivery problem with transshipment. Çapar (2013) also propose exact approach for joint shipment consolidation and inventory decisions in a two-stage distribution system. Despite the benefit of providing optimal solution of exact methods, these methods as mentioned above have serious disadvantages due to long computational times especially for large sized problems. Therefore one way forward is to embrace heuristics and metaheuristics, which though they do not guarantee optimality, they are simpler to understand and to modify besides requiring a reasonable computational time. In addition these approaches provide the flexibility to researchers to adjust and enhance their performances by adding new attributes such as learning. These are described in (Salhi & Thompson 2022*a*).

Recently, it is also found interesting and useful to hybridize exact methods with heuristics or metaheuristics as highlighted by (Salhi & Thompson 2022*b*).

Exact Methods

Exact methods are mathematically based approaches with the aim to produce an optimal solution. These are often used for small or medium sized combinatorial optimization problems. The common one is branch and bound which is an approach used for integer linear programming(ILP) problems. It works with two tools: splitting and branching procedures. There are several enhancements on this original method, some of these are integrated in the commercial optimisation software IBM ILOG CPLEX.

In this study, we formulate the problem as an ILP but adopted a set partitioning formulation as our exact approach. Set partitioning formulation is originally proposed by (Balinski, M. L., & Quandt 1964). For instance, Baldacci et al. (2008) used set partitioning formulation for the vehicle routing problem. The formulation is given in the following:

Let a set of feasible routes R, with their costs c_k, $k \in R$ and let N be the set of customers,

$$a_{ik} = \begin{cases} 1 & \text{if customer } i,\ i{\in}N \text{ belongs to route } k,\ k{\in}R \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

$$x_k = \begin{cases} 1 & \text{if } k^{th} \text{ route is used, } k{\in}R; \\ 0 & else \end{cases} \tag{2.2}$$

$$\min \sum_{k \in R} c_k x_k \tag{2.3}$$

subject to

$$\sum_{k \in R} a_{ik} x_k = 1; \quad \forall i \in N \tag{2.4}$$

$$x_k \in \{0, 1\}; \quad \forall k \in R \tag{2.5}$$

Objective function refers to the minimization of the total cost. Eq. 2.4 refers that each customer is covered by one route. Last constraint, eq. 2.5 refers that x_k is a binary variable.

Note that, if we replace equal sign(=) in eq. 2.4 with $\geq$, the set partitioning formulation becomes the the set covering formulation which is relatively much easier to solve. Note the above problem has N constraints and $|R|$ binary variables. In this study we adopt this type of formulation when exploring the optimal solution of our problem in chapter 6.

Metaheuristics

Metaheuristics are common approaches in vehicle routing problems due to requiring a reasonable computational time, simplicity and flexibility. Metaheuristics can be classified according to Salhi (2017) under (i) improving only heuristics such as GRASP, composite heuristics, variable neighbourhood search (VNS), large neighbourhood search (LNS), etc (ii) not necessarily improving heuristics such as simulated annealing, tabu search, etc and (iii) population based heuristics such as GA, Ant Colony (AC), Particle Swarm (PS), etc. As we adopted VNS, LNS and hybrid of VNS and LNS in our study, we present a brief summary of these approaches in the following:

Variable Neighbourhood Search

Mladenović, N., & Hansen (1997) introduced the VNS algorithm for the location problem in the case of the p-median problem. VNS works based on a systematic change of neighborhoods within a local search. It moves from

one neighbourhood to another neighbourhood if there is no improvement in the current solution to escape from local optimality. Neighbourhoods are usually designed from smallest to largest. If there is no improvement in the last and largest neighbourhood, the search returns back to the first and smallest neighbourhood Salhi (2017). As long as the neighbourhoods and local searches are designed effectively, VNS gives effective results with practical implementation as shown by the excellent work given in (Simeonova et al. 2018, Brandão 2020).
In chapter 7, we will revisit VNS and provide more details as we present our implementation of VNS to our problem.

Large Neighbourhood Search

Large neighbourhood search was first introduced by (Shaw 1997). It works with destroy and repair operators. At each iteration, a certain number of attributes are removed from the solution and then re-inserted into the solution using repair mechanisms in order to obtain a new feasible solution which may or may not be improved (Wolfinger 2021). There are different removal strategies adopted by researchers according to the type of the problem, the common one includes random removal or worst removal (Pisinger & Ropke 2007).

Adaptive LNS is effective in vehicle routing problems as shown in (Lahyani et al. 2019, Sacramento et al. 2019, Liu, Tao & Xie 2019). The idea here is to choose a pair of removal/repair operator at each iteration depending on the quality of such pair that could have been assessed at the learning stage. This strategy is effective as it speeds up the search while guiding the search to a better choice of ruin/repair. A basic implementation of LNS will be explored in chapter 7 with some suggestions provided in the final chapter.

Hybridisation

The hybridisation of several heuristics is a powerful tool as hybridised methods compensate the weaknesses of each other. Though the choice is not as easy as it sound, a good strategy that embrace the idea is welcome. Some details on this subject can be found in (Salhi & Thompson 2022*b*). Here, VNS/LNS is also attempted in chapter 7 to assess whether or not this strategy can be useful.

2.5 Summary

A literature review on supplier selection and order consolidation scheduling are presented. A brief of the techniques that are used in this study are also presented. From the literature review that we conducted, we can see that there is a gap on the literature for handling both of supplier selection and order consolidation scheduling problems together. In supplier selection studies, data availability is a challenge. Therefore most of the studies, researchers prefer to rely on expert knowledge. However, full expert knowledge is also not always available. There is not any supplier selection study in the literature which handles this issue. Our approach is able to make supplier selection decision with partial expert knowledge systematically and probabilistically. For the order consolidation scheduling problem in the literature, possible consolidation configurations are computed in the mathematical model each and every iteration. In our proposed exact approach, all possible consolidation configurations are computed out of the model and only once. And the proposed metaheuristic approach perfroms better than the exact method. On the other hand, we handle the speacial case of order consolidation whichh allows consolidation of the orders upto 3. The next chapter will present an integrated deterministic rule used in MCDM for the case of supplier selection.

Chapter 3

An Effective Integrated MCDM approach

3.1 Introduction

In this chapter, we present a deterministic selection rule that considers a set of techniques to produce an overall and robust selection of suppliers instead of making a decision based on one method only.

In the literature, there is a tendency to use integrated approaches for many type of problems as different approaches perform effectively in different aspects. In this study, we mainly focus on the combination of the results of the different ranking methods. In ranking methods for problems as supplier selection problem, different lists obtained from different methods can be superior to the other lists obtained from the other methods in different aspects. Hence, we offer to combine the results of different approaches by an effective deterministic rule. According to (Govindan et al. 2015), researchers might be biased for relying on only one approach. They also propose to investigate the acceptance level of the approaches by the decision makers and building experimental designs. We offer to combine the results of different approaches with a deterministic approach to prevent the biasedness which might come from relying on one approach. We also consider the preference level of the decision-makers for each individual method among the other method.

We propose two different types of deterministic rule; rank-based deterministic rule and score based deterministic rule. Score based deterministic rule consists of three different versions; deterministic rule only based on score of the alternatives, deterministic rule based on score and the weight of the position of the alternatives, deterministic rule based on the score and the weight of the alternatives and weight of the methods. In the literature, MCDM methods are most common approaches for supplier selection problem due to decision making ability with expert knowledge. We explained the proposed deterministic rule among three common MCDM methods; AHP, TOPSIS and VIKOR for supplier selection problem. We generated data for AHP, TOPSIS and VIKOR methods and compared the performances of these methods by regret based measurement methods that we developed. In different cases, different methods dominate the other methods. We justify that there is no need to use single method, results of the different methods can be combined by the proposed deterministic rule. Note that we offer to combine the results of methods not combining the methods. We also compare the performance of the proposed deterministic rules among the individual methods by regret based methods. Computational results from 10 data sets show that deterministic rules perform better than each individual method.

3.2 Brief Review on Integration of Different MCDM Methods

Integration of different methods for multi criteria decision problems is common in the literature (Jain et al. 2018, Liu, Quan, Li & Wang 2019). Different methods are effective in different aspects (Rodrigues et al. 2014, Opricovic 2004). Very recently, Watróbski et al. (2019) provided an interesting review and highlighted this important issue that a selection of a particular MCDM method could be misleading for a given decision problem. The authors presented a useful framework using a decision rule to identify a subset of promising methods for a given problem. In this study, we aim to support this important claim by introducing an integrated approach that focuses on the combination of the results of such a subset of different ranking methods instead of the combination of these methods themselves. For example, in the supplier selection problem, different selection lists are obtained from different methods, some are better than the others depending on various aspects. To overcome this anomaly, we offer to combine the results of a subset of ranking methods (e.g., those that could be identified from Watróbski et al. (2019) or other means) by incorporating a simple but effective deterministic integrated approach. For simplicity, our subset consists of three commonly used Multi Criteria Decision Making Methods(MCDM), namely, AHP, TOPSIS and VIKOR. We propose two types of deterministic approaches which we describe as score based and ranked based deterministic approaches.

Govindan et al. (2015) state that researchers could be biased for one approach. They propose a way where the acceptance level of the approaches is investigated by the decision makers leading to the building of experimental designs. In this study, we propose a more effective approach that prevents the biasedness which may arise from using a single method. We also incorporate into our approach, the preference level of the decision-makers for each individual method with reference to the others.

In this chapter, we develop a novel integrated deterministic approach for the supplier selection problem. However, the proposed approach can be applicable for other related selection problems. In this chapter, we handle the supplier selection problem in case of the availability of the expert knowledge. The case of lack of knowledge will be treated in the next chapter. Multi-Criteria Decision Making (MCDM) methods are among the most commonly used approaches for the supplier selection problem due to the consideration of multi-criteria and decision making ability with expert knowledge. Expert knowledge is essential source of decision making when data is not available. In supplier selection problem, experts have knowledge about suppliers based on their past experience or reputation of the suppliers on the market. We present the proposed approach based on three commonly used MCDM methods, namely, AHP, TOPSIS and VIKOR.

The contributions of this chapter are fourfolds:

(ii) The reduction in the biasedness of relying on a single method only and hence provide an added flexibility to the decision maker.

(i) The design and analysis of an effective deterministic integrated MCDM approach that takes into account the results of a subset of MCDM methods.

(iii) A regret-based evaluation mechanism to assess the performance of the individual methods as well as the proposed approach.

(iv) The generation of robust data sets that could be used for future benchmarking.

The rest of the chapter is organised as follows; the MCDM methods used in the proposed methodology are first presented in Section 3 with their respective illustrative examples. The effect of relying on a single method including the data generation schemes and the evaluation mechanisms to assess each of the three MCDM methods are outlined in Section 4 with their respective illustrative examples. We describe the proposed integrated approach and its three variants in Section 5 and provide the computational results in Section 6. A brief summary on this chapter is given in Section 7.

3.3 MCDMs used in the Proposed Methodology

We present three MCDM approaches namely, the Analytical Hierarchical Process(AHP), Technique for Order-Preference by the Similarity to Ideal Solution(TOPSIS) and the Vlsekriterijumska Optimizacija I KOmpromisno Resenje(VIKOR). These already known techniques will be followed by illustrative examples while emphasizing their pros and cons. These will make the basis for our analysis in the next section.

3.3.1 Analytical Hierarchy Process (AHP)

Analytical Hierarchy Process (AHP) is a well-known MCDM method. It works based on the pairwise comparison of criteria and alternatives. The relative importance of criteria and then relative preferences between alternatives for each criterion are provided by decision makers. Decision makers submit their preferences in AHP pairwise comparison scale (Saaty 1980) which is presented in Table 3.1.

Numerical Rating	Definition
1	Equally important
3	Moderate importance of one over another
5	Essential or strong importance
7	Very strong importance
9	Extreme importance
2,4,6,8	Intermadiate values between the two adjacent judgements

Table 3.1: AHP Pairwise Comparison Scale

The pairwise comparison matrix in AHP is a reciprocal matrix with its diagonal elements equal to 1. The experts submit their preferences between the criteria with the values from AHP Pairwise Comparison Scale and the lower triangular of the matrix is reciprocal of the upper triangular matrix. An example pairwise comparison matrix is presented in Table 3.2.

	A	B	C
A	**1**	3	1/5
B	1/3	**1**	1/7
C	5	7	**1**

Table 3.2: Example Pairwise Comparison Matrix

The pairwise comparison matrix of criteria is used to compute the priorities (weights) of criteria. On the other hand, the pairwise comparison of alternatives for each criterion is used to compute the priorities (preferences) of alternatives for each criterion. In the last step, the priority vector of each alternative is multiplied by the corresponding priority value (weight) of the criterion and the alternatives are then ranked based on the overall priority values. Since AHP works based on the subjective judgement of the decision makers, a consistency check is performed to assess if the judgements of decision makers are consistent. If a decision maker submits that alternative A is more preferable than alternative B (pairwise comparison matrix value of A vs. B:3) and alternative B is more preferable than alternative C(B vs. C:5), then the decision maker can not state that alternative C is more preferable than alternative A(C vs. A:5). The consistency check is performed by computing the Consistency Index (CI). If the value of CI is below 0.10, the judgements are accepted as consistent. The main notations used in the AHP process are given in Figure 3.1 and the main steps of the method are summarized in Figure 3.2.

Notation:
c: number of decision criteria
s: number of alternatives
A: pairwise comparison matrix of criteria valued by a_{ij} which showing the relative importance of the i^{th} criterion compared to j^{th} criterion; $i = 1, \ldots, c; j = 1, \ldots, c$.
N: normalized pairwise comparison matrix valued by n_{ij}; $i = 1, \ldots, c; j = 1, \ldots, c$.
w: weights of the criteria indexed by $i = 1, \ldots, c$.
r: weighted sum vector indexed by $i = 1, \ldots, c$.
λ_{max}: principal eigenvalue
CI: Consistency Index
RI: Random Consistency Index
CR: Consistency Ratio
B: comparison matrix of alternatives for each criterion, $B_k; k = 1, \ldots, c$ each B_k valued by b_{ij}, $i = 1, \ldots, s; j = 1, \ldots, s$.
P: matrix of priorities of alternatives based on the criteria valued by p_{ij},$i = 1, \ldots, c; j = 1, \ldots, s$.
o: overall priority value of alternatives indexed by $i = 1, \ldots, c$.

Figure 3.1: Notation of Steps of AHP

1. Set c decision criteria, s alternatives and RI.
2. Create the cxc pairwise comparison matrix $A = [a_{ij}]$.
3. Determine the priority values(weights) of the criteria as follows:

   (a) Compute the normalized pairwise comparison matrix say $N = [n_{ij}]$.

   $$n_{ij} = \frac{a_{ij}}{\sum_{i=1}^{i=c} a_{ij}} \quad i = 1, \ldots, c; \quad j = 1, \ldots, c; \tag{3.1}$$

   (b) Calculate the priorities(weights) of the criteria by following formula:

   $$w_i = \frac{\sum_{j=1}^{i=c} n_{ij}}{c} \quad i = 1, \ldots, c; \tag{3.2}$$

4. Compute the consistency index (CI) and consistency ratio (CR) to check the consistency of the judgements,

   $$r_i = \sum_{j=1}^{j=c} a_{ij} w_i \quad i = 1, \ldots, c \tag{3.3}$$

   $$\lambda_{max} = \frac{1}{c} \sum_{i=1}^{i=c} \frac{r_i}{w_i} \tag{3.4}$$

   $$CI = (\lambda_{max} - s)/(s - 1) \tag{3.5}$$

   $$CR = CI/RI \tag{3.6}$$

   If CR $\geq$ 0.10 go back to step 2 and repeat the pairwise comparison.

5. Create c B_k matrices $B_k = [b_{ij}]$.
6. Apply Step 3 to the B_k matrices and compute the priorities of the alternatives based on the criteria, $P = [p_{ij}]$.
7. Apply Step 4 to the B_k matrices to check the consistency of judgments. If CR $\geq$ 0.10 go back to step 5.
8. Compute the overall priorities of alternatives among the criteria, o_i.

   $$o_i = p_{ij} \ltimes w_i \quad i = 1, \ldots, n \tag{3.7}$$

9. Rank the alternatives from the highest to the lowest o_i values.

Figure 3.2: Steps of AHP

Random Consistency Index (RI) is determined based on the number of items being compared and table of RI is shown in Table 3.3.

n	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

Table 3.3: Random Consistency Index (RI)

Illustrative Example 1

In this subsection we provide the following supplier selection example, which have 4 decision criteria and 3 supplier alternatives. We follow the steps of Figure 3.2.

1. The decision criteria include: product quality, price, delivery performance and reputation. The alternatives are given as supplier A, supplier B and supplier C.

2. Criteria are compared in a pairwise manner. The decision makers submit their preferences on *AHP Pairwise Comparison Scale* which is presented in Table 3.1. The results of the pairwise comparison matrix are given in Table 3.4.

Criteria	Product Quality	Price	Delivery Performance	Reputation
Product Quality	1	5	3	9
Price	1/5	1	1/3	5
Delivery Performance	1/3	3	1	5
Reputation	1/9	1/5	1/5	1

Table 3.4: Pairwise Comparison Matrix

3. Priorities of the criteria are calculated based on the pairwise comparison matrix shown in Table 3.4.

   (a) Normalized pairwise comparison matrix, N is calculated by eq.(3.1) and the results are summarized in Table3.5.

Criteria	Product Quality	Price	Delivery Performance	Reputation
Product Quality	0.61	0.54	0.66	0.45
Price	0.12	0.11	0.07	0.25
Delivery Performance	0.20	0.33	0.22	0.25
Reputation	0.07	0.02	0.04	0.05

Table 3.5: Normalized Pairwise Comparison Matrix

(b) After normalization, priorities(weights) of the criteria are calculated by eq.(3.2) and results shown in Table 3.6.

Criteria	Priority Value
Product Quality	0.57
Price	0.14
Delivery Performance	0.25
Reputation	0.05

Table 3.6: Priorities(weights) of the criteria

4. The consistency ratio is calculated with equations (3.3) to (3.6) to check the consistency of the judgements. If consistency level is not enough, pairwise comparison matrix is revised. The calculation of λ_{max} is presented in Table 3.7 with r_i and w_i shown in their respective columns

Criteria	r_i	w_i	$r_i/\ w_i$
Product Quality	2.42/	0.57	= 4.28
Price	0.56/	0.14	= 4.07
Delivery Performance	1.08/	0.25	= 4.34
Reputation	0.19/	0.05	= 4.06
		Total	16.75
		λ_{max}	4.19

Table 3.7: λ_{max} calculation

(a) The consistency index(CI) is calculated using eq.3.5 leading to

$$CI = (4.19 - 4)/(4 - 1) = 0.063$$

(b) The consistency ratio(CR) is derived using eq.3.6 with RI being 0.90 (see Table 3.3 for the case of m=4). This leads to

$$CR = 0.063/0.90 = 0.07$$

Since the consistency ratio is under 0.10, the pairwise comparison is considered as consistent.

5. After pairwise comparison of criteria, the alternatives are compared pairwise for each criterion. The decision makers submit their relative preferences between the alternatives for every criteria. Pairwise comparison matrix of alternatives with regard to product quality is presented in Table 3.8. The same pairwise comparison matrix is also built for other criteria.

Product Quality	A	B	C
A	1	3	1/5
B	1/3	1	1/7
C	5	7	1

Table 3.8: Pairwise Comparison Matrix of Alternatives for Product Quality

6. The priority values of alternatives for every criteria are calculated according to equations (3.1), (3.2). The results are summarized in Table 3.9.

Alternatives	Product Quality	Price	Delivery Performance	Reputation
A	0.19	0.26	0.11	0.15
B	0.08	0.63	0.78	0.78
C	0.72	0.11	0.11	0.07

Table 3.9: Priority values of Alternatives based on the criteria

7. Consistencies of the judgements are checked with the same procedure as shown in step 4. The consistency check of judgements in pairwise comparison of alternatives with regard to product quality is followed as follows:

$$\lambda_{max} = 3.07$$

$$CI = (3.07–3)/(3-1) = 0.03$$

RI is determined based on the number of criteria which is n=3, RI is 0.58 (see Table 3.3 for n=3). Therefore,

$$CR = 0.03/0.58 = 0.03$$

The consistency check of judgements in pairwise comparison of alternatives with regard to other criteria are also conducted and results are presented below:

CR for pairwise comparison of alternatives with regard to cost, delivery performance and reputation are 0.03, 0 and 0.07 respectively.

As consistency ratios for pairwise comparison of alternatives with regard to each criterion are lower than 0.1, judgements are accepted as consistent.

8. Overall priority values of all alternatives among the criteria, o_i are calculated with eq.(3.7) and presented in Table 3.10.

Alternatives	Overall Priority
Supplier A	0.18
Supplier B	0.36
Supplier C	0.45

Table 3.10: AHP Scores of Alternatives

9. Alternatives are ranked based on the overall priority values leading to the following result with C being at the top followed by B and then A.

3.3.2 Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)

TOPSIS differs from AHP as it is a distance based MCDM method. It ranks the alternatives based on the distance from the ideal-solution and negative-ideal solution. In other words, the closer to the ideal, the better. The notation used in TOPSIS are given in Figure 3.3 and the main steps of the method are summarised in Figure 3.4.

Notation:
c: number of decision criteria
s: number of alternatives
T: TOPSIS decision matrix in which alternatives are evaluated based on criteria valued by t_{ij}, $i = 1, \ldots, c; j = 1, \ldots, s$.
K: normalized TOPSIS decision matrix valued by k_{ij}, $i = 1, \ldots, c; j = 1, \ldots, s$.
w: weights of the criteria indexed by $i = 1, \ldots, c$.
M: weighted normalized TOPSIS decision matrix valued by $m_{ij}; i = 1, \ldots, c; j = 1, \ldots, s$.
b_i: weighted normalized value of best alternative for criterion $i = 1, \ldots, c$.
w_i: weighted normalized value of worst alternative for criterion i=1,..c.
d_j^b: distance of alternative j to the best alternative for criterion $i; i = 1, \ldots, c; j = 1, \ldots, s$.
d_j^w: distance of alternative j to the worst alternative for criterion $i; i = 1, \ldots, c; j = 1, \ldots, s$.
c_j^w: similarity of alternative j to the ideal solution, $j = 1, \ldots, s$.

Figure 3.3: Notations of TOPSIS

1. Set c decision criteria, s alternatives and weights of the criteria w_i (i=1,...,c).

2. Create the cxs decision matrix $T = [t_{ij}]$ shows the evaluation of alternative i for criterion j.

3. Compute the normalized evaluation matrix K, k_{ij} denoting the element of K.

$$k_{ij} = \frac{t_{ij}}{\sqrt{\sum_{i=1}^{i=c} t_{ij}^2}} \quad i = 1, \ldots, c; \quad j = 1, \ldots, s; \tag{3.8}$$

4. Compute the weighted normalized decision matrix M with m_{ij} being the element of M.

$$m_{ij} = w_i * k_{ij} \quad i = 1, \ldots, c; \quad j = 1, \ldots, s; \tag{3.9}$$

5. Compute the best b_i and worst w_i values for each attribute. If criterion is benefit criterion which means it has positive impact,

$$b_i = \max_j m_{ij} \quad i = 1, \ldots, c \tag{3.10}$$

$$w_i = \min_j m_{ij} \quad i = 1, \ldots, c \tag{3.11}$$

   If criterion is cost criterion which means it has negative impact, then,

$$b_i = \min_j m_{ij} \quad i = 1, \ldots, c \tag{3.12}$$

$$w_i = \max_j m_{ij} \quad i = 1, \ldots, c \tag{3.13}$$

6. Compute the geometric distance of each alternative from the best solution d_j^b.

$$d_j^b = \sqrt{\sum_{i=1}^{i=c} (m_{ij} - b_i)^2} \quad j = 1, \ldots, s \tag{3.14}$$

7. Compute the geometric distance of each alternative from the worst solution d_j^w

$$d_j^w = \sqrt{\sum_{i=1}^{i=c} (m_{ij} - w_i)^2} \quad j = 1, \ldots, s \tag{3.15}$$

8. Compute the similarities of alternatives to the ideal solution c_j^w as follows,

$$c_j^w = \frac{d_j^w}{(d_j^w + d_j^b)} \quad j = 1, \ldots, s \tag{3.16}$$

9. Alternatives are ranked from the highest to the lowest values based on the c_j^w values.

Figure 3.4: Steps of TOPSIS

Illustrative Example 2

1. The same example given earlier is followed here. Alternatives vs criteria evaluation matrix is created. Alternatives are evaluated and scored based on the criteria and presented in Table 3.11

Alternatives	Product Quality	Price	Delivery Performance	Reputation
A	5	4	6	9
B	4	7	8	5
C	9	3	4	3

Table 3.11: TOPSIS initial Decision Matrix

2. The evaluation matrix is normalized by the eq.3.8 and presented in Table 3.12

Alternatives	Product Quality	Price	Delivery Performance	Reputation
A	0.45	0.47	0.56	0.84
B	0.36	0.81	0.74	0.47
C	0.82	0.35	0.37	0.28

Table 3.12: Normalized Decision Matrix

3. The weighted normalized matrix is calculated by eq.3.9 and presented in Table 3.13. In this example, weights are obtained from AHP.

Alternatives	Product Quality	Cost	Delivery Performance	Reputation
A	0.26	0.06	0.14	0.04
B	0.21	0.11	0.19	0.02
C	0.46	0.05	0.09	0.01

Table 3.13: Weighted Normalized Decision Matrix

4. The best and the worst value of each attribute are determined by eqs 3.10 to 3.13 and the results are given in Table 3.14.

Alternatives	Product Quality	Price	Delivery Performance	Reputation
Best	0.46	0.05	0.19	0.04
Worst	0.21	0.11	0.09	0.01

Table 3.14: Best and Worst values of each attribute

5. The geometric distance of each alternative from the best solution is calculated by eq.3.14 and results are presented in Table 3.15.

Alternatives	d_j^b
A	0.21
B	0.27
C	0.10

Table 3.15: Distance of each alternative from the best solution

6. The geometric distance of each alternative from the worst solution is calculated by eq.3.15 and results are given in Table 3.16.

Alternatives	d_j^w
A	0.09
B	0.09
C	0.26

Table 3.16: Distance of each alternative from the worst solution

7. The similarities of the alternatives to the worst solution are obtained using eq.3.15 and the scores are provided in Table 3.17.

Alternatives	TOPSIS Score
A	0.30
B	0.26
C	0.73

Table 3.17: TOPSIS Scores of the Alternatives

8. According to the similarities to the ideal solution, the alternatives are ranked as follows: C at the top followed by A and then by B.

3.3.3 VIKOR

VIKOR is an MCDM method that searches for the compromised solution for maximum group utility of majority and minimum regret of the individuals. It works similarly as TOPSIS. The notation used in VIKOR are given in Figure 3.5 and the main steps of the method are summarised in Figure 3.6.

Notation:
c: number of decision criteria
s: number of alternatives
V: VIKOR decision matrix in which alternatives are evaluated based on criteria valued by $v_{ij}; i = 1, \ldots, c; j = 1, \ldots, s$.
L: normalized VIKOR decision matrix valued by $m_{ij}; i = 1, \ldots, c; j = 1, \ldots, s$.
w_i: weights of the criteria indexed by $i = 1, \ldots, c$.
u: weight of maximum group utility
S : Utility measure
R : Regret measure
Q : VIKOR index

Figure 3.5: Notation of VIKOR

1. Set c decision criteria, s alternatives and weights of the criteria w_i.

2. Create the mxn decision matrix V with v_{ij} being the evaluation of alternative i for criterion j.

3. Compute the normalized evaluation matrix L with l_{ij} denoting the element of L.

   v_i^* is maximum value for benefit attributes, minimum value for cost attributes, v_i^- is minimum value for benefit attributes, maximum value for cost attributes,

$$l_{ij} = \frac{v_i^* - v_{ij}}{v_i^* - v_i^-} \quad i = 1, \ldots, c \quad j = 1, \ldots, s \tag{3.17}$$

4. The utility measure(S) and the regret measure(R) are calculated for each alternative with the following formulas:

$$S_i = \sum_{i=1}^{i=s} w_i l_{ij} \quad i = 1, \ldots, c \tag{3.18}$$

$$R_i = \max_i w_i l_{ij} \quad i = 1, \ldots, c \tag{3.19}$$

$$S^* = \min_i S_i \quad S^- = \max_i S_i \tag{3.20}$$

$$R^* = \min_i R_i \quad R^- = \max_i R_i \tag{3.21}$$

5. VIKOR index(Q) is calculated with the following formula: u is the weight of the maximum group utility, and it is determined based on the compromise level of decision makers.

$$Q_i = \frac{u(S_i - S^-)}{(S^* - S^-)} + \frac{(1-u)(R_i - R^*)}{(R^* - R^-)} \tag{3.22}$$

6. Alternatives are ranked based on the S, R and Q values from smallest to the largest. According to this ranking list, A_1 and A_2 is accepted first and second best compromised alternative if the following conditions are satisfied:

a) A_1 is also best ranked by S or/and R .

$$b)\ Q(A_2) - Q(A_1) \geq DQ; DQ = 1/(c-1) \tag{3.23}$$

If all conditions are satisfied, the solution is accepted as a compromise solution.

Otherwise (i.e., if one of the conditions is not satisfied), we check the following

a) If only the first condition is not satisfied, A_1 and A_2 are accepted as first and second compromised best alternatives.

b) If the second condition is not satisfied, the alternatives are ranked based on the distance relation by the following formula;

$$Q(A_m) - Q(A_1) > DQ \tag{3.24}$$

Figure 3.6: Steps of VIKOR

Illustrative Example 3

The earlier example is also used here.

1. Each alternative is evaluated with regard to each criterion and scored to build the VIKOR decision matrix in Table 3.18.

Alternatives	Product Quality	Cost	Delivery Performance	Reputation
A	5	4	7	9
B	3	6	8	5
C	9	3	4	3

Table 3.18: VIKOR initial Decision Matrix

2. The normalized decision matrix is calculated by eq.3.17.

Alternatives	Product Quality	Cost	Delivery Performance	Reputation
A	0.38	0.05	0.06	0
B	0.57	0.14	0	0.03
C	0	0	0.25	0.05

Table 3.19: VIKOR Normalized Decision Matrix

3. The utility measure(S), regret measure(R) and VIKOR index(Q) are calculated by equations 3.18, 3.19 and 3.22 respectively.

   w_j represents the relative importance(priority value) of each criterion.

Alternatives	S	R
A	0.49	0.38
B	0.73	0.57
C	0.30	0.25

Table 3.20: Utility and Regret Measures

4. VIKOR index(Q) is calculated by 3.23. u is assumed as 0.5.

Alternatives	Q
A	0.42
B	1
C	0

Table 3.21: VIKOR Index Values

5. The alternatives are ranked based on the S, R and Q values ranking from smallest to the largest.

Alternatives	S	Alternatives	R	Alternatives	Q
C	0.30	C	0.25	C	0
A	0.68	A	0.47	A	0.78
B	0.74	B	0.57	B	1

Table 3.22: VIKOR List

If the conditions in step 6 a) and b) are satisfied, the ranking list will lead to a compromise solution.

a) C is also best ranked in S or/and Q.

b) $(0.78 - 0) \geq 1/(3 - 1)$

All conditions are satisfied. Thus, the solution can be accepted as compromise solution. The alternatives are ranked as follows: C at the top followed by A and then by B.

3.4 Effect of Relying on Single Method-Based Selection Rules

In this section we demonstrate, as also shown in earlier studies including the recent interesting review by Watróbski et al. (2019), the risk of producing misleading results for a given decision problem if one entirely relies on one single method only. In this work, for simplicity the three MCDM methods namely, AHP, TOPSIS, VIKOR are used. We provide a simple evaluation measure to assess the effectiveness of each of the three individual methods. This assessment is based on the measure of regret, also known as opportunity cost. As there is no expert knowledge is available for these three methods, we propose a survey data set generator. This assessment measure and these new data sets will be used not only for testing these three individual methods but also the proposed approach which is presented in the following section.

3.4.1 Data Generation Schemes

We generated data sets for AHP, TOPSIS and VIKOR. These survey data sets are compiled in the MS Visual Studio environment using C++ language. In this experiment and for illustration purposes, we consider 10 alternative suppliers and 4 supplier selection criteria. Alternative suppliers are known as A to J for simplicity while the four supplier selection criteria include product quality, price, delivery performance and reputation. The three generator schemes, though strongly related, are presented separately one for each method. Note that these schemes could be easily modified to cater for the characteristics of other MCDM ranking methods if need be.

Data Generation for AHP

The following steps constitute the generation mechanism for AHP.

1. We firstly generate a survey data for AHP by creating a pairwise comparison matrix for the criteria. Here, experts submit their relative preferences between the criteria in the AHP Pairwise Comparison Scale (Saaty 2008) which is presented in Table 3.1. The pairwise comparison matrix in AHP is a reciprocal matrix with its diagonal elements equal to 1, and the lower triangular of the matrix is reciprocal of the upper triangular matrix. To illustrate how the generation is performed, we present an example of a pairwise comparison matrix in Table 3.2 using the following two steps.

   (a) We define an array of this scale which includes 1, 3, 5, 7, 9 and their respective reciprocal values 1/3, 1/5, 1/7, 1/9.

   (b) We then assign random numbers from this array for the first row of the matrix which represents the relative importance of the first criterion vs. the other criteria for the decision makers.

2. The remaining rows of the pairwise comparison matrix of the criteria are generated pseudo-randomly based on the consistency rule which is described as follows;

   Consistency Check in AHP

   In the above table, product quality vs. price is 5 which means product quality is relatively 5 times more important than the price. On the other hand, product quality vs. delivery performance is 9. According to these two values, price vs delivery performance is expected to be relatively 1.8(9/5). In the proposed data generation mechanism, we assign the closest value from the array which is 1. While delivery performance and price have different relative importance values compared to product quality, price vs. delivery performance is 1 which means they have relatively equal importance. It shows that we generate the data randomly as long as they remain consistent.

In case inconsistent results are obtained (e.g., the consistency ratio as defined in AHP is larger than 0.10), the process is repeated again.

For each criterion, the same procedure is performed for the pairwise comparison of alternatives. For example, for the first criterion, alternatives are compared in a pairwise manner. The first row of the pairwise comparison of alternatives for this criterion is generated by assigning the values from the array randomly. The other rows of the matrix are generated to compute the consistency rule of the AHP. For illustration, an example of a pairwise comparison matrix of 10 suppliers for product quality criterion is presented in Table 3.23.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	1	5	9	1/3	1	3	7	1/3	1/7	9
S2	1/5	1	1	1/9	1/5	1/3	1	1/9	1/9	1
S3	1/9	1	1	1/9	1/9	1/3	1	1/9	1/9	1
S4	3	9	9	1	3	9	9	1	1/3	9
S5	1	5	9	1/3	1	3	7	1/3	1/7	9
S6	1/3	3	3	1/9	1/3	1	3	1/9	1/9	3
S7	1/7	1	1	1/9	1/7	1/3	1	1/9	1/9	1
S8	3	9	9	1	3	9	9	1	1/3	9
S9	7	9	9	3	7	9	9	3	1	9
S10	1/9	1	1	1/9	1/9	1/3	1	1/9	1/9	1

Table 3.23: Example Pairwise Comparison Matrix of 10 Suppliers for Product Quality

3. For the pairwise comparison matrices of the alternatives for the other criteria, these are obtained based on the generation of the pairwise comparison matrix of alternatives for the first criterion. They can obviously be generated randomly but we preferred to generate them pseudo randomly by incorporating expert knowledge. For example if experts submit that alternative 1 is 3 times better than alternative 2 in terms of product quality, price of the alternative 1 is expected to be higher than alternative 2. They are closely related. So we assigned the first row of the pairwise comparison of the alternatives for the second criterion using the values from the first row of the pairwise comparison of alternatives for the first criterion by allowing a perturbation of ± 1 range in the array. On the other hand, product quality vs delivery performance and product quality and reputation vs reputation are

less relative. Therefore we keep the range ±2 for the first row of these pairwise comparison matrices which means, while alternative 1 vs alternative 2 is 1 for product quality, alternative 1 vs alternative 2 can be one of 1/5, 1/3, 1, 3 and 5 values for delivery performance. For the ten suppliers, the perturbation is carried out using the random range values for delivery performance as 1 -1 2 -2 2 2 1 2 2 whereas those for reputation as -1 0 1 0 0 2 -1 0 1.

So the first row of the pairwise comparison matrix of the alternatives for the other criteria are assigned based on the relationship between the first criterion pseudo randomly.

4. The remaining rows of the matrices are generated based on the consistency rule. The pairwise comparison of 10 suppliers for delivery performance criterion is presented in Table 3.24.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	1	7	7	3	1/5	7	9	1	1/3	9
S2	1/7	1	1	1/3	1/9	1	1	1/7	1/9	1
S3	1/7	1	1	1/3	1/9	1	1	1/7	1/9	1
S4	1/3	3	3	1	1/3	3	3	1/3	1/3	3
S5	5	9	9	3	1	9	9	1	1	9
S6	1/7	1	1	1/3	1/9	1	1	1/9	1/9	1
S7	1/9	1	1	1/3	1/9	1	1	1/9	1/9	1
S8	1	7	7	3	1	9	9	1	1	9
S9	3	9	9	3	1	9	9	1	1	9
S10	1/9	1	1	1/3	1/9	1	1	1/9	1/9	1

Table 3.24: Example Pairwise Comparison Matrix of 10 Suppliers for Delivery Performance

5. The weights of criteria, and the overall evaluation matrix of alternatives based on the criteria are then obtained. The overall evaluation matrix is presented in Table 3.25 which will be used for the data generation for TOPSIS.

	Product Quality	Price	Delivery Performance	Reputation
S1	0.09789	0.14266	0.15515	0.12320
S2	0.01961	0.02341	0.02360	0.04310
S3	0.01848	0.02341	0.02360	0.01730
S4	0.18252	0.22806	0.06822	0.15681
S5	0.09789	0.06794	0.24190	0.15681
S6	0.03955	0.02315	0.02299	0.05200
S7	0.01886	0.02262	0.02272	0.01730
S8	0.18252	0.19457	0.19544	0.20183
S9	0.32419	0.25155	0.22367	0.21434
S10	0.01848	0.02262	0.02272	0.01730

Table 3.25: Overall Evaluation of Suppliers Based on Criteria in AHP

Data Generation for TOPSIS

TOPSIS requires, as one of its inputs, a decision matrix which includes the evaluation of the alternatives based on the criteria and their respective weights. To incorporate subjectivity which can arise when using these kind of methods such as AHP, TOPSIS and VIKOR, we introduce into our scheme two additional attributes, namely, normalisation and some form of perturbation. These are necessary to obtain consistent and meaningful values that can then be used analytically in any subsequent evaluation.

1. The data about the alternatives such as age, price, etc are first provided which are then normalized.

2. In case there is a lack of data in step 1, expert judgements are used to construct the decision matrix. Experts score the alternatives on a scale of 1 to 10. Note that the same alternatives are evaluated based on the same criteria for these three methods. Though their judgements may differ for different methods, their results are expected to be consistent. The evaluation matrix of TOPSIS is generated based on the overall evaluation matrix of alternatives using the criteria from AHP. This matrix for TOPSIS is then perturbed accordingly to be consistent with the one used for AHP. Here, these values are turned on a scale 1 to 10

(i.e., timing these values by 100) then multiplying them by 0.25 and then perturbing them using a $\pm$ random number between 0 and 1 while keeping them integer. This is an important step that aims to alleviate the subjectivity element often encountered in these methods.

In brief, for a given supplier and a given criterion let β_{AHP} be the original AHP value. We then compute the new value for TOPSIS, say β_{TOPSIS}, using the following steps.

a) Generate a random number $\alpha \in [0, 1]$.

b) For each element of the TOPSIS matrix do the following:

- If $\alpha < 0.5$, set $\beta_{TOPSIS} = Max(1, \lfloor (25 \times \beta_{AHP}) - \alpha \rfloor)$
- Otherwise set $\beta_{TOPSIS} = \lceil (0.25 \times (100 \times \beta_{AHP})) + \alpha \rceil$

For example, consider $\alpha = 0.2159 < 0.5$.

For S1 and product quality, the value for TOPSIS would be $(0.09789 \times 25) - 0.2159 = 2.4472$ and hence $\beta_{TOPSIS} = \lfloor 2.4472 \rfloor = 2$.

The evaluation of the other elements of the TOPSIS matrix can then be computed in the same way.

3. The weights of the criteria in TOPSIS are then set to those from AHP. This choice is commonly used in the literature as shown in the following recent studies (Jain et al. 2018, Akgün & Erdal 2019).

Data Generation For VIKOR

VIKOR also needs a decision matrix as an input. We therefore perturb the decision matrix of TOPSIS to obtain a decision matrix for VIKOR in case the experts might submit slightly different values for the evaluation of the alternatives with different methods.

A similar mechanism as the one given for TOPSIS is performed here.
a) Generate a random number $\alpha \in [0, 1]$.
b) For each element of the VIKOR matrix do the following:

- If $\alpha < 0.5$ set $\beta_{VIKOR} = Max(1, \lfloor \beta_{TOPSIS} - (0.2 \times \beta_{TOPSIS}) \rfloor)$,
- Otherwise we set $\beta_{VIKOR} = Min(10, \lceil \beta_{TOPSIS} + (0.2 \times \beta_{TOPSIS} \rceil))$.

For example, consider $\alpha = 0.5628 > 0.5$.
For S1 and product quality, the value for VIKOR would be
$(2 \times 0.2) + 2 = 2.4$ and hence $\beta_{VIKOR} = \lceil 2.4 \rceil = 2$
The evaluation of the other elements of the VIKOR matrix can then be computed in the same way.

It is worth noting that our generation rule, though used for the supplier selection problem in this instance, can easily be made appropriate for other type of decision problems. Also, in this experiment, we opted for a lighter perturbation by injecting the same noise (the random number α to all elements of the TOPSIS and the VIKOR matrices). This is based on the view that a decision maker will apply his/her subjectivity toward all suppliers and all criteria in the same way. However, if one wishes to have a stronger perturbation, for each supplier and each criteria, the noise could then be injected to each element of the two matrices. The above generation code and the data sets can be requested from the author.

3.4.2 Regret-Based Assessment Methods

In this section, we provide the definitions of two assessment measures and use them to show that it is not always possible to identify the best performing individual method. Later in Section 4, we will use these measures to illustrate how much the integrated approach can improve the results, and how it can outperform all individual methods.

We first provide the notation and the adjustment scheme, followed by the two regret measures with illustrative examples for the three MCDM methods.

Notation
s : number of suppliers indexed by $i = 1, \ldots, s$
m : number of methods used indexed by x and y; $x, y = 1, \ldots, m$
S_x^i : the score of the i^{th} supplier by method x; $i = 1, \ldots, s; x = 1, \ldots, m$
R_x^i : the rank of the i^{th} supplier by method x; $i = 1, \ldots, s; x = 1, \ldots, m$

Figure 3.7: Notations of Regret Based Methods

Adjustment Scheme

We first ran AHP, TOPSIS and VIKOR methods on 10 generated data sets. The results of each method are evaluated using the regret based method, also known as the opportunity cost measure. These three methods give different ranking lists. AHP ranks the alternatives from the largest to the lowest score where the sum of the scores equals 1. TOPSIS also ranks the alternatives from the largest to the lowest but the sum of the scores does not equal 1. VIKOR, on the other hand, ranks the alternatives from the lowest to the largest where the sum of the scores is also not necessarily 1. As our assessment measures use the scores of alternatives from different methods together, it is necessary to bring the scores to a common scale. We adopt the scores of AHP for this purpose, and adjust the results of other methods so that the alternatives are ranked in decreasing order of scores and the sum of scores is equal to 1. We designed the following adjustment scheme to perform such a task:

a) We adjust the scores in the list of TOPSIS to have the same scale as the ones in AHP by setting $S^i_{TOPSIS} = \tilde{S}^i_{TOPSIS} / \sum_{i=1}^{i=s} \tilde{S}^i_{TOPSIS}$, $\forall i = 1, \ldots, s$ with $\tilde{S}^i_{TOPSIS}$ being the original score of supplier i found by TOPSIS.

b) As VIKOR ranks the alternatives from lowest to largest, we substracted the values of the list from 1 and divide them by the sum of the scores. This simple transformation leads to scores being on the same scale used in AHP and TOPSIS. We set $S^i_{VIKOR} = (1 - \tilde{S}^i_{VIKOR}) / \sum_{i=1}^{i=s}(1 - \tilde{S}^i_{VIKOR})$ $\forall i = 1, \ldots, s$ with $\tilde{S}^i_{VIKOR}$ being the original score of supplier i found by VIKOR.

The new results of these three methods are put in descending order and Table 3.26 provides an example for one of the data sets. There are some suppliers with the same score in the list. As the suppliers with same score in the lower ranks, they dont have big impact on the results. Therefore we chose which comes first randomly and rank all the suppliers as the total number of ranks will be 10 for all three methods to apply the regret based methods. These results though informative with respect to each method cannot be used to confirm whether or not the result of one method is better than the other. In the following subsections, we will present an evaluation scheme to achieve this goal by introducing a regret-based measure. This is explored under two variants, namely, a rank-based and a score-based measure.

	AHP		TOPSIS		VIKOR	
Order	Supplier	Score	Supplier	Score	Supplier	Score
1	Supplier E	0.284	Supplier E	0.306	Supplier E	0.343
2	Supplier J	0.185	Supplier D	0.170	Supplier D	0.218
3	Supplier D	0.182	Supplier J	0.167	Supplier J	0.201
4	Supplier A	0.155	Supplier A	0.138	Supplier A	0.190
5	Supplier G	0.051	Supplier F	0.045	Supplier F	0.024
6	Supplier F	0.048	Supplier G	0.042	Supplier G	0.021
7	Supplier B	0.027	Supplier B	0.033	Supplier B	0.02
8	Supplier H	0.023	Supplier H	0.033	Supplier C	0
9	Supplier I	0.023	Supplier I	0.033	Supplier H	0
10	Supplier C	0.021	Supplier C	0.033	Supplier I	0

Table 3.26: Normalized Results of the three methods for Data Set1

In brief, we compare the effectiveness of the methods based on the deviation between the ranking results of one method and the others. Each method gives the ranked list of alternatives and corresponding scores.

For simplicity, consider x as the reference method in which the other method, say y, is being evaluated against.

A weight function-

Let w_x^i be the weight of supplier i for being ranked R_x^i in method x. This is expressed as a strictly decreasing discontinuous function of the rank of the suppliers. In this particular setting, the top supplier is associated with a weight of unity whereas the bottom supplier a weight of $1/s$.

$$w_x^i = \frac{1}{R_x^i};\ i = 1, \ldots, s; x = 1, \ldots, m$$

The above function is one way to discriminate the position of the suppliers, though other functions may be attempted as well. We opted for this function as discrepancies in higher ranks are usually more important to the decision maker. For example, if a supplier comes first in one method and second in the other one, the difference between its weights in the methods will be higher as opposed to the case of it coming seventh and eighth. This is reasonable since alternatives in higher ranks carry significant importance for decision makers and more attention should be paid to those ranks.

The deviation of the result of method y with respect to the result of method x is then given by Δ_{xy} in Eq 3.25.

$$\Delta_{xy} = \sum_{i=1}^{i=s}(S_x^i \times w_x^i) - \sum_{i=1}^{i=s}(S_x^i \times w_y^i) \qquad \forall x, y = 1, \ldots, m \;\; (x \neq y) \quad (3.25)$$

Note that we include both the rank weights and the scores of alternatives in calculating the deviation. This is useful because it may be possible for two consecutive alternatives in the rank list to have quite different scores. In addition, there may be very close scores in consecutive ranks in the list. Considering both the rank and the score helps us to account for such situations when calculating our deviation measures.

In our example, we have $s = 10$ representing Supplier A to Supplier J and $m = 3$ referring to AHP, TOPSIS and VIKOR.

In the following subsections, the rank-based and the score-based regret measures are presented respectively.

Rank-Based Regret Measure

In this subsection, we evaluated the result of each method based on the results of the other methods via the calculation of the deviations. The method which has the smallest deviation is considered to be the best performer based on the reference method. These calculations are performed for all the reference methods (i.e., all the methods), taken one at a time. For instance, when we consider VIKOR as the reference method and calculate the deviation of AHP from VIKOR and the deviation of TOPSIS from VIKOR, we can find out whether AHP or TOPSIS has the smallest deviation and then be considered as the best and the other method will be the second with respect to VIKOR. If there were m methods, this ranking is then done for all the $m - 1$ other methods.

Illustrative Example 1

We ran the three methods with data set1 whose results are presented in Table 3.26 and calculated the deviations of each method with respect to the other methods.

Firstly, we evaluated the list of AHP based on the list of TOPSIS using eq.3.25. For computing the deviation of the results of AHP from the results of TOPSIS, $(x = 2, y = 1)$

$$\Delta_{21} = ((0.306{\times}1{+}0.170{\times}0.5{+}...0.033{\times}0.1){-}(0.306{\times}1{+}0.167{\times}0.5{+}...0.033{\times}0.1))$$

$$= 0.000614073$$

For the deviation of result of AHP from the result of VIKOR, $(x = 3, y = 1)$

$$\Delta_{31} = ((0.343\times 1+0.218\times 0.5+...0\times 0.1)-(0.343\times 1+0.201\times 0.5+...0\times 0.1))$$
$$= 0.002812460$$

For the deviation of result of TOPSIS from the result of AHP, $(x = 1, y = 2)$

$$\Delta_{12} = ((0.284\times 1+0.185\times 0.5+...0.021\times 0.1)-(0.284\times 1+0.182\times 0.5+...0.021\times 0.1))$$
$$= 0.000673790$$

For the deviation of result of TOPSIS from the result of VIKOR, $(x = 3, y = 2)$

$$\Delta_{32} = ((0.343\times 1+0.218\times 0.5+...0\times 0.1)-(0.343\times 1+0.201\times 0.5+...0\times 0.1))$$
$$= 0$$

For the deviation of result of VIKOR from the result of AHP,$(x = 1, y = 3)$

$$\Delta_{13} = ((0.284\times 1+0.185\times 0.5+...0.021\times 0.1)-(0.284\times 1+0.182\times 0.5+...0.021\times 0.1))$$
$$= 0.000723549$$

For the deviation of result of VIKOR from the result of TOPSIS, $(x = 2, y = 3)$

$$\Delta_{23} = ((0.306\times 1+0.170\times 0.5+...0.033\times 0.1)-(0.306\times 1+0.170\times 0.5+...0.033\times 0.1))$$
$$= 0$$

In summary, using the reference method AHP, the deviation of the results of TOPSIS and VIKOR are 0.000673790 and 0.000723549 respectively. In terms of the effectiveness of TOPSIS and VIKOR based on AHP, TOPSIS is ranked as first and VIKOR as second. For each of the other reference methods, the deviations of the other two methods are computed in the same way and summarized in Table 3.27. The methods in rows refer reference methods denoted by x, and the methods in columns refer the evaluated methods, denoted by y.

	AHP	TOPSIS	VIKOR
AHP	-	0.000673790	0.000723549
TOPSIS	0.000614073	-	0
VIKOR	0.002812460	0	-

Table 3.27: Deviations of the methods with respect to each of the reference methods with Data Set 1

For a given reference method $x, x = 1, \ldots, m$, we evaluate the comparative effectiveness of the other methods using the following rank-based algorithm.

The Rank-Based Regret Algorithm (RBRA)

Step 1: For a given reference method $x = 1, \ldots, m$ do

- Compute $\Delta_{xy} \quad \forall y = 1, \ldots, m$
- Sort the vector Δ in ascending order and let $Rank_{xy}$ be the rank of the y^{th} method with respect to the reference method x.

Step 2: Compute the total rank for method y as $TRank_y = \sum_{x=1}^{x=m} Rank_{xy}$

Step 3: Find y^* as the selected method using $y^* = \text{ArgMin}_{y=1,\ldots,m} TRank_y$.

The rankings of the methods based on the deviation values found in Table 3.27 are presented in Table 3.28 with bold displaying the smallest total score. In this particular example, TOPSIS has the smallest total rank of 2 and hence it is the best performer, with VIKOR as the second best, followed by AHP in the last position.

	AHP	TOPSIS	VIKOR
AHP	-	1	2
TOPSIS	2	-	1
VIKOR	2	1	-
Total Rank	4	**2**	3

Table 3.28: The ranking of the methods based on the deviations with Data Set 1

Some Basic Statistical Results

We ran the three methods with ten generated data sets and summarized their respective rankings including ties in Table 3.29. For example if the best configuration for instance 1 found by AHP is used in TOPSIS and VIKOR (see row 1 in Table 3.29), TOPSIS has a smaller deviation than VIKOR and hence TOPSIS earns the position of 1 for that data set and VIKOR earns 2. If there is a tie, as shown for instances 5, 7, 8 and 10, they both earn the position of 1.

	AHP	TOPSIS	VIKOR
AHP	-	1,1,1,2,1,2,1,1,1,1	2,2,2,1,1,1,1,1,2,1
TOPSIS	2,1,2,2,2,1,2,2,2,2	-	1,1,1,1,1,2,1,1,1,1
VIKOR	2,1,2,2,2,1,2,2,2,2	1,1,1,1,1,2,1,1,1,1	-
Overall Total Rank	36	**23**	25

Table 3.29: Ranking of the methods based on the deviations over the 10 data sets

To obtain an overall result we summed up these rankings and record the results under 'Total' in Table 3.29. The method with the smallest value is the most effective: In our case, TOPSIS is the best followed by VIKOR then AHP.

In the next subsection, we evaluate the effectiveness of the methods using the score based approach instead.

Score-Based Regret Measure

Here, we evaluated the effectiveness of the methods using their deviation scores directly instead of the ranks they achieve with these scores. This is similar in principle to the rank based except a score is used instead of rank. The method is as follows:

The Score-Based Regret Algorithm (SBRA)

Step 1: For a given reference method $x = 1, \ldots, m$ do

- Compute $\Delta_{xy} \quad \forall y = 1, \ldots, m$
- Compute the total sum for the method y as

$$\tilde{\Delta}_y = \sum_{\substack{x=1 \\ x \neq y}}^{m} \Delta_{xy}$$

Step 2: Find y^* as the selected method using $y^* = \text{ArgMin}_{y=1,\ldots,m} \tilde{\Delta}_y$.

Illustrative Example 2

The total deviation score of AHP is the sum of the deviation of AHP from TOPSIS and from VIKOR:

$\tilde{\Delta}_1 = \Delta_{21} + \Delta_{31} = 0.000614073 + 0.00281246 = 00342654$

The total deviation score of TOPSIS is then:

$\tilde{\Delta}_2 = \Delta_{12} + \Delta_{32} = 0.00067379$
Finally, the total deviation score of VIKOR is:
$\tilde{\Delta}_3 = \Delta_{13} + \Delta_{23} = 0.000723549$

$$y^* = \underset{y=1,2,3}{\text{ArgMin}}(\tilde{\Delta}_y) = \underset{y=1,2,3}{\text{ArgMin}}(0.00342654, 0.00067379, 0.000723549) = 2$$

In other words, TOPSIS is the most effective method as it has the smallest total deviation, $\tilde{\Delta}_2 = \text{Min}(\tilde{\Delta}_1, \tilde{\Delta}_2, \tilde{\Delta}_3)$

Some Basic Statistical Results

The analysis is performed over the 10 generated data sets. For each method the average deviation score and the number of best (including ties) are also reported in Table 3.30 where the best results are shown in bold. Based on these empirical results, TOPSIS tends to perform better compared to the other methods in terms of average deviation score. However, AHP produces a slightly larger number of best solutions (i.e., 5 compared to 4) though it yields the worst average deviation. On the other hand, VIKOR's average performance in terms of deviation score sits in between the two methods while it comes last in producing the number of best solutions.

The above inconclusive results demonstrate that there is not necessarily one method that dominates the others. This demonstrates the negative effect of relying on one method only as also strongly shown by Watróbski et al. (2019). This led us to a technique that aim to combine the results of the individual methods.

Data Set	AHP	TOPSIS	VIKOR
Data Set 1	0.003430	**0.000067**	0.000072
Data Set 2	**0**	0.000280	0.000740
Data Set 3	0.212390	**0.053030**	0.073120
Data Set 4	0.110700	0.094880	**0.094020**
Data Set 5	**0.000580**	0.001000	0.001000
Data Set 6	**0.000820**	0.021810	0.001890
Data Set 7	**0.001430**	0.002180	0.002180
Data Set 8	0.078890	**0.071390**	**0.071390**
Data Set 9	0.128970	**0.116700**	0.225740
Data Set 10	**0.242820**	0.265910	0.262170
Average	*0.078000*	***0.062790***	*0.073300*
Number of Best (incl ties)	**5**	*4*	*2*

Table 3.30: Deviation scores of the three methods

3.5 The Deterministic Integrated Approach

Here, we propose an integrated and deterministic approach to combine the different MCDM methods through their respective ranking lists of suppliers. In this study for completeness we use AHP, TOPSIS and VIKOR methods which are described in the two previous sections. In other words, the lists which are obtained from the different MCDM methods are considered in the design of the following deterministic method. We adopt two rules one using the ranks and the other the scores instead. These two rules are described in the following sub-sections. Additionally, we offer three variants for the latter rule.

3.5.1 RBDR

Here, the positions (or ranks) of the suppliers in the lists of the different methods are summed up and put in ascending order. The steps of this technique which we refer to as RBDR, short for the ranked based deterministic rule algorithm, are summarized below:

The RBDR algorithm

Step 1: Solve the problem for all methods x, $x = 1, \ldots, m$.

Step 2: Record the R_x^i ($i = 1, \ldots, s$; $x = 1, \ldots, m$).

Step 3: For each supplier i($i = 1, \ldots, s$) compute

$$\delta_i = \sum_{x=1}^{m} R_x^i$$

Step 4: Sort the list (δ) in ascending order with

$$i^* = ArgMin_{i=1,\ldots,s}\delta_i$$

being the supplier at the top of the list and $\tilde{\delta}$ its corresponding ordered list with

$$\tilde{\delta}(1) = i^*$$

As an example consider data set1 where positions of the suppliers in the lists of AHP, TOPSIS and VIKOR are presented in Table 3.31. The alternative suppliers are ranked based on the sum of the position values in ascending order in Table 3.32. Here, supplier E receives the smallest total value of 3 resulting in supplier E being at the top of the list whereas suppliers C and I at the bottom. It is worth noting that it may not, in some situations, be possible to achieve full ranking with RDBR because of the existence of ties as shown by I and C. Though these two suppliers happen to be at the bottom of the list and hence may not affect the decision, such an occurrence of tie could also occur with suppliers at the top of the list where the decision could be more critical. The score-based methods that we present next are able to differentiate more between the alternatives.

Alternatives	AHP	TOPSIS	VIKOR	TOTAL
Supplier A	4	4	4	12
Supplier B	7	7	7	21
Supplier C	10	10	8	28
Supplier D	3	3	2	7
Supplier E	1	1	1	3
Supplier F	6	5	5	16
Supplier G	5	6	6	17
Supplier H	8	8	9	25
Supplier I	9	9	10	28
Supplier J	2	3	3	8

Table 3.31: Positions of the suppliers in the list of the methods for Data Set1

Alternatives (i)	TOTAL (δ_i)
Supplier E	3
Supplier D	7
Supplier J	8
Supplier A	12
Supplier F	16
Supplier G	17
Supplier B	21
Supplier H	25
Supplier C	28
Supplier I	28

Table 3.32: Rank of alternatives based on total position for Data Set 1

In brief, the new list is given in Table 3.32. It is worth noting that this list differs from each of the best lists found by the individual methods.

3.5.2 SBDR

In this section, we use the score as our measure instead of the rank. Three different versions of the score based rule are presented. In the first version, the score of the alternative in each method is considered only. On the other hand, in the second version, the position of the alternative is also considered. In the last version, in addition to the score and the position of the alternative in the lists, a preference weight of the methods is introduced. These three versions are presented in the following subsections.

Additional notations:

α_x: weight of method x; $\quad x = 1, \ldots, m$.

S_x: Ranked scores of suppliers according to the x^{th} method with S_x^i denoting the score of the i^{th} supplier by method x.

β_i: Total score of i^{th} supplier based on the score based deterministic rule.

Variant 1- The SBDR Algorithm

In this initial version, we sum up the scores of the alternatives in results of the methods and rank the alternatives in descending order based on these total scores. The steps of this technique which we call SBDR for short are summarized as follows:

Step 1: For each method x find the ranked list S_x; $\quad x = 1, \ldots, m$

Step 2: Record $S_x^i (x = 1, \dots, m; i = 1, \dots, s)$

Step 3: Compute

$$\beta_i = \sum_{x=1}^{m} S_x^i \quad i = 1, \dots, s \tag{3.26}$$

Step 4: Sort the list in descending order with

$$i^* = \operatorname*{ArgMax}_{i=1,\dots,m} \beta_i \tag{3.27}$$

and $\tilde{\beta}$ being the ordered list of the suppliers with $\tilde{\beta}(1) = i^*$

Illustrative Example 3

The same example is considered here. Alternatives are ranked based on equations 3.26 and 3.27. The individual scores for each supplier under each of the three methods including their total score are given in Table 3.33. The suppliers are then sorted based on their total score and a new ranking list of suppliers is shown in Table 3.34. Here, supplier E is at the top of the list whereas C is at the bottom.

Alternatives	AHP	TOPSIS	VIKOR	TOTAL
Supplier A	0.1546	0.1381	0.1903	0.4830
Supplier B	0.0271	0.0331	0.0023	0.0625
Supplier C	0.0215	0.0329	0	0.0543
Supplier D	0.1819	0.1701	0.2176	0.5696
Supplier E	0.2840	0.3059	0.3430	0.9329
Supplier F	0.0476	0.0452	0.0242	0.1170
Supplier G	0.0513	0.0418	0.0211	0.1143
Supplier H	0.0235	0.0329	0	0.0563
Supplier I	0.0234	0.0329	0	0.0563
Supplier J	0.1852	0.1671	0.2014	0.5536

Table 3.33: Total Scores of Alternatives in Different Methods for Data Set 1

Alternatives	TOTAL
Supplier E	0.9329
Supplier D	0.5696
Supplier J	0.5536
Supplier A	0.4830
Supplier F	0.1170
Supplier G	0.1143
Supplier B	0.0625
Supplier H	0.0563
Supplier I	0.0563
Supplier C	0.0543

Table 3.34: Ranking based on Total Score for Data Set 1

Variant 2- The incorporation of the positions of the alternatives in SBDR (SBDRP)

In this version, position of the alternative is also considered. In other words, this variant is variant one with the addition of the positions of the alternatives. The position weight of each alternative is multiplied by the score of the alternative and alternatives are then ranked in descending order.

This algorithm which, we call the SBDRP algorithm, is similar to the SBDR algorithm described earlier except in Step 3, β_i of eq. 3.26 is replaced by eq. 3.28.

$$\beta_i = \sum_{x=1}^{m} w_x^i S_x^i \tag{3.28}$$

Illustrative Example 4

Using the same example the alternatives are ranked based on the β_i values found by eq. 3.28 instead. The summary results are given in Table 3.35.

Alternatives	TOTAL
Supplier E	0.9329
Supplier D	0.2545
Supplier J	0.2154
Supplier A	0.1207
Supplier F	0.0218
Supplier G	0.0208
Supplier B	0.0089
Supplier H	0.0070
Supplier I	0.0063
Supplier C	0.0054

Table 3.35: Ranking based on Total Score and Position for Data Set 1

Variant 3- The incorporation of the position of the alternatives and the preferences for the MCDM methods in SBDR (SBDRPP)

In this version, in addition to using the position and score of the alternatives as in variant 2, the weights of the methods are also considered. This is in case the decision makers have some preferences among the methods. It is important to stress that solving MCDM problems, often requires additional preference information from a decision maker. In this experiment, for simplicity we reflect this aspect by generating random weights a decision maker would assign to a given method and hence to its criteria. Due to the high degree of subjectivity of the decision makers and the appropriateness of a given method for a given decision problem, different ways of determining these weights could be worth examining, some of which are highlighted in the suggestion Section.

This algorithm which we call the SBDRPP algorithm is also similar to the SBDR algorithm where Steps 1,2 and 4 are left unchanged except Step 3 where eq. 3.26 is replaced by eq. 3.29.

$$\beta_i = \sum_{x=1}^{m} \alpha_x w_x^i S_x^i \tag{3.29}$$

Illustrative Example 5

We also used the same example here. For this data set, we generated preferences of the decision makers(α) randomly as in the below: $\alpha_1 = 0.166$ $\alpha_2 = 0.666$ $\alpha_3 = 0.166$

The final results are summarized in Table 3.36.

Alternatives	TOTAL
Supplier E	0.3085
Supplier D	0.0849
Supplier J	0.0637
Supplier A	0.0374
Supplier F	0.0082
Supplier G	0.0069
Supplier B	0.0039
Supplier H	0.0032
Supplier I	0.0029
Supplier C	0.0025

Table 3.36: Ranking based on Total Score, Position and Weight of the Methods for Data Set 1

3.6 Computational Experiments

The proposed deterministic and integrated approaches, namely, SBDR, SBDRP and SBDRPP are tested on the 10 data sets obtained by our data set generator described in the previous section. New ranking lists are obtained and their respective deviations from the original lists derived by the individual methods, namely, AHP, TOPSIS and VIKOR are evaluated. Note that the results of the rank based algorithm RBDR will provide a set of lists each of which dominates the ones found by the individual methods whose performances, excluding RBDR, is shown in Table 3.29. For simplicity, a similar table with RBDR scoring 10 and the others having their original values increased accordingly, is not reproduced here. This section will therefore focus on the performance of the three score-based variants when compared against the individual methods.

3.6.1 Individual Methods vs. SBDR

The total deviations of the lists according to the score-based deterministic rule and the three individual methods are presented in Table 3.37. As shown in this table, in every run, the deviation of the list found by the proposed algorithm SBDR is less than or equal to the deviations of the other individual methods. For convenience, the average results of the deviations of each method are also computed, with the best represented by 'bold'. The overall average percentage deviation from the best is computed as

Overall Average Deviation (%) = 100×(Average(method)-Best average)/Best average
with Average(method) representing the average deviation of a given method and the Best average refers to the smallest average over all the four methods. The results in Table 3.37 demonstrate that SBDR is robust and outperforms the other individual methods by a large margin. For instance, TOPSIS which is the best individual method is 45% worse than SBDR whereas AHP and VIKOR are 90% and 74% worse than SBDR respectively.

	AHP	TOPSIS	VIKOR	SBDR
Data Set 1	0.00434	0.00067	0.00074	0.00067
Data Set 2	0	0.00037	0.00099	0
Data Set 3	0.26752	0.05504	0.08184	0.04699
Data Set 4	0.11932	0.09823	0.09708	0.08484
Data Set 5	0.00066	0.00122	0.00122	0.00033
Data Set 6	0.00082	0.02881	0.00225	0.00082
Data Set 7	0.00143	0.00243	0.00243	0.00143
Data Set 8	0.09275	0.08276	0.08276	0.03729
Data Set 9	0.13509	0.11873	0.26411	0.11061
Data Set 10	0.29880	0.31251	0.30785	0.20029
Average	0.09207	0.07008	0.08413	**0.04833**
Overall Average Deviation (%)	90.51	45.00	74.28	**0.00**

Table 3.37: Deviation results of AHP, TOPSIS, VIKOR and SBDR

3.6.2 Individual Methods vs. SBDRP

The SBDRP algorithm is also tested on the same 10 generated data sets resulting in new ranking lists. A comparison of the deviations of the proposed algorithm SBDRP and the individual methods is given in Table 3.37. The deviation obtained by SBDRP is less than or equal to the deviations of the other three methods in each run. This is a similar pattern shown by the original SBDR algorithm. According to the average deviations over the 10 runs, SBDRP is more effective than the AHP, TOPSIS and VIKOR as it yields the smallest deviation individually and obviously on average. For clarity, the best value is shown in 'bold'. With respect to the overall average deviation TOPSIS scores again as the best individual performer with over 18% worse than SBDRP whereas AHP and VIKOR have a relatively poorer performances with 110% and over 41% worse than SBDRP respectively.

	AHP	TOPSIS	VIKOR	SBDRP
Data Set 1	0.02594	0.01069	0.01077	0.01069
Data Set 2	0.00001	0.00012	0.00062	0.00001
Data Set 3	0.85326	0.25020	0.39873	0.25020
Data Set 4	0.57181	0.25416	0.25767	0.25767
Data Set 5	0.00048	0.00041	0.00041	0.00025
Data Set 6	0.01957	0.06041	0.02443	0.01744
Data Set 7	0.04214	0.01839	0.01839	0.01839
Data Set 8	0.66163	0.30965	0.30965	0.28615
Data Set 9	0.59683	0.49525	0.77675	0.45946
Data Set 10	0.80481	0.61952	0.61751	0.40284
Average	0.35765	0.20188	0.24149	**0.17031**
Overall Average Deviation (%)	110.00	18.54	41.79	**0.00**

Table 3.38: Deviation results of AHP, TOPSIS, VIKOR and SBDRP

3.6.3 Individual Methods vs. SBDRPP

The third variant namely the SBDRPP algorithm is also tested on the 10 generated data sets. The comparison of the deviations is given in Table 3.37. The deviation found with this version is also equal or less than the other three methods in each data set except data set3. In this particular data set, the deviation of TOPSIS is slightly smaller than our proposed method. This is mainly due to the large weight associated with TOPSIS. However, the average deviation of SBDRPP is still relatively smaller than the other three MCDM methods, with the best result shown in 'bold'. Again with respect to the overall average performance, TOPSIS is found to be the best individual method with over 11% worse than SBDRPP while AHP and VIKOR produce nearly 89% and over 40% relatively poorer performances compared to SBDRPP.

	AHP	TOPSIS	VIKOR	SBDRPP
Data Set 1	0.02867	0.01069	0.01077	0.01069
Data Set 2	0.00001	0.00012	0.00062	0.00001
Data Set 3	0.88673	0.25135	0.39162	0.25241
Data Set 4	0.57831	0.25462	0.25758	0.24978
Data Set 5	0.00049	0.00040	0.00040	0.00025
Data Set 6	0.02222	0.05348	0.02787	0.01996
Data Set 7	0.03975	0.01839	0.01839	0.01839
Data Set 8	0.70054	0.30729	0.30729	0.28570
Data Set 9	0.56942	0.48809	0.92110	0.46245
Data Set 10	0.74836	0.72077	0.71858	0.59221
Average	0.35745	0.21052	0.26542	**0.18919**
Overall Average Deviation (%)	88.94	11.27	40.29	**0.00**

Table 3.39: Deviation results of AHP, TOPSIS, VIKOR and SBDRPP

3.7 Summary

We presented an effective deterministic and integrated approach which takes into account the results of the different individual ranking methods with the aim to demonstrate that relying on one individual method can be misleading to decision makers. For illustration, the performances of three commonly used MCDM approaches, namely, AHP, TOPSIS and VIKOR are investigated. This is achieved by designing an approach that considers the score and rank based regret measures. We show that our approach outperforms the individual methods resulting in providing a relatively more robust outcome for the decision maker.

For example, in terms of the overall average deviation our approach outperforms the best individual method, namely, TOPSIS by at least 11% and at most by 45%. On the other hand, AHP and VIKOR produced relatively poorer performances compared to the proposed approach reaching, in some cases, over 100% deterioration for AHP and over 41% for VIKOR.

In summary, through these empirical experiments, we demonstrate that it is much better not to rely on one MCDM method only when it comes to supplier selection. Some aspects on how to enhance the proposed approach are provided in the last chapter of this dissertation under the suggestion section.

The next chapter will deal with the case where there is a lack of expert knowledge using Bayesian networks.

Chapter 4

Integration of MCDM Methods & Bayesian Networks for Supplier Selection with Incomplete Expert Knowledge

The integrated and deterministic approach in the previous chapter is based on the assumption that there is complete knowledge. In this chapter, we explore the case when that assumption is not always valid leading to a lack of expert knowledge. To achieve that we will integrate MCDM methods and Bayesian Networks. Illustrative example is provided throughout and computational results given. We first start by presenting the strengths and weakness of some of the MCDM methods that we use in this work, followed by the main ingredients that form the basis of the methodology. Our novel approach is then presented in Section 3 and the case study in Section 4 and a scenario analysis in Section 5 using new performance measures.

4.1 Overview of the owning Expert Knowledge and Dynamic Structure of Supplier Selection Problem

In many logistical problems such as the supplier selection problem, data is not always readily available or is often rather limited. For instance, when new suppliers are considered, some performance criteria cannot be truly determined and some sellers may not want to share all information or spend resources on gathering information sought by buyers (Igarashi et al. 2013).

One way forward is for the decision makers to take advantage of the experts' knowledge and perception in such circumstances. The buyer may be able to gather data about some criteria, such as price and reputation through market research. On the other hand, the buyer may form a judgement about some criteria like cooperation and communication abilities based on the purchasing interviews. In addition, a buyer may ask for sample parts which can be used to evaluate the suppliers in terms of product quality, delivery performance, among others. Though it is not possible to gain knowledge about the supplier performance based on the sample parts, such as reliability and price stability. Our aim is to develop a robust approach which takes into account incomplete expert knowledge. We believe this approach is important for effective decision making. Multi criteria decision making methods (MCDM) as shown earlier consider expert knowledge, however,they are not able to work with incomplete expert knowledge and these are often deterministic methods that are not designed to deal with uncertainty. However, supplier selection problem has uncertain and dynamic environment. Uncertainty must be considered and alternative suppliers must be evaluated dynamically. Bayesian Networks(BNs) are able to work with incomplete expert knowledge. According to the prior belief of the buyers about some evaluation criteria, BNs estimate the other criteria. When buyers have some new evidence about criteria, it updates the network based on the obtained evidence dynamically.

One philosophy is that experts are encouraged to submit their knowledge quantitatively although they may feel more comfortable providing such information qualitatively. The tendency of a qualitative submission and the uncertainty in expert knowledge prompt to integrate MCDMs with fuzzy approach (Jain et al. 2018, Chang 2019, Mohammed, Harris, Soroka & Nujoom 2019). Govindan et al. (2015) emphasize the fuzzy prevalence in the review of multi-criteria decision making approaches for green supplier evaluation and selection. There is a lot of research on fuzzy approaches. Here, the idea is to transform linguistic expressions of experts such as low, medium and high into a trapezoidal or triangular type membership (Zhang et al. 2017, Liu, Eckert, Yannou-Le Bris & Petit 2019, Darbari et al. 2019). On the other hand, Liu & Li (2019) use multi attribute decision making method based on generalized maclaurin symmetric mean aggergation operators for probabilistic linguistic information. In this paper, we propose to use Ranked Nodes to model qualitative judgements of experts in Bayesian Networks (BNs). This type of network has been shown to provide effective tools for decision making based on expert knowledge (Fenton et al. 2007). BNs have the advantage of being able to deal with uncertainty while considering the causal relationship between the decision criteria (Jensen & Nielsen 2013). For example, Dogan & Aydin (2011*b*) integrated BNs and Total Cost of Ownership for the supplier selection problem in the automotive industry while Hosseini & Ivanov (2019) used BNs for the resilience measure of supply networks.

In case data is not available or limited, expert knowledge becomes an invaluable alternative source of information. However, full expert knowledge is not always available. BNs has the flexibility to work even if there is incomplete knowledge. This is usually achieved by estimating the missing knowledge based on the causal relationship between the criteria. This can be considered as a dynamic supplier evaluation tool. In other words, when a buyer has a new evidence about a given criterion, BNs update the entries of the network based on the entered evidence as posterior probabilities. This demonstrates that BNs are effective tools to evaluate alternative suppliers based on the causal relationship between the selection criteria. However, it is worth stressing that there is no systematic way of determining the causal structure of BNs. Kaya & Yet (2019) adopted DEMATEL to resolve this issue with DEMATEL being used to determine the cause-effect relationship between the criteria. Very recently, Li et al. (2020) also used DEMATEL for the supplier selection problem in a Chinese Textile Industry to determine the most influential criteria for the evaluation of suppliers based on the cause-effect relationship between the criteria. Kaya & Yet (2019) determined the causal relationship between the criteria based on DEMATEL and then parameterized the BN with Ranked Nodes systematically. DEMATEL results are also used for parameters of Ranked Nodes. In this network, the buyer can evaluate alternative suppliers based on each criterion. However, it is worth noting that the network does not rank the alternatives based on the overall performance. In this study, we propose to extend the work of Kaya & Yet (2019) by producing a ranking of the suppliers. To achieve this, we incorporate TOPSIS into our methodology. TOPSIS is one of the commonly used MCDM methods for the supplier selection problem. It works based on distance to the ideal solution, see Wang et al. (2009), Mohammed (2019). Inputs of TOPSIS are weights of the criteria and the evaluation matrix of the alternatives based on the criteria. In case there is a lack of data, the evaluation of the alternatives based on criteria is carried out on a scale that needs to be determined. Mostly, as experts prefer to submit their knowledge by linguistic expressions, fuzzy approach is usually integrated with TOPSIS (Nilashi et al. 2019, Venkatesh et al. 2019). As the full expert knowledge is not always possible, BN provides a probabilistic full evaluation matrix for TOPSIS based on the expert knowledge which is submitted with linguistic expressions. The missing knowledge is incorporated based on the causal relationship between the criteria which then provides a complete and an updated evaluation matrix.

Another input of TOPSIS is the weights of the decision criteria. It is hard to elicit these weights quantitatively. Liu & Wan (2019) use ELECTRE I and III for the weights of criteria. One way forward, which is commonly adopted in the literature, is to integrate AHP with TOPSIS for the elicitation of the weights (Jain et al. 2018, Akgün & Erdal 2019). AHP is the most common MCDM method for supplier selection (Chai et al. 2013, Govindan et al. 2015, Kahraman et al. 2003). It works based on the pairwise comparison of criteria and alternatives (Saaty 2008). Rodrigues et al. (2014) compares TOPSIS and AHP methods for the supplier selection problem and shows that TOPSIS is more practical in terms of the number of criteria, alternatives and decision makers, and the agility in decision as AHP has a hierarchical structure and it process pairwise comparison at each level. In this study, we propose TOPSIS as a ranking approach and AHP for the elicitation of the weights of criteria for TOPSIS. Recently, Singh et al. (2018), Venkatesh et al. (2019) also used AHP for the elicitation of weights of the criteria for the upplier selection problem. The elicitation of the weights of the criteria can also be determined by DEMATEL (Baykasoglu et al. 2013).

In the case study performed in this paper, the weights of the criteria are initially calculated by both DEMATEL and AHP for a better understanding. However, the discussion with the experts led to a conclusion that the AHP-based results represent their preferences much more than those derived by DEMATEL. It is also worth noting that DEMATEL aims to determine the weights based on the cause and effect relationships and prioritizes the cause attributes than the effect attributes and AHP on the other hand prioritizes the criteria based on the relative preferences of the experts. It is therefore appropriate to take advantages of the strengths of these two methods by using DEMATEL for building causal graph and AHP for the calculation of the weights of the criteria.

Integrated approaches compensate the limitations of individual methods. Büyüközkan & Ifi (2012) integrated fuzzy DEMATEL, fuzzy ANP and fuzzy TOPSIS methods for the evaluation of green suppliers. DEMATEL determines causal relations between criteria, ANP compares alternatives pairwise and TOPSIS ranks the alternatives. Fuzzy hybridization is also used in this study due to the vagueness of expert judgements. On the other hand, Mathematical Programming techniques (MP) are effective methods as they give exact and objective solutions (Moheb-Alizadeh & Handfield 2019, Izadikhah & Farzipoor Saen 2019). However, they are not easily extendable to deal with uncertainty. AI techniques are effective techniques for dealing with uncertainty and giving updated and dynamic solutions(Luan et al. 2019). The integration of MCDM, MP and AI methods provide more effective methods in which the weakness of a given method is compensated by the others,

(Ramanathan 2007, Dogan & Aydin 2011*b*, Liu, Quan, Li & Wang 2019)

In this study, we combined the strengths of MCDM and BN in a most effective way. We use DEMATEL to elicit the causal graph of the BN based on the causal knowledge of the experts. BN provides the evaluation of alternatives based on the decision criteria which make up the initial decision matrix of TOPSIS. We then parameterize BN using Ranked Nodes which allows the experts to submit their knowledge with linguistic expressions. We propose AHP to determine the weights of the decision criteria and TOPSIS to rank the alternatives. A supplier selection case study is conducted to illustrate the effectiveness of the proposed approach. Two evaluation measures, namely, the number of mismatches and the distance due to the mismatch are developed to assess the performance of the proposed approach. A scenario analysis with 5% to 20% of missing values with an increment of 5% is conducted to demonstrate that our approach remains robust as the level of missing values increases. The strengths and limitations of the methods used in the proposed approach are summarised in Figure 4.1. This integration provides a systematic and user friendly way to evaluate the alternatives. It helps the experts to submit their available knowledge with linguistic expressions and calculates the relative performances of the suppliers in a probabilistic way. One of the important contributions of this study is the ability of working with incomplete knowledge and making reliable probabilistic estimations.

Method	Strengths	Limitations
AHP	Ability to work with expert Judgement Elicitation of weights of criteria Consistency Check	Impractical for many alternatives Elicitation of decision matrix Inefficient in dealing with uncertainty Inability to analyze causal relations between criteria
TOPSIS	Ability to work with expert Judgement Ranking based on similarity to ideal solution	Elicitation of full numerical decision matrix Inefficient in dealing with uncertainty Inability to analyze causal relations between criteria
DEMATEL	Analysis of causal relations	No ranking
Bayesian Networks	Estimation with partial expert knowledge Dealing with uncertainty with probabilistic approach Analysis of causal relations Dynamic structure	Elicitation of probabilities(NPTs)

Figure 4.1: Strengths and Limitations of the Methods used in the Proposed Approach

The main contributions of this study are as follows:

(i) An effective integration of MCDM methods and BNs for multi criteria decision problems is proposed.

(ii) This novel approach is able to work with both complete and incomplete expert knowledge.

(iii) Ranked Nodes are used for easy elicitation of probabilities of BN based on linguistic expressions of experts.

(iv) The value of the knowledge is analysed and a scenario analysis is conducted with interesting results.

The rest of the chapter is organised as follows. The techniques that are used in this study are outlined in Section 2 followed by the proposed approach in Section 3. An illustrative example using a case study is provided in Section 4 and a scenario analysis is presented in Section 5. Our conclusion and suggestions are provided in the final section.

4.2 Main Ingredients of the Methodology

The proposed approach will be described in the next section. Here, we present some of the techniques that will be used. The following three MCDM approaches namely, the AHP, TOPSIS and DEMATEL will be incorporated into our methodology. We also use Bayesian Networks (BNs) with Ranked Nodes (RNs). As these are relatively less known, we first provide a brief description of BNs and how RNs are implemented.

4.2.1 MCDM Methods

AHP and TOPSIS methods are explained in details in the previous chapter. In addition these methods, we use DEMATEL in the proposed approach.

DEMATEL

DEMATEL is an MCDM method used for determining the causal relationship between the criteria and strengths of the relationship. It works based on two matrices: direct relation matrix which shows the direct relationship between the criteria and total relation matrix which shows the total direct and indirect influences between the criteria (Chang et al. 2011). The detailed steps of DEMATEL is given in the following (Kaya & Yet 2019):

1. The direct relation matrix A is built by asking experts the influence of decision criteria on each other in scale of 0-4. The average of the responses of the experts are recorded.

2. A normalized direct relation matrix M is obtained by dividing values of the direct relation matrix A with the maximum of sum of rows and columns:

$$M = A * \min(\frac{1}{\max \sum_i^n a_{ij}}, \frac{1}{\max \sum_j^n a_{ij}}) \tag{4.1}$$

   where a_{ij} is the average direct relation matrix value for row i and column j.

3. The total relation matrix T represents the sum of direct and indirect relations:

$$T = M + M^2 + M^3 + M^4 + \dots \tag{4.2}$$

   It is calculated as follows:

$$T = M(I - M)^{-1} \tag{4.3}$$

   where I is the identity matrix.

4. For each criterion, the sum of the associated row R and column C is computed. If $(R - C) > 0$, the criterion is defined as cause criterion, else the criterion is defined as effect criterion. Strengths of the criteria are defined by (R+C) values.

5. The threshold value is determined by the experts to determine the causal effects between the criteria. The influence values which are greater than the threshold value accepted as causal influences and causal network is built.

   Figure 4.2 shows an example causal graph built by DEMATEL(Kaya & Yet 2019). The causal graph with DEMATEL is built based on the total relation matrix of DEMATEL. However, the causal graph of BN cannot be built based on total relation matrix as the total relation matrix includes direct and indirect relations and the causal graph of BN can only include direct relations. DEMATEL results need to be transformed into the causal graph of BN. We adopted the methodology of (Kaya & Yet 2019). Direct relation matrix which consists of only direct relations is used for building causal graph of BN and matrix values are used for parameterization of BN.

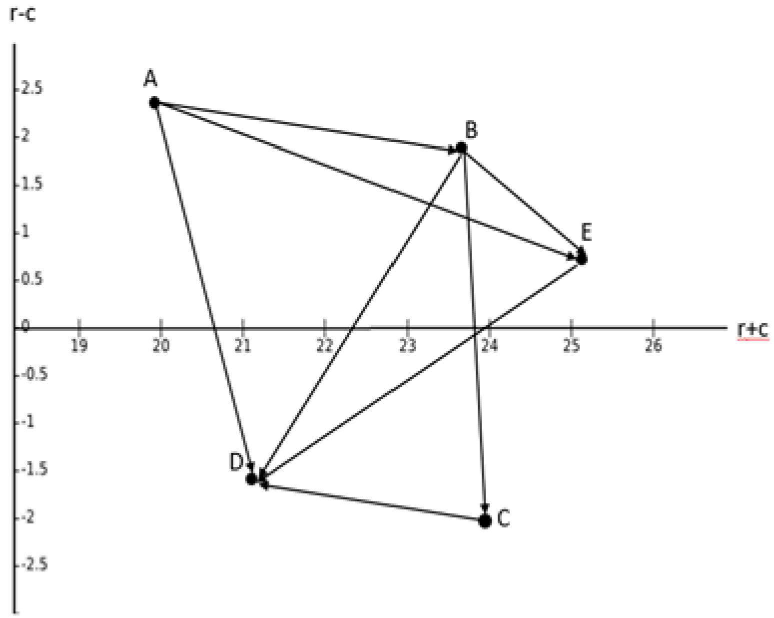


Figure 4.2: Example DEMATEL Graph

4.2.2 Bayesian Networks

Bayesian Networks are probabilistic graphical decision making tools (Fenton & Neil 2013). They work based on Bayes' Theorem and make inferences based on the prior beliefs of experts. They are able to make inferences even with partial evidence. When new evidence is obtained, BNs calculate the posterior probabilities and update all the network based on the new evidence. BNs are comprised of nodes and arcs which represent variables and causal relationship between variables respectively. Each node has its own Node Probability Table(NPT) which includes the conditional probability distribution parameters of that node (Kaya & Yet 2019). It calculates the joint probability distributions based on Bayes Theorem. Joint probability distributions are calculated based on the parent and child relations of the nodes. Example BN for X, Y, Z and T events is given in Figure 4.3.

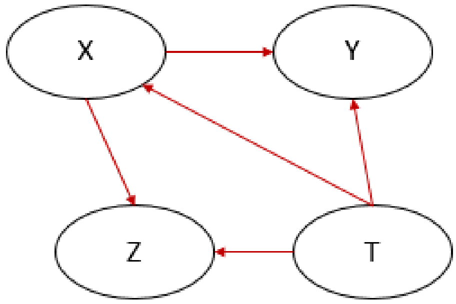


Figure 4.3: Example BN

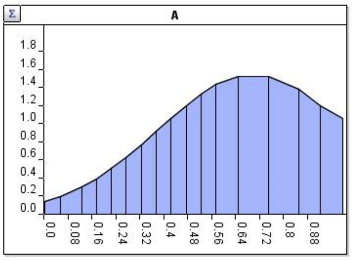
The joint probability distribution of these events is calculated as below:
$P(X,Y,Z,T) = P(X|T)P(Y|X,T)P(Z|X,T)P(T)$

BNs are acyclic directed graphical networks. Cycles are not allowed between the nodes. BNs have the advantage in terms of representation of causal relationships between variables graphically. It analyses the causal relationship between variables probabilistically and systematically. BNs have also the flexibility in working with expert knowledge resulting in producing updated results based on the obtained new evidences.

4.2.3 Ranked Nodes

Ranked Nodes (RNs) are expert friendly tools of BNs for decision making based on human judgement. RNs work based on doubly Truncated Normal (TNormal) distribution with scaled states [0-1] and approximate this distribution with a discrete BN node with equal width intervals (Fenton et al. 2007). RNs work with weighted functions such as the weighted mean (WMEAN), the weighted minimum (WMIN) and the weighted maximum (WMAX). These three measures are used to determine the central tendency of child node depending on the parent nodes.

An illustrative example is displayed in Figure 4.4 by (Kaya & Yet 2019). This shows a TNormal distribution with mean 0.7 and variance 0.1 on the left and ranked node approximation of this distribution on the right. This ranked node has 5 states, so it approximates the probability density under 5 equal width intervals (i.e., [0,0.2), [0.2,0.4), [0.4,0.6), [0.6,0.8) and [0.8,1]).



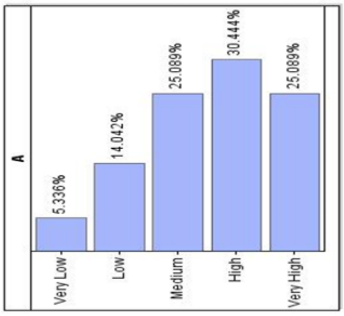


Figure 4.4: Graph of an example node with TNormal Distribution(left) and with Ranked Nodes(right)

The main advantage of ranked nodes is that they require fewer parameters than their node probability tables (NPTs) counterparts. Besides, RNs are flexible enough to define a wide variety of shapes. An NPT has probability values of a node for each state combination of its parents. Therefore, the number of parameters in an NPT is the cartesian product of the number of its parents' states and its states. For example, the BN model in Figure 4.5 has three variables X, Y, Z where X is dependent on Y and Z, and each node has 5 states. Without using RNs, the number of probability values that need to be elicited from experts for the NPT of X is therefore $5^3 = 125$ as seen in Figure 4.6. This is not only a time consuming task for the experts but can also be confusing resulting in misleading information.

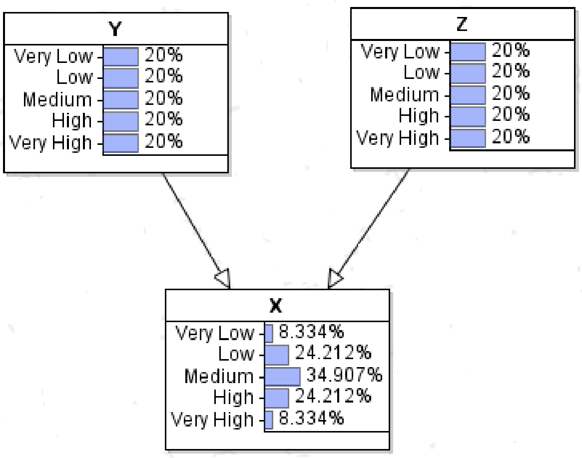


Figure 4.5: Example Network

Y	Very Low					Low		
Z	Very Low	Low	Medium	High	Very High	Very Low	Low	Medium
Very Low	0.96763027	0.4377141	0.010672107	2.5189054	0.0	0.5622859	0.03236970	2.1454595E
Low	0.03236970	0.56228495	0.91646445	0.3193347	0.00233947	0.43771407	0.9352606	0.4377140
Medium	0.0	9.68281E-7	0.07286344	0.6806418	0.8634297	2.1454595E-	0.03236970	0.5622849
High	0.0	0.0	8.881784E-18	2.3433357	0.13423082	0.0	0.0	9.68281E-7
Very High	0.0	0.0	0.0	0.0	5.0315307E-	0.0	0.0	0.0

Figure 4.6: NPT of X

The construction of NPTs by ranked nodes consists of the following steps.
- Firstly, the states of a ranked node are determined.
- The type of the weighted function to be adopted is selected.
- The weights and variances of its parents are determined.
- NPTs are then calculated based on the TNormal approximation. This is performed automatically by the software AgenaRisk (Fenton et al. (2007)).

For instance, if we use ranked nodes for our example model in Figure 4.5, we need to define 3 parameters. These include the weights of Y and Z and the variance of X to define NPT of X as shown in Figure 4.7.

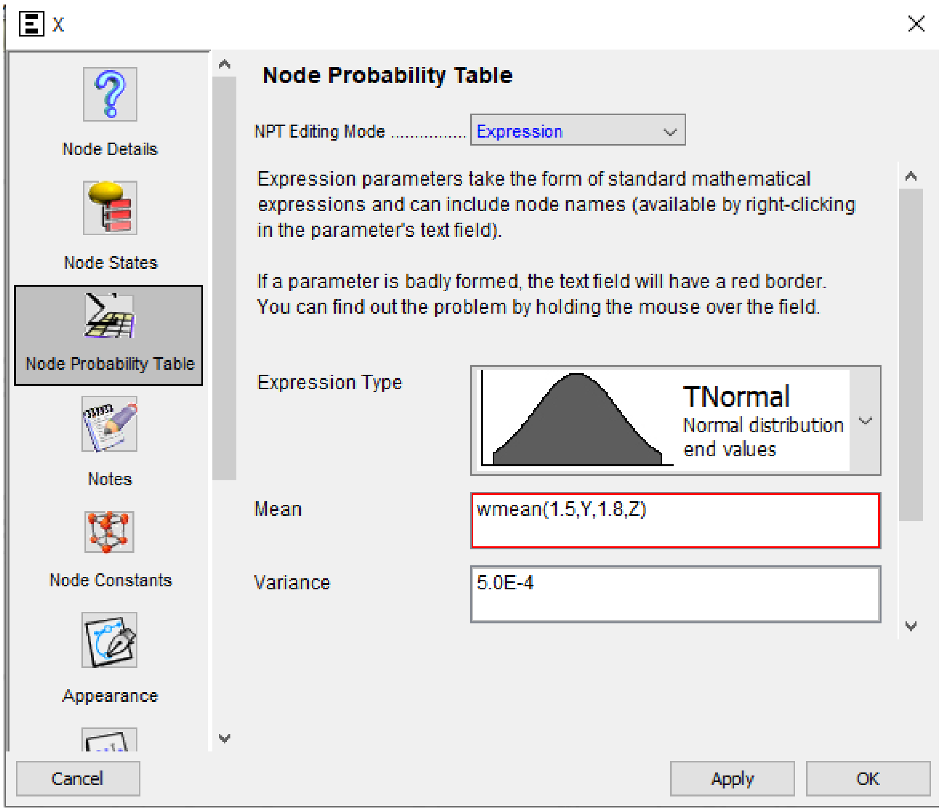


Figure 4.7: Parameters for NPT of X with ranked nodes

4.3 A Novel Integrated Approach

We develop an integrated approach that combines DEMATEL, AHP, TOPSIS and BNs for the supplier selection problem. We first provide an overview of the algorithm, followed by the algorithm itself and some explanation of the main steps.

4.3.1 An Overview

We adopt an approach that consists of four stages,

1. Use DEMATEL to determine the causal graph of BN,
2. Apply AHP to find the weights of the criteria,
3. Implement BNs to provide the evaluation matrix of alternatives for TOPSIS,
4. Use TOPSIS to rank the alternatives.

A basic flow chart is given in Figure 4.8 and the main steps of the algorithm which we refer to as "MCDM-BN" are summarized in Figure 4.9. It is worth noting that this approach can easily be made applicable for other multi-criteria type problems.

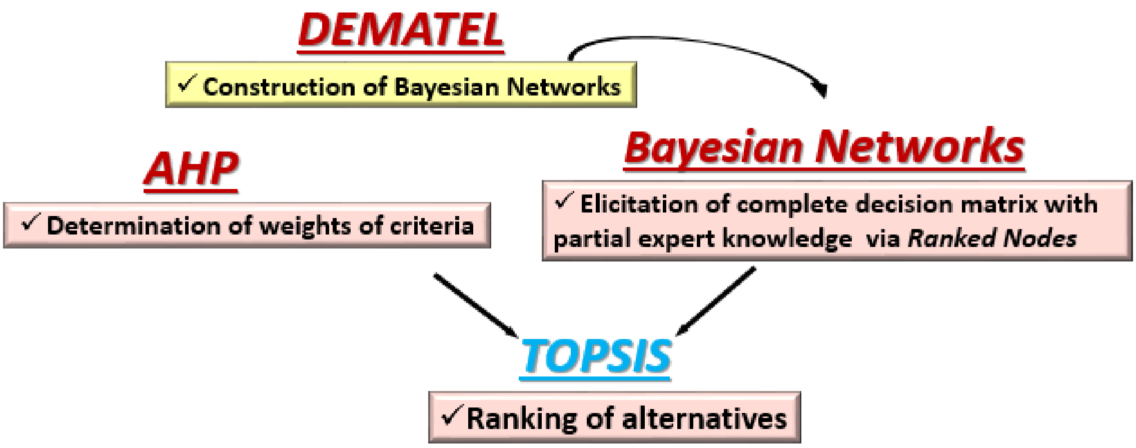


Figure 4.8: Flow Chart of the MCDM and BN Integrated Approach(MCDM-BN)

4.3.2 The MCDM-BN Algorithm

Figure 4.9 describes the summary of the MCDM-BN algorithm.

1. Identify the decision criteria
2. Find the weights of the criteria using AHP
3. Determine the causal relationship between the decision criteria by DEMATEL
4. Define the states of BN and construct the BN
5. Parameterize the BN with Ranked Nodes
6. Elicit the decision matrix from the experts
7. Estimate the missing values in the decision matrix using BN
8. Rank the alternatives with TOPSIS.

Figure 4.9: The Integrated MCDM-BN Algorithm

4.3.3 Explanation of the Main Steps of the MCDM-BN Algorithm

In Step 1, experts determine the main criteria for the supplier selection decision.

In Step 2, the relative importance of the decision criteria are found using the pairwise comparison of criteria of AHP. There are obviously several other available methods for determining the weights of the criteria. These include for instance the entropy method, Step-wise Weight Assessment Ratio Analysis (SWARA) and Simos method (**?**). DEMATEL can also be applied as a weighting method (Baykasoglu et al. 2013). As we already use DEMATEL in our proposed approach for the construction of BNs, we also elicit the weights of the criteria based on DEMATEL and evaluate the results with the experts in the case study. In the last step of DEMATEL, the total relation matrix(T) is obtained. This matrix has two important indicators, namely, the importance indicator(t^+) and the relation indicator(t^-) which are sums of and differences between the rows and columns of the total relation matrix, respectively. The weights of the criteria which are represented by w_i are then calculated with the following formulas:

$$z_i = \left((t^{+})^2 + (t^{-})^2\right)^{1/2} \tag{4.4}$$

$$w_i = \frac{z_i}{\sum_{i=1}^{n} z_i} \tag{4.5}$$

We conducted a DEMATEL survey with purchasing experts. The total relation matrix(T) is presented in Table 4.1 and the weights of the criteria which are calculated based on DEMATEL are given in Table 4.2.

	Product Quality	Delivery Performance	Price	Cooperation	Reputation
Product Quality	0	0.261	0.526	0	0.505
Delivery Performance	0	0	0.399	0	0.296
Price	0	0	0.015	0	0.088
Cooperation	0.217	0.405	0.276	0	0.345
Reputation	0	0	0.177	0	0.015

Table 4.1: Total Relation Matrix

	t^{+}	t^{-}	w_i
Product Quality	1.509	1.075	0.210
Delivery Performance	1.360	0.029	0.160
Price	1.498	-1.290	0.230
Cooperation	1.243	1.243	0.200
Reputation	1.441	-1.057	0.200

Table 4.2: Weights of the criteria by DEMATEL

Kobryń (2017) produced an interesting modification of the formulas of the calculation of the weights of the criteria using DEMATEL. These are defined in the following equations:

$$\bar{t_i} = \frac{1}{2}(t^{+} + t^{-}) \tag{4.6}$$

$$w_i = \frac{\bar{t_i}}{\sum_{i=1}^{n} \bar{t_i}} \tag{4.7}$$

The corresponding results related to the modified DEMATEL rule as defined by Kobryń (2017) are presented in Table 4.3.

	$\bar{t}_i$	w_i
Product Quality	1.292	0.370
Delivery Performance	0.695	0.200
Price	0.104	0.030
Cooperation	1.243	0.350
Reputation	0.192	0.050

Table 4.3: Weights of the criteria by DEMATEL using the modified rule

We also calculated the weights of the criteria based on the AHP in Step 2 of the case study. The results are given in Table 4.5.

We evaluated the results of these two strategies with the experts. They state that the weights elicited by AHP represent better their preferences. DEMATEL is used to determine the cause-effect relationships and gives more importance to the criteria which have cause-effect on the other criteria. Based on the above information, we therefore opted to use AHP for the elicitation of the weights of the criteria.

In Step 3, BNs evaluate the alternatives based on the causal relationship between the criteria. When any evidence is entered to any criterion, BNs update the rest of the network based on the causal relationship between the criteria which are determined by DEMATEL. However, it is worth noting that the initial causal graph which was obtained by DEMATEL may not be a convenient causal network for BNs as it may include cycles. After the construction of the initial causal graph, cycles are eliminated using the interesting rules constructed by Kaya & Yet (2019).

In Step 4, the causal graph of BN is built based on the causal graph obtained from DEMATEL where the states of the nodes in BN are also determined.

In Step 5, we use Ranked Nodes to parameterize the BN. This is mainly because these are easy to elicit the expert knowledge from experts as a parameter of BN. The mean function of Ranked Nodes, known as the WMEAN function, is used as the Ranked Node function. The central tendency and variance of child nodes are determined with the weights and variances of the parent nodes via the WMEAN function. The weights of the parent nodes are then elicited from the direct relation matrix of DEMATEL. On the other hand, the variance values are summed for each child node and normalized to the unit scale of TNormal Distribution. In this study, AgenaRisk software is used to compute the BN model automatically where ranked nodes are already inserted in the software.

In Step 6, the decision matrix is elicited from experts as it is one of the inputs of TOPSIS. In this matrix, alternatives are evaluated based on the selection criteria by the experts. Experts may not have a complete knowledge about all attributes of the suppliers. In this case, they submit their available knowledge about the alternatives.

In Step 7, in case there are missing values in the decision matrix for TOPSIS, BNs estimate these elements to complete the decision matrix.

In Step 8, TOPSIS uses the weights of the criteria and the decision matrix as an input and proceed with the matrix calculations to compute the geometric distance to the best and the worst alternatives. The alternatives are then ranked based on the smallest distance to the best alternative and the largest distance to the worst alternative.

4.4 Case Study

To illustrate the approach, we use the following example based on a case study carried out with a forging company in Turkey. This company outsources machining operation and they have alternative suppliers. In this case study, we evaluated the eight alternative suppliers used by the company, For simplicity, we refer to these as Supplier A,..., Supplier H. We follow the step by step of the approach as given in Figure 4.9.

1. *Identify decision criteria*: Experts determined the decision criteria as product quality, delivery performance, price, cooperation and reputation based on the supplier selection criteria given in Kaya & Yet (2019).

2. *Determine the weights of criteria*: AHP was conducted to determine the weights of the criteria. The pairwise comparison matrix of criteria is presented in Table 4.4.

Criteria	Product Quality	Delivery Performance	Price	Cooperation	Reputation
Product Quality	1	1	1	5	5
Delivery Performance	1	1	1	3	5
Price	1	1	1	3	5
Cooperation	1/5	1/3	1/3	1	3
Reputation	1/5	1/5	1/5	1/3	1

Table 4.4: Pairwise Comparison Matrix of the Criteria

After processing the matrix computations, the weights of the criteria are obtained and presented in Table 4.5.

Criteria	Weight
Product Quality	0.31
Delivery Performance	0.27
Price	0.27
Cooperation	0.10
Reputation	0.05

Table 4.5: Weights of the Criteria

A consistency check is then conducted and the consistency index(CI) is found as 0.028. As $CI < 0.1$, the judgements are considered as consistent.

3. *Determine the causal relationship between the decision criteria*: Causal relationships between criteria are determined by DEMATEL. We conducted DEMATEL survey with purchasing experts from the company. The direct relation matrix of the DEMATEL is presented in Table 4.6.

Criteria	Product Quality	Delivery Performance	Price	Cooperation	Reputation
Product Quality	0	2	2.67	0	3
Delivery Performance	0	0	2.67	0	2
Price	0	0	0	0	0.67
Cooperation	1.67	2.67	0	0	1
Reputation	0	0	1.33	0	0

Table 4.6: Direct Relation Matrix

The threshold value is set to 2 by the experts in this case study. The values which are greater than or equal to 2 are considered accepted as direct causal relation between the corresponding criteria. According to Table 4.6, there is a direct causal relation between product quality and price, delivery performance and price, product quality and delivery performance, cooperation and delivery performance, product quality and reputation, and finally delivery performance and reputation. The causal graph of the criteria is presented in Figure 4.10.

4. *Construct BN and define its states*: As mentioned earlier we use Ranked Nodes. The states of the nodes are therefore determined in an ordinal scale, namely, very high, high, medium, low and very low.

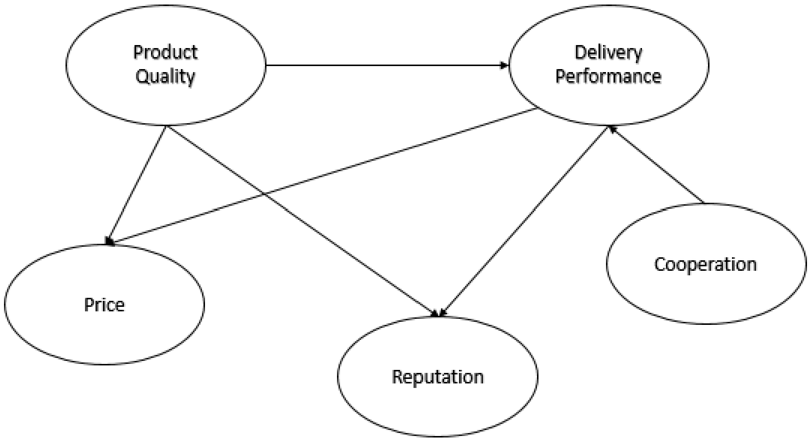


Figure 4.10: Causal Graph

5. *Parameterize the BN with Ranked Nodes*: We propose to use Ranked Nodes to parameterize the BN with WMEAN as the Ranked Node function as we also noted earlier. Weights of the parent nodes were elicited from the direct relation matrix of DEMATEL. For example, according to causal relationship between the criteria, the parents of price are product quality and delivery performance where the weights of product quality and delivery performance for mean of price node are 2.67 and 2.67. On the other hand, the variance values are summed for each child node and normalized to the unit scale of the TNormal Distribution[0-1]. the variances of the values in the matrix are presented in Table 4.7.

Criteria	Product Quality	Delivery Performance	Price	Cooperation	Reputation
Product Quality	0	1	0.33	0	0
Delivery Performance	0	0	0.33	0	1
Price	0	0	0	0	1.33
Cooperation	0.33	0.33	0	0	0
Reputation	0	0	2.33	0	0

Table 4.7: Variances of Criteria

The software AgenaRisk software (Fenton et al. 2007) is used automatically computes the BN model. As an example, see a snapshot in Figure 4.11.

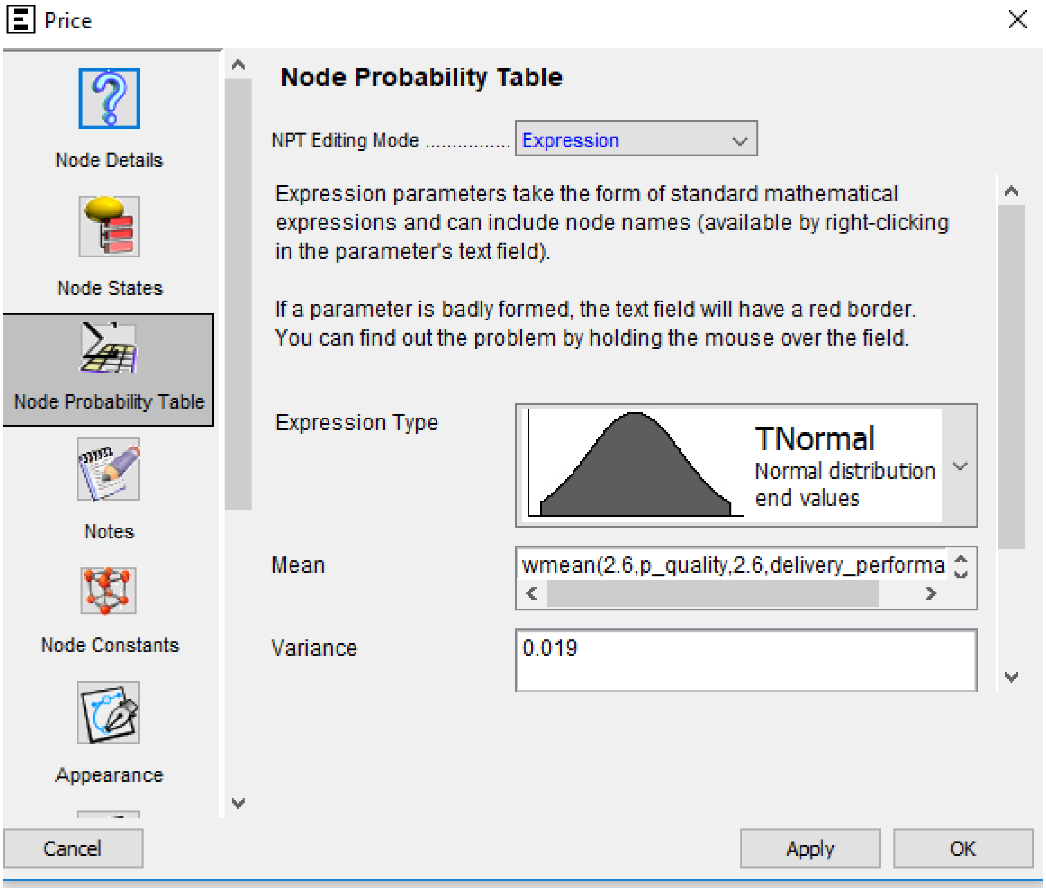


Figure 4.11: Ranked Nodes

6. *Elicit the decision matrix from the experts*: Decision makers submitted their knowledge about the alternatives for each criterion on the five point ordinal scale as very high, high, medium, low and very low. This information is presented in Table 4.8. We then randomly deleted some of the knowledge to make BN estimate the missing knowledge. The evaluation of the alternatives with missing knowledge is given in Table 4.9. We finally entered the knowledge of the experts as evidence into the BN as shown in Figure 4.12.

Alternatives	Product Quality	Delivery Performance	Price	Cooperation	Reputation
Supplier A	Very High	Very High	Very High	High	Very High
Supplier B	Very High	High	Very High	High	Very High
Supplier C	Very High	High	Very High	Medium	Very High
Supplier D	High	Medium	High	Medium	High
Supplier E	High	Medium	High	High	Medium
Supplier F	Medium	Medium	High	Medium	Medium
Supplier G	Medium	High	High	High	Low
Supplier H	Low	Low	Low	High	Low

Table 4.8: Evaluation of Suppliers based on Criteria

Alternatives	Product Quality	Delivery Performance	Price	Cooperation	Reputation
Supplier A	Very High	Very High	Very High	-	Very High
Supplier B	Very High	High	Very High	-	Very High
Supplier C	Very High	High	Very High	Medium	Very High
Supplier D	High	Medium	High	-	-
Supplier E	-	Medium	High	High	Medium
Supplier F	Medium	Medium	High	Medium	Medium
Supplier G	Medium	High	High	-	Low
Supplier H	Low	Low	Low	High	Low

Table 4.9: Evaluation of Experts with missing knowledge denoted by '-'

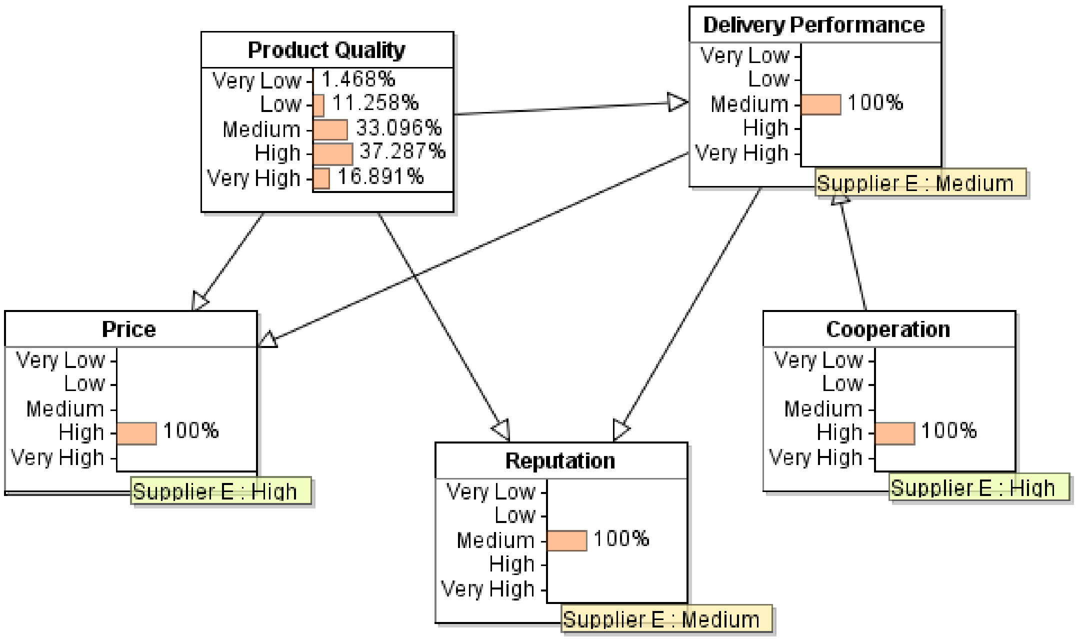


Figure 4.12: Example BN with Ranked Nodes for Supplier E

7. *Estimate the missing values in the decision matrix*: BNs submit their knowledge about the alternative suppliers as provided in Table 4.9.

   If there is missing knowledge, BN estimates the missing values. In this case study, to validate this process we purposely deleted some of these knowledge values randomly and make BN estimate these missing values. For example, the elicitation of probability of product quality for supplier E with BN by AgenaRisk software is presented in Figure 4.13. The full estimation of BN with incomplete knowledge is presented Table 4.10. More explanation on this issue will be discussed in the next section.

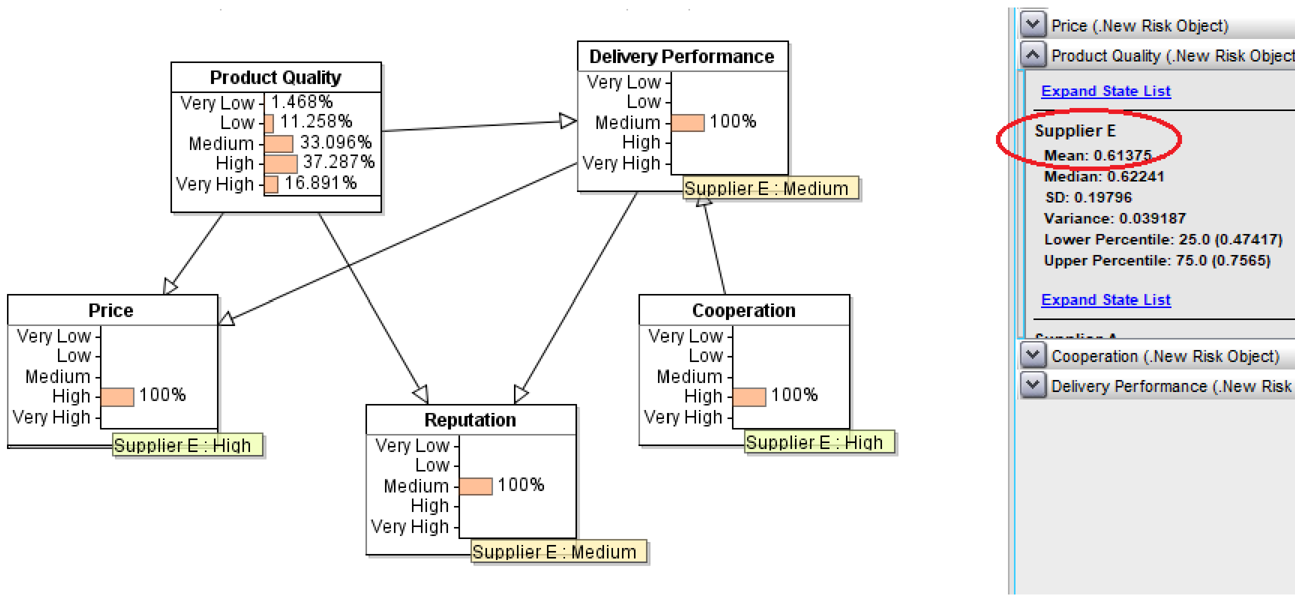


Figure 4.13: Estimation of Probability of Product Quality for Supplier E

Alternatives	Product Quality	Delivery Performance	Price	Cooperation	Reputation
Supplier A	0.90	0.90	0.90	0.70	0.90
Supplier B	0.90	0.70	0.90	0.55	0.90
Supplier C	0.90	0.70	0.90	0.50	0.90
Supplier D	0.70	0.50	0.70	0.44	0.61
Supplier E	0.61	0.50	0.70	0.50	0.50
Supplier F	0.50	0.50	0.70	0.50	0.50
Supplier G	0.50	0.70	0.70	0.66	0.30
Supplier H	0.30	0.30	0.30	0.70	0.30

Table 4.10: Probabilities of Criteria elicited from BN

8. *Rank alternatives with TOPSIS*: The weights of the criteria are obtained by AHP and the decision matrix is given by BN as inputs for TOPSIS which finally ranks the alternative suppliers. The results are given in Table 4.11.

Alternatives	TOPSIS Score
Supplier A	0.63
Supplier B	0.57
Supplier C	0.57
Supplier G	0.46
Supplier D	0.46
Supplier E	0.41
Supplier H	0.37
Supplier F	0.33

Table 4.11: Ranking of Suppliers

4.5 Scenario Analysis for Knowledge Value

To assess the effects of the knowledge value a sensitivity analysis is conducted by using various scenarios with different levels of missing values.

4.5.1 Performance Measures

We assess the robustness of the approach using the following two evaluation measures:

a) Total number of mismatch (T_m)

Let n refer to the number of suppliers,

$\tilde{P}_i$ refers to the original or default position of the i^{th} supplier(i.e., no missing value); $i = 1, \ldots, n$

P_i the position of the i^{th} supplier in the new list when there are some missing values; $i = 1, \ldots, n$

$$T_m = \sum_{i=1}^{n} m_i$$

$$where \quad m_i = \begin{cases} 1, & \text{if } P_i \neq \tilde{P}_i \quad i = 1, \ldots, n \\ 0, & \text{otherwise} \end{cases} \tag{4.8}$$

b)Total distance of mismatch D_m

This refers to the sum of mismatch for each supplier which is defined as

$$D_m = \sum_{i=1}^{n} d(P_i, \tilde{P}_i) = \sum_{i=1}^{n} \mid P_i - \tilde{P}_i \mid$$

with $\mid P_i - \tilde{P}_i \mid$ is the distance between P_i and $\tilde{P}_i$; $i = 1, \ldots, n$.

Illustrative Example

Firstly, we ranked the alternatives with complete knowledge and obtained the list in Table 4.12. When we compare with the list in Table 4.11 which has missing values, 3 mismatch items are generated which include Supplier E, Supplier G, Supplier D. Their total position distance of mismatch items is 4 as E mismatched by 2, G by 1 and D by 1.

Alternatives	TOPSIS Score
Supplier A	0.63
Supplier B	0.57
Supplier C	0.57
Supplier E	0.46
Supplier G	0.46
Supplier D	0.46
Supplier H	0.37
Supplier F	0.33

Table 4.12: Ranking of Suppliers with Complete Knowledge

4.5.2 Statistical Analysis

We replicated the approach 10 times for these 8 supplier and 5 criteria while varying the number of missing value from 5% to 20% with an increment of 5. Figure 4.14 shows the total number of mismatch whereas Figure 4.15 displays the total position distance of mismatch as the number of missing value increases. According to both graphs, as one may expect, the total distance and the number of mismatch increase with the number of missing values. The marginal change appears to be relatively higher when the % of missing values(m_i) is low (ie 5%, 10%) and they slowly stabilise. In other words, we have for $m_i = 5\%$, $T_m = 2.4$ & $D_m = 3.2$, these values increase for $m_i = 10\%$ to $T_m = 3.6$ & $D_m = 4.8$, and then their relative increase slow down afterward.

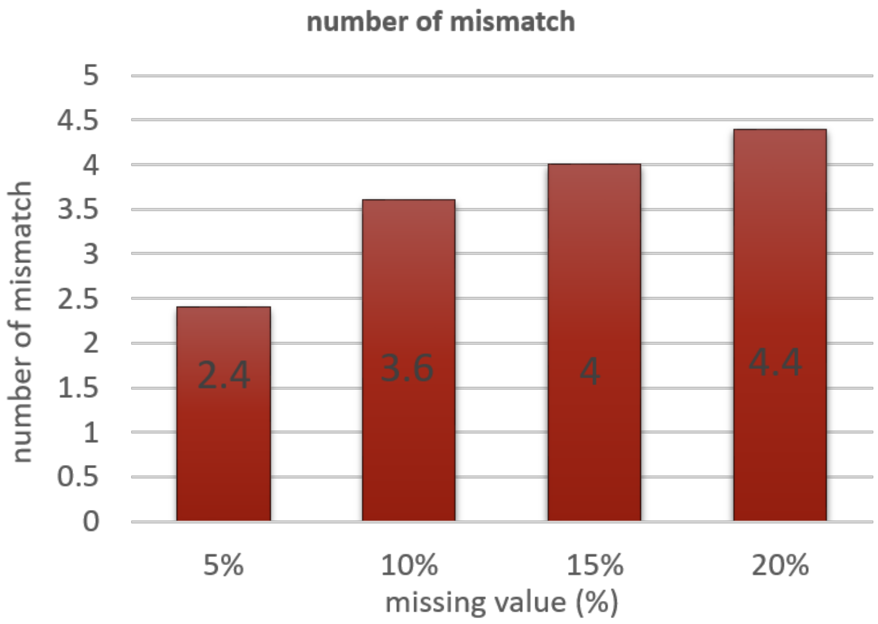


Figure 4.14: Effect of mismatch on % of missing values

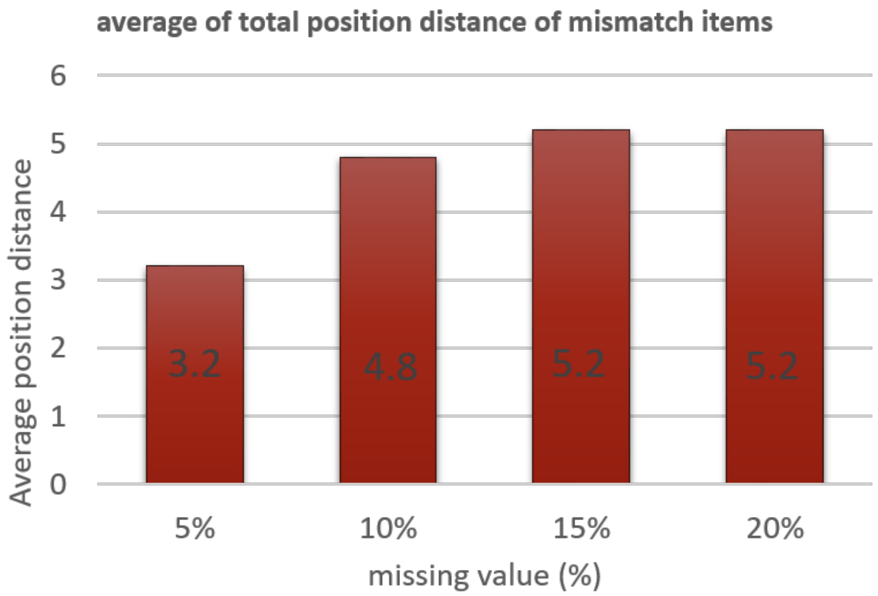


Figure 4.15: Effect of average position on % of missing values

4.6 Summary

In this study, we present an approach which integrates MCDM and BNs for the supplier selection problem in case of incomplete knowledge. This is an important facet within MCDM as one may not always have the chance to acquire all the information required. Our approach uses TOPSIS to rank the alternative suppliers with the weights of the selection criteria for TOPSIS obtained by AHP. The initial evaluation matrix for TOPSIS is estimated by BN. DEMATEL is used for determining the causal graphical structure and parameterization of the BN. We developed two performance criteria of mismatch, namely, the distance as well as the number of mismatches. A sensitivity analysis using several levels of missing values, ranging from 5% to 20% with an increment of 5%, is conducted. Interesting results show that our approach is robust as the degree of mismatch does not deteriorate significantly with the increase in the number of missing values.

The next chapter treats the order consolidation scheduling by first provide the introduction of the problem and the necessary items that are required to conduct the research.

Chapter 5

Time-Critical Scheduling: Problem Definition and the Construction of all configurations for consolidation of pairs and triplets

5.1 Introduction

In this chapter, the order consolidation scheduling problem when time is critical is investigated. We consider that the suppliers, here named the freight or truck companies, are already identified either by the company or as by-product of the results of the earlier chapters. The aim here is to optimise this kind of scheduling problem. We first introduce the problem, then provide the necessary notation, followed by the identification of all the configurations. We first present the case of pair consolidation and extend the methodology to triplets where we construct all the necessary configurations and define their respective costs. A small example is also given to illustrate the construction of these consolidations.

5.2 Problem Definition

In this study, we specifically work on time-critical freight logistics where the orders have to be met by third party logistics companies in a short time. Third party logistics companies (3PL) receive urgent orders and they have to offer a competitive quote in a short time around 10-15 minutes. The company has to offer a quote which will make the customer choose this company but at the same time yields the company profit. An illustration of such a typical problem is given below in Figure 5.1.

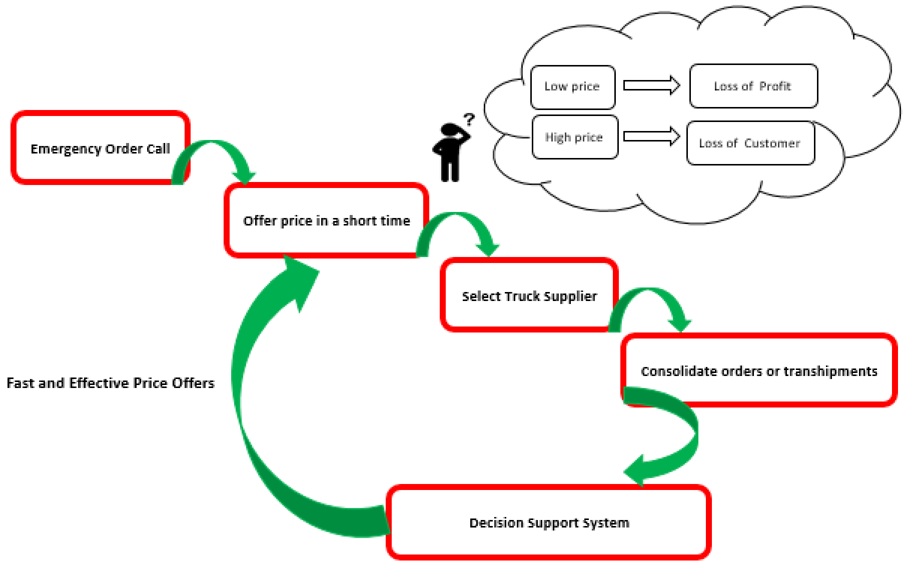


Figure 5.1: Illustration of a typical problem definition

To be able to offer a competitive quote, scheduling of the orders must be done effectively using freight companies known as truck suppliers. All available orders have to be considered and scheduled so to minimize the total distance while maintaining the high level of service level. Another critical point is the response time of the quote offer which has to be rather quick. Therefore, we offer to consolidate orders and determine suitable transshipment points and perform consolidation of orders with and without transshipment in an effective and fast way. Note that this 3PL company can identify all their truck companies before hand using their experience or the results of our earlier chapters. In other words, we assume that these truck companies are available to respond to the call of the company. In the first stage of our study, we consider pair consolidation of orders as it is easier to implement in practice while requiring relatively less computation time. This limitation to pairs only was initially required by the company as they usually opt for direct shipments instead to maintain their high level of quality of service. This will be then extended to cater for triplets. For the sake of completeness we consider that the chosen suppliers are at negligible distance to the collection points.

5.3 Case of Pair Consolidation

In this case, we propose to consolidate the orders as pairs with different consolidation configurations to minimize the cost and maximize the service level. The latter factor is defined by the quality of service that is provided as well as the speed by which the quote offer is given. In other words, we aim to determine the pairs to be consolidated and the best consolidation configuration for the pairs which give the maximum saving. We first compute the savings for all consolidation configurations of each possible pair of shipments, as recently shown by Salhi et al. (2020). We perform a similar task by extending to the case of triplets. This is, to the best of our knowledge, the first time this is formally explained and formulated. Once these configurations and their savings are determined, these will be used as input in our mathematical formulations which will be developed in the next chapter. The consolidation configurations and the corresponding costs and savings from the consolidations among the non-consolidation case are given below after the necessary notations are defined.

5.3.1 Notation

n : number of shipments, indexed by $i(i = 1, .., n)$
C_i : the collection location of shipment $i(i = 1, .., n)$
D_i : the delivery location of shipment $i(i = 1, .., n)$
$V = (C_i, D_i); i = 1, .., n$: the set of shipments with $|V| = n$ and each shipment i being defined by its collection and delivery locations C_i and D_i; $i = 1, ..., n$ respectively
$\Re$: the set of regions, indexed by $R_r; R_r \in \Re; r = 1, ..., |\Re|$
K: the set of potential configurations of serving any two shipments ($K = \{0, 1, 2, ..., 8\}$), indexed by $v \in K$ with $v = 0$ referring to the original configuration (no consolidation), $v \in \{1, ..., 4\}$ for en-route consolidation and $v \in \{5, ..., 8\}$ for transshipment consolidation.
$d_{L_iL_j}$: the distance(cost) between locations L_i and L_j where a location refers to origin and destination of shipments $i, j(i, j = 1, ..., n)$
S_{ij} : the cost saving over the original configuration when using the best consolidation configuration of shipment i with shipment $j(i, j = 1, ..., n)$.
π^v_{ij} : the cost of serving shipments i and j using configuration $v \in K$
Ω^l_{rs}: set of potential transshipment points throughout the business area $\Re$, indexed by transshipment point $T_k; T_k \in \Omega^l_{rs}(k = 1, ..., |\Omega^l_{rs}|)$ that can be used between shipments with collection (delivery) points in regions R_r and R_s and the delivery (collection) point in region $R_l; R_r, R_s, R_l \in \Re$

$\gamma_{(T_{k*})}$: This is the nearest transshipment point to R_r and R_s, which is determined based on the base location of each region as a reference point, say $(\bar{R}_r, \bar{R}_s)$, $\gamma_{(T_{k*})} = Arg \min_{T_k \in \Omega^l_{rs}} (d_{\bar{R}_r T_k} + d_{\bar{R}_s T_k})$

Γ^l_{rs} : the subset of transshipment points which are the nearest transshipment points to R_r and R_s ($\Gamma^l_{rs} \subseteq \Omega^l_{rs}$) that can be used between collection (delivery) points in regions R_r and R_s to go to (come from) region $R_l; R_r, R_s, R_l \in \Re$

$F_{\gamma_{(T_{k*})}}$: the financial cost incurred to consolidate the two shipments at the transshipment point $\gamma_{(T_{k*})}$ (i.e., the additional fixed cost).

Figure 5.2: Notations for Pair Consolidation

5.3.2 Consolidation Configurations and Computation of Consolidation Costs

In case of shipment of two orders, the third party logistics companies mostly prefer to use two TL(truckload) for each shipment to make sure that the customer needs are met urgently. If we assume two shipments i and j whose collection points are C_i and C_j with their delivery destinations being D_i and D_j respectively, using two different truck for each shipment

$$C_i \rightarrow D_i \text{ and } C_i \rightarrow D_j$$

Cost of original configuration which is non-consolidation is:

$$\pi^0_{ij} = d_{C_i D_i} + d_{C_j D_j}$$

However, instead of following this tradition, we offer to consolidate orders with and without transshipment to minimize the cost and maximize the service level.

(a) En-route Order Consolidation Configurations

In en-route order consolidation case, we offer to use only one truck for collection and delivery of two orders. The truck firstly will pickup the orders from the collection points and after that deliver them to the delivery locations. There are four possible en-route consolidation configurations for shipment of two orders which are given in Figure 5.3.

$$C_i \longrightarrow C_j \longrightarrow D_i \longrightarrow D_j \qquad (1)$$

$$C_i \longrightarrow C_j \longrightarrow D_j \longrightarrow D_i \qquad (2)$$

$$C_j \longrightarrow C_i \longrightarrow D_j \longrightarrow D_i \qquad (3)$$

$$C_j \longrightarrow C_i \longrightarrow D_i \longrightarrow D_j \qquad (4)$$

Figure 5.3: En-route consolidation configurations

Costs of en-route consolidation configurations which are given below respectively.

$$\pi_{ij}^1 = d_{C_iC_j} + d_{C_jD_i} + d_{D_iD_j}$$

$$\pi_{ij}^2 = d_{C_iC_j} + d_{C_jD_j} + d_{D_jD_i}$$

$$\pi_{ij}^3 = d_{C_jC_i} + d_{C_iD_j} + d_{D_jD_i}$$

$$\pi_{ij}^4 = d_{C_jC_i} + d_{C_iD_i} + d_{D_iD_j}$$

(b) Transshipment Consolidation Configurations

In case collection and delivery points are close, en-route consolidation configurations are effective. However, in case there is a large distance between either collection or delivery points, using only one truck cannot be an effective option. In this case, we offer to use transshipment point for merging either collections or deliveries.

Case 1: Merging Deliveries

If delivery points of the orders are close to each other and collection points are far from each other, collection of the orders with two different trucks and merging the collections in one of the trucks at a transshipment point and going through the delivery points with one truck can be more cost productive consolidation configuration. Possible merging consolidation configurations are given in Figure 5.4.

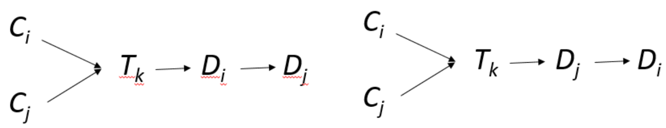


Figure 5.4: Merging delivery configurations (5) left, (6) right

The cost for merging deliveries, as shown by the configurations in Figure 5.4, depending on (5) and (6) are given below respectively.

$$\pi_{ij}^{5} = \min_{\gamma(T_k) \in \Gamma_{rs}^{r'}} ([d_{C_i\gamma(T_k)} + d_{C_j\gamma(T_k)} + F_{\gamma(T_k)}] + [d_{\gamma(T_k)D_i} + d_{D_iD_j}];$$
$$\{C_i \in R_r, C_j \in R_s, D_i \in R_{r'}, D_j \in R_{r'}\}$$

$$\pi_{ij}^{6} = \min_{\gamma(T_k) \in \Gamma_{rs}^{r'}} ([d_{C_i\gamma(T_k)} + d_{C_j\gamma(T_k)} + F_{\gamma(T_k)}] + [d_{\gamma(T_k)D_j} + d_{D_jD_i}];$$
$$\{C_i \in R_r, C_j \in R_s, D_i \in R_{r'}, D_j \in R_{r'}\ \}$$

If the used trucks of both shipments are not determined, then a vehicle which has enough capacity to carry both shipments is chosen for one of the shipments. If it is possible, this large truck is preferred to take the shortest distance which means it starts from the collection point which is close to the transshipment point. After merging the collections in this large size vehicle at the transshipment point, this vehicle delivers both orders after the transshipment point.

Case 2: Merging Collections

This case is the opposite of the merging delivery case. Here, the collection points are close to each other but delivery points are far from each other. One large truck can collect both orders and at a transshipment point transfer one of the orders to another truck. After the transshipment point, the largest truck takes the shortest distance among the distances between each delivery points and transshipment point.This is performed to reduce transportation cost as the largest vehicle is likely to consume more fuel.

The merging collection consolidation configurations are given in Figure 5.5.

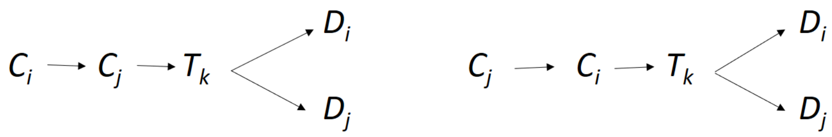


Figure 5.5: Merging collection configurations (7) left, (8) right

The cost for merging deliveries with the configurations given in Figure 5.5 is based on (7) and (8) and are defined below respectively.

$$\pi_{ij}^{7} = \min_{\gamma(T_k)\in\Gamma_{rs}^{r'}} ([d_{\gamma(T_k)D_i} + d_{\gamma(T_k)D_j)} + F_{\gamma(T_k)}] + [d_{C_jC_i} + d_{C_i\gamma(T_k)}]$$
$$\{C_i \in R_{r'}, C_j \in R_{r'}, D_i \in R_r, D_j \in R_s\}$$

$$\pi_{ij}^{8} = \min_{\gamma(T_k)\in\Gamma_{rs}^{r'}} ([d_{\gamma(T_k)D_i} + d_{\gamma(T_k)D_j} + F_{\gamma(T_k)}] + [d_{C_iC_j} + d_{C_j\gamma(T_k)}]$$
$$\{C_i \in R_{r'}, C_j \in R_{r'}, D_i \in R_r, D_j \in R_s\}$$

Note that to eliminate those configurations that are unlikely to lead to a promising outcome, neighbourhood schemes are also introduced which will be developed in the next chapter as we aim here to provide the general framework.

Computation of the cost saving

For calculation of the cost saving, firstly we define the least cost of consolidation among all possible consolidation configurations including non-consolidation ones. The least cost of consolidating two orders, namely, i and j is computed as follows:

$$\hat{C}_{ij}^{1} = \begin{cases} \min\limits_{v=1,\dots,8} \pi_{ij}^{v} & \text{if shipments i and j are feasible to consolidate} \\ \pi_{ij}^{0} & \text{otherwise} \end{cases} \tag{5.1}$$

The best consolidation configuration is determined in the following:

$$v^* = Arg \min_{v\in\{1,\dots,8\}} \pi_{ij}^{v}$$

To find the best consolidation configuration among non-consolidation, en-route consolidation and transshipment consolidation configurations, the following saving formulation is used.

$$S_{ij} = \pi_{ij}^{0} - \hat{C}_{ij}^{1}; \qquad i, j = 1, .., n$$

The total saving is then computed as

$$TS = \sum_{(i,j) \in E^p} S_{ij}$$

,

with E^p representing those pairs of shipments that are chosen in the final solution configuration.

It is important to stress that these calculations are performed from the outset and once only. This extra effort is worthwhile as it provides competitive advantage among the classical mathematical formulations in the literature which are replicated each and every run inside the formulation.

5.3.3 Data Generation and Illustrative Example

We generated 10 data sets of each sample sizes of 50, 100, 150 and 200 to run tests with the proposed mathematical model.

Firstly, we generated the coordinates of the collection and delivery locations of the orders and transshipment points. These coordinates are generated randomly in range of 0 and 100 as decimal numbers. Sample data set for 10 orders is presented in Table 5.1.

Request No	Pickup location		Delivery Location	
	(X	Y)	(X	Y)
0	35.3099	6.10065	94.3724	12.5462
1	4.34889	54.1429	86.523	22.0008
2	34.0617	43.3027	26.5267	52.3637
3	73.278	90.4233	23.8899	45.1003
4	37.6751	78.1945	91.3144	29.7128
5	85.876	39.4086	54.4969	6.90023
6	32.4229	30.4575	94.1252	35.0444
7	50.856	34.7118	98.2269	85.6532
8	69.8813	86.401	53.0107	83.3399
9	15.8757	55.6505	50.3037	96.0479

Table 5.1: Sample Data Set Size of 10

Then savings were computed for all possible pairs of orders based on consolidation configurations; en-route consolidation configurations and transshipment consolidation configurations. Then maximum savings for each pair of orders and the corresponding consolidation configuration and the corresponding transshipment point if the consolidation configuration is the consolidation configuration with transshipment were determined. All these algorithms are programmed in Visual Studio C++ and executed on an Intel Core i5-7300U CPU @ 2.60GHz PC with 64-bit operating system and 8 GB RAM.

Savings of sample data set is given in Table 5.2. Configurations which give the max saving for the sample data set is given in Table 5.3.

No	0	1	2	3	4	5	6	7	8	9
0	0	24.58	-56.48	-66.32	-13.44	-15.43	15.37	-25.63	-110	-88.23
1	24.58	0	-10.63	-10.91	38.06	-2.26	43.52	4.81	-54.70	-5.82
2	-56.48	-10.63	0	-0.54	-31.41	-60.16	-20.86	-28.35	-79.65	-18
3	-66.32	-10.91	-0.54	0	-3.32	-34.46	-31.55	-33.54	13.70	-17.64
4	-13.44	38.06	-31.41	-3.32	0	1.19	21.23	-0.70	-25.86	-5.84
5	-15.43	-2.26	-60.16	-34.46	1.19	0	-5.08	-38.72	-80.91	-107.41
6	15.37	43.52	-20.86	-31.55	21.23	-5.08	0	18.47	-68.88	-43.38
7	-25.63	4.81	-28.35	-33.54	-0.70	-38.72	18.47	0	-30.79	-20.24
8	-110	-54.70	-79.65	13.70	-25.86	-80.91	-68.88	-30.79	0	-22.06
9	-88.23	-5.82	-18.00	-17.64	-5.84	-107.41	-43.38	-20.24	-22.06	0

Table 5.2: Savings of Sample Data Set

No	0	1	2	3	4	5	6	7	8	9
0	0	3	2	8	6	3	5	1	2	2
1	3	0	2	3	2	1	1	1	2	2
2	2	2	0	3	4	4	4	4	4	1
3	8	3	3	0	1	2	1	3	2	1
4	6	2	4	1	0	1	1	1	3	3
5	3	1	4	2	1	0	3	1	2	5
6	5	1	4	1	1	3	0	1	2	2
7	1	1	4	3	1	1	1	0	2	2
8	2	2	4	2	3	2	2	2	0	4
9	2	2	1	1	3	5	2	2	4	0

Table 5.3: Max Saving Configurations of Sample Data Set

No	0	1	2	3	4	5	6	7	8	9
0	0	0	0	2	2	0	2	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	2	0	0	0	0	0	0	0	0	0
4	2	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	2
6	2	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	2	0	0	0	0

Table 5.4: Transshipment Points of max Saving of the small Data Set of 10 shipments

We also propose to extend the consolidation of orders pair to triplet which is discussed in the next section. This will be followed by the set partitioning formulation.

5.4 Case of Triplet Consolidation

In case of triplet consolidation, we consolidate the orders as triplets with and without transshipment consolidation configurations to save cost and time among non-consolidation configuration.

In case of no consolidation (original configuration) this is given as follows:

$$C_i \to D_i \;\&\; C_j \to D_j \;\&\; C_k \to D_k$$

The cost of this original configuration is

$$C^0_{ijk} = d_{C_i D_i} + d_{C_j D_j} + d_{C_k D_k}$$

5.4.1 (a) Consolidation of 3 requests en-route(no transshipment)

There are 6 cases to explore.

Case (a-1) First possible en-route triplet consolidation is given in Figure 5.6.

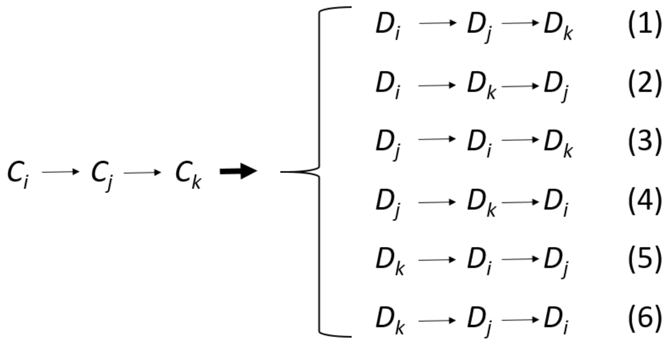


Figure 5.6: En-route consolidation for triplet case (a-1)

Cost of (1)

$$\delta_1 = d_{D_i D_j} + d_{D_j D_k}$$

Cost of (2)

$$\delta_2 = d_{D_i D_k} + d_{D_k D_j}$$

Cost of (3)

$$\delta_3 = d_{D_j D_i} + d_{D_i D_k}$$

Cost of (4)

$$\delta_4 = d_{D_j D_k} + d_{D_k D_i}$$

Cost of (5)

$$\delta_5 = d_{D_k D_i} + d_{D_i D_j}$$

Cost of (6)

$$\delta_6 = d_{D_k D_j} + d_{D_j D_i}$$

Note (δ_1,..,δ_6) will be used later in other scores given below

Compute all the 6 costs for (a-1). We will do the same for all cases of (a).

$$\Delta^1_{C_i C_j C_k} = d_{C_i C_j} + d_{C_j C_k} + d_{C_k D_i} + \delta_1$$

as $\delta_0^{'} = d_{C_i C_j} + d_{C_j C_k}$ will be used as well

$$\Delta^2_{C_i C_j C_k} = \delta_0^{'} + \delta_2 + d_{C_k D_i}$$

$$\Delta^3_{C_i C_j C_k} = \delta_0^{'} + \delta_3 + d_{C_k D_j}$$

$$\Delta^4_{C_i C_j C_k} = \delta_0^{'} + \delta_4 + d_{C_k D_j}$$

$$\Delta^5_{C_iC_jC_k} = \delta_0' + \delta_5 + d_{C_kD_k}$$

$$\Delta^6_{C_iC_jC_k} = \delta_0' + \delta_6 + d_{C_kD_k}$$

Least cost for (a-1):

$$T^{(1)}_{ijk} = \min_{r=1,\ldots,6}(\Delta^r_{C_iC_jC_k})$$

Case (a-2)

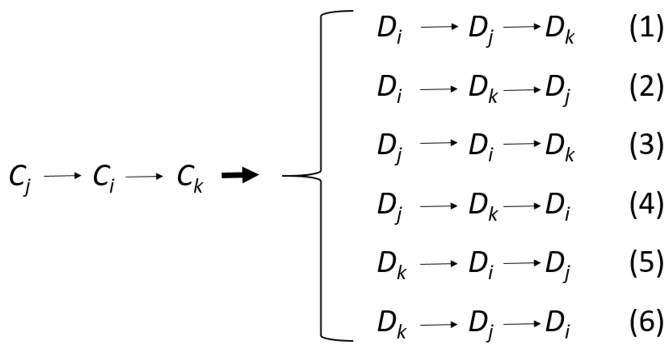


Figure 5.7: En-route consolidation for triplet case (a-2)

Compute all costs for (a-2)

$$\Delta^1_{C_jC_iC_k} = d_{C_jC_i} + d_{C_iC_k} + d_{C_kD_i} + \delta_1$$

$$\Delta^1_{C_jC_iC_k} = \delta_0^2 + \delta_1 + d_{C_kD_i}$$

(as $\delta_0^2 = d_{C_jC_i} + d_{C_iC_k}$)

$$\Delta^2_{C_jC_iC_k} = d_{C_kD_i} + \delta_2 + \delta_0^2$$

$$\Delta^3_{C_jC_iC_k} = d_{C_kD_j} + \delta_3 + \delta_0^2$$

$$\Delta^4_{C_jC_iC_k} = d_{C_kD_j} + \delta_4 + \delta_0^2$$

$$\Delta^5_{C_jC_iC_k} = d_{C_kD_k} + \delta_5 + \delta_0^2$$

$$\Delta^6_{C_jC_iC_k} = d_{C_kD_k} + \delta_6 + \delta_0^2$$

$$T^2_{ijk} = \min_{r=1,\ldots,6}(\Delta^r_{C_jC_iC_k})$$

Same calculations are done for cases of (a-3), (a-4), (a-5), (a-6) in Appendix B and $T^3_{ijk}, T^4_{ijk}, T^5_{ijk}$ and T^6_{ijk} are calculated.

The best en-route cost consolidation configuration ijk is therefore,

$$C^1_{ijk} = \min_{s=1,\ldots,6}(T^s_{ijk})$$

After consolidation of triplets as en-route, we consider the consolidation of triplets with the transshipment points.

5.4.2 (b) Consolidation of Triplets(ijk) with only one transshipment point(T)

Let $\Lambda_{P_1P_2P_3} = \{T \in \Omega$ s.t T is in neighborhood of points $P_1, P_2, P_3\} \subset \Omega$ with Ω set of all transshipments . Note that there are two cases, namely, the first dealing with having transshipment points before delivery and the other after delivery instead.

(b-1) Case of transshipment points before delivery

In this case, we consider the transshipment points before delivery points. We classified the configurations in this case under two sub-cases which we refer to as subsections b-1-1 and b-1-2.

Subcase (b-1-1) In this case, two of the orders collected and then combined at the transshipment point then collect the third order then go towards the delivery points of the orders. Possible configurations for this case are given below as b-1-1-1, b-1-1-2 and b-1-1-3.

<u>b-1-1-1)</u>

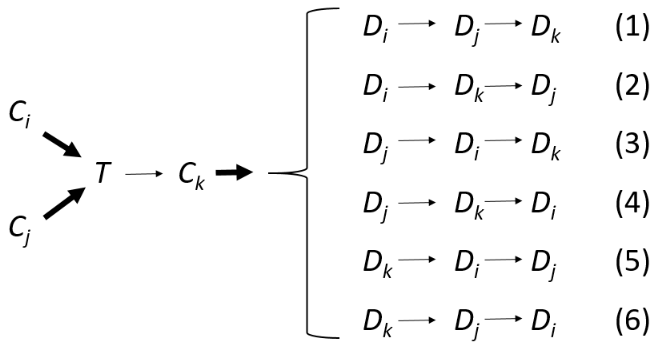


Figure 5.8: Triplet case with transshipment of (b-1-1)

$$\Delta^1_{C_iTC_jTC_k} = \min_{T\in\Lambda_{C_iC_jC_k}} (d_{C_iT} + d_{C_jT} + d_{TC_k}) + \delta_1 + d_{C_kD_i}$$

$$\Delta^1_{C_iTC_jTC_k} = \delta_0^{1'} + \delta_1 + d_{C_kD_i}$$

as $\delta_0^{1'} = \min_{T\in\Lambda_{C_iC_jC_k}} (d_{C_iT} + d_{C_jT} + d_{TC_k})$
Note $\delta_1, ..., \delta_6$ are unchanged.

$$T^{1'}_{ijk} = \min_{r=1,...,6} (\Delta^r_{C_iTC_jTC_k})$$

b-1-1-2)

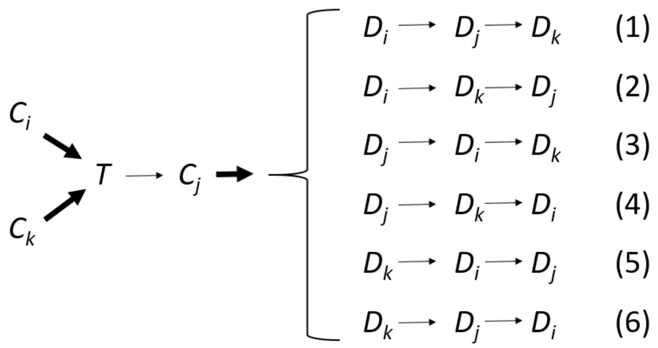


Figure 5.9: Triplet case with transshipment of (b-1-2)

$$\Delta^1_{C_iTC_jTC_j} = \delta_0^{1'} + \delta_1 + d_{C_jD_i}$$

$$T^{2'}_{ijk} = \min_{r=1,\ldots,6}(\Delta^r_{C_iTC_jTC_j})$$

<u>b-1-1-3)</u>

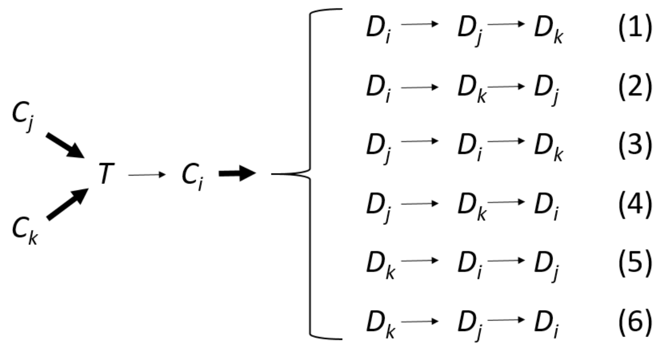


Figure 5.10: Triplet case with transshipment of (b-1-3)

$$\Delta^{1'}_{C_jTC_iTC_k} = \delta^{1'}_0 + \delta_1 + d_{C_iD_i}$$

$$T^{3'}_{ijk} = \min_{r=1,\ldots,6}(\Delta^r_{C_jTC_iTC_k})$$

$$C^2_{ijk} = \min_{s=1,\ldots,3}(T^{s'}_{ijk})$$

Subcase (b-1-2) In this case, all orders collected, two of them by one truck and the third one by another truck and all collections are consolidated at the transshipment point as off towards to delivery points.

b-1-2-1)

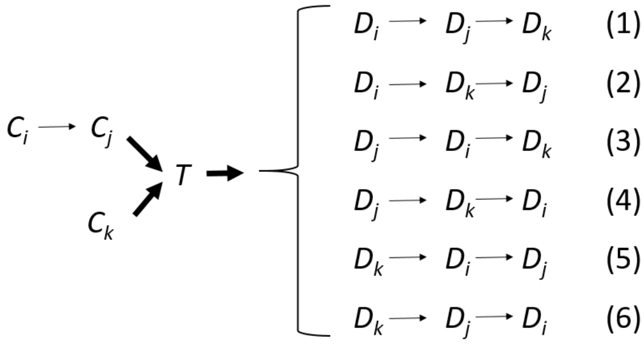


Figure 5.11: Triplet case with transshipment of (b-2-1)

$$\Delta^1_{C_iC_jTC_k} = d_{C_iC_j} + \min_{T\in\Delta_{C_jC_kD_i}} (d_{C_jT} + d_{C_kT} + d_{D_iT}) + \delta_1 = d_{C_iC_j} + \delta_0^{2'} + \delta_1$$

$$\Delta^2_{C_iC_jTC_k} = d_{C_iC_j} + \min_{T\in\Delta_{C_jC_kD_i}} (d_{C_jT} + d_{C_kT} + d_{D_iT}) + \delta_2 = d_{C_iC_j} + \delta_0^{2'} + \delta_2$$

$$\Delta^3_{C_iC_jTC_k} = d_{C_iC_j} + \min_{T\in\Delta_{C_jC_kD_j}} (d_{C_jT} + d_{C_kT} + d_{D_jT}) + \delta_3 = d_{C_iC_j} + \delta_0^{3'} + \delta_3$$

$$\Delta^4_{C_iC_jTC_k} = d_{C_iC_j} + \min_{T\in\Delta_{C_jC_kD_j}} (d_{C_jT} + d_{C_kT} + d_{D_jT}) + \delta_4 = d_{C_iC_j} + \delta_0^{3'} + \delta_4$$

$$\Delta^5_{C_iC_jTC_k} = d_{C_iC_j} + \min_{T\in\Delta_{C_jC_kD_k}} (d_{C_jT} + d_{C_kT} + d_{D_kT}) + \delta_5 = d_{C_iC_j} + \delta_0^{4'} + \delta_5$$

$$\Delta^6_{C_iC_jTC_k} = d_{C_iC_j} + \min_{T\in\Delta_{C_jC_kD_k}} (d_{C_jT} + d_{C_kT} + d_{D_kT}) + \delta_6 = d_{C_iC_j} + \delta_0^{4'} + \delta_6$$

$$T^{1''}_{ijk} = \min_{r=1,\dots,6} (\Delta^r_{C_iC_jTC_k})$$

Note $\delta_1, .., \delta_6$are unchanged.
We do the same for the other 5 cases in Appendix B.
Least cost

$$T_{ijk}^{6''} = \min_{r=1,\ldots,6} (\Delta_{C_k C_j T C_i T^r})$$

$$C_{ijk}^3 = \min_{s=1,\ldots,6} (T_{ijk}^{s''})$$

(b-2) Case of transshipment after collection points

In the second case of triplet consolidation with transshipments, transshipment point is assigned after collection points in the delivery points region. This case has two subcases.

Subcase (b-2-1)

In this subcase, collections are collected and deliver one of the orders then head to transshipment points while transferring one of the deliveries to the other truck at the transshipment after that head to the delivery point of the third order.

b-2-1-1)

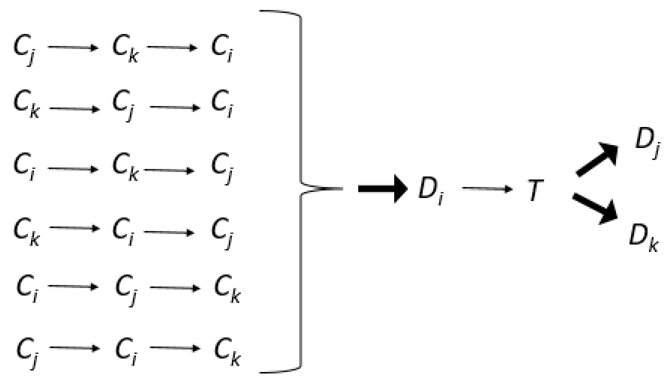


Figure 5.12: Triplet case with transshipment of (c-1-1)

Cost of (1)

$$\delta_1^{'} = d_{C_j C_k} + d_{C_k C_i}$$

Cost of (2)

$$\delta_2^{'} = d_{C_k C_j} + d_{C_j C_i}$$

Cost of (3)

$$\delta_3^{'} = d_{C_i C_k} + d_{C_k C_j}$$

Cost of (4)

$$\delta_4^{'} = d_{C_k C_i} + d_{C_i C_j}$$

Cost of (5)

$$\delta_5^{'} = d_{C_iC_j} + d_{C_jC_k}$$

Cost of (6)

$$\delta_6^{'} = d_{C_jC_i} + d_{C_iC_k}$$

Note ($\delta_1^{'}$,..,$\delta_6^{'}$) will be used later in other scores given below

$$\Delta^1_{D_iTD_jTD_k} = \min_{T\in\Lambda_{D_iD_jD_k}} (d_{D_iT} + d_{TD_j} + d_{TD_k}) + \delta_1 + d_{C_iD_i}$$

$$\Delta^1_{D_iTD_jTD_k} = \delta_0^{1^{''}} + \delta_1^{'} + d_{C_iD_i}$$

as $\delta_0^{1^{''}} = \min_{T\in\Lambda_{D_iD_jD_k}} (d_{D_iT} + d_{D_jT} + d_{TD_k})$

We computed all the 6 costs for (b-2-1-1). We will do the same for all cases of (b-2-1).

$$T_{ijk}^{1^{'''}} = \min_{r=1,\ldots,6} (\Delta_{D_iTD_jTD_k{}^r})$$

b-2-1-2)

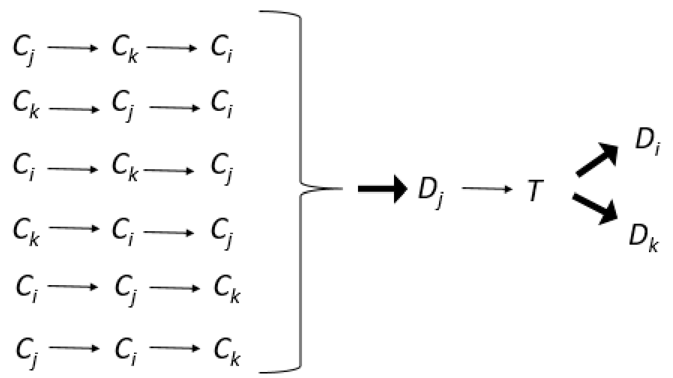


Figure 5.13: Triplet case with transshipment of (c-1-2)

$$\Delta^1_{D_jTD_jTD_k} = \delta_0^{1^{''}} + \delta_1^{'} + d_{C_iD_j}$$

$$T_{ijk}^{2^{'''}} = \min_{r=1,\ldots,6} (\Delta_{D_jTD_jTD_k{}^r})$$

b-2-1-3)

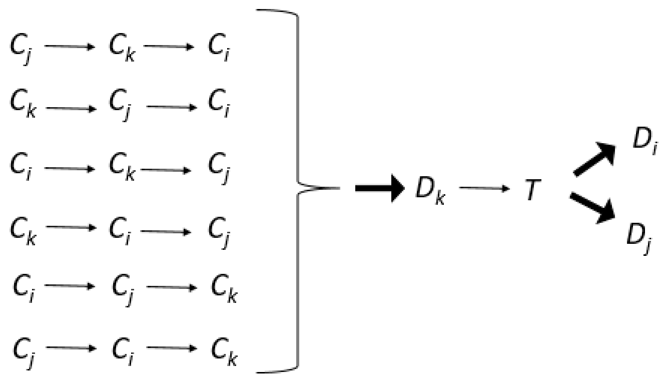


Figure 5.14: Triplet case with transshipment of (c-1-3)

$$\Delta^1_{D_kTD_iTD_j} = \delta_0^{1''} + \delta_1^{'} + d_{C_iD_k}$$

$$T^{3'''}_{ijk} = \min_{r=1,\ldots,6}(\Delta^r_{D_kTD_iTD_j})$$

$$C^4_{ijk} = \min_{s=1,\ldots,3}(T^{s'''}_{ijk})$$

Subcase (b-2-2) In this subcase, after collection of orders, truck head to the transshipment point and then one of the three orders is transferred to the another truck at this point and then head to deliver the two other orders.

b-2-2-1)

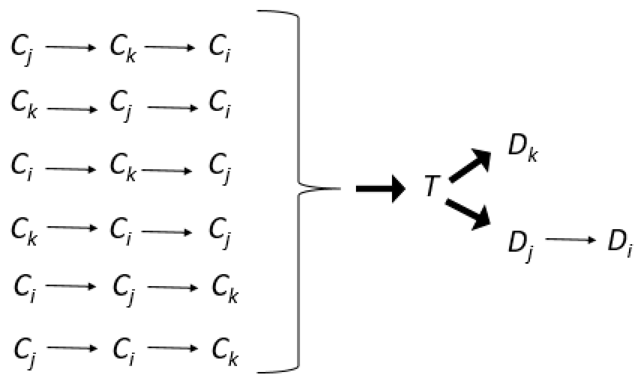


Figure 5.15: Triplet case with transshipment of (c-2-1)

$$\Delta^1_{TD_kD_jD_i} = d_{D_jD_i} + \min_{T\in\Delta_{C_iD_kD_j}} (d_{C_iT} + d_{TD_k} + d_{TD_j}) + \delta_1^{'} = d_{D_jD_i} + \delta_0^{2''} + \delta_1^{'}$$

$$\Delta^2_{TD_kD_jD_i} = d_{D_jD_i} + \min_{T\in\Delta_{C_iD_kD_j}} (d_{C_iT} + d_{TD_k} + d_{TD_j}) + \delta_2^{'} = d_{D_jD_i} + \delta_0^{2''} + \delta_2^{'}$$

$$\Delta^3_{TD_kD_jD_i} = d_{D_jD_i} + \min_{T\in\Delta_{C_jD_kD_j}} (d_{C_jT} + d_{TD_k} + d_{TD_j}) + \delta_3^{'} = d_{D_jD_i} + \delta_0^{3''} + \delta_3^{'}$$

$$\Delta^4_{TD_kD_jD_i} = d_{D_jD_i} + \min_{T\in\Delta_{C_jD_kD_j}} (d_{C_jT} + d_{TD_k} + d_{TD_j}) + \delta_4^{'} = d_{D_jD_i} + \delta_0^{3''} + \delta_4^{'}$$

$$\Delta^5_{TD_kD_jD_i} = d_{D_jD_i} + \min_{T\in\Delta_{C_kD_kD_j}} (d_{C_kT} + d_{TD_k} + d_{TD_j}) + \delta_5^{'} = d_{D_jD_i} + \delta_0^{4''} + \delta_5^{'}$$

$$\Delta^6_{TD_kD_jD_i} = d_{D_jD_i} + \min_{T\in\Delta_{C_kD_kD_j}} (d_{C_kT} + d_{TD_k} + d_{TD_j}) + \delta_6^{'} = d_{D_jD_i} + \delta_0^{4''} + \delta_6^{'}$$

$$T^{1''''}_{ijk} = \min_{r=1,\ldots,6} (\Delta_{TD_kD_jD_i{}^r})$$

The other 5 configurations are also computed and presented in Appendix B.

Least cost

$$T^{6''''}_{ijk} = \min_{r=1,\ldots,6} (\Delta^r_{TD_iD_jD_k})$$

$$C^5_{ijk} = \min_{s=1,\ldots,6} (T^{s''''}_{ijk})$$

5.4.3 Computation of the cost saving for the case of triplets

The least cost of consolidating three orders, namely, i, j and k is computed as follows:

$$\hat{C}^1_{ijk} = \begin{cases} \min_{s=0,\ldots,5} \mathrm{C}^s_{ijk} & \text{if shipments i and j are feasible to consolidate} \\ \mathrm{C}^0_{ijk} & \text{otherwise} \end{cases} \tag{5.2}$$

The best consolidation configuration is determined in the following:

$$s^* = Arg \min_{s \in \{0,...,5\}} C^s_{ijk}$$

To find the best consolidation configuration among non-consolidation, en-route consolidation and transshipment consolidation configurations, the following saving formulation is used.

$$S_{ijk} = C^0_{ijk} - \hat{C}^1_{ijk}; \qquad i, j, k = 1, .., n$$

The total saving is then computed as

$$TS = \sum_{(i,j,k) \in E^t} S_{ijk}$$

,

with E^t representing those triplets of shipments that are chosen in the final solution configuration.

As stressed in pair consolidation case, all these calculations are performed once before running the mathematical model, while the mathematical models in the literature carry out all calculations inset and replicated each and every run.

5.5 Summary

In this chapter we extend on the earlier study by Salhi et al. (2020) on constructing configurations for consolidation for pairs to the case of triplets. We then defined their corresponding costs and associated savings.

The next chapter will cover the mathematical formulation on how to solve this scheduling problem optimally and produce some computational results accordingly.

Chapter 6

Mathematical Models and a Computational Analysis

6.1 Introduction

In this chapter, we formulate the problem using two mathematical approaches, namely, a standard Integer 0-1 Linear Programming (ILP) and a set partitioning based(SPB). The set partitioning formulation is extended to the case of triplets where all configurations including singleton, pairs and triplets are first defined. To tighten the formulation, three types of tightening using appropriate valid inequalities are developed. Computational experiments using instances varying in size from 50 to 200 shipments with an increment of 50 are carried out and the results are analysed. A computational comparison between the classical 0-1 ILP and the SPB is first produced for the case of pairs using ILOG CPLEX. Ten random instances for each size (n=50, 100,150 and 200) are considered and the average results in terms of CPU are recorded and discussed.

6.2 Initial Mathematical Formulation

We first start by providing an existing 0-1 formulation as proposed by (Salhi et al 2020) for the case of pairs and then produce a simple but powerful alternative formulation based on the set partitioning problem.

To find the pairs of orders that can be consolidated for maximizing the total cost saving, a mathematical formulation based on 0-1 integer linear program is proposed. More details can be found in Salhi et al. 2020. Recall from the previous chapter that S_{ij} is the saving by consolidating shipments i with j. The decision variable X_{ij} of whether or not to consolidate shipments

i with j is defined as follows.

$$X_{ij} = \begin{cases} 1 & \text{if shipments i and j are consolidated} \\ 0 & \text{otherwise} \end{cases} \tag{6.1}$$

$$\text{Maximise } TS = \sum_{j=1}^{n} \sum_{i=1}^{n} S_{ij} X_{ij} \tag{6.2}$$

subject to

$$\sum_{j=1}^{n} X_{ij} = 1; \quad i = 1, ..., n \tag{6.3}$$

$$\sum_{i=1}^{n} X_{ij} = 1; \quad j = 1, ..., n \tag{6.4}$$

$$X_{ij} = X_{ji}; \quad i, j = 1, ..., n \tag{6.5}$$

$$X_{ij} \in \{0, 1\}; \quad i, j = 1, ..., n \tag{6.6}$$

The objective function (6.2) is to find the maximum total saving among consolidation configurations. Constraint (6.3) enforces that each shipment i can consolidate with another shipment including itself only and (6.4) shows that each shipment j can be consolidated with another shipment only. Constraint (6.5) refers that an assigned order cannot be assigned to another order as a pair. Constraint (6.6) refers to the binary decision variables.

Note that the TS value obtained from this formulation needs to be halved to the assignment problem structure of the formulation so to avoid double counting.

This mathematical model has $n^2 + 2n$ constraints and n^2 binary variables. This can be solved using standard commercial optimization software such as GAMS, LINDO, ILOG CPLEX, among others. For convenience and availability, we will be using the latter one in this study.

An illustrative example

We ran the above proposed mathematical model for the illustrative example given in Section 5.3.3. The savings S_{ij} are already computed in Section 5.3.3 and their values are given in Table 5.2. The above model is run with these computed savings and the total maximum saving as well as the consolidated pairs which give the optimal result were obtained. The total saving was found as 70.2337 and the optimal consolidated pairs are given in Table

6.1 with their corresponding consolidation configuration type and transshipment points. The configuration number represents the order of pairs as for example configuration number 13 represents consolidation pair of 1 and 4 and vice versa. The consolidation type shows the consolidation configuration type among the 8 pair consolidation type described in the previous chapter. The transshipment point column gives the best transshipment point for the consolidation of the corresponding pair. This happens if the consolidation type is one of the consolidation types with the transshipment case which are given by $v \in \{5, ..., 8\}$ as explained in the earlier chapter.

Configuration No	i	j	Consolidation Type	Transshipment Point
0	0	0	0	0
13	1	4	2	0
19	2	2	0	0
32	3	8	2	0
13	4	1	2	0
40	5	5	0	0
46	6	7	1	0
46	7	6	1	0
32	8	3	2	0
54	9	9	0	0

Table 6.1: Consolidated Pairs for the Max Saving of the Sample Data Set (illustrative example)

To provide an overall computational time, we ran the model 10 times by ILOG CPLEX. These CPU times for the 10 runs are given in Table 6.2 where it can be observed that most values are around the 50 milisecs except the last two runs which seems to deviate with an increase. This could be due to the computer being slightly slower by doing something else in the background and in parallel exactly at that moment such as basic computer updates. Normally there is no need to carry out a few runs as the problem is deterministic but in this occasion we just want to be sure if there is any discrepancies between the runs.

	Run-1	Run-2	Run-3	Run-4	Run-5	Run-6	Run-7	Run-8	Run-9	Run-10
CPU Time	46	45	47	52	56	59	52	73	103	46

Table 6.2: CPU time (in milisecs) of 10 runs of the Sample Data Set

6.3 Set Partitioning-based formulation

The above problem can be formulated differently if we can identify all the possible consolidation configurations a priory. The problem will turn into determining the optimal subset of these configurations that cover all the shipments. In other words, the problem can be formulated as a set partitioning type model. For this particular scheduling we can take advantage of the insight of the problem by introducing valid inequalities which we will refer to as tightening.

We first define the various configurations and then produce the set partitioning formulation followed by some tightening of the model.

6.3.1 Defining the various configurations

Consider we have n requests with respective collection, delivery, quantity and time windows. As we are restricting our consolidations to include up to most triplets only, we can summarise these configurations as follows:

Singletons type configurations

As we have n requests, this leads to exactly n configurations with each request served on its own. In other words, no consolidation is allowed.

Let n_0 denotes the number of singletons and

$E_1 = \{k : k = 1, \dots, n\}$ be the set of such configurations with $n_0 = n = |E_1|$ its cardinality. This is equivalent to having one out of n combinations which is exactly $n_0 = C_n^1$.

Let

$$y_k = \begin{cases} 1 & \text{if } k^{th} \text{ configuration is used;} \\ 0 & else \end{cases} \qquad k = 1, \dots, n_0 \tag{6.7}$$

Consolidation with pairs

Here, we consider the consolidation of requests made up by pairs. In the worst case scenario we have

$$n_1 = C_n^2 = \frac{n(n-1)}{2} \tag{6.8}$$

Let $E_2 = \{k : k = n_0 + 1, \dots, n_0 + n_1\}$ be the set of such configurations and $n_1 = |E_2|$ be its cardinality.

Let

$$y_k = \begin{cases} 1 & k^{th} \text{ consolidation of pairs is chosen} \\ 0 & else \end{cases} \quad k = n_0+1, \ldots, n_0+n_1 \quad (6.9)$$

Consolidation with triplets

In this case, we consider those consolidations made up of trips. Using the same approach, in the worst case we have the following number

$$n_2 = C_n^3 = \frac{n(n-1)(n-2)}{6} \quad (6.10)$$

Let

$$y_k = \begin{cases} 1 & k^{th} \text{ consolidation of pairs is chosen} \\ 0 & else \end{cases} \quad k = n_0+n_1+1, \ldots, n_0+n_1+n_2 \quad (6.11)$$

Let $E_3 = \{k : k = n_0 + n_1 + 1, \ldots, n_0 + n_1 + n_2\}$ be the set of such configurations and $n_2 = |E_3|$ its cardinality.

6.3.2 Formulation 1- A simple set partitioning model

Let $N = n_0+n_1+n_2$ be the total number of all possible consolidations including singleton, pairs and triplets. Let also define the following input matrix to represent whether a given request is covered by a given configuration. Note that the elements of this matrix are computed a priory. This is defined as

$$a_{ik} = \begin{cases} 1 & \text{if request } i \text{ is covered by configuration } k \\ 0 & else \end{cases} \quad i = 1, ...n; k = 1, ...N \quad (6.12)$$

Let S_k be the saving due to having the k^{th} consolidation; $k = 1, \ldots N = n_0 + n_1 + n_1$

Note that the total number of configurations is in practice much reduced. This is due to feasibility constraints in terms of capacity and time windows for the case of pairs and triplets. The saving for such infeasible configurations can be set to $-M$, very large negative number (in other this will never be chosen).

In other words, note if the k^{th} consolidation is not feasible ($k = n_0 + 1, \ldots, n_0 + n_1 + n_2$), we set $S_k = -M$ (M being a large positive number)

Let $N = n_0 + n_1 + n_2$ be the total number of alternatives including singletons, pairs and triplets as defined in the previous subsection. Let the following formulation (P) be given as follows:

$$\max \sum_{k=1}^{N} S_k y_k \tag{6.13}$$

$$\sum_{k=1}^{N} a_{ik} y_k = 1; i = 1, \ldots, n \tag{6.14}$$

$$y_k = \{0, 1\}; k = 1, \ldots, N \tag{6.15}$$

(6.13) define the objective function which is to maximize the total saving, constraints (6.14) guarantee that each request belongs to one configuration only including singletons, and (6.15) refer to the binary nature of the decision variables. The problem has n constraints and $B = n + (n-1).n/2 + (n.(n-1).(n-2)/6$ binary variables. Note that the partitioning model with triplets increased the number of binary variables from an order of n^2 in case of pairs up to n^3.

6.3.3 Formulation 2 - Tighter set partitioning models

Th above model can be tightened given the structure of the problem. We first produce a basic tightening followed by specific valid inequalities as well as other subset based constraints to yield other forms of tightening.

(a) Basic Tightening

Note that above formulation can be tightened by adding the following valid inequality. It can be observed that in any feasible solution, in the worst case, there will be at most n configurations of singletons. In other words no consolidations of pairs or triplets would occur. This can be guaranteed by adding the following constraints which states that among all the possible consolidations (i.e., N) including singletons, we can always guarantee to have at most n configurations. This explains why this constraint is a valid inequality.

$$\sum_{k=1}^{N} y_k <= n \tag{6.16}$$

(b) Stronger Tightening via valid inequalities

We can tighten the formulation even further by replacing (6.16) with the following set of valid inequalities related to each type of consolidation, namely, singleton, double and triplets.

$$\sum_{k=1}^{n} y_k <= n \tag{6.17}$$

$$\sum_{k=n+1}^{n+n_1} y_k <= n/2 \tag{6.18}$$

$$\sum_{k=n+n_1+1}^{n+n_1+n_2} y_k <= n/3 \tag{6.19}$$

(6.17) guarantees that if a solution configuration is made up of singletons only the solution can not have more than n. Using the same idea, (6.18) limits the solution made of pairs to $n/2$ only as we cannot have more than that. The same logic is also applied to triplets where (6.19) guarantees that any solution made up of triplets only will have at most $n/3$ configurations. All these three sets of constraints are valid inequalities as explained above.

Note that though the model has 3 extra constraints, the feasible region is now much restricted which could lead to obtaining the optimal solution relatively quickly though this is not always guaranteed. In other words, we expect the solver to use less iterations but each iteration may require slightly more time due to the extra three constraints. So there is a balance between the number of iterations saved against the extra computing time required per iteration. Unfortunately striking the right balance is not an easy task to identify from the outset. A more formal analytical discussion will be provided later.

(c) Stronger tightening via Subset inclusion

In the basic tightening as defined by constraint 6.16, we consider all the N configurations. However, the idea is still valid if we use any random subset of size n, the selected configurations within that subset will be always less than or equal n though it will be most of the time strictly less if configurations of pairs or triplets were part of the random subset selected. In the worst case no configuration from that random subset will be chosen making such new constraint a valid inequality. This is given as follows.

$$\sum_{k \in E_n} y_k <= n \tag{6.20}$$

where E_n is a subset of random n configurations chosen from the N configurations.

It is also worth noting that this kind of constraints can be added as many times as we wish, say M times.

Let E_n^r be the r^{th} random subset of configurations of size n. Therefore the new set of valid inequalities can be defined as

$$\sum_{k \in E_n^r} y_k <= n; r = 1, \ldots, M \tag{6.21}$$

As mentioned earlier, the choice of M can be tricky as adding too many may restrict the feasible region which is good but could consume more CPU time while solving each time a more compact formulation as will be shown later. In our computational results we will investigate a suitable value of M by analysing the results for the case of $r = 1, ..., M'$ with $M' <= M$.

6.4 Computational Analysis

We first provide a comparison between the 0-1 ILP and the set partitioning formulation for the case of pairs to demonstrate that the latter is more efficient and hence worth pursuing. The results of the tightening is then assessed and interesting results are summarised. The analysis is performed on 4 sets of data ranging in size from 50 to 200 in an increment of 50, each instance is run 10 times and average results produced.

Note that all the formulations are coded in Visual Studio C++ and executed on an Intel Core i5-7300U CPU @ 2.60GHz 64-bit operating system with 8 GB RAM. The commercial optimisation software, namely, IBM ILOG CPLEX 12.9 is used for solving the mathematical model.

6.4.1 Comparison of 0-1 ILP vs Basic set partitioning formulation for the case of pairs

We first ran 10 instances of each size of 50, 100, 150 and 200 with the 0-1 ILP original formulation and the basic set partitioning for the case of pairs. In addition within each set we run for 10 runs and record the average CPU time. This is done just to record explicitly the CPU time though that is usually similar. As an example we show the case of $n = 50$ when using the 0-1 ILP formulation in Table 6.3. A similar pattern is found in all other instances and also when using the set partitioning.

CPU Time	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	Average
Data Set 1	123	156	133	393	423	180	356	124	170	170	222.8
Data Set 2	102	125	142	117	107	113	114	100	77	77	107.4
Data Set 3	106	127	123	116	147	134	125	132	116	110	123.6
Data Set 4	73	86	92	110	129	104	89	93	108	111	99.5
Data Set 5	85	94	98	123	124	117	117	144	123	95	112
Data Set 6	109	103	153	135	136	116	161	134	104	125	127.6
Data Set 7	103	112	155	145	332	327	237	122	133	124	179
Data Set 8	81	74	74	93	118	83	90	78	131	134	95.6
Data Set 9	65	80	86	93	135	81	78	87	100	111	91.6
Data Set 10	104	126	105	158	128	125	122	158	113	122	126.1
Overall Average											**128.52**

Table 6.3: An Illustration of CPU Times (milisecs) and averages for Paired Consolidation with 0-1 ILP Formulation for data sets of size $n = 50$

We ran the same data sets with the set partitioning formulation for the paired consolidation case. As expected the results are similar but the corresponding computational burden is very different with the set partitioning model consuming massively less CPU time. The average CPU times of 10 runs of 10 samples of size of 50, 100, 150 and 200 are presented in Figure 6.1. For example, the value reported for the $n = 50$ (i.e., size 50) for the 0-1 ILP is the overall average value provided in Table 6.3. These is performed for all the data sets and for both approaches. We can see that the set partitioning model performs much more effective compared to 0-1 ILP formulation.

The graph of average CPU improvement by the set partitioning formulation over its counterpart is also given for each size in Figure 6.1. As we can see from the graph, the average improvement for size 50, 100 and 150 is around 57% and for size 200 is 85%, with the increase of the sample size, the set partitioning performs dramatically even better.

Mathematically this can also be due to the fact that the original 0-1 ILP model requires over n^2 constraints whereas the set partitioning one has n constraints only. However, it is also worth noting that the former uses n^2 binary variables whereas the latter needs $n + n(n-1)/2 + n(n-1)(n-2)/6$ binary variables.

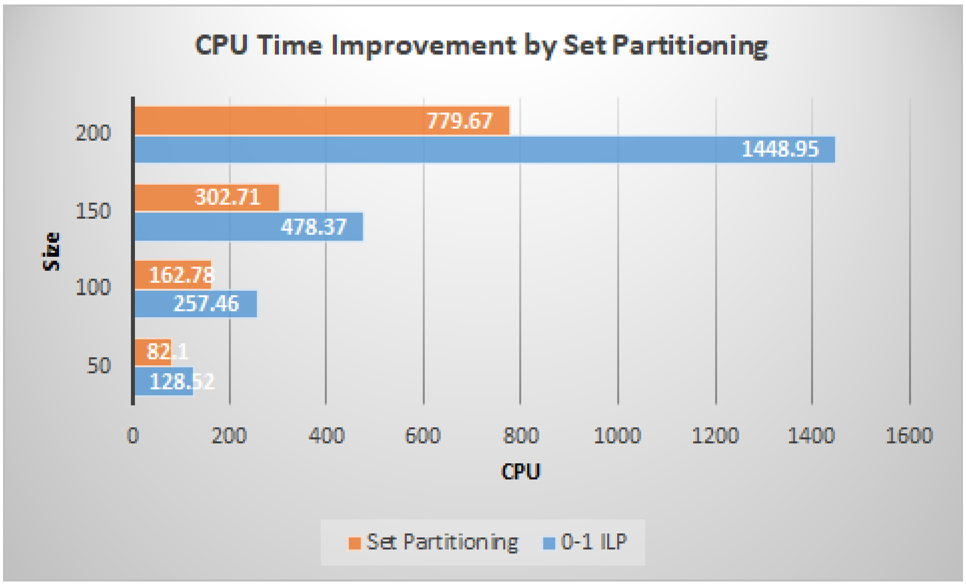


Figure 6.1: CPU time for 0-1 ILP and the set partitioning

Some error plot graphs

We plot the error bar graphs based on standard deviation (std) of CPU times for paired consolidation from 0-1 ILP and set partitioning formulations for each sample size. Standard deviation graphs of CPU time for size 50 from 0-1 ILP and set partitioning formulations are given in Figure 6.2-Figure 6.3.

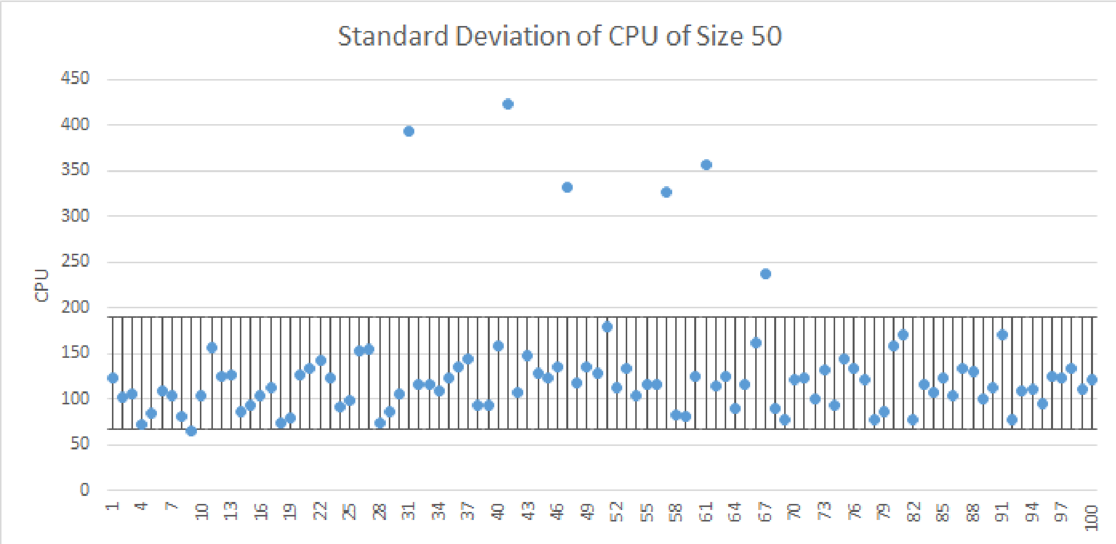


Figure 6.2: Standard Deviation of CPU of 0-1 ILP Formulation for Paired Consolidation-50

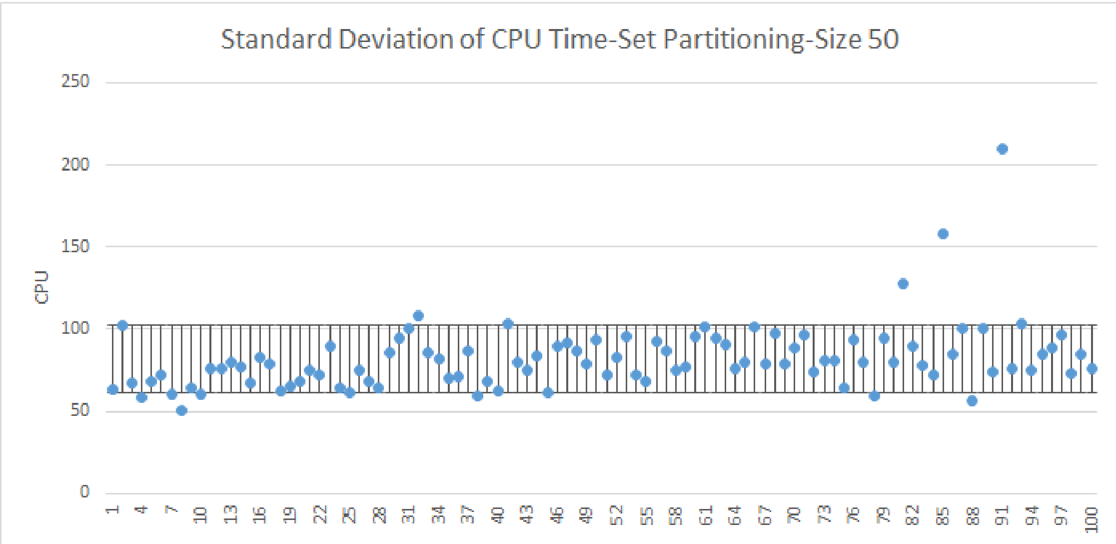


Figure 6.3: Standard Deviation of CPU of Set Partitioning for Paired Consolidation-50

Error plot graphs for the other sizes are given in Appendix C. According to these graphs we can say that the set partitioning formulation is slightly more reliable than its counterpart due to the smaller spread of its CPU times.

6.4.2 Results including triplets of the Set Partitioning and its variants

The above results demonstrate that the set partitioning is more efficient when tested in the case of pairs. For this reason this is the formulation which we will explore further in terms of computational results for the case of the whole problem including the triplets. Obviously, the 0-1 ILP formulation could be extended to triplets as a mathematical exercise but with no real gain given that it was outperformed by the set partitioning model.

Effect of using triplets over pairs

We ran the same data sets for triplet consolidation including pair consolidations with set partitioning formulation. The savings obtained from triplet consolidation vs paired consolidation and improvement from paired consolidation to triplet consolidation for size 50 is given in Table 6.4, and summary for all sizes is given 6.5.

Size 50	**Pair**	**Triplet(Incl Pair)**	**Improvement(%)**
Data Set 1	854.20	1171.38	37.13
Data Set 2	860.05	1179.08	37.09
Data Set 3	605.51	836.57	38.16
Data Set 4	524.68	706.02	34.56
Data Set 5	674.59	912.36	35.25
Data Set 6	639.14	903.16	41.31
Data Set 7	786.74	1094.44	39.11
Data Set 8	522.44	721.99	38.20
Data Set 9	762.39	972.99	27.62
Data Set 10	785.85	1049.00	33.49
Average	701.56	954.70	**36.08**

Table 6.4: Paired vs Triplet Consolidation Savings of Size of 50

Size	Pair	Triplet(Incl Pair)	Average Improvement(%)
50	701.56	954.70	**36.08**
100	1639.84	2228.37	**35.89**
150	2584.37	3523.99	**36.36**
200	3536.87	4929.47	**39.37**

Table 6.5: Paired vs Triplet Consolidation Savings of All Sizes

Results from basic tightening

We tested the set partitioning model with addition of basic tightening constraint and we obtained the average CPU times. As obviously seen in Figure 6.4, by the increase of sample size set partitioning model with the addition of basic tightening formulation performs better than the other formulations.

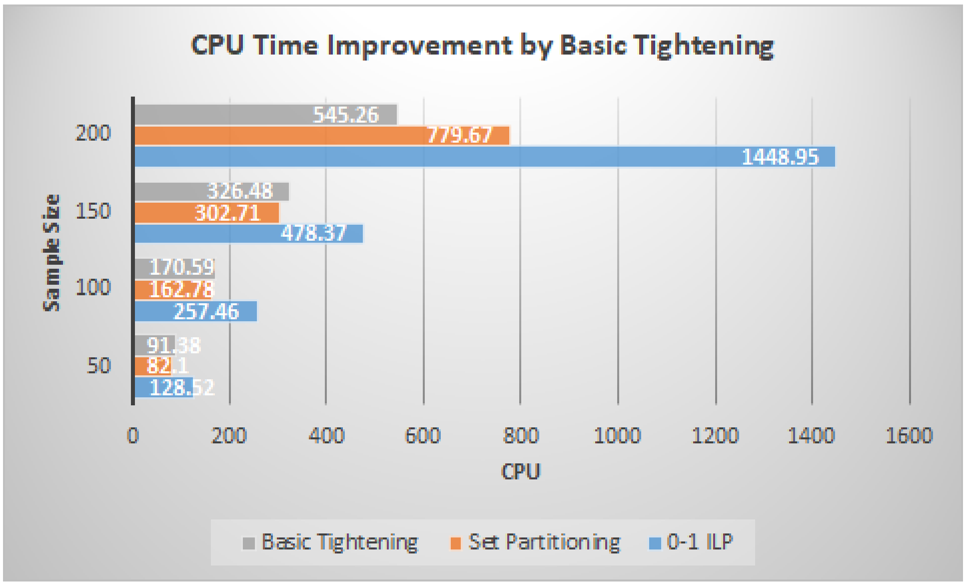


Figure 6.4: CPU of Set Partitioning with Basic Tightening vs other formulations for Paired Consolidation

Results from strong tightening (valid inequalities)

We replaced basic tightening constraint in set partitioning formulation with 3 level strong tightening constraints for pair consolidation of orders and we obtained the average CPU times based on sample size. We illustrate the improvement in CPU time with the addition of strong tightening constraints we plotted a comparative bar graph given in Figure 6.5, by the increase of the sample size, set partitioning formulation with the addition of strong tightening constraints performs much better than set covering, set partitioning and set partitioning with basic tightening formulations.

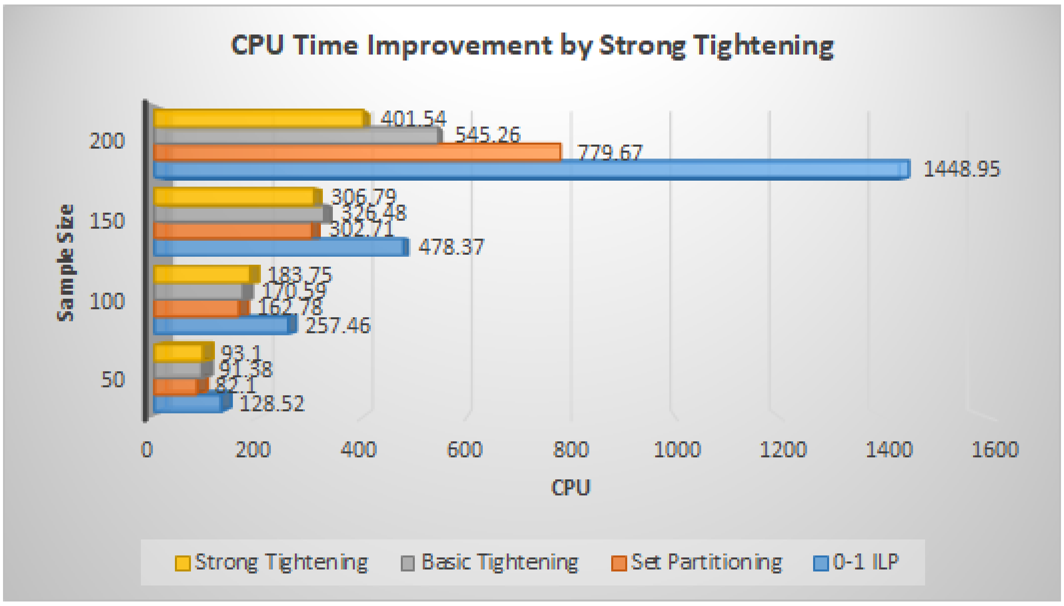


Figure 6.5: Improvement in CPU with strong tightening constraints

Results from the subset based tightening

We added 6 sub-basic tightening constraints for 6 subsets for paired consolidation of size 50, 100, 150 and 200. 10 instances of each size was run 10 times and CPU times are computed. As an example, random numbers for 2 subsets and CPU times for size 100 are given in Table C.7. For completeness, all other subsets for size 100 are given in Appendix B each sub-basic tightening constraints up to 6. As there are 4950($100\times99/2$) possible configurations for size 100, random numbers are generated between 0 and 4950. r_1,...r_6 denotes random numbers. Subset constraints are added within the range of 100, additional 2 subset constraints for size 100 are as in the following(where r_1=41, r_2=3617 from Table C.7):

$$\sum_{k=41}^{41+100} y_k <= n \tag{6.22}$$

$$\sum_{k=3617}^{3617+100} y_k <= n \tag{6.23}$$

We plotted comparative bar graph to illustrate the CPU performance of subset-based tightenings. It is given in Figure 6.6, Figure 6.7, Figure 6.8 and Figure 6.9. For all sizes we can say that adding 3 or 4 subsets work most effectively. After 4 subsets it cause increase in CPU as seen for 5 subsets for size 100 and size 200 in Figure 6.7 and Figure 6.9 respectively. Especially size 100 shows significant increase for subset 5. Summary of the subset based tightenings based on each size and each subset with deviation values from set partitioning formulation is given in Table 6.7. According to the average improvement based on each subset, we can justify the same result that 3 subset and 4 subset are more likely to provide better CPU time.

$r_1 - r_2$	1	2	3	4	5	6	7	8	9	10	Average
41-3617	199	187	195	198	197	223	437	291	244	190	236.1
61-491	109	160	99	107	106	92	111	96	99	95	107.4
755-3395	114	95	112	94	104	97	118	96	104	92	102.6
1384-1750	183	206	213	176	150	168	265	267	154	123	190.5
1578-4608	116	143	147	107	114	110	106	96	116	112	116.7
2212-4664	110	106	123	98	107	98	107	93	111	103	105.6
2995-2042	115	104	110	122	119	164	101	114	99	120	116.8
3481-1977	112	105	107	114	131	103	167	90	99	94	112.2
4319-874	157	114	119	99	138	96	118	92	99	94	112.6
4827-486	106	106	178	137	96	105	93	97	102	133	115.3
Overall Average											131.58

Table 6.6: CPU with Set Partitioning with Subbasic Tightening-2 for size 100

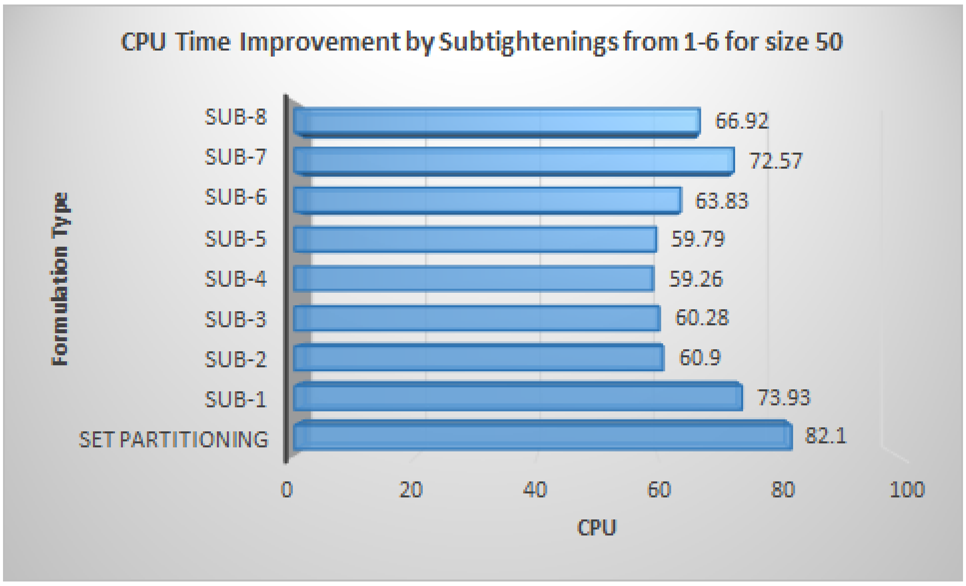


Figure 6.6: Comparative CPU Time for Subset-based Tightenings for Size 50

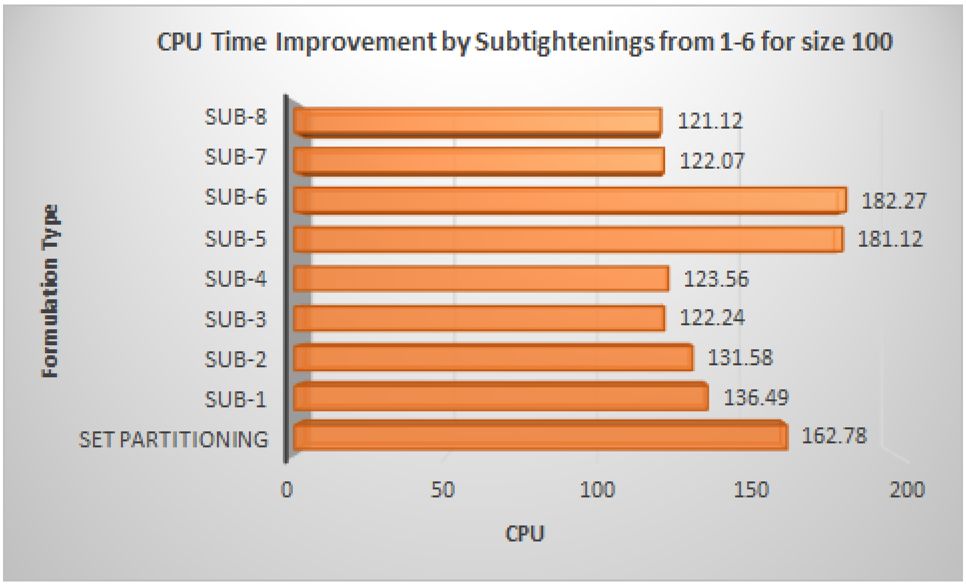


Figure 6.7: Comparative CPU Time for Subset-based Tightenings for Size 100

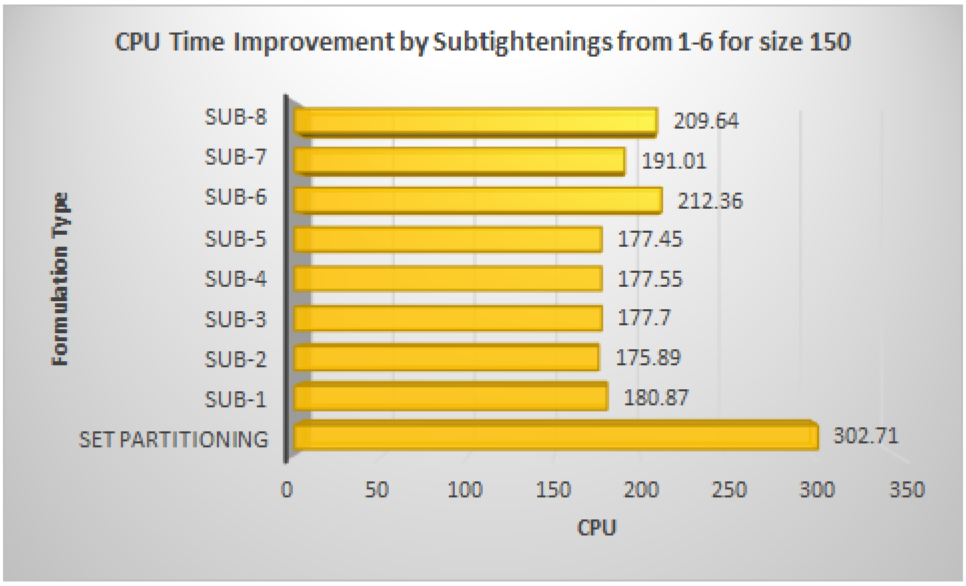


Figure 6.8: Comparative CPU Time for Subset-based Tightenings for Size 150

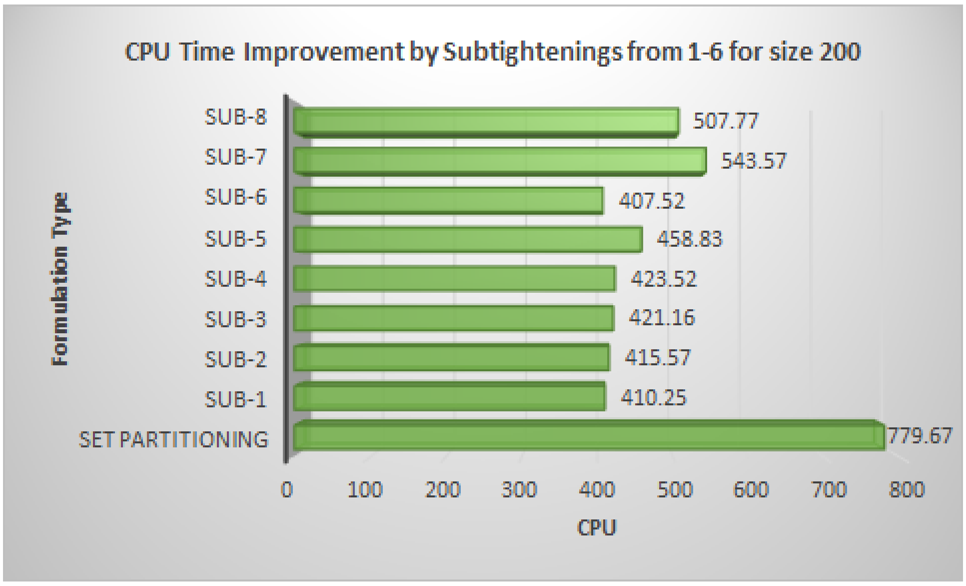


Figure 6.9: Comparative CPU Time for Subset-based Tightenings for Size 200

Sample Size \ Subset Number	1	2	3	4	5	6	7	8
50	0.10	0.26	0.27	0.28	0.27	0.22	0.12	0.18
100	0.16	0.19	0.25	0.24	-0.11	-0.12	0.25	0.26
150	0.40	0.42	0.41	0.41	0.41	0.30	0.37	0.31
200	0.47	0.47	0.46	0.46	0.41	0.48	0.30	0.35
Overall Average	0.28	0.33	0.35	0.35	0.25	0.22	0.26	0.27

Table 6.7: Average deviation over 10 runs from the original set partitioning formulation (in %) for n=50,100,150 and 200

We conducted same set partitioning computations for triplet consolidation case. Comparative CPU times are given in Table C.12 for set partitioning and set partitioning with addition of basic tightening constraint and set partitioning with the addition of strong tightening constraints. It is illustrated in Figure 6.10.

Reasoning behind possible increase in CPU in tightening

The increase in CPU time when adding constraints (ie tightening) can be seen as counter intuitive. However, if we dig deep into how the Simplex method works that could easily happen. In our case as an example, tightening constraints required less CPU time as in the case of pair consolidation especially with the increase of sample size it performed significantly better. However, as there are so many operations in case of triplet consolidation, the number of requests has to be large enough to compensate with the extra effort. Here below we provide a simple mathematical explanation that could contribute to this increase.

Let n_0 and t_0 be the original number of iterations required and the approximate time to perform one iteration respectively. We refer to one iteration as the Simplex iteration within CPLEX. So the approximate total time, say T_0, to complete the original implementation is:

$$T_0 = n_0 * t_0 \tag{6.24}$$

Once we add additional constraints, the approximate time per iteration is now t_1 which will be larger, $t_1 > t_0$ as the Simplex tableau is bigger. The number of iterations required due to tightening say n_1 will be $n_1 <= n_0$ as tighter bounds could help to avoid having unnecessary iterations. The new total time with tightening, say T_1, will be:

$$T_1 = n_1 * t_1 \tag{6.25}$$

In summary, tightening is worth to do if $T_1 < T_0$ which leads to

$$n_1 < n_0 * (\frac{t_0}{t_1}) \tag{6.26}$$

as $t_1 * n_1 <= n_0 * t_0$

Therefore it can be said that depending on the ratio of the two times and that of the two number of iterations. As an example, if we could save 5% of the number of iterations via tightening but if each new iteration requires more than 5% cpu time than its original one, we do not really gain anything but lose more. However if the new iteration needs only say 4% more CPU, we will save about 1% overall. In our experiments, we observe that if the number of request becomes greater than 200, tightening would start to perform better.

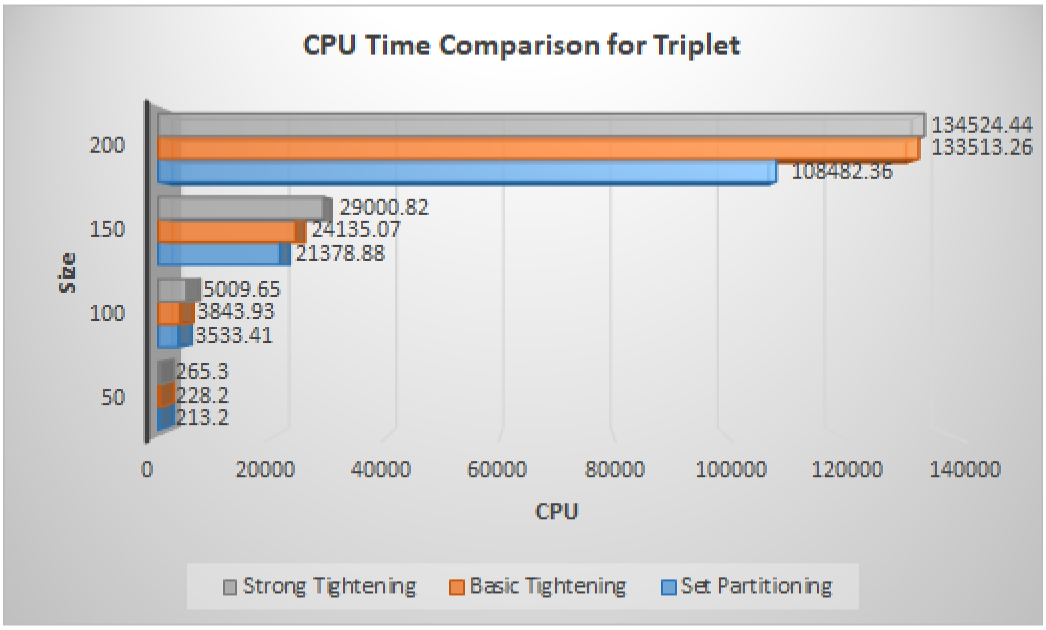


Figure 6.10: Comparative CPU Graph for Triplet Consolidation

Figure 6.10 shows that tightening constraints do not bring gain in terms of CPU time for triplet consolidation case with these sample sizes due to the reason given above. If larger sizes of data sets are tested for triplet consolidation case then it might worth to add additional constraint to save CPU time.

Note that all detailed tables are given in Appendix C.

6.5 Summary

In this chapter, two mathematical models are developed, one using a 0-1 ILP and the other based on set partitioning. For the latter, all the configurations using singleton, pairs and triplets are first defined. Besides, the latter is made more efficient by incorporating interesting tightening constraints, also known as valid inequalities, into the model. A computational analysis is conducted using four set of instances varying in size from 50 to 200 with a step size of 50. To provide statistical evidence, 10 random instances for each size are used and average results in terms of CPU recorded and analysed. It was found that set partitioning based formulation is relatively much more efficient and the tightening is useful though not in all cases. A mathematical reasoning behind this potential weakness is also presented.

In the next chapter, we treat the same scheduling problem by developing two powerful metaheuristics to respond to the excessive CPU time that could be required for larger instances.

Chapter 7

Meta-heuristic approaches: Variable Neighbourhood Search, Large Neighbourhood Search and their hybridisation

7.1 Introduction

In this chapter, we exploit the power of two well known meta-heuristics to address the same order consolidation scheduling problem studied in the previous chapter. This includes variable neighbourhood search (VNS) and large neighbourhood search (LNS). For the VNS, appropriate neighbourhood structures and local searches are developed and for the LNS, novel removal and insertion operators are examined. To provide the strengths of the two approaches we also investigate the effect of hybridisation. Computational results based on the same instances given in the earlier chapter are used for testing, which are then analysed and discussed.

7.2 Variable Neighbourhood Search (VNS)

We first provide the general framework of VNS. We then discuss the various neighbourhood structures and the local searches used. As we analyse some variants empirical testing will be used to choose the most appropriate one.

7.2.1 Basic VNS algorithm

The main idea of VNS is to systematically change the neighbourhood whenever a solution is not improved. It usually starts with the smallest neighbourhood to evaluate and keeps increasing the neighbourhood depending on the complexity of the neighbourhood. The method is made up of three basic steps, known as shaking (generation of a neighbouring solution), local search (try to improve the solution) and move/not move (acceptance criterion to whether or not to accept the solution and then change or not the neighbourhood). There are obviously several variants as discussed in the review chapter by Salhi & Thompson (2022*a*). In this study, we explore a simple VNS version to see how it behaves against the optimal solution as more complex ones could easily be extended and investigated.

The VNS Algorithm

Step1: Generate an initial solution X, select the set of neighbourhood structures $N_k(X)$, k=1,... k_{max}.(k_{max} is the number of neighbourhood structures). Define the local searches LS(l); l=1,... l_{max} with l_{max} being the number of local searches. Set $k = 1$

Step 2: While stopping criteria not met apply the following

2a) Generate the solution X' in $N_k(X)$ *[shaking step]*

2b) Apply a local search engine (made up of $LS(l); l = 1, \ldots, l_{max}$ to X' to find X''. *[local search step]*

2c) If X'' is better than X(ie. $F(X'') > F(X)$; case of maximization), set $X = X''$ and $k = 1$ else if $k = k_{max}$ set $k = 1$ and return to N_1 else set $k = k + 1$. *[Move or not move step]*

2d) Go back to step 2

In this study, we set $k_{max} = 6$ and $l_{max} = 2$. The details of the neighbourhood structures and local searches are given in the subsequent sections.

7.2.2 The Neighbourhood structures

For this problem, six neighbourhood structures are defined.
Let C be the number of routes in solution X, c_{ij} the j^{th} request of the i^{th} route, and cr_i the number of requests in route i. An illustrative example with the number of requests (i.e., $n = 9$), indexed by $i = 1, ...9$ and the number of routes being $C = 5$ is given below:

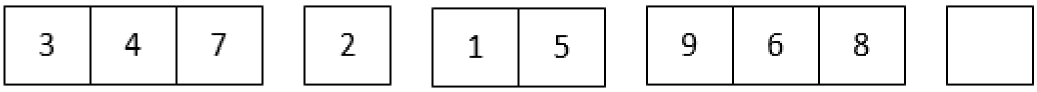


Figure 7.1: An Illustrative Example for Routes

$cr_1 = 3$, $cr_2 = 1$, $cr_3 = 2$, $cr_4 = 3$ and $cr_5 = 0$. Note that we always keep an empty route with one idle node at each iteration to create opportunity to attract singleton in case it is worthwhile. The concept of providing an empty route during the search has proved to be crucial in earlier metaheuristics that are applied to routing problems.

For instance, in Figure 7.1 we have as an example $c_{13} = 7$ and $c_{32} = 5$

Neighbourhood structures and corresponding illustrative examples are given below:

Neighbourhood 1 (1-0): Here the idea is to remove a random request from a random route and insert it in another random route which has requests less than 3 to guarantee feasibility as the maximum number of requests is 3.

Let $c_{12} = 4$ being selected randomly to be removed from route 1, and the second route i=2 is chosen randomly for the insertion of this removed request, namely request 4. This will be removed from the first route and will be inserted to the end of the second route. The representation of the solution configuration before and after the change is given in Figure 7.2.

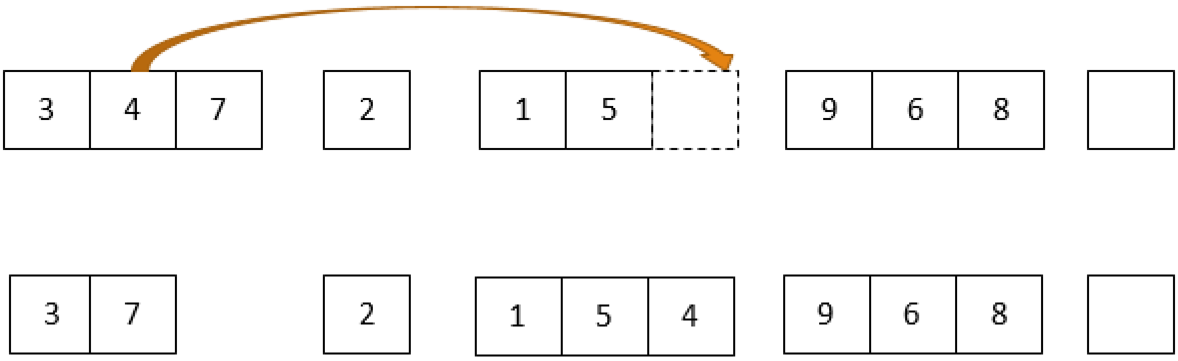


Figure 7.2: An Illustrative Example for Neighbourhood 1

Neighbourhood 2 (1-1): In this neighbourhood, random requests of 2 random routes are swapped.

Let $c_{21} = 2$ and $c_{41} = 9$ are selected to be swapped.

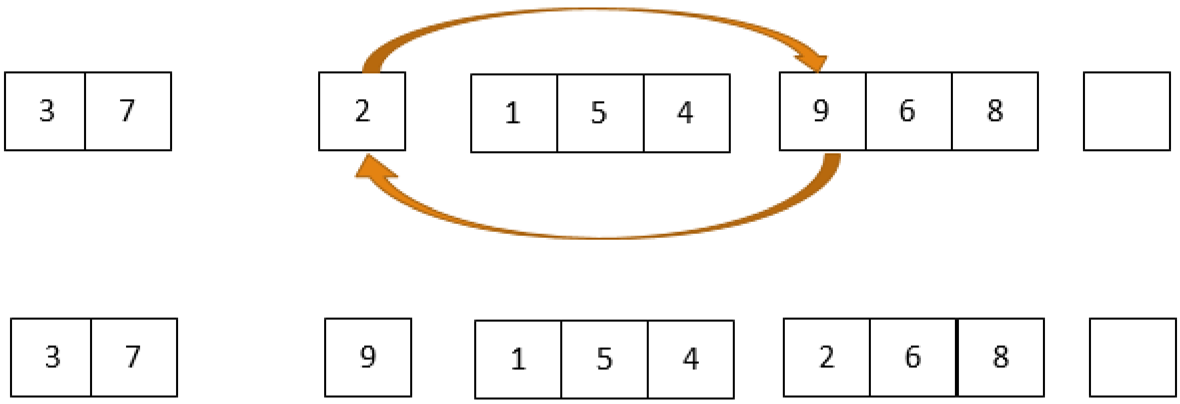


Figure 7.3: An Illustrative Example for Neighbourhood 2

Neighbourhood 3 (2-0): Here the idea is to remove 2 random requests from a random route which has at least 2 requests and insert the removed requests in another random route which has requests less than 2.

Let $i = 3$ is selected randomly as route to be removed from and $j = 2, 3$ are selected as requests to be removed from route $i = 3$, which are $c_{32} = 5$ and $c_{33} = 4$ and $i = 2$ is selected randomly as route to inserted.

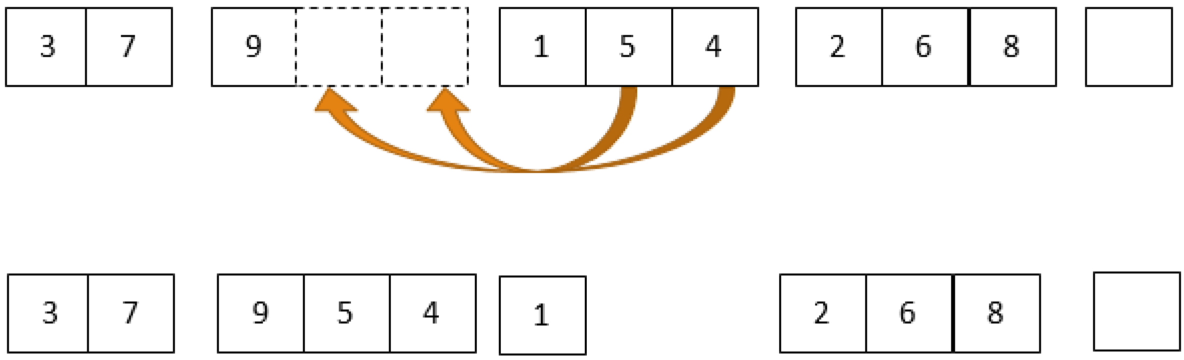


Figure 7.4: An Illustrative Example for Neighbourhood 3

Neighbourhood 4 (2-0): In this neighbourhood 2 random requests are removed from a random route as in neighbourhood 3 but here each of the removed requests are inserted into 2 different random routes.

Let $i = 2$ is selected randomly as route to be removed from and $j = 1, 2$ are selected as requests to be removed from route $i = 2$, which are $c_{21} = 9$ and $c_{22} = 5$ and $i = 1$ and $i = 5$ are selected randomly as route to inserted.

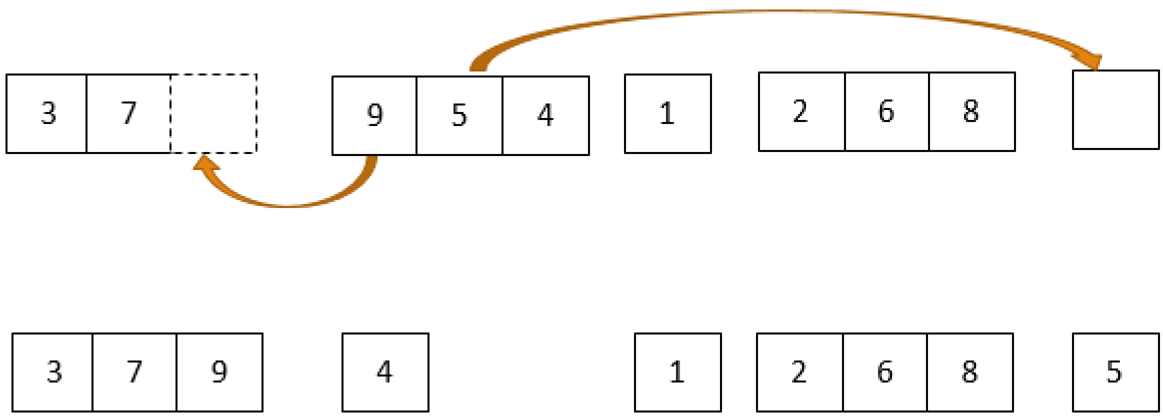


Figure 7.5: An Illustrative Example for Neighbourhood 4

Neighbourhood 5 (2-1): Here, 2 random requests from a random route are removed, while one of these random routes swapped with a random request from a random route and the other removed request is inserted into the same route. Therefore the random route which will be inserted into has to have at least a request to be swapped with one of the removed requests and maximum 2 requests as the other removed request will be inserted into this route to guarantee feasibility as the maximum number of requests is 3.

Let $i = 4$ is randomly selected as route to be removed from and swapped with and $i = 2$ is selected as route to be inserted into and swapped with. Let $c_{43} = 8$ is randomly selected to be be swapped with $c_{21} = 4$ which is randomly selected to be swapped too, and $c_{41} = 2$ is randomly selected to be removed and inserted into route $i = 2$.

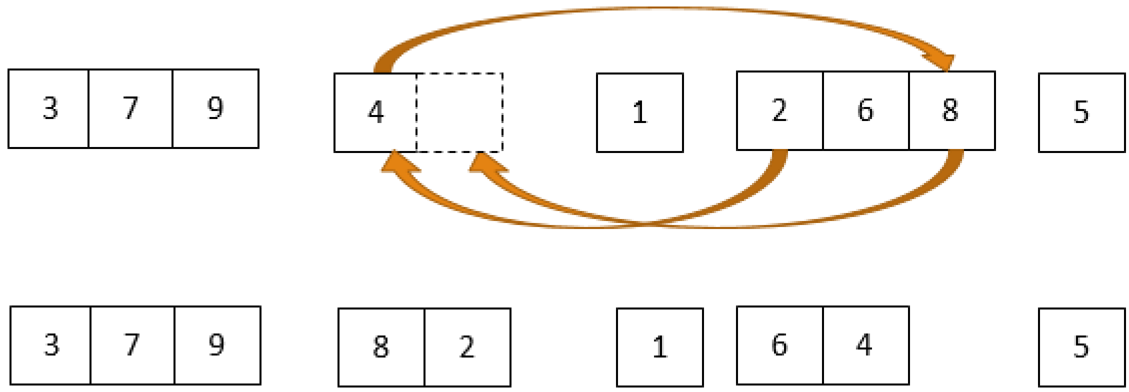


Figure 7.6: An Illustrative Example for Neighbourhood 5

Neighbourhood 6 (2-1): Here, 2 random requests from a random route are removed. The removed requests are processed separately in different routes. One of the removed requests is swapped with a random request of a random route and the other removed request is inserted in a different random route.

Let $i = 1$ is randomly selected as route to be removed from and swapped with and $i = 2$ is selected as route to be inserted into and $i = 5$ is randomly selected as route swapped with. Let $c_{13} = 9$ is randomly selected to be swapped with $c_{51} = 5$ and $c_{11} = 3$ is randomly selected to be removed and inserted into route $i = 2$.

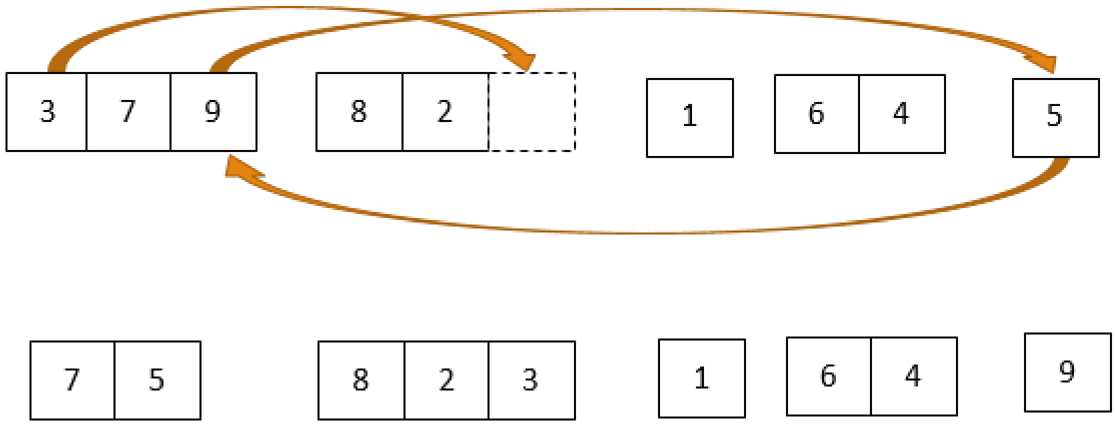


Figure 7.7: An Illustrative Example for Neighbourhood 6

7.2.3 The local searches

We used two local searches. One is based to improve routes within routes whereas the second is to improve routes by using more than one route.

Local Search 1 (LS1)

LS1(X): (Within routes) This local search is designed for the search of the best consolidation configuration for a given candidate route. As there are several possible consolidation configurations for a given route (set of two or three orders), LS1 finds the best configuration for the given route.
Note that a route is made up of its requests and when evaluated using LS1, the best configuration of such a route is then defined by the order in which the collections and the deliveries including the transshipment if there are made, with its corresponding saving recorded. This is carried out by evaluating all the combinations for the case of pairs and triplets with or without transshipment as already defined in chapter 5 where the consolidations are obtained.

For instance, if we consider the first route in Figure 7.1. This route includes 3 requests as request 3, 4 and 7. This local search computes the saving values for each consolidation configuration, as explained in chapter 5, and finds the one that produces the largest saving, and hence offers the corresponding consolidation configuration. For example, the best consolidation configuration for requests 3, 4 and 7 is shown in Figure 7.8 where the collection for request 4, is followed by the collection for request 7 then the collection for request 3 in sequence. These are then split at the transshipment request 3 receives its delivery on its own route, whereas requests 4 and 7 get their deliveries in a second route by first delivering to request 4 then finally request 7.

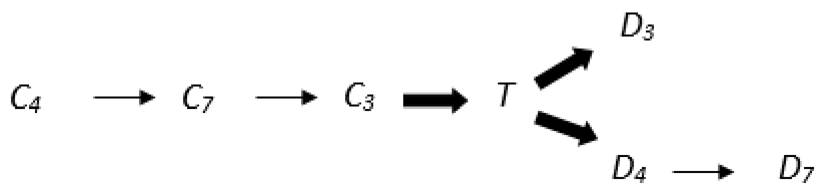


Figure 7.8: An Illustrative Example for Local Search 1

Note that this local search, LS1, also acts as the objective function evaluator as it provides for each route the best configuration with its maximum saving.

Local Search 2 (LS2)

LS2(X): (Between routes) This local search is designed to explore the possibility of shifting requests between routes. For a given set of requests, this local search works to design the best routes with respect to obtaining the maximum saving. In brief, a random request is removed from the first route and saving of the case of insertion of this removed request into other routes which are suitable for insertion and which have less than 3 requests are computed respectively by using LS1. The request is then inserted into where improvement is obtained first. In other words, we are adopting the first improvement strategy instead of the best improvement strategy which can be too time consuming. This procedure is applied throughout all the available routes. A formal step by step algorithm for LS2 is given below followed by an illustrative example.

The LS2 algorithm

For $k = 1, ..., C$

Record its saving (already known) S_{R_k}.

Choose a random request from R_k, say c_{kj} and evaluate the saving for the reduced route say $R'_k = R_k - c_{kj}$ using LS1 to obtain the saving say $S(R_{k'})$

For l =1,...,C

IF ($l \neq k$ and $cr_l < 3$) do the following

Insert c_{kj} in R_l to get $R'_l = R_l + c_{kj}$ and evaluate the saving on R'_s using LS1 to obtain $S_{R'_l}$.

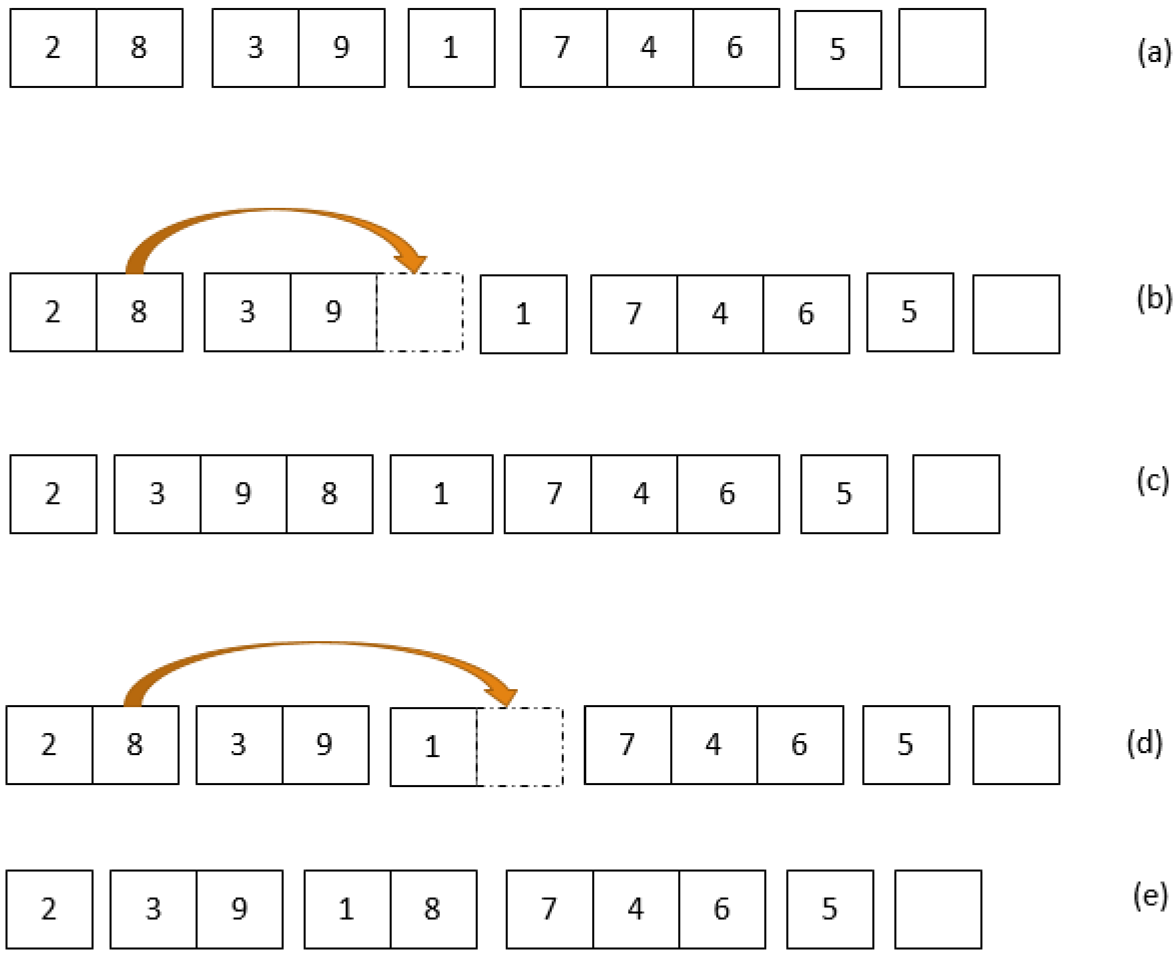
IF $(S_{R'_l} + S_{R'_k}) - (S_{R_k} + S_{R_l}) > 0$

Set $R_k = R'_k, R_l = R'_l, S_{R_k} = S_{R'_k}$ and $S_{R_l} = S(R'_s)$;

Set $cr_k = cr_k - 1$ and $cr_l = cr_l + 1$
Stop

An Illustrative Example

An illustrative example is given Figure 7.9 with the necessary steps explained below.



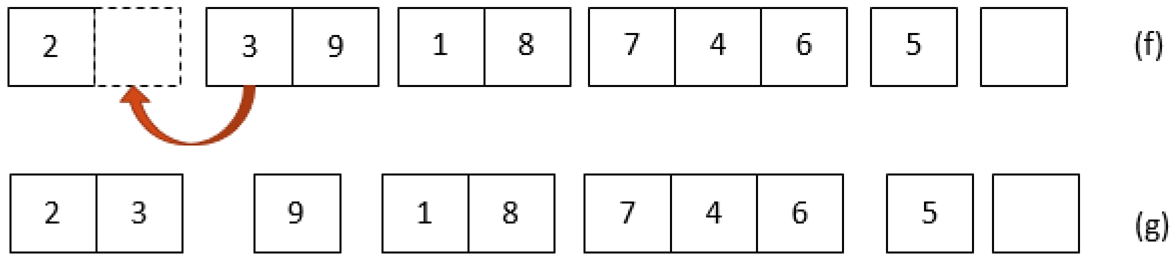


Figure 7.9: An Illustrative Example for Local Search 2

Let say after application of a neighbourhood and local search 1 a candidate solution say x is obtained as given in Figure 7.9a). We will apply local search 2 on x. We use the same notations in neighbourhood structures section. In the above example, $C = 6$, as there are 6 routes.

We apply local search 2 by removing a random request from first route, $c_{12} = 8$ selected randomly as the request to be removed. We first try to insert into the second route. As the effected routes are only first and second route $i = 1$ and $i = 2$, we will compare the saving values of these routes before and after change.

Before the change, first and second routes are as given in Figure 7.9.b), $cr_1 = 2$, $c_{11} = 2$, $c_{12} = 8$ and $cr_2 = 2$, $c_{21} = 3$, $c_{22} = 9$, after the change, the routes are adjusted as in Figure 7.9.c).

Let S_i saving of the route i before change and $S_i^{'}$ saving of the route i after the change.

Saving values of the first and second route before the change,

S_1= 32 and S_2= 15; $S_1 + S_2 = 47$

Saving values of the first and second route after the change

$S_1^{'}$=0 and $S_2^{'}$=40; $S_1^{'} + S_2^{'} = 40$

As the saving values after the change is less than the one before,

$S_1^{'} + S_2^{'} < S_1 + S_2$,

this insertion move is not accepted.

We move on the next route $i = 3$, before the change first and third routes are as given in Figure 7.9.d) and after the change they are as given in Figure 7.9.e). We compare the savings before and after,

S_1= 32 and S_3= 0; $S_1 + S_3 = 32$

$S_1^{'}$=0 and $S_3^{'}$=45; $S_1^{'} + S_3^{'} = 45$

As the saving values after the change is bigger than the one before, $S_1^{'} + S_3^{'} > S_1 + S_3$ which means we find the first improvement, this insertion move is accepted and new solution say $X^{'}$ as given in Figure 7.9.e) is accepted as solution X.

After random request removal from the first route and insertion with first improvement strategy we move on next route $i = 2$. (We do not go through the routes that we changed to save time). $c_{21} = 3$ is selected randomly as request to be removed from route $i = 2$ and we again start from the first route for the insertion of the removed request. The routes before the change are as given in Figure 7.9.f) and after the change as given in Figure 7.9.g). Savings of the affected routes are,

S_1= 0 and S_2= 40; $S_1 + S_3 = 32$

$S_1^{'}$=60 and $S_3^{'}$=0; $S_1^{'} + S_3^{'} = 60$

As $S_1^{'} + S_2^{'} > S_1 + S_2$, first improvement is found, this insertion is accepted and new solution say $X^{''}$ as given in Figure 7.9.g) is accepted as solution X.

The same procedure is applied through the all routes.

7.2.4 Stopping Criterion

The stopping criterion adopted here is based on the number of iterations as well as the number of successive non improvements. Given that n is the number of requests, VNS will stop after

$Maxiterations = n * 0.25$ iterations are performed or $Min(5, 0.05 * n)$ consecutive non-improvements are found whichever is reached first.

The first part of the above rule is introduced to give a upper limit on the number of runs whereas the second is based on the solution quality.

7.2.5 Selection the various VNS implementation variants

We investigate four variants of VNS. These variants are based on two items, namely, (a) initial solution and (b) the way we revert back to step 2.

Variant 1: Start with random initial solution and continue with same initial solution at each run.

Variant 2: Start with best initial solution and continue with the same initial solution at each run.

Variant 3: Start with random initial solution and continue with the last best solution at each run.

Variant 4: Start with best initial solution and continue with the last best solution at each run.

We ran the above four variants of VNS with 10 randomly generated instances of size 50. Each instance was run 10 times and the average performance was recorded. The detailed results of the four variants and their corresponding graph of comparison is given in Figure 7.10.

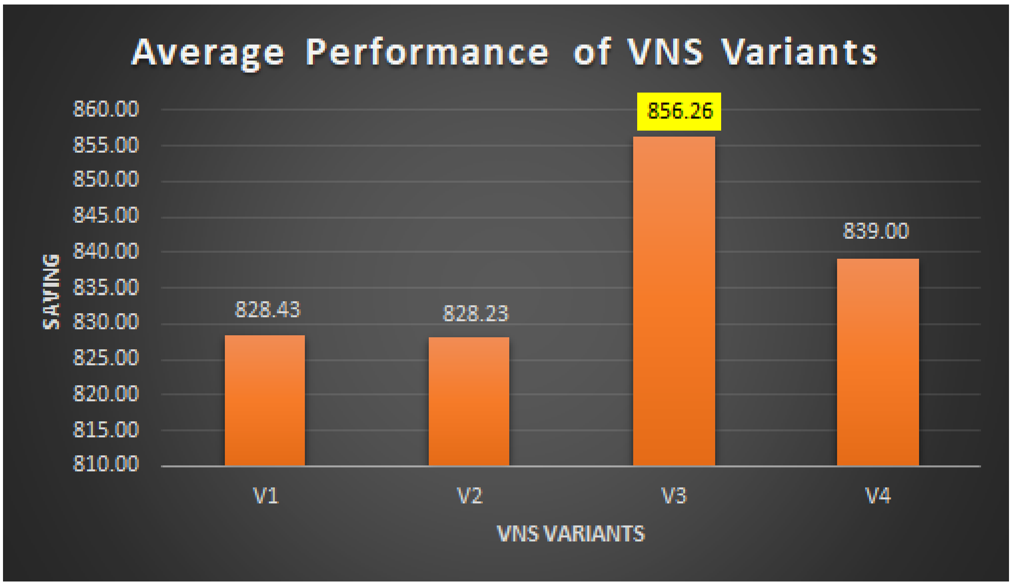


Figure 7.10: Performance Comparison of VNS Variants

According to the results in Table D.1, the best variant is variant 3 with the average result of 856.26 shown highlighted. In other words, this variant starts with a random initial solution and continues with the last best solution at each run.

7.2.6 Adjusted Stopping Criterion

To assess whether the solution found is of good quality or not, we recorded the optimal solution for these instances (as found in the previous chapter) and also presented in Table 7.1 (TS denotes the total saving). It is observed that the optimal average solution for size 50 is 954.70. This is equivalent to a loss in quality of 10.31% (i.e 100*(954.70-856.26)/954.70). A formal mathematical expression of the deviation (in %) is defined in Eq (7.1). In other words, there is a significant gap between the optimal and the heuristic results.

$$Dev_H(\%) = 100 * (Saving(Optimal) – Saving(H))/Saving(Optimal) \quad (7.1)$$

One way forward is to re-examine the way the stopping criterion is designed and implemented. Therefore, we adjusted the stopping rule to see whether the first stopping criterion was a bit restrictive in terms of the number of non-improvements, especially for larger sizes where more running time to improve the solution could be required. This stopping criterion 2 is given next and empirical results derived accordingly.

In this adjusted stopping criterion which we refer as stopping criterion 2, the VNS will stop after
$Maxiterations = n * 0.25$ iterations are performed or
$Min(10, Max(5, (0.05*n))$ consecutive non-improvements are found whichever is reached first.

Note that the first part of the above rule referring to the upper limit on the number of iterations is not changed.

We ran the best VNS variant, namely, VNS-Variant 3, with this new stopping rule for the same 10 instances of size 50.

Instance	TS	Optimal	Dev(%)
50-1	1067.43	1171.38	8.87
50-2	1046.74	1179.08	11.22
50-3	786.63	836.57	5.97
50-4	636.71	706.02	9.82
50-5	807.99	912.36	11.44
50-6	783.44	903.16	13.26
50-7	992.13	1094.44	9.35
50-8	657.74	721.99	8.90
50-9	910.49	972.99	6.42
50-10	933.01	1049.00	11.06
Average	862.23	954.70	9.63

Table 7.1: Saving of size 50 instances with second stopping rule

According to the results, the deviation of the average solutions decreased to 9.68% from 10.31%. (Note that average of the deviations is slightly different, recorded at 9.63% instead).

7.3 Large Neighbourhood Search (LNS)

In this study, we also explore a perturbation based metaheuristic, namely, LNS, which was briefly described in the literature review section. For more details, see the overview chapter by (Salhi & Thompson 2022*a*). In brief, the idea behind LNS is made up of two steps which can be repeated several times:

(a) destroy a given solution by removing a certain amount of attributes, say r_{max}, using removal operators, say R_k, $k = 1.., r_{max}$ resulting in an incomplete solution and therefore not feasible.

(b) reinsert these attributes back into the solution using repair mechanisms, say M_l, $l = 1, ..l_{max}$ where l_{max} denote the possible repair mechanisms that can be used to repair the solution, turning it into a feasible solution again.

In this section, we first provide the general framework of LNS followed by the removal operators and the repair mechanisms that we adopted. Two variants will be investigated using an initial experiment to identify empirically which one is the most promising to be used on larger instances.

7.3.1 Basic LNS algorithm

The main steps of the algorithm are given below. It is also worth stressing that the operators, namely, R_k and M_l, incorporate randomness to avoid cycling in case the incumbent solution X is not changed.

Step 1: Define removal mechanisms $R_k(k = 1..r_{max})$, repair operators $M_k(k = 1, ..m_{max})$ and maximum number of attributes can be removed, N_{max}. Start with a given solution X.

Step 2: While the stopping criteria is not met do the following

2a) Destroy solution X using removal mechanism $R_k(k = 1, ..R_{max})$ by removing N_{max} attributes to get partial solution X'

2b) Insert those N_{max} removed attributes from 2a) using repair operators $M_k(k = 1, ..M_{max})$ to obtain a full solution X''

2c) (optional step) apply local search to find the improved X''

2d) If X'' is better than X set $X = X''$

2e) Go back to step 2 with solution X

7.3.2 The Removal Operators

We have adopted two removal strategies which we denote as the random removal strategy and the guided removal strategy.

Removal Strategy 1(Random Removal Strategy)- In this strategy, attributes to be removed are selected randomly. In this work, we opted to remove $N_{max} = 0.4 * n$ requests randomly.

Removal Strategy 2(Guided Removal Strategy) In this strategy, instead of removing attributes randomly, we follow a rule. We remove the routes whose saving $S_k < 0; k = 1, ..N$. We start from the first $S_k < 0$ and repeat this step until total number of removed requests either reach $N_{max} = 0.4 * n$ or there is no more $S_k < 0$.

7.3.3 The Repair Operator

In this work, we used one repair mechanism, based on the best insertion within a given route but not over all routes, resulting in adopting a first improvement strategy. This is carried out as it is simpler and relatively much faster than the best improvement which would require investigating all routes.

Step 1: Randomly select one of the removed attributes, say request A

Step 2:

2a)Starting from the first incomplete route and insert request A in the best place.

2b) If the solution is improved, insert A in that chosen position and go back to step 1, otherwise go back to step 2a)

2c) If A cannot be inserted in all the incomplete routes, then keep it as singleton.

Step 3: While there are still some removed attribute go back to step 1 otherwise record the new complete solution X' and stop.

7.3.4 Selection between the two removal strategies

We ran the two variants of LNS, one with the random removal strategy (denoted by RMS for short) and the other using the guided removal strategy (GRS). We used the average CPU time of the best variant of VNS (VNS3) as our stopping criterion. Here, we tested the same 10 randomly generated instances of size 50. The detailed results are given in Table D.3 with the average summary results shown in the last row.

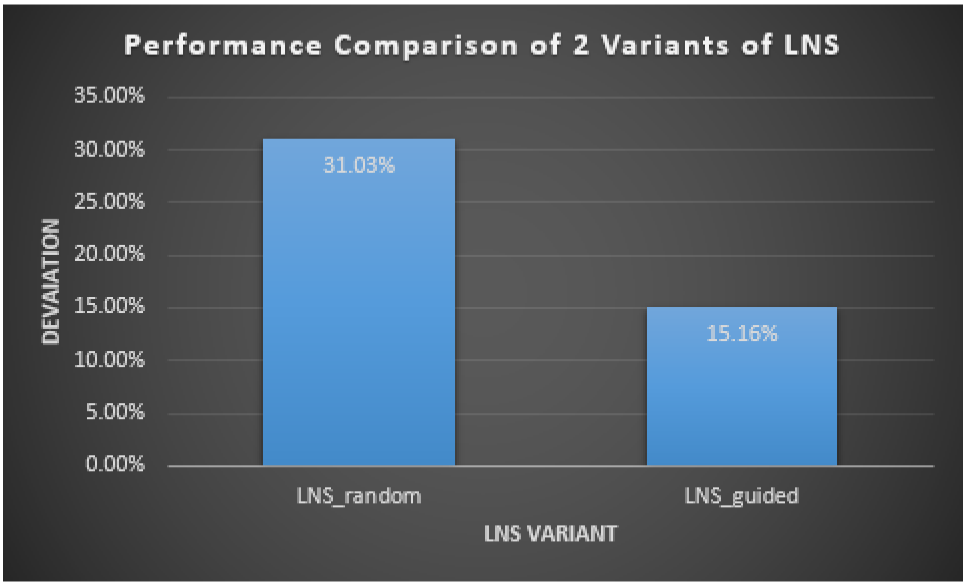


Figure 7.11: Performance Comparison of LNS Variants

It can be shown that the guided variant of the LNS performed relatively much better than the random variant with a massive average improvement of 18.60% (i.e., 100*(808.72-658.26)/808.72). This is also shown in Figure 7.11 where the average of the deviations from the optimal solutions for each variant is displayed (15.16% vs 31.03% for the Guided vs the Random LNS respectively). We therefore adopted the Guided LNS for the rest of the experiments as will be shown in the next section.

7.4 Hybridisation of VNS/LNS

As both the VNS and the LNS have their strengths and weaknesses, it is useful to attempt to combine them while obviously using the same amount of total CPU time. We adopted a simple hybridisation by integrating them in series, starting with VNS. Obviously we could have also started with LNS however as guided variant of LNS's removal mechanism work on the route which have $S_k < 0$, we used VNS first and perturb the solution which obtained after VNS by destroying the routes $S_k < 0$ instead of improving the solution with VNS after LNS's repair mechanism for routes which have S_k. Note that in this study we did not hybridise the two metaheuristics based on useful information which we could have gathered during the search, this could be an interesting research avenue to pursue. Here, the following basic algorithm of the hybrid metaheuristic is given below:

Step 1: Start with a random initial solution X

Step 2: While the stopping criterion is not met perform the following:

2a) Apply VNS-V3 starting with X to get new solution X' and set X=X' (note that the solution never gets worse in VNS)

2b) Apply LNS-guided on X to get X'.

2c) If X' is better than X set $X = X'$.

The performance of this integration is assessed in the computational result section which is given next.

7.5 Computational Results

In this section, we test our methodology on instances with larger size as carried out in the earlier chapter. These refer to size 100, 150 and 200 requests where 10 random instances for each size are tested and their respective average saving recorded. All the algorithms are coded in Visual Studio C++ and executed on an Intel Core i5-7300U CPU @ 2.60GHz 64-bit operating system with 8 GB RAM.

Note that the deviation measure as defined in the previous section using the optimal solution found in the previous chapter is adopted for assessing the performance of each of the metaheuristics.

We first provide a more detailed analysis on the choice of the second VNS stopping criterion, followed by the presentation of two experiments where in experiment 1, the CPU time of the VNS is used whereas in experiment 2, the stopping criterion of VNS based criterion 2 is adopted instead of its corresponding VNS CPU time. These results are provided in the subsections of LNS and Hybrid respectively. Finally, the overall results are then summarised and analysed.

7.5.1 More analysis on the choice of VNS stopping criterion 2

We first provide a summary of empirical result based on these larger instances to reiterate the choice of the second stopping criteria used in VNS which was already assessed using size 50 only. This is carried out by running the best variant of VNS, namely VNS3.

The average deviation results are shown Figure 7.12 and support our earlier choice of opting for stopping criterion 2. Adoption of second stopping stopping criterion 2 instead of stopping criterion 1 obviously decreases the deviation from the optimal solution especially for the larger sizes it performs significantly better. For more information, the detailed results for all instances for each stopping criterion can be found in Appendix C.

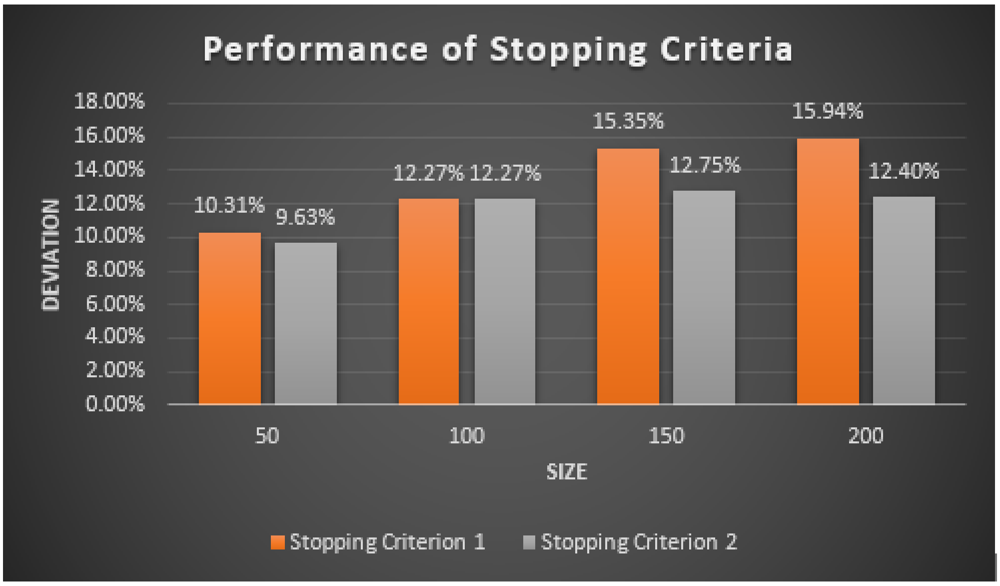


Figure 7.12: Performance Comparison of Stopping Criteria of VNS

7.5.2 VNS3 Criterion 2 results

The detailed results of the VNS3 with criterion 2 which showed to be relatively better as empirically demonstrated in the earlier subsection are also reproduced and presented in Appendix D. and their average results in Table 7.2 though these are also given as part of Appendix C. The results are for n=50,100, 150 and 200 requests where 10 random instances are tested for each size and the average results produced.

Size	TS	Optimal	Deviation(%)
50	862.23	954.70	9.63
100	1955.39	2228.37	12.27
150	3073.70	3523.99	12.75
200	4318.42	4929.47	12.40

Table 7.2: Summary Results of VNS using the second stopping criterion

7.5.3 The LNS results

We already tested the two proposed LNS approaches namely, LNS Random and LNS Guided with the first experiment for size 50 in the previous section and we determined the best variant of LNS as LNS Guided. Here we provide the results of the LNS Guided, using the instances (n=50,100, 150 and 200) based on the two experiments:

Results of LNS using Experiment 1

Here, the CPU time of VNS3 found for each instance is adopted. The LNS Guided stop where the algorithm reaches the VNS-3 CPU time. We conducted first experiment already for size 50 to determine the best variant. Here we conduct the first experiment for other sizes, 100, 150 and 200. The results including the optimal solutions and the corresponding deviations in % are reported in Table D.8. For completeness, the average summary results are also provided for in Table 7.3.

Results of LNS using Experiment 2

To be consistent in terms of comparison between the methods, we also ran the guided variant of LNS with second stopping criterion of VNS. This is used to relate to the number of iterations etc instead of the CPU time used. This comparison can also be seen to be independent from the CPU time of VNS3 but consistent in terms of reasoning behind the rule itself. In addition, this does not require to have the VNS results first which can be a handicup if one is interested to explore LNS or any other method. The average summary results is also provided in Table 7.4 for completeness.

The deviation values of LNS based on second experiment is less than the deviation values based on the first experiment. However, we also need to compare these deviation values against the deviation values from the other methods too. These results are given to present the details of the results. The discussion about the performance of the algorithm compared to the other methods based on the deviation from the optimal and computational time complexity is given in the computational results section.

Size	TS	Optimal	Deviation(%)
50	808.72	954.70	15.16
100	1812.50	2228.37	18.74
150	2844.08	3523.99	19.25
200	3884.48	4929.47	21.20

Table 7.3: Average summary results of LNS: Case of experiment 1

Size	TS	Optimal	Deviation(%)
50	775.10	954.70	18.83
100	1720.56	2228.37	22.73
150	2762.93	3523.99	21.51
200	3741.79	4929.47	24.08

Table 7.4: Average summary results of LNS: Case of experiment 2

7.5.4 The Hybrid VNS/LNS results

We hybridized the VNS and LNS as LNS perturbs the solution comes from VNS and improve with repair mechanisms. Here we provide the results of the hybrid using the same instances under the two experiments related to the VNS3 stopping criteria.

Summary results of hybrid using experiment 1 and experiment 2 are provided in Table 7.5 and Table 7.6 respectively.

The results obtained from hybrid approach is similar to the results obtained from VNS. The detailed analyses and comparison with the other results are given in computational results.

Size	TS	Optimal	Deviation(%)
50	868.13	954.70	9.01
100	1960.07	2228.37	12.04
150	3073.40	3523.99	12.79
200	4274.22	4929.47	13.31

Table 7.5: Average summary of Hybrid: Case of experiment 1

Size	TS	Optimal	Deviation(%)
50	875.41	954.70	8.14
100	1995.07	2228.37	10.47
150	3134.20	3523.99	11.02
200	4232.26	4929.47	14.16

Table 7.6: Average summary of Hybrid: Case of experiment 2

Note that the detailed additional tables are given in Appendix D.

7.5.5 Overall computational Comparison

In summary, as said earlier, we ran 10 randomly generated instances of size 50, 100, 150 and 200 with the chosen methods which are VNS3 with stopping criterion 2, the guided LNS and Hybrid. We carried out two experiments based on stopping criteria where in the first experiment, VNS3 cpu time is used as the stopping criterion for the other two and in the second experiment, the second stopping criterion of VNS, which is related to the rule rather than the time required, is used instead.

Solution Quality

The overall performances of these metaheuristics are measured based on the deviation in % from the optimal solutions found in the earlier chapter.

The detailed results of these methods using experiment 1 and experiments 2 are presented in Appendix D. Note that for VNS it is the same experiment. Also for an overall comparisons based on the average deviation under experiment 1 and experiment 2 are given in Figure 7.13 and Figure 7.14 respectively.

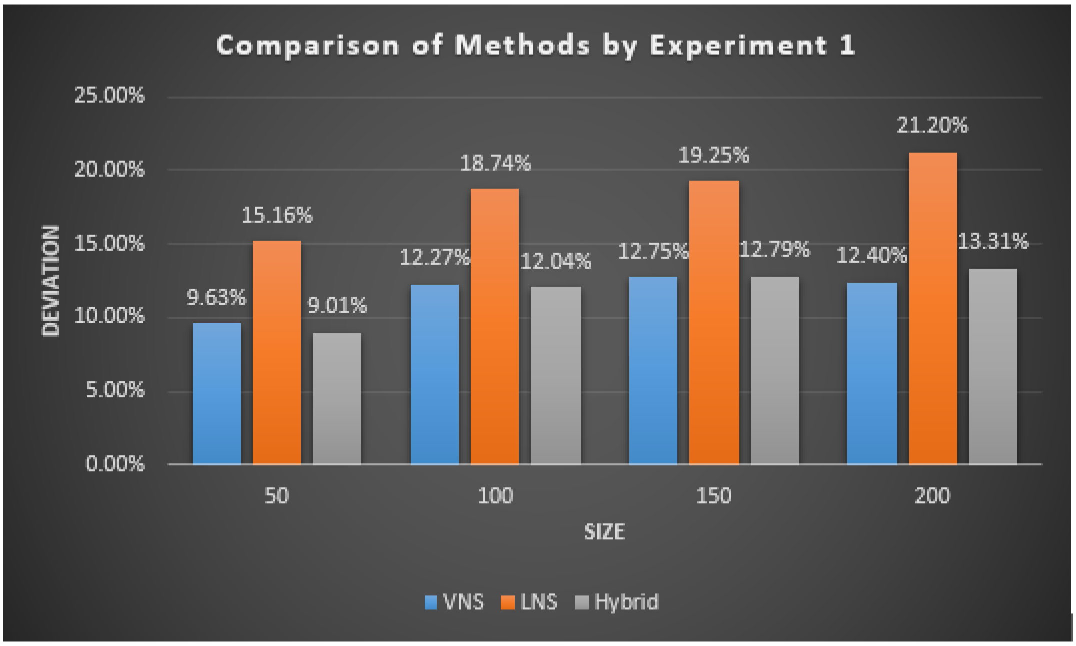


Figure 7.13: Comparison of the Methods by Experiment 1

According to experiment 1 results, both VNS and Hybrid produce similar results and outperform the LNS. However, for size 50 and 100 Hybrid performs slightly better than VNS but for larger instances such as size 150 and 200, VNS performs slightly better than hybrid.

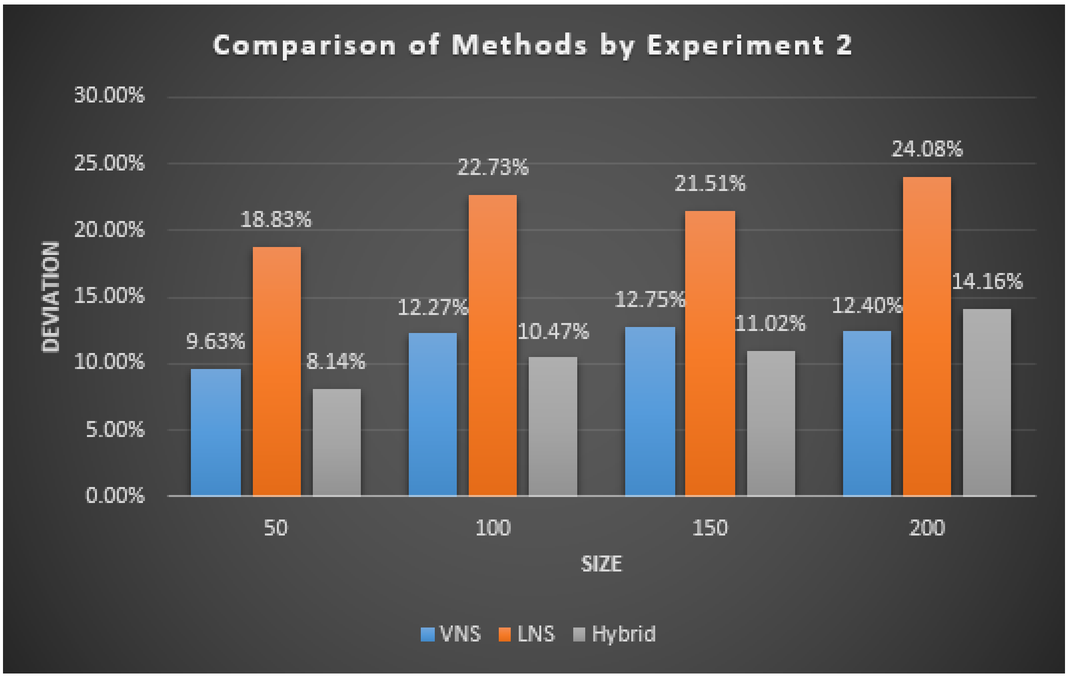


Figure 7.14: Comparison of the Methods by Experiment 2

Results of experiment 2 is not also so different than the results of experiment 1. VNS and Hybrid methods still outperform the LNS. However, for the largest size, say size 200, VNS performs better than the other methods producing the smallest deviation value.

CPU Time of the methods

We also compared the computational time performance of the methods. Note that the computational time of the exact method consists of the computational time of the mathematical model and computational time of the saving generation which is quite large. For the exact methods, savings of all possible consolidation configurations for all requests are computed. However, for the metaheuristics, the saving calculation is carried out only the relevant(candidate) route. The CPU time (in milisecs) comparison of VNS, Hybrid and exact methods are given in Figure 7.15. We only presented the cpu time of VNS and Hybrid from the metaheuristics that we adopted as they were considered to be the best in terms of deviation from the optimal result. It can be shown that the CPU for the exact method is massively larger as the size of the problem increases. Note that the detailed additional tables are given in Appendix E.

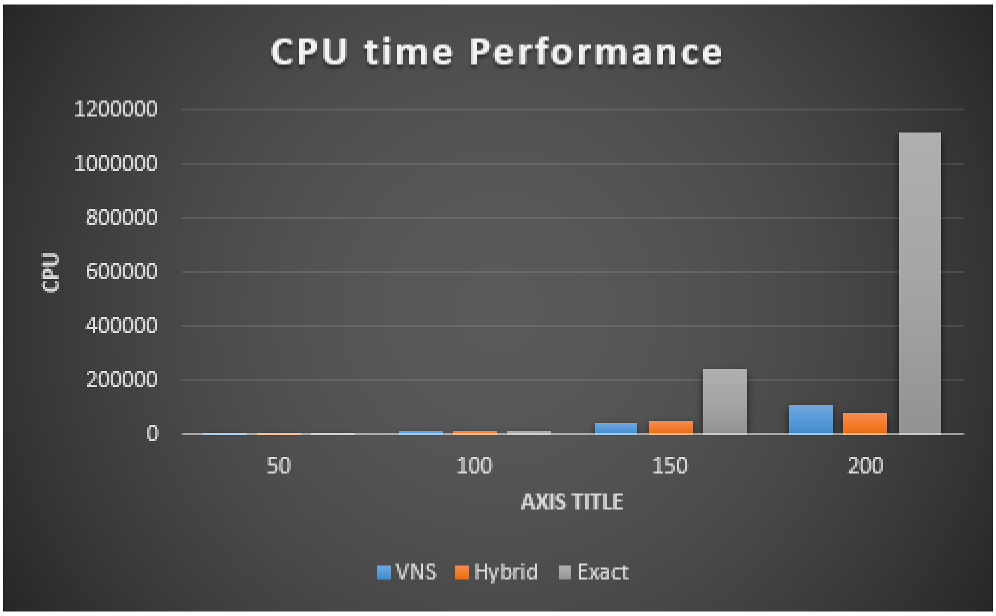


Figure 7.15: CPU Performance of the methods for all Sizes

7.6 Summary

In this chapter, we addressed the consolidation scheduling problem using meta-heuristics, namely, VNS and LNS and their hybridisation. This is mainly carried mainly to respond to the potential increase in CPU time that could be required to solve larger instances optimally. For VNS, appropriate and novel neighbourhood structures and local searches are exploited. Similarly, in LNS we presented removal and repair operators that take into account the structure of the problem. Interesting computational results are also provided and analysed with the VNS and the Hybrid showing better performances. Note the obtained results are not as competitive to the optimal solution in terms of quality though they are obtained in relatively less computational effort. The quality of these metaheuristics could eventually be enhanced as will be highlighted in the next chapter under the section limitations and suggestions

The following and final chapter will summarise our findings, highlight potential limitations of the approaches that we adopted while providing suggestions that could be worth examining.

Chapter 8

Conclusion, Limitations and Suggestions

8.1 Introduction

In this chapter, we summarise the findings which we have achieved in this work, highlight some limitations as well as providing an outline of the potential research avenues that we believe to be worthwhile.

8.2 Conclusion

The thesis is made up of eight chapters. The first one deals with the problem we are examining, the aim and contribution of the work and also outlines the structure of the thesis. Chapter two provides a review on two related areas, namely, supplier selection which here represent the freight logistic companies and freight order consolidation scheduling. In the first area we review those multi-criteria decision approaches that are used independently, then discuss those integrated approaches and then highlight the use of uncertainty that could be tackled through fuzzy set approaches and Bayesian networks. The second part concentrates on freight consolidation scheduling based on exact and heuristic methods. Our first two contributions are provided in the next two chapters, namely chapter three and chapter four. Chapter three deals with the design and analysis of a novel and effective integrated MCDM approach that aims to combine the results of individual ones such as AHP, TOPSIS and VIKOR. Though these are chosen as an example, the approach can be used and adapted for any number of individual ones. A new data set generator is also produced to reflect the various scenarios. The results demonstrate, though empirically, that it is not effective to rely on using one individual MCDM method only. Our findings also supports earlier studies in this area such as the recent survey by Watróbski et al. (2019).

Chapter four treats uncertainty when there is a lack of knowledge. We treat this issue by designing a novel approach that combines MCDM methods, AHP, TOPSIS and DEMATEL with Bayesian Network(BN).In this proposed approach, TOPSIS is used for ranking the suppliers. TOPSIS has two input; weights of the criteria and initial decision matrix. Weights of the criteria is provided by AHP. Initial decision matrix is elicitated by BN. The causal graph of BN is determined by DEMATEL.

Chapter five defines the freight order consolidation scheduling problem. Here we first enumerate all the possible order consolidation configurations for pair and triplets and produce their corresponding saving due to consolidation. A simple 0-1 ILP is also provided here for the case of pairs. The next two chapters present the approaches that we developed to solve this scheduling problem. The first part which is covered in chapter six deals with a development of a set partitioning based approach and provides some interesting results compared to the standard 0-1 formulation. In this chapter we also introduce some new and efficient valid inequalities and create a data set generator to produce ten random instances with 50 to 200 requests with a step size of 50. This data set is used for the empirical testing in this chapter and in the subsequent one as well. The next chapter, namely, chapter seven, treats the same problem but from a heuristic search view point. Though the exact method was fast for small instances, it requires a significant increase in CPU for larger ones. Though optimal solutions could be found, due to the time urgency of the problem, we wanted to examine two powerful and well known meta-heuristics, namely, VNS and LNS. Interesting neighbourhood structures and local searches are designed for VNS, while specially built removal and repair operators are presented for LNS. Some variants for both VNS and LNS are explored and their hybridisation was also empirically examined. Computational performance between the method is carried out showing that the VNS and the hybrid are the best performers. The final chapter, chapter eight, is the current chapter which presents some limitations of the study and highlights several suggestions for future research.

8.3 Limitations/Suggestions

The following limitations are worth discussing. Some are based on methodology whereas others on applications. These shortcomings have also led to some interesting suggestions. For simplicity we organise the discussion under two headers, namely the supplier selection (chapters 3 and 4) and the consolidation scheduling. (Chapters 5, 6 and 7)

Supplier selection part

In this part, other MCDM approaches besides the three that are used to determine the integrated approach can include several other ones. This can result to an extended analysis with more robust results. The criteria that we considered could also be revisited.

In Chapter 3 for example, we showed, through empirical experiments, that it is much better not to rely on one MCDM method only when it comes to supplier selection. This important result reinforces earlier studies that support such arguments including the recent and informative review by Watróbski et al. (2019) who provided an interesting framework to identify a subset of promising methods for a given decision problem instead of one method only. Our approach though deals with the problem from another angle it overcomes the problem by incorporating invaluable information that are gathered by the individual MCDM methods to generate a global result. This also differs from classical integrated methods that aim to construct a new method which combines elements of the individual methods.

One way forward would be to merge the framework of Watróbski et al. (2019) to identify a subset of methods for a given decision problem and then adapt the proposed approach to get even more robust solution from the outcomes of these selected individual methods.

Also in chapter 3, we carried out the elicitation of the weights of criteria from AHP to be consistent in all methods. However, other weighting methods can also be proposed. Besides, we generated the weights associated with the individual methods randomly but in practice this can be explored further. One possibility would be to adopt an ordinal range as low (say [0.05-0.4]), medium (say [0.4, 0.65]) and high (say [0.65, 0.95]). These can be submitted to the decision makers for elicitation of their preferences among the available methods and a random number within these small ranges are chosen instead. The concept of interval preference is an interesting idea that is recently explored by Ahn (2017) for the case of AHP but can be adopted to cater for the individual methods instead. Another way would be to calculate the probability of success of an individual method based on their respective regret based measure results found in Subsection 3.2 or through similar empirical experiments.

The integration of more complex but powerful approaches such as those using evolutionary methods and also mathematical programming could be revisited to incorporate this type of learning which is based on useful information found by other methods (see Salhi (2017)).

The data set used for our supplier selection testing in both chapters 3 and 4 is also small and larger data sets supported by more case studies could enhance the usefulness of both MCDM methods proposed in these two chapters. For instance, in chapter four, the results were based on one case study using three experts from the company. A data gathering that invites more experts or similar type of experts from various companies could provide more insight. The current study is limited to one case study. One way forward would be to identify a set of case studies that have different levels of missed information to assess whether the approach we developed will remain robust. In this work, we used TOPSIS as a ranking method but other MCDM ranking methods such VIKOR, ELECTRE or PROMETHEE can also be adopted. Our approach could also be extended to incorporate invaluable information derived from commonly used statistical techniques on missing values.

Another interesting though challenging avenue would be to incorporate fuzzy information in some of these techniques by considering a mixture of memberships to better represent each of the criteria from a fuzzy environment view point. Exciting research aspects on fuzziness in MCDM that are discussed in the recent informative review chapter by (Zeshui & Zhang 2022) could be worth exploring. One way would be to incorporate some ideas from the interesting study of Rodrigues et al. (2014) where AHP and TOPSIS, are hybridized. Such an investigation would be worthwhile not only for academics but practitioners too.

Consolidation scheduling part

For the consolidation scheduling part, our mathematical formulation is based on set partitioning. This could be limited if the company wishes to incorporate more than triplets. The generation of all combinations especially if more than one transshipment point for a given consolidation is also required, this will increase the size of the problem drastically. However, it is worth stressing that not all combinations will be needed as some could be eliminated due to feasibility constraints and hence developing a scheme to identify these non feasible ones would be worth exploring.

In the meta-heuristic chapter, the number of local searches in VNS could be increased leading to a more powerful local search engine. This would explore a wider search space that could enhance the quality of the results but at the expense of extra CPU. However, this computational burden could be managed if neighbourhood reductions that cut unnecessary computations for non promising moves could be developed. Such implementation could be made even more efficient by introducing suitable data structure that avoid recomputing already found information. These key aspects are highlighted in (Salhi & Thompson 2022*b*) and could be worth investigating.

An adaptive search which uses learning could also be introduced where a subset of local searches will be called for at a given iteration instead. The same observation could be made for LNS where additional removal operators could be introduced and analysed. Here pairs of operators (removal/repairs) could be analysed at a learning stage and then an adaptive LNS can then be used that involves a given pair at a given iteration depending on its probability of success. Some of these ideas could be attempted as successfully implemented by (Sze et al. 2016, 2017) for the case of the classical VRP and the cumulative capacitated VRP respectively.

In this study, the supplier selection and the scheduling tasks can be considered dependent due to the chosen suppliers location and the respective collection points. Here, we assumed that the extra time/cost to the first chosen collection point in any consolidation is a constant regardless of the supplier chosen. This is mainly due to having a large number of available suppliers that are chosen based on earlier supplier selection. If this is not the case, the computation of the configuration for consolidation may be affected by the choice of the suppliers. This aspect can be considered and our approach need to be revisited by reexamining all the configurations between the collection points and the possible suppliers around that collection points. This could yield to the best configuration consolidation between two or three requests being different to the ones we obtained. It is worth noting that this will have no bearing in the methodology behind either the exact method or the metaheuristics though the results may obviously be different due to the new configurations and their respective savings.

Given the urgency of providing quotes to customers, it may be interesting though challenging to examine the effect of probable arrival of new requests using past information. This can lead to offering competitive quotes with the expectation that order requests that are not too far from the paths chosen will materialise making the consolidation relatively much more effective. The incorporation of such stochastic aspect would not only advance the theoretical and academic knowledge but also provide a competitive advantage to those 3PLs that embrace this innovative and data driven approach by putting the resource in collecting as much information as possible so the exploration could be made worthwhile.

Appendices

Appendix A

Contribution to Knowledge

In this appendix we highlight the main contributions as demonstrated through the conferences and the publications.

A.1 Conferences

1. OR61 (September 2019) University of Kent, The Selection of Freight Transportation Suppliers with presence of Scheduling
2. EURO (July 2019) Dublin, A VNS approach for the Freight Transportation Supplier Selection Problem
3. ECCO XXXV –CO 2022 Joint Conference (June 2022) Online Hosted, Mathematical Models and Metaheuristic Approaches for the Order Consolidation Scheduling Problem

A.2 Publications

1. S. Salhi, B. Gutierrez, N. Wassan, S. Wu R. Kaya (2020). An effective real time GRASP-based metaheuristic: Application to order consolidation and dynamic selection of transship-ment points for time-critical freight logistics," Expert Systems With Applications, 158, 1–36. [this has some small parts from chapters 2, 5 and 6]
2. R. Kaya, S.Salhi, V. Spileger. A Novel Integration of MCDM Methods and BN: The Case of Incomplete Expert Knowledge. Paper submitted from Chapter 3 (under review, 2nd revision) [mainly based on chapters 2 and 4]

3. R. Kaya, S.Salhi, V. Spileger. An Effective Integrated Ranking Approach for Multi Criteria Decision Making [submitted], [mainly based on chapters 2 and 3]

4. R.Kaya, S.Salhi, V. Spielgler. Exact Approaches for Order Consolidation Scheduling Problem for Third Party Logistics Companies; in preparation, [based on chapters 5 and 6]

5. R Kaya, S.Salhi and V. Spielgler Metaheuristics for for Order Consolidation Scheduling Problem for Third Party Logistics; in preparation. [based on chapters 5 and 7]

Appendix B

Triplet Consolidation-Additional Configurations

B.1 Consolidation of 3 requests en-route(no transshipment

Case (a-3)

$$C_i \longrightarrow C_k \longrightarrow C_j \;\Rightarrow\; \begin{cases} D_i \longrightarrow D_j \longrightarrow D_k & (1) \\ D_i \longrightarrow D_k \longrightarrow D_j & (2) \\ D_j \longrightarrow D_i \longrightarrow D_k & (3) \\ D_j \longrightarrow D_k \longrightarrow D_i & (4) \\ D_k \longrightarrow D_i \longrightarrow D_j & (5) \\ D_k \longrightarrow D_j \longrightarrow D_i & (6) \end{cases}$$

Figure B.1: En-route consolidation for triplet case (a-3)

Compute all costs for (a-3)

$$\Delta^1_{C_iC_kC_j} = d_{C_iC_k} + d_{C_kC_j} + d_{C_jD_i} + \delta_1$$

$$\Delta^1_{C_iC_kC_j} = \delta^3_0 + \delta_1 + d_{C_jD_i}$$

(as $\delta_0^3 = d_{C_iC_k} + d_{C_kC_j}$)

$$\Delta^2_{C_iC_kC_j} = d_{C_jD_i} + \delta_2 + \delta_0^3$$

$$\Delta^3_{C_iC_kC_j} = d_{C_jD_j} + \delta_3 + \delta_0^3$$

$$\Delta^4_{C_iC_kC_j} = d_{C_jD_j} + \delta_4 + \delta_0^3$$

$$\Delta^5_{C_iC_kC_j} = d_{C_jD_k} + \delta_5 + \delta_0^3$$

$$\Delta^6_{C_iC_kC_j} = d_{C_jD_k} + \delta_6 + \delta_0^3$$

$$T^3_{ijk} = \min_{r=1,\ldots,6} (\Delta^r_{C_iC_kC_j})$$

Case (a-4)

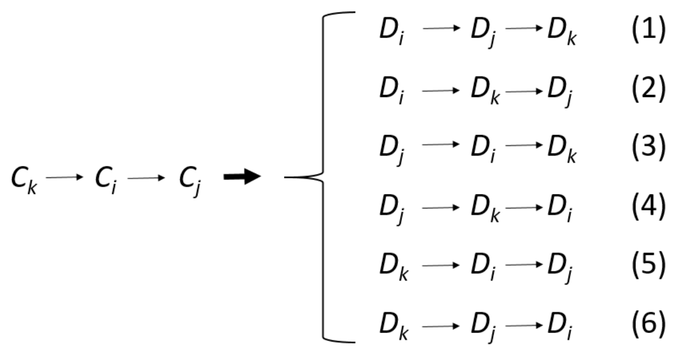


Figure B.2: En-route consolidation for triplet case (a-4)

Case (a-5)

$$C_j \longrightarrow C_k \longrightarrow C_i \quad \Rightarrow \quad \left\{ \begin{array}{ll} D_i \longrightarrow D_j \longrightarrow D_k & (1) \\ D_i \longrightarrow D_k \longrightarrow D_j & (2) \\ D_j \longrightarrow D_i \longrightarrow D_k & (3) \\ D_j \longrightarrow D_k \longrightarrow D_i & (4) \\ D_k \longrightarrow D_i \longrightarrow D_j & (5) \\ D_k \longrightarrow D_j \longrightarrow D_i & (6) \end{array} \right.$$

Figure B.3: En-route consolidation for triplet case (a-5)

Case (a-6)

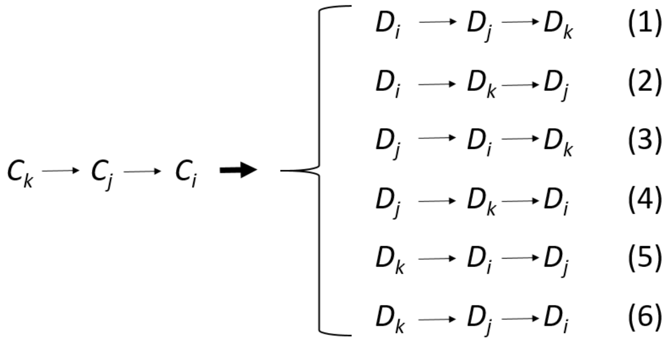


Figure B.4: En-route consolidation for triplet case (a-6)

B.2 Consolidation of Triplets(ijk) with only one transshipment point(T)

b-1) Case of transshipment points before delivery

b-1-2-2)

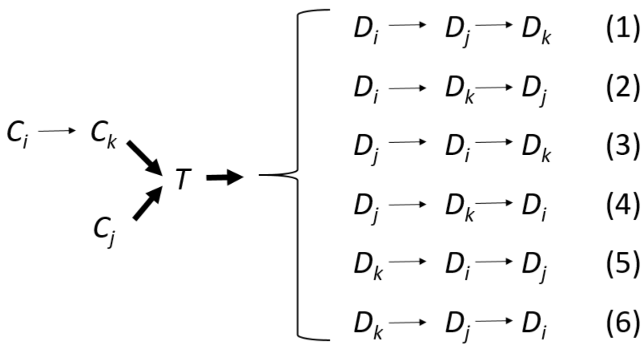


Figure B.5: Triplet case with transshipment of (b-2-2)

$$\Delta_{C_iC_kTC_jT^1} = d_{C_iC_k} + \min_{T \in \Delta_{C_jC_kD_i}} (d_{C_kT} + d_{C_jT} + d_{D_iT}) + \delta_1 = d_{C_iC_k} + \delta_0^{2'} + \delta_1$$

$$\Delta^2_{C_iC_kTC_jT} = d_{C_iC_k} + \delta_0^{2'} + \delta_2$$

$$\Delta^3_{C_iC_kTC_jT} = d_{C_iC_k} + \delta_0^{3'} + \delta_3$$

$$\Delta^4_{C_iC_kTC_jT} = d_{C_iC_k} + \delta_0^{3'} + \delta_4$$

$$\Delta^5_{C_iC_kTC_jT} = d_{C_iC_k} + \delta_0^{4'} + \delta_5$$

$$\Delta^6_{C_iC_kTC_jT} = d_{C_iC_k} + \delta_0^{4'} + \delta_6$$

$$T^{2''}_{ijk} = \min_{r=1,\ldots,6}(\Delta^r_{C_iC_kTC_jT})$$

b-1-2-3)

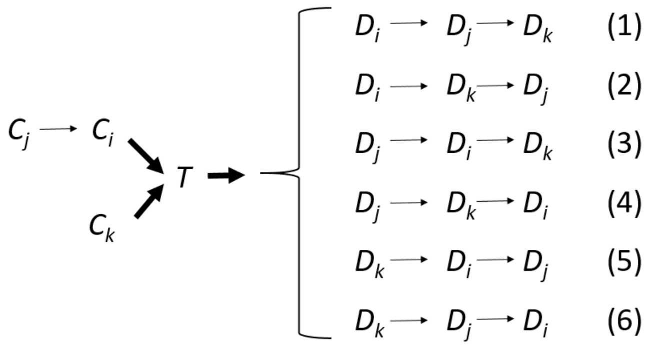


Figure B.6: Triplet case with transshipment of (b-2-3)

$$\Delta^1_{C_jC_iTC_kT} = d_{C_jC_i} + \min_{T\in\Delta_{C_iC_kD_i}}(d_{C_iT} + d_{C_kT} + d_{D_iT}) + \delta_1 = d_{C_jC_i} + \delta_0^{5'} + \delta_1$$

$$\Delta^2_{C_jC_iTC_kT} = d_{C_jC_i} + \min_{T\in\Delta_{C_iC_kD_i}}(d_{C_iT} + d_{C_kT} + d_{D_iT}) + \delta_2 = d_{C_jC_i} + \delta_0^{5'} + \delta_2$$

$$\Delta^3_{C_jC_iTC_kT} = d_{C_jC_i} + \min_{T\in\Delta_{C_iC_kD_j}}(d_{C_iT} + d_{C_kT} + d_{D_jT}) + \delta_3 = d_{C_jC_i} + \delta_0^{6'} + \delta_3$$

$$\Delta^4_{C_jC_iTC_kT} = d_{C_jC_i} + \min_{T\in\Delta_{C_iC_kD_j}}(d_{C_iT} + d_{C_kT} + d_{D_jT}) + \delta_4 = d_{C_jC_i} + \delta_0^{6'} + \delta_4$$

$$\Delta^5_{C_jC_iTC_kT} = d_{C_jC_i} + \min_{T\in\Delta_{C_iC_kD_k}}(d_{C_iT} + d_{C_kT} + d_{D_kT}) + \delta_5 = d_{C_jC_i} + \delta_0^{7'} + \delta_5$$

$$\Delta^6_{C_jC_iTC_kT} = d_{C_jC_i} + \min_{T\in\Delta_{C_iC_kD_k}}(d_{C_iT} + d_{C_kT} + d_{D_kT}) + \delta_6 = d_{C_jC_i} + \delta_0^{7'} + \delta_6$$

$$T^{3''}_{ijk} = \min_{r=1,\dots,6}(\Delta^r_{C_jC_iTC_kT})$$

b-1-2-4)

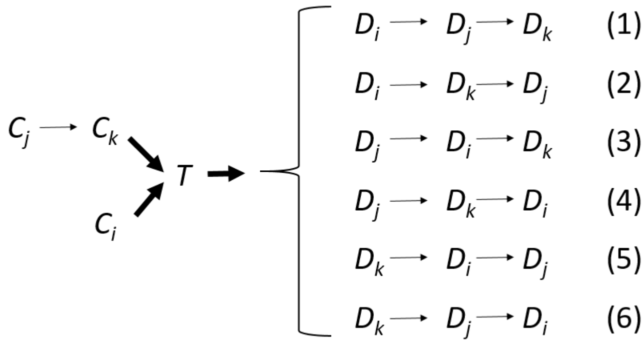


Figure B.7: Triplet case with transshipment of (b-2-4)

$$\Delta^1_{C_jC_kTC_iT} = d_{C_jC_k} + \delta_0^{5'} + \delta_1$$

$$\Delta^2_{C_jC_kTC_iT} = d_{C_jC_k} + \delta_0^{5'} + \delta_2$$

$$\Delta^3_{C_jC_kTC_iT} = d_{C_jC_k} + \delta_0^{6'} + \delta_3$$

$$\Delta^4_{C_jC_kTC_iT} = d_{C_jC_k} + \delta_0^{6'} + \delta_4$$

$$\Delta^5_{C_jC_kTC_iT} = d_{C_jC_k} + \delta_0^{7'} + \delta_5$$

$$\Delta^6_{C_jC_kTC_iT} = d_{C_jC_k} + \delta_0^{7'} + \delta_6$$

$$T^{4''}_{ijk} = \min_{r=1,\ldots,6}(\Delta^r_{C_jC_kTC_iT})$$

b-1-2-5)

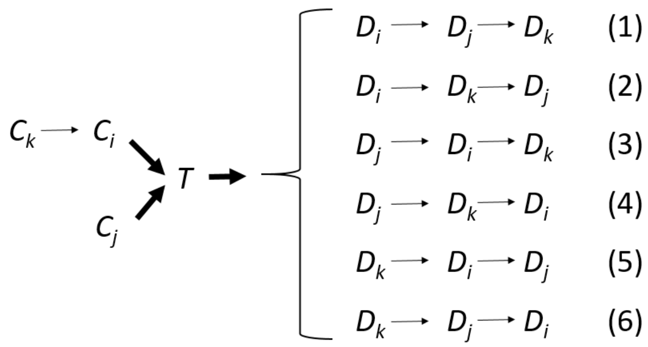


Figure B.8: Triplet case with transshipment of (b-2-5)

$$\Delta^1_{C_kC_iTC_jT} = d_{C_kC_i} + \min_{T\in\Delta_{C_iC_jD_i}} (d_{C_iT} + d_{C_jT} + d_{D_iT}) + \delta_1 = d_{C_kC_i} + \delta_0^{8'} + \delta_1$$

$$\Delta^2_{C_kC_iTC_jT} = d_{C_kC_i} + \min_{T\in\Delta_{C_iC_jD_i}} (d_{C_iT} + d_{C_jT} + d_{D_iT}) + \delta_2 = d_{C_kC_i} + \delta_0^{8'} + \delta_2$$

$$\Delta^3_{C_kC_iTC_jT} = d_{C_kC_i} + \min_{T\in\Delta_{C_iC_jD_j}} (d_{C_iT} + d_{C_jT} + d_{D_jT}) + \delta_3 = d_{C_kC_i} + \delta_0^{9'} + \delta_3$$

$$\Delta^4_{C_kC_iTC_jT} = d_{C_kC_i} + \min_{T\in\Delta_{C_iC_jD_j}} (d_{C_iT} + d_{C_jT} + d_{D_jT}) + \delta_4 = d_{C_kC_i} + \delta_0^{9'} + \delta_4$$

$$\Delta^5_{C_kC_iTC_jT} = d_{C_kC_i} + \min_{T\in\Delta_{C_iC_jD_j}} (d_{C_iT} + d_{C_jT} + d_{D_kT}) + \delta_5 = d_{C_kC_i} + \delta_0^{10'} + \delta_5$$

$$\Delta^5_{C_kC_iTC_jT} = d_{C_kC_i} + \min_{T\in\Delta_{C_iC_jD_j}} (d_{C_iT} + d_{C_jT} + d_{D_kT}) + \delta_6 = d_{C_kC_i} + \delta_0^{10'} + \delta_6$$

$$T^{5''}_{ijk} = \min_{r=1,\ldots,6} (\Delta^r_{C_kC_iTC_jT})$$

b-1-2-6)

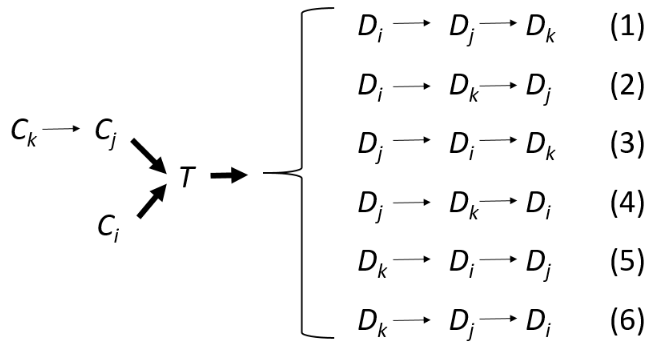


Figure B.9: Triplet case with transshipment of (b-2-6)

$$\Delta^1_{C_kC_jTC_iT} = d_{C_kC_j} + \delta_0^{8'} + \delta_1$$

$$\Delta^2_{C_kC_jTC_iT} = d_{C_kC_j} + \delta_0^{8'} + \delta_2$$

$$\Delta^3_{C_kC_jTC_iT} = d_{C_kC_j} + \delta_0^{9'} + \delta_3$$

$$\Delta^4_{C_kC_jTC_iT} = d_{C_kC_j} + \delta_0^{9'} + \delta_4$$

$$\Delta^5_{C_kC_jTC_iT} = d_{C_kC_j} + \delta_0^{10'} + \delta_5$$

$$\Delta^6_{C_kC_jTC_iT} = d_{C_kC_j} + \delta_0^{10'} + \delta_6$$

(b-2) Case of transshipment after collection points

b-2-2-2)

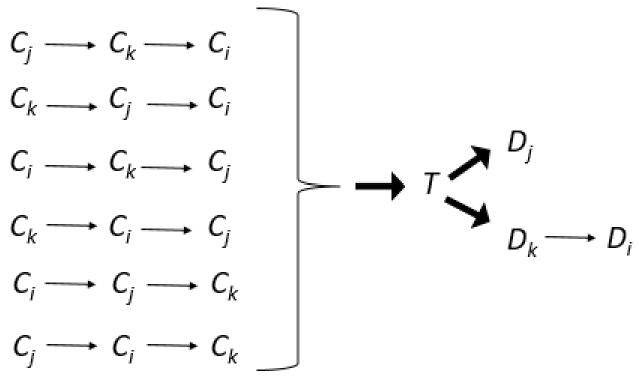


Figure B.10: Triplet case with transshipment of (c-2-2)

$$\Delta^1_{TD_jD_kD_i} = d_{D_kD_i} + \min_{T\in\Delta_{C_iD_kD_j}}(d_{C_iT} + d_{TD_k} + d_{TD_j}) + \delta_1^{'} = d_{D_kD_i} + \delta_0^{2''} + \delta_1^{'}$$

$$\Delta^2_{TD_jD_kD_i} = d_{D_kD_i} + \delta_0^{2''} + \delta_2^{'}$$

$$\Delta^3_{TD_jD_kD_i} = d_{D_kD_i} + \delta_0^{3''} + \delta_3^{'}$$

$$\Delta^4_{TD_jD_kD_i} = d_{D_kD_i} + \delta_0^{3''} + \delta_4^{'}$$

$$\Delta^5_{TD_jD_kD_i} = d_{D_kD_i} + \delta_0^{4''} + \delta_5^{'}$$

$$\Delta^6_{TD_jD_kD_i} = d_{D_kD_i} + \delta_0^{4''} + \delta_6^{'}$$

$$T_{ijk}^{2''''} = \min_{r=1,\ldots,6}(\Delta_{TD_jD_kD_i^r})$$

b-2-2-3)

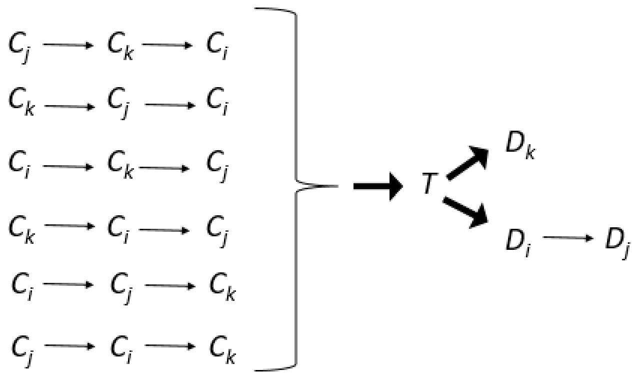


Figure B.11: Triplet case with transshipment of (c-2-3)

$$\Delta^1_{TD_kD_iD_j} = d_{D_iD_j} + \min_{T \in \Delta_{C_iD_kD_i}} (d_{C_iT} + d_{TD_k} + d_{TD_i}) + \delta_1^{'} = d_{D_kD_i} + \delta_0^{5''} + \delta_1^{'}$$

$$\Delta^2_{TD_kD_iD_j} = d_{D_iD_j} + \min_{T \in \Delta_{C_iD_kD_i}} (d_{C_iT} + d_{TD_k} + d_{TD_i}) + \delta_2^{'} = d_{D_kD_i} + \delta_0^{5''} + \delta_2^{'}$$

$$\Delta^3_{TD_kD_iD_j} = d_{D_iD_j} + \min_{T \in \Delta_{C_jD_kD_i}} (d_{C_jT} + d_{TD_k} + d_{TD_i}) + \delta_3^{'} = d_{D_kD_i} + \delta_0^{6''} + \delta_3^{'}$$

$$\Delta^4_{TD_kD_iD_j} = d_{D_iD_j} + \min_{T \in \Delta_{C_jD_kD_i}} (d_{C_jT} + d_{TD_k} + d_{TD_i}) + \delta_4^{'} = d_{D_kD_i} + \delta_0^{6''} + \delta_4^{'}$$

$$\Delta^5_{TD_kD_iD_j} = d_{D_iD_j} + \min_{T \in \Delta_{C_kD_kD_i}} (d_{C_kT} + d_{TD_k} + d_{TD_i}) + \delta_5^{'} = d_{D_kD_i} + \delta_0^{7''} + \delta_5^{'}$$

$$\Delta^6_{TD_kD_iD_j} = d_{D_iD_j} + \min_{T \in \Delta_{C_kD_kD_i}} (d_{C_kT} + d_{TD_k} + d_{TD_i}) + \delta_6^{'} = d_{D_kD_i} + \delta_0^{7''} + \delta_6^{'}$$

$$T^{3''''}_{ijk} = \min_{r=1,\ldots,6} (\Delta_{TD_kD_iD_j^r})$$

b-2-2-4)

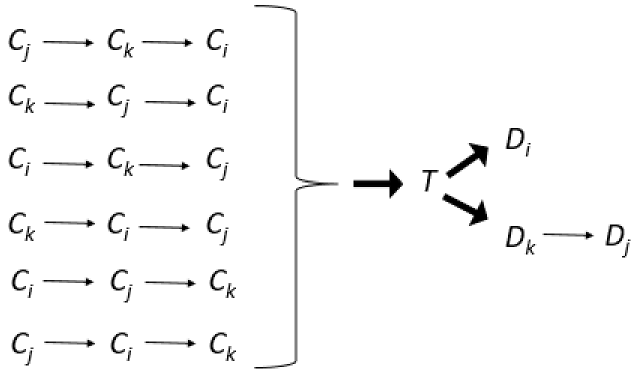


Figure B.12: Triplet case with transshipment of (c-2-4)

$$\Delta^1_{TD_iD_kD_j} = d_{D_kD_j} + \min_{T \in \Delta_{C_iD_kD_i}} (d_{C_iT} + d_{TD_k} + d_{TD_i}) + \delta_1^{'} = d_{D_kD_j} + \delta_0^{5''} + \delta_1^{'}$$

$$\Delta^2_{TD_iD_kD_j} = d_{D_kD_j} + \delta_0^{5''} + \delta_2^{'}$$

$$\Delta^3_{TD_iD_kD_j} = d_{D_kD_j} + \delta_0^{6''} + \delta_3^{'}$$

$$\Delta^4_{TD_iD_kD_j} = d_{D_kD_j} + \delta_0^{6''} + \delta_4^{'}$$

$$\Delta^5_{TD_iD_kD_j} = d_{D_kD_j} + \delta_0^{7''} + \delta_5^{'}$$

$$\Delta^6_{TD_iD_kD_j} = d_{D_kD_j} + \delta_0^{7''} + \delta_6^{'}$$

$$T^{4''''}_{ijk} = \min_{r=1,\ldots,6} (\Delta_{TD_iD_kD_j^r})$$

<u>b-2-2-5)</u>

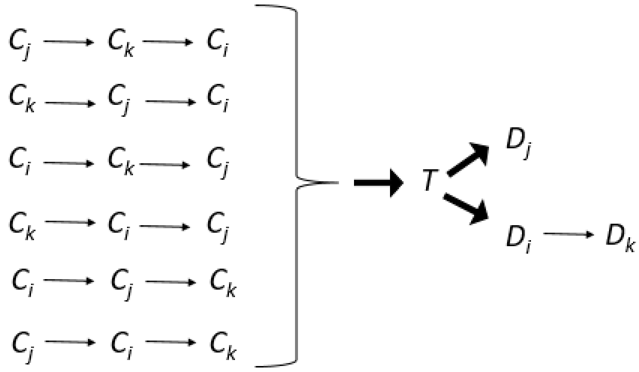


Figure B.13: Triplet case with transshipment of (c-2-5)

$$\Delta^1_{TD_jD_iD_k} = d_{D_iD_k} + \min_{T\in\Delta_{C_iD_jD_i}} (d_{C_iT} + d_{TD_j} + d_{TD_i}) + \delta_1^{'} = d_{D_iD_k} + \delta_0^{8''} + \delta_1^{'}$$

$$\Delta^2_{TD_jD_iD_k} = d_{D_iD_k} + \min_{T\in\Delta_{C_iD_jD_i}} (d_{C_iT} + d_{TD_j} + d_{TD_i}) + \delta_2^{'} = d_{D_iD_k} + \delta_0^{8''} + \delta_2^{'}$$

$$\Delta^3_{TD_jD_iD_k} = d_{D_iD_k} + \min_{T\in\Delta_{C_jD_jD_i}} (d_{C_jT} + d_{TD_j} + d_{TD_i}) + \delta_3^{'} = d_{D_iD_k} + \delta_0^{9''} + \delta_3^{'}$$

$$\Delta^4_{TD_jD_iD_k} = d_{D_iD_k} + \min_{T\in\Delta_{C_jD_jD_i}} (d_{C_jT} + d_{TD_j} + d_{TD_i}) + \delta_4^{'} = d_{D_iD_k} + \delta_0^{9''} + \delta_4^{'}$$

$$\Delta^5_{TD_jD_iD_k} = d_{D_iD_k} + \min_{T\in\Delta_{C_kD_jD_i}} (d_{C_kT} + d_{TD_j} + d_{TD_i}) + \delta_5^{'} = d_{D_iD_k} + \delta_0^{10''} + \delta_5^{'}$$

$$\Delta^6_{TD_jD_iD_k} = d_{D_iD_k} + \min_{T\in\Delta_{C_kD_jD_i}} (d_{C_kT} + d_{TD_j} + d_{TD_i}) + \delta_6^{'} = d_{D_iD_k} + \delta_0^{10''} + \delta_6^{'}$$

$$T_{ijk}^{5''''} = \min_{r=1,\ldots,6} (\Delta_{TD_jD_iD_k^r})$$

b-2-2-6)

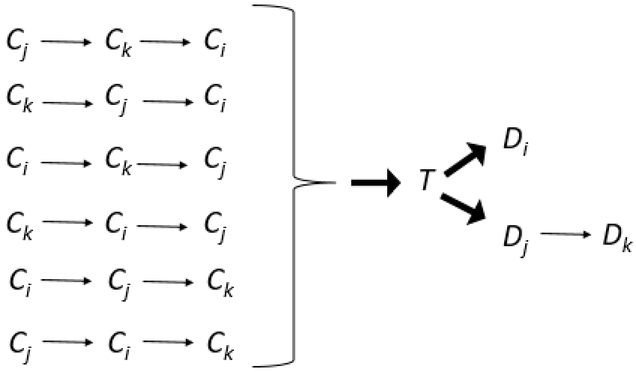


Figure B.14: Triplet case with transshipment of (c-2-6)

$$\Delta^1_{TD_iD_jD_k} = d_{D_jD_k} + \min_{T\in\Delta_{C_iD_iD_j}} (d_{C_iT} + d_{TD_i} + d_{TD_j}) + \delta_1^{'} = d_{D_jD_k} + \delta_0^{8''} + \delta_1^{'}$$

$$\Delta^2_{TD_iD_jD_k} = d_{D_jD_k} + \delta_0^{8''} + \delta_2^{'}$$

$$\Delta^3_{TD_iD_jD_k} = d_{D_jD_k} + \delta_0^{9''} + \delta_3^{'}$$

$$\Delta^4_{TD_iD_jD_k} = d_{D_jD_k} + \delta_0^{9''} + \delta_4^{'}$$

$$\Delta^5_{TD_iD_jD_k} = d_{D_jD_k} + \delta_0^{10''} + \delta_5^{'}$$

$$\Delta^6_{TD_iD_jD_k} = d_{D_jD_k} + \delta_0^{10''} + \delta_6^{'}$$

Appendix C

Additional Tables of Chapter 6

Size 100	**Pair**	**Triplet(Incl Pair)**	**Improvement(%)**
Data Set 1	1525.25	2146.10	40.70
Data Set 2	1695.07	2277.53	34.36
Data Set 3	1935.62	2591.78	33.90
Data Set 4	1473.63	1992.24	35.19
Data Set 5	1694.29	2284.95	34.86
Data Set 6	1823.18	2503.98	37.34
Data Set 7	1651.04	2262.12	37.01
Data Set 8	1369.99	1865.89	36.20
Data Set 9	1707.26	2332.18	36.60
Data Set 10	1523.10	2026.91	33.08
Average	1639.84	2228.37	**35.89**

Table C.1: Paired vs Triplet Consolidation Savings of Size of 100

.

Size 150	**Pair**	**Triplet(Incl Pair)**	**Improvement(%)**
Data Set 1	2622.40	3514.90	34.03
Data Set 2	2534.48	3457.17	36.41
Data Set 3	2904.82	3926.58	35.17
Data Set 4	2619.91	3582.24	36.73
Data Set 5	2639.62	3618.08	37.07
Data Set 6	2395.04	3218.28	34.37
Data Set 7	2744.51	3762.66	37.10
Data Set 8	2517.76	3490.74	38.64
Data Set 9	2483.7	3423.94	37.86
Data Set 10	2381.42	3245.32	36.28
Average	2584.37	3523.99	**36.36**

Table C.2: Paired vs Triplet Consolidation Savings of Size of 150

Size 200	**Pair**	**Triplet(Incl Pair)**	**Improvement(%)**
Data Set 1	3693.61	5046.66	36.63
Data Set 2	3637.89	4678.47	28.60
Data Set 3	3684.87	5135.21	39.36
Data Set 4	3293.50	4857.01	47.47
Data Set 5	3673.58	4663.41	26.94
Data Set 6	3168.36	5042.89	59.16
Data Set 7	3620.08	4890.30	35.09
Data Set 8	3667.24	5012.42	36.68
Data Set 9	3508.98	5040.74	43.65
Data Set 10	3420.54	4927.63	44.06
Average	3536.87	4929.47	**39.37**

Table C.3: Paired vs Triplet Consolidation Savings of Size of 200

	Set Partitioning	Basic Tightening
Size 50	82.1	91.38
Size 100	162.78	170.59
Size 150	302.71	326.48
Size 200	779.67	545.26

Table C.4: CPU time Improvement by Set Partitioning-Basic Tightening

	Set Covering	Set Partitioning	Basic Tightening	Strong Tightening
Size 50	128.52	82.1	91.38	93.1
Size 100	257.46	162.78	170.59	183.75
Size 150	478.37	302.71	326.48	306.79
Size 200	1448.95	779.67	545.26	401.54

Table C.5: CPU time Improvement by Set Partitioning-Strong Tightening

Subset tightenings for size 100 with random numbers for 10 instances are given up to 6 subsets in the following tables.

r_1	1	2	3	4	5	6	7	8	9	10	Average
41	140	165	162	199	241	208	187	149	138	135	172.4
874	122	109	110	106	116	106	115	107	130	114	113.5
1384	266	258	170	97	116	113	105	102	109	111	144.7
1578	111	110	103	112	107	111	103	132	110	115	111.4
1750	130	122	110	113	124	127	111	123	119	124	120.3
2212	179	162	156	151	143	168	157	150	145	175	158.6
3617	114	120	167	309	210	154	142	129	149	142	163.6
4319	128	118	152	145	107	108	108	112	116	106	120
4608	113	115	116	115	115	102	158	198	190	192	141.4
4664	166	97	112	117	114	126	120	122	107	109	119
Overall Average											136.49

Table C.6: CPU with Set Partitioning with Subbasic Tightening-1 for size 100

$r_1 - r_2$	1	2	3	4	5	6	7	8	9	10	Average
41-3617	199	187	195	198	197	223	437	291	244	190	236.1
61-491	109	160	99	107	106	92	111	96	99	95	107.4
755-3395	114	95	112	94	104	97	118	96	104	92	102.6
1384-1750	183	206	213	176	150	168	265	267	154	123	190.5
1578-4608	116	143	147	107	114	110	106	96	116	112	116.7
2212-4664	110	106	123	98	107	98	107	93	111	103	105.6
2995-2042	115	104	110	122	119	164	101	114	99	120	116.8
3481-1977	112	105	107	114	131	103	167	90	99	94	112.2
4319-874	157	114	119	99	138	96	118	92	99	94	112.6
4827-486	106	106	178	137	96	105	93	97	102	133	115.3
Overall Average											131.58

Table C.7: CPU with Set Partitioning with Subbasic Tightening-2 for size 100

$r_1 - r_2 - r_3$	1	2	3	4	5	6	7	8	9	10	Average
41-3617-1384	189	173	170	163	166	174	150	168	172	149	167.4
292-2482-2571	104	104	98	109	103	97	162	131	98	101	110.7
491-2995-2042	116	117	104	98	105	170	137	95	167	161	127
1578-4608-2212	97	122	102	117	105	116	107	108	102	107	108.3
1750-4319-874	184	164	122	127	119	111	111	129	103	120	129
3481-1977-61	92	103	103	104	134	98	166	98	111	112	112.1
3866-4868-95	149	154	106	155	112	118	107	146	105	106	125.8
4664-755-3395	101	112	101	109	109	136	99	116	136	103	112.2
4704-3902-153	109	115	112	119	111	152	99	108	99	100	112.4
4827-486-2691	106	106	106	158	118	119	128	113	115	106	117.5
Overall Average											122.24

Table C.8: CPU with Set Partitioning with Subbasic Tightening-3 for size 100

$r_1 - r_2 - r_3 - r_4$	1	2	3	4	5	6	7	8	9	10	Average
41-3617-1384-1750	122	154	112	177	116	115	116	144	184	147	138.7
292-2482-2571-3866	108	132	148	145	147	171	112	116	92	104	127.5
917-1549-2185-4944	107	177	97	117	97	104	110	112	133	92	114.6
2212-4664-755-3395	190	169	107	100	112	122	107	97	160	115	127.9
2691-4704-3902-153	106	160	106	118	95	98	101	101	93	106	108.4
2995-2042-4827-486	113	165	97	103	90	105	104	122	103	100	110.2
3481-1977-61-491	141	156	196	109	99	96	132	160	131	97	131.7
4319-874-1578-4608	172	128	115	104	116	199	159	136	156	148	143.3
4868-95-497-1926	106	107	108	117	110	165	110	94	89	114	112
4871-1638-1869-112	115	115	98	166	145	108	99	159	95	113	121.3
Overall Average											123.56

Table C.9: CPU with Set Partitioning with Subbasic Tightening-4 for size 100

$r_1 - r_2 - r_3 - r_4 - r_5$	1	2	3	4	5	6	7	8	9	10	Average
41-3617-1384-1750-4319	219	177	189	188	184	184	157	166	145	157	176.6
112-917-1549-2185-4944	103	114	156	194	133	134	204	118	125	155	143.6
491-2995-2042-4827-486	192	200	269	276	236	223	217	217	199	210	223.9
497-1926-4871-1638-1869	117	184	158	135	155	204	195	195	109	191	164.3
755-3395-3481-1977-61	195	231	247	244	134	115	141	158	110	128	170.3
874-1578-4608-2212-4664	221	223	173	172	162	118	159	165	151	169	171.3
2482-2571-3866-4868-95	187	208	215	226	225	148	141	201	204	133	188.8
2691-4704-3902-153-292	300	281	114	123	202	212	201	212	207	180	203.2
3953-4011-1622-633-2823	176	202	214	212	222	172	204	215	207	210	203.4
4664-291-2761-3503-1918	212	208	209	159	207	145	124	107	105	182	165.8
Overall Average											181.12

Table C.10: CPU with Set Partitioning with Subbasic Tightening-5 for size 100

$r_1 - r_2 - r_3 - r_4 - r_5 - r_6$	1	2	3	4	5	6	7	8	9	10	Average
41-3617-1384-1750-4319-874	343	296	237	261	253	359	190	180	150	153	242.2
237-2959-3773-4791-2779-778	211	137	135	162	141	127	208	219	231	176	174.7
292-2482-2571-3866-4868-95	112	130	143	136	147	230	159	122	104	120	140.3
497-1926-4871-1638-1869-112	194	205	110	125	161	175	159	151	152	254	168.6
917-1549-2185-4944-3953-4011	104	113	211	160	134	108	206	109	117	137	139.9
1578-4608-2212-4664-755-3395	136	253	295	188	173	164	163	188	168	234	196.2
1622-633-2823-4664-291-2761	219	234	223	223	134	117	215	193	153	111	182.2
3481-1977-61-491-2995-2042	153	468	125	249	216	240	148	152	285	117	215.3
3503-1918-797-2894-2962-3057	210	182	209	189	111	144	222	212	233	227	193.9
4827-486-2691-4704-3902-153	120	133	150	190	170	214	128	263	133	193	169.4
Overall Average											182.27

Table C.11: CPU with Set Partitioning with Subbasic Tightening-6 for size 100

CPU	Set Partitioning	Basic Tightening	Strong Tightening
Size 50	213.2	228.2	265.3
Size 100	3533.41	3843.93	5009.65
Size 150	21378.88	24135.07	29000.82
Size 200	108482.36	118171.1	133513.26

Table C.12: Comparative CPU Time for Triplet Consolidation

We ran the 10 instances of each size 10 times with set covering formulation for paired consolidation case. The average CPU times of the 10 runs of each sample sizes are given in below tables respectively.

CPU Time	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	Average
Data Set 1	123	156	133	393	423	180	356	124	170	170	222,8
Data Set 2	102	125	142	117	107	113	114	100	77	77	107,4
Data Set 3	106	127	123	116	147	134	125	132	116	110	123,6
Data Set 4	73	86	92	110	129	104	89	93	108	111	99,5
Data Set 5	85	94	98	123	124	117	117	144	123	95	112
Data Set 6	109	103	153	135	136	116	161	134	104	125	127,6
Data Set 7	103	112	155	145	332	327	237	122	133	124	179
Data Set 8	81	74	74	93	118	83	90	78	131	134	95,6
Data Set 9	65	80	86	93	135	81	78	87	100	111	91,6
Data Set 10	104	126	105	158	128	125	122	158	113	122	126,1
Overall Average											**128.52**

Table C.13: CPU Times for Paired Consolidation with Set Covering Formulation of Size of 50

CPU time	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	Average
Data Set 1	268	317	311	275	257	319	287	191	188	206	261,9
Data Set 2	250	272	283	265	277	265	297	198	203	178	248,8
Data Set 3	281	349	321	279	318	502	315	198	207	193	296,3
Data Set 4	294	285	280	234	269	275	236	182	172	165	239,2
Data Set 5	260	339	276	266	287	300	260	198	188	181	255,5
Data Set 6	246	203	196	191	270	326	251	270	254	269	247,6
Data Set 7	366	388	356	388	382	260	235	221	223	226	304,5
Data Set 8	263	304	378	252	278	326	222	194	189	178	258,4
Data Set 9	183	272	215	226	196	246	274	221	195	184	221,2
Data Set 10	190	247	240	236	229	449	265	236	161	159	241,2
Overall Average											**257.46**

Table C.14: CPU Times for Paired Consolidation with Set Covering Formulation of Size of 100

CPU time	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	Average
Data Set 1	417	456	466	516	347	329	320	313	307	297	376,8
Data Set 2	456	498	530	544	375	341	334	333	311	331	405,3
Data Set 3	614	671	630	429	406	394	389	395	381	411	472
Data Set 4	437	538	473	474	336	310	302	310	312	289	378,1
Data Set 5	758	742	666	501	506	541	463	477	470	432	555,6
Data Set 6	437	520	496	566	395	338	346	354	342	321	411,5
Data Set 7	774	745	675	550	522	494	492	510	498	460	572
Data Set 8	716	784	786	539	533	545	655	528	485	449	602
Data Set 9	478	526	524	512	410	356	358	345	319	348	417,6
Data Set 10	736	738	904	544	514	516	485	510	510	471	592,8
Overall Average											**478.37**

Table C.15: CPU Times for Paired Consolidation with Set Covering Formulation of Size of 150

CPU time	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	Average
Data Set 1	1170	2026	2560	1597	1315	1725	1518	1558	1469	1513	1645,1
Data Set 2	1273	1316	1242	1289	1310	1344	1271	1287	1253	1235	1282
Data Set 3	1277	1277	2447	1612	1282	1176	1053	1341	1124	1330	1391,9
Data Set 4	1460	1679	1685	1907	1679	1511	1387	1512	1481	1448	1574,9
Data Set 5	1267	1325	1528	1342	1307	1406	1293	1343	1452	1882	1414,5
Data Set 6	1215	873	940	1135	1274	1175	980	1024	923	970	1050,9
Data Set 7	1306	1375	1657	1869	1692	1593	1526	1537	1685	1675	1591,5
Data Set 8	1857	1846	2099	1900	1734	1887	1693	1669	1639	1767	1809,1
Data Set 9	1615	1576	1606	1470	1224	1929	1772	1435	1481	1228	1533,6
Data Set 10	1092	1254	1265	1331	1243	1179	1190	999	1129	1278	1196
Overall Average											**1448.95**

Table C.16: CPU Times for Paired Consolidation with Set Covering Formulation of Size of 200

Appendix D

Additional Tables from Chapter 7

Saving	Variant 1	Variant 2	Variant 3	Variant 4
50-1	1029.46	1034.40	1052.91	1050.19
50-2	987.66	1000.47	1059.68	1033.22
50-3	750.75	752.62	751.45	720.02
50-4	625.21	614.30	623.41	643.67
50-5	798.42	803.68	824.51	807.05
50-6	778.71	757.85	785.16	764.08
50-7	942.79	938.13	967.00	938.21
50-8	620.78	624.34	654.69	655.60
50-9	865.40	855.74	902.09	871.05
50-10	885.17	900.78	941.70	906.91
Average	828.43	828.23	**856.26**	839.00

Table D.1: Saving of size 50 instances with first stopping rule

Variant	TS	Deviation(%)
LNS random	658.26	31.03
LNS guided	808.72	15.16

Table D.2: LNS comparison

Instance	TS	Optimal	Dev(%)
50-1	725.73	1171.38	38.04
50-2	818.80	1179.08	30.56
50-3	483.12	836.57	42.25
50-4	499.68	706.02	29.23
50-5	645.51	912.36	29.25
50-6	668.43	903.16	25.99
50-7	729.33	1094.44	33.36
50-8	490.60	721.99	32.05
50-9	723.73	972.99	25.62
50-10	797.72	1049.00	23.95
Average	658.26	954.70	31.03

a) LNS Random

Instance	TS	Optimal	Dev(%)
50-1	979.72	1171.38	16.36
50-2	1018.89	1179.08	13.59
50-3	691.04	836.57	17.40
50-4	581.43	706.02	17.65
50-5	752.71	912.36	17.50
50-6	800.92	903.16	11.32
50-7	871.40	1094.44	20.38
50-8	667.71	721.99	7.52
50-9	777.05	972.99	20.14
50-10	946.34	1049.00	9.79
Average	**808.72**	954.70	**15.16**

b) LNS Guided

Table D.3: LNS variants second rule vns cpu for size 50

Instance	TS	Optimal	Dev(%)
50-1	1067.43	1171.38	8.87
50-2	1046.74	1179.08	11.22
50-3	786.63	836.57	5.97
50-4	636.71	706.02	9.82
50-5	807.99	912.36	11.44
50-6	783.44	903.16	13.26
50-7	992.13	1094.44	9.35
50-8	657.74	721.99	8.90
50-9	910.49	972.99	6.42
50-10	933.01	1049.00	11.06
Average	862.23	954.70	9.63

a) Size 50

Instance	TS	Optimal	Dev(%)
100-1	1929.24	2146.10	10.10
100-2	1988.74	2277.53	12.68
100-3	2308.69	2591.78	10.92
100-4	1695.55	1992.24	14.89
100-5	1989.68	2284.95	12.92
100-6	2219.95	2503.98	11.34
100-7	1938.29	2262.12	14.32
100-8	1619.46	1865.89	13.21
100-9	1999.91	2332.18	14.25
100-10	1864.43	2026.91	8.02
Average	1955.39	2228.37	12.27

b) Size 100

Instance	TS	Optimal	Dev(%)
150-1	3147.38	3514.90	10.46
150-2	3046.97	3457.17	11.87
150-3	3336.44	3926.58	15.03
150-4	3102.10	3582.24	13.40
150-5	3059.88	3618.08	15.43
150-6	2808.01	3218.28	12.75
150-7	3357.40	3762.66	10.77
150-8	3066.27	3490.74	12.16
150-9	2940.60	3423.94	14.12
150-10	2871.98	3245.32	11.50
Average	3073.70	3523.99	12.75

c) Size 150

Instance	TS	Optimal	Dev(%)
200-1	4487.64	5046.66	11.08
200-2	3956.66	4678.47	15.43
200-3	4474.55	5135.21	12.87
200-4	4210.55	4857.01	13.31
200-5	4187.20	4663.41	10.21
200-6	4470.62	5042.89	11.35
200-7	4309.52	4890.30	11.88
200-8	4406.52	5012.42	12.09
200-9	4367.02	5040.74	13.37
200-10	4313.92	4927.63	12.45
Average	4318.42	4929.47	12.40

d) Size 200

Table D.6: VNS Results by Second Stopping Criterion for size 50, 100, 150, 200

Instance	TS	Optimal	Dev(%)
50-1	1052.91	1171.38	10.11
50-2	1059.68	1179.08	10.13
50-3	751.45	836.57	10.17
50-4	623.41	706.02	11.70
50-5	824.51	912.36	9.63
50-6	785.16	903.16	13.07
50-7	967.00	1094.44	11.64
50-8	654.69	721.99	9.32
50-9	902.09	972.99	7.29
50-10	941.70	1049.00	10.23
Average	856.26	954.70	10.31

a) Size 50

Instance	TS	Optimal	Dev(%)
100-1	1929.24	2146.1	10.10
100-2	1988.74	2277.53	12.68
100-3	2308.69	2591.78	10.92
100-4	1695.55	1992.24	14.89
100-5	1989.68	2284.95	12.92
100-6	2219.95	2503.98	11.34
100-7	1938.29	2262.12	14.32
100-8	1619.46	1865.89	13.21
100-9	1999.91	2332.18	14.25
100-10	1864.43	2026.91	8.02
Average	1955.39	2228.37	12.27

b) Size 100

Instance	TS	Optimal	Dev(%)
150-1	2984.30	3514.90	15. 10
150-2	2902.36	3457.17	16.05
150-3	3244.38	3926.58	17.37
150-4	3118.07	3582.24	12.96
150-5	3077.8	3618.08	14.93
150-6	2604.76	3218.28	19.06
150-7	3335.30	3762.66	11.36
150-8	3122.25	3490.74	10.56
150-9	2787.91	3423.94	18.58
150-10	2676.98	3245.32	17.51
Average	2985.41	3523.99	15.35

c) Size 150

Instance	TS	Optimal	Dev(%)
200-1	4245.23	5046.66	15.88
200-2	4019.10	4678.47	14.09
200-3	4428.12	5135.21	13.77
200-4	3961.86	4857.01	18.43
200-5	3892.52	4663.41	16.53
200-6	4093.13	5042.89	18.83
200-7	4177.55	4890.30	14.57
200-8	4213.08	5012.42	15.95
200-9	4182.71	5040.74	17.02
200-10	4224.42	4927.63	14.27
Average	4143.77	4929.47	15.94

d) Size 200

Table D.4: VNS Results by First Stopping Criterion for size 100, 150, 200

Size	TS	Optimal	Dev(%)
50	856.26	954.70	10.31
100	1955.39	2228.37	12.27
150	2985.41	3523.99	15.35
200	4143.77	4929.47	15.94

Table D.5: Summary Results of VNS using the first stopping criterion

Size	TS	Optimal	Deviation(%)
50	862.23	954.70	9.63
100	1955.39	2228.37	12.27
150	3073.70	3523.99	12.75
200	4318.42	4929.47	12.40

Table D.7: Summary Results of VNS using the second stopping criterion

Instance	TS	Optimal	Dev(%)
50-1	979.72	1171.38	16.36
50-2	1018.89	1179.08	13.59
50-3	691.04	836.57	17.40
50-4	581.43	706.02	17.65
50-5	752.71	912.36	17.50
50-6	800.92	903.16	11.32
50-7	871.40	1094.44	20.38
50-8	667.71	721.99	7.52
50-9	777.05	972.99	20.14
50-10	946.34	1049.00	9.79
Average	808.72	954.70	15.16

a) Size 50

Instance	TS	Optimal	Dev(%)
100-1	1772.26	2146.10	17.42
100-2	1843.63	2277.53	19.05
100-3	2126.37	2591.78	17.96
100-4	1567.89	1992.24	21.30
100-5	1762.61	2284.95	22.86
100-6	2161.04	2503.98	13.70
100-7	1803.54	2262.12	20.27
100-8	1541.63	1865.89	17.38
100-9	1922.53	2332.18	17.57
100-10	1623.54	2026.91	19.90
Average	1812.50	2228.37	18.74

b) Size 100

Instance	TS	Optimal	Dev(%)
150-1	2842.35	3514.90	19.13
150-2	2785.48	3457.17	19.43
150-3	3113.26	3926.58	20.71
150-4	2906.00	3582.24	18.88
150-5	2942.76	3618.08	18.67
150-6	2691.13	3218.28	16.38
150-7	3051.59	3762.66	18.90
150-8	2689.29	3490.74	22.96
150-9	2726.53	3423.94	20.37
150-10	2692.45	3245.32	17.04
Average	2844.08	3523.99	19.25

c) Size 150

Instance	TS	Optimal	Dev(%)
200-1	3875.44	5046.66	23.21
200-2	3665.74	4678.47	21.65
200-3	4137.25	5135.21	19.43
200-4	3735.78	4857.01	23.08
200-5	3738.18	4663.41	19.84
200-6	3925.13	5042.89	22.17
200-7	3799.90	4890.30	22.30
200-8	4125.04	5012.42	17.70
200-9	3842.82	5040.74	23.76
200-10	3999.55	4927.63	18.83
Average	3884.48	4929.47	21.20

d) Size 200

Table D.8: LNS results for large instances- Case of experiment 1

Instance	TS	Optimal	Dev(%)
50-1	942.68	1171.38	19.52
50-2	1018.07	1179.08	13.66
50-3	698.69	836.57	16.48
50-4	573.49	706.02	18.77
50-5	759.77	912.36	16.72
50-6	638.86	903.16	29.26
50-7	816.21	1094.44	25.42
50-8	596.93	721.99	17.32
50-9	828.67	972.99	14.83
50-10	877.60	1049.00	16.34
Average	775.10	954.70	18.83

a) Size 50

Instance	TS	Optimal	Dev(%)
100-1	1614.71	2146.10	24.76
100-2	1688.88	2277.53	25.85
100-3	2103.41	2591.78	18.84
100-4	1698.20	1992.24	14.76
100-5	1769.59	2284.95	22.55
100-6	1905.31	2503.98	23.91
100-7	1648.01	2262.12	27.15
100-8	1390.63	1865.89	25.47
100-9	1722.30	2332.18	26.15
100-10	1664.52	2026.91	17.88
Average	1720.56	2228.37	22.73

b) Size 100

Instance	TS	Optimal	Dev(%)
150-1	2758.92	3514.90	21.51
150-2	2742.06	3457.17	20.68
150-3	3004.98	3926.58	23.47
150-4	2711.03	3582.24	24.32
150-5	2963.49	3618.08	18.09
150-6	2590.72	3218.28	19.50
150-7	2798.82	3762.66	25.62
150-8	2763.48	3490.74	20.83
150-9	2676.28	3423.94	21.84
150-10	2619.56	3245.32	19.28
Average	2762.93	3523.99	21.51

c) Size 150

Instance	TS	Optimal	Dev(%)
200-1	3707.25	5046.66	26.54
200-2	3598.42	4678.47	23.09
200-3	3945.24	5135.21	23.17
200-4	3681.62	4857.01	24.20
200-5	3587.01	4663.41	23.08
200-6	3653.34	5042.89	27.55
200-7	3684.77	4890.30	24.65
200-8	3832.52	5012.42	23.54
200-9	3928.33	5040.74	22.07
200-10	3799.35	4927.63	22.90
Average	3741.79	4929.47	24.08

d) Size 200

Table D.9: LNS Results on large instances: Case of Experiment 2

Instance	TS	Optimal	Dev(%)
50-1	1076.63	1171.38	8.09
50-2	1081.82	1179.08	8.25
50-3	788.96	836.57	5.69
50-4	659.29	706.02	6.62
50-5	830.88	912.36	8.93
50-6	784.87	903.16	13.10
50-7	974.98	1094.44	10.92
50-8	685.57	721.99	5.05
50-9	918.95	972.99	5.55
50-10	952.12	1049.00	9.24
Average	875.41	954.70	8.14

a) Size 50

Instance	TS	Optimal	Dev(%)
100-1	1990.62	2146.10	7.24
100-2	2061.70	2277.53	9.48
100-3	2327.27	2591.78	10.21
100-4	1782.11	1992.24	10.55
100-5	1982.80	2284.95	13.22
100-6	2250.96	2503.98	10.10
100-7	2041.18	2262.12	9.77
100-8	1626.99	1865.89	12.80
100-9	2026.11	2332.18	13.12
100-10	1861.00	2026.91	8.19
Average	1995.07	2228.37	10.47

b) Size 100

Instance	TS	Optimal	Dev(%)
150-1	3183.43	3514.90	9.43
150-2	3210.68	3457.17	7.13
150-3	3413.18	3926.58	13.07
150-4	3104.11	3582.24	13.35
150-5	3185.93	3618.08	11.94
150-6	2849.65	3218.28	11.45
150-7	3346.9	3762.66	11.05
150-8	3046.06	3490.74	12.74
150-9	3068.55	3423.94	10.38
150-10	2933.54	3245.32	9.61
Average	3134.20	3523.99	11.02

c) Size 150

Instance	TS	Optimal	Dev(%)
200-1	4214.03	5046.66	16.50
200-2	3974.13	4678.47	15.05
200-3	4540.53	5135.21	11.58
200-4	4200.80	4857.01	13.51
200-5	3926.89	4663.41	15.79
200-6	4370.62	5042.89	13.33
200-7	4281.23	4890.30	12.45
200-8	4238.73	5012.42	15.44
200-9	4326.69	5040.74	14.17
200-10	4248.98	4927.63	13.77
Average	4232.26	4929.47	14.16

d) Size 200

Table D.10: Hybrid Results for larger instances: Case of experiment 2

Appendix E

Detail Results of Experiment 1 and Experiment 2- All methods

E.1 Results with Experiment 1

Size 50		VNS-3		LNS-Guided		Hybrid	
Instance	Optimal	TS	Dev(%)	TS	Dev(%)	TS	Dev(%)
50-1	1171.38	1067.43	8.87	979.72	16.36	1089.70	6.97
50-2	1179.08	1046.74	11.22	1018.89	13.59	1076.97	8.66
50-3	836.57	786.63	5.97	691.04	17.40	797.32	4.69
50-4	706.02	636.71	9.82	581.43	17.65	662.32	6.19
50-5	912.36	807.99	11.44	752.71	17.50	795.67	12.79
50-6	903.16	783.44	13.26	800.92	11.32	761.80	15.65
50-7	1094.44	992.13	9.35	871.40	20.38	985.13	9.99
50-8	721.99	657.74	8.90	667.71	7.52	660.37	8.53
50-9	972.99	910.49	6.42	777.05	20.14	918.76	5.57
50-10	1049.00	933.01	11.06	946.34	9.79	933.25	11.03
Average	954.70	862.23	9.63	808.72	15.16	868.13	9.01

Table E.1: Summary Results of All Methods with Experiment 1 for Size 50

Size 100		VNS-3		LNS-Guided		Hybrid	
Instance	Optimal	TS	Dev(%)	TS	Dev(%)	TS	Dev(%)
100-1	2146.10	1929.24	10.10	1772.26	17.42	1889.59	11.95
100-2	2277.53	1988.74	12.68	1843.63	19.05	1966.32	13.66
100-3	2591.78	2308.69	10.92	2126.37	17.96	2324.92	10.30
100-4	1992.24	1695.55	14.89	1567.89	21.30	1806.14	9.34
100-5	2284.95	1989.68	12.92	1762.61	22.86	1970.91	13.74
100-6	2503.98	2219.95	11.34	2161.04	13.70	2124.38	15.16
100-7	2262.12	1938.29	14.32	1803.54	20.27	2030.44	10.24
100-8	1865.89	1619.46	13.21	1541.63	17.38	1643.99	11.89
100-9	2332.18	1999.91	14.25	1922.53	17.57	2044.24	12.35
100-10	2026.91	1864.43	8.02	1623.54	19.90	1799.80	11.20
Average	2228.37	1955.39	12.27	1812.50	18.74	1960.07	12.04

Table E.2: Summary Results of All Methods with Experiment 1 for Size 100

Size 150		VNS-3		LNS-Guided		Hybrid	
Instance	Optimal	TS	Dev(%)	TS	Dev(%)	TS	Dev(%)
150-1	3514.90	3147.38	10.46	2842.35	19.13	3085.97	12.20
150-2	3457.17	3046.97	11.87	2785.48	19.43	2978.89	13.83
150-3	3926.58	3336.44	15.03	3113.26	20.71	3361.70	14.39
150-4	3582.24	3102.10	13.40	2906.00	18.88	3161.51	11.74
150-5	3618.08	3059.88	15.43	2942.76	18.67	3211.40	11.24
150-6	3218.28	2808.01	12.75	2691.13	16.38	2741.14	14.83
150-7	3762.66	3357.40	10.77	3051.59	18.90	3317.64	11.83
150-8	3490.74	3066.27	12.16	2689.29	22.96	3038.09	12.97
150-9	3423.94	2940.60	14.12	2726.53	20.37	2955.27	13.69
150-10	3245.32	2871.98	11.50	2692.45	17.04	2882.42	11.18
Average	3523.99	3073.70	12.75	2844.08	19.25	3073.40	12.79

Table E.3: Summary Results of All Methods with Experiment 1 for Size 150

Size 200		VNS-3		LNS-Guided		Hybrid	
Instance	Optimal	TS	Dev(%)	TS	Dev(%)	TS	Dev(%)
200-1	5046.66	4487.64	11.08	3875.44	23.21	4356.33	13.68
200-2	4678.47	3956.66	15.43	3665.74	21.65	4007.34	14.35
200-3	5135.21	4474.55	12.87	4137.25	19.43	4480.02	12.76
200-4	4857.01	4210.55	13.31	3735.78	23.08	4229.15	12.93
200-5	4663.41	4187.20	10.21	3738.18	19.84	4000.81	14.21
200-6	5042.89	4470.62	11.35	3925.13	22.17	4491.42	10.94
200-7	4890.30	4309.52	11.88	3799.90	22.30	4232.55	13.45
200-8	5012.42	4406.52	12.09	4125.04	17.70	4216.76	15.87
200-9	5040.74	4367.02	13.37	3842.82	23.76	4456.47	11.59
200-10	4927.63	4313.92	12.45	3999.55	18.83	4271.36	13.32
Average	4929.47	4318.42	12.40	3884.48	21.20	4274.22	13.31

Table E.4: Summary Results of All Methods with Experiment 1 for Size 200

E.2 Results with Experiment 2

Size 50		VNS-3		LNS-Guided		Hybrid	
Instance	Optimal	TS	Dev(%)	TS	Dev(%)	TS	Dev(%)
50-1	1171.38	1067.43	8.87	942.68	19.52	1076.63	8.09
50-2	1179.08	1046.74	11.22	1018.07	13.66	1081.82	8.25
50-3	836.57	786.63	5.97	698.69	16.48	788.96	5.69
50-4	706.02	636.71	9.82	573.49	18.77	659.29	6.62
50-5	912.36	807.99	11.44	759.77	16.72	830.88	8.93
50-6	903.16	783.44	13.26	638.86	29.26	784.87	13.10
50-7	1094.44	992.13	9.35	816.21	25.42	974.98	10.92
50-8	721.99	657.74	8.90	596.93	17.32	685.57	5.05
50-9	972.99	910.49	6.42	828.67	14.83	918.95	5.55
50-10	1049.00	933.01	11.06	877.60	16.34	952.12	9.24
Average	954.70	862.23	9.63	775.10	18.83	875.41	8.14

Table E.5: Summary Results of All Methods with Experiment 2 for Size 50

Size 100		VNS-3		LNS-Guided		Hybrid	
Instance	Optimal	TS	Dev(%)	TS	Dev(%)	TS	Dev(%)
100-1	2146.10	1929.24	10.10	1614.71	24.76	1990.62	7.24
100-2	2277.53	1988.74	12.68	1688.88	25.85	2061.70	9.48
100-3	2591.78	2308.69	10.92	2103.41	18.84	2327.27	10.21
100-4	1992.24	1695.55	14.89	1698.20	14.76	1782.11	10.55
100-5	2284.95	1989.68	12.92	1769.59	22.55	1982.80	13.22
100-6	2503.98	2219.95	11.34	1905.31	23.91	2250.96	10.10
100-7	2262.12	1938.29	14.32	1648.01	27.15	2041.18	9.77
100-8	1865.89	1619.46	13.21	1390.63	25.47	1626.99	12.80
100-9	2332.18	1999.91	14.25	1722.30	26.15	2026.11	13.12
100-10	2026.91	1864.43	8.02	1664.52	17.88	1861.00	8.19
Average	2228.37	1955.39	12.27	1720.56	22.73	1995.07	10.47

Table E.6: Summary Results of All Methods with Experiment 2 for Size 100

Size 150		VNS-3		LNS-Guided		Hybrid	
Instance	Optimal	TS	Dev(%)	TS	Dev(%)	TS	Dev(%)
150-1	3514.90	3147.38	10.46	2758.92	21.51	3183.43	9.43
150-2	3457.17	3046.97	11.87	2742.06	20.68	3210.68	7.13
150-3	3926.58	3336.44	15.03	3004.98	23.47	3413.18	13.07
150-4	3582.24	3102.10	13.40	2711.03	24.32	3104.11	13.35
150-5	3618.08	3059.88	15.43	2963.49	18.09	3185.93	11.94
150-6	3218.28	2808.01	12.75	2590.72	19.50	2849.65	11.45
150-7	3762.66	3357.40	10.77	2798.82	25.62	3346.90	11.05
150-8	3490.74	3066.27	12.16	2763.48	20.83	3046.06	12.74
150-9	3423.94	2940.60	14.12	2676.28	21.84	3068.55	10.38
150-10	3245.32	2871.98	11.50	2619.56	19.28	2933.54	9.61
Average	3523.99	3073.70	12.75	2762.93	21.51	3134.20	11.02

Table E.7: Summary Results of All Methods with Experiment 2 for Size 150

Size 200		VNS-3		LNS-Guided		Hybrid	
Instance	Optimal	TS	Dev(%)	TS	Dev(%)	TS	Dev(%)
200-1	5046.66	4487.64	11.08	3707.25	26.54	4214.03	16.50
200-2	4678.47	3956.66	15.43	3598.42	23.09	3974.13	15.05
200-3	5135.21	4474.55	12.87	3945.24	23.17	4540.53	11.58
200-4	4857.01	4210.55	13.31	3681.62	24.20	4200.80	13.51
200-5	4663.41	4187.20	10.21	3587.01	23.08	3926.89	15.79
200-6	5042.89	4470.62	11.35	3653.34	27.55	4370.62	13.33
200-7	4890.30	4309.52	11.88	3684.77	24.65	4281.23	12.45
200-8	5012.42	4406.52	12.09	3832.52	23.54	4238.73	15.44
200-9	5040.74	4367.02	13.37	3928.33	22.07	4326.69	14.17
200-10	4927.63	4313.92	12.45	3799.35	22.90	4248.98	13.77
Average	4929.47	4318.42	12.40	3741.79	24.08	4232.26	14.16

Table E.8: Summary Results of All Methods with Experiment 2 for Size 200

Bibliography

Abdel-Basset, M., Saleh, M., Gamal, A. & Smarandache, F. (2019), 'An approach of TOPSIS technique for developing supplier selection with group decision making under type-2 neutrosophic number', *Applied Soft Computing Journal* **77**, 438–452.
URL: *https://doi.org/10.1016/j.asoc.2019.01.035*

Aguezzoul, A. (2014), 'Third-party logistics selection problem : A literature review on criteria', *Omega* **49**, 69–78.
URL: *http://dx.doi.org/10.1016/j.omega.2014.05.009*

Ahn, B. S. (2017), 'The analytic hierarchy process with interval preference statements $', *Omega* **67**, 177–185.
URL: *http://dx.doi.org/10.1016/j.omega.2016.05.004*

Akgün, & Erdal, H. (2019), 'Solving an ammunition distribution network design problem using multi-objective mathematical modeling, combined AHP-TOPSIS, and GIS', *Computers and Industrial Engineering* **129**, 512–528.

Alinaghian, M. & Shokouhi, N. (2018), 'Multi-depot multi-compartment vehicle routing problem, solved by a hybrid adaptive large neighborhood search', *Omega (United Kingdom)* **76**, 85–99.
URL: *https://doi.org/10.1016/j.omega.2017.05.002*

Allahyari, S., Salari, M. & Vigo, D. (2015), 'A hybrid metaheuristic algorithm for the multi-depot covering tour vehicle routing problem', *European Journal of Operational Research* **242**(3), 756–768.
URL: *http://dx.doi.org/10.1016/j.ejor.2014.10.048*

Amin, S., Haeri, S. & Rezaei, J. (2019), 'A grey-based green supplier selection model for uncertain environments', *Journal of Cleaner Production* **221**, 768–784.
URL: *https://doi.org/10.1016/j.jclepro.2019.02.193*

Anand, N., van Duin, R. & Tavasszy, L. (2021), ‘Carbon credits and urban freight consolidation: An experiment using agent based simulation’, *Research in Transportation Economics* **85**(November 2019), 100797.
URL: *https://doi.org/10.1016/j.retrec.2019.100797*

Baldacci, R., Christofides, N. & Mingozzi, A. (2008), ‘An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts’, *Mathematical Programming* **115**(2), 351–385.

Balinski, M. L., & Quandt, R. E. (1964), ‘On an integer program for a delivery problem’, *Operations research* **12**(2), 300–304.

Baños, R., Ortega, J., Gil, C., Fernández, A. & De Toro, F. (2013), ‘A Simulated Annealing-based parallel multi-objective approach to vehicle routing problems with time windows’, *Expert Systems with Applications* **40**(5), 1696–1707.

Baykasoglu, A., Kaplanoglu, V., Durmusoglu, Z. D. U. & Sahin, C. (2013), ‘Integrating fuzzy DEMATEL and fuzzy hierarchical TOPSIS methods for truck selection’, *Expert Systems with Applications* **40**, 899–907.

Brandão, J. (2020), ‘A memory-based iterated local search algorithm for the multi-depot open vehicle routing problem’, *European Journal of Operational Research* **284**, 559–571.

Büyüközkan, G. & Ifi, G. (2012), ‘A novel hybrid MCDM approach based on fuzzy DEMATEL, fuzzy ANP and fuzzy TOPSIS to evaluate green suppliers’, *Expert Systems with Applications* **39**(3), 3000–3011.

Çapar, I. (2013), ‘Joint shipment consolidation and inventory decisions in a two-stage distribution system’, *Computers and Industrial Engineering* **66**(4), 1025–1035.

Chai, J., Liu, J. N. K. & Ngai, E. W. T. (2013), ‘Application of decision-making techniques in supplier selection: A systematic review of literature’, *Expert Systems with Applications* **40**(10), 3872–3885.
URL: *http://dx.doi.org/10.1016/j.eswa.2012.12.040*

Chang, B., Chang, C. W. & Wu, C. H. (2011), ‘Fuzzy DEMATEL method for developing supplier selection criteria’, *Expert Systems with Applications* **38**(3), 1850–1858.
URL: *http://dx.doi.org/10.1016/j.eswa.2010.07.114*

Chang, K. H. (2019), 'A novel supplier selection method that integrates the intuitionistic fuzzy weighted averaging method and a soft set with imprecise data', *Annals of Operations Research* **272**(1-2), 139–157.
URL: *https://doi.org/10.1007/s10479-017-2718-6*

Che, Z. H., Chiang, T. A. & Lin, T. T. (2021), 'A multi-objective genetic algorithm for assembly planning and supplier selection with capacity constraints', *Applied Soft Computing* **101**, 107030.
URL: *https://doi.org/10.1016/j.asoc.2020.107030*

Cunha, C. B. & Silva, M. R. (2007), 'A genetic algorithm for the problem of configuring a hub-and-spoke network for a LTL trucking company in Brazil', *European Journal of Operational Research* **179**(3), 747–758.

Darbari, J. D., Kannan, D., Agarwal, V. & Jha, P. C. (2019), 'Fuzzy criteria programming approach for optimising the TBL performance of closed loop supply chain network design problem', *Annals of Operations Research* **273**(1-2), 693–738.
URL: *https://doi.org/10.1007/s10479-017-2701-2*

Dogan, I. & Aydin, N. (2011*a*), 'Combining Bayesian Networks and Total Cost of Ownership method for supplier selection analysis', *Computers & Industrial Engineering* **61**(4), 1072–1085.
URL: *http://linkinghub.elsevier.com/retrieve/pii/S0360835211001719*

Dogan, I. & Aydin, N. (2011*b*), 'Combining Bayesian Networks and Total Cost of Ownership method for supplier selection analysis', *Computers & Industrial Engineering* **61**(4), 1072–1085.
URL: *http://linkinghub.elsevier.com/retrieve/pii/S0360835211001719*

Dohale, V., Gunasekaran, A., Akarte, M. & Verma, P. (2021), 'An integrated Delphi-MCDM-Bayesian Network framework for production system selection', *International Journal of Production Economics* **242**(January), 108296.
URL: *https://doi.org/10.1016/j.ijpe.2021.108296*

Ecer, F. (2022), *Multi-criteria decision making for green supplier selection using interval type-2 fuzzy AHP: a case study of a home appliance manufacturer*, Vol. 22, Springer Berlin Heidelberg.
URL: *https://doi.org/10.1007/s12351-020-00552-y*

Fenton, N. E., Neil, M. & Caballero, J. G. (2007), 'Using Ranked Nodes to Model Qualitative Judgments in Bayesian Networks', *IEEE Transactions on Knowledge and Data Engineering* **19**(10), 1420–1432.

Fenton, N. & Neil, M. (2013), *Risk Assessment and Decision Analysis with Bayesian Networks*, CRC Press.

Govindan, K., Rajendran, S., Sarkis, J. & Murugesan, P. (2015), 'Multi criteria decision making approaches for green supplier evaluation and selection : a literature review', *Journal of Cleaner Production* **98**, 66–83.
URL: *http://dx.doi.org/10.1016/j.jclepro.2013.06.046*

Gupta, S., Soni, U. & Kumar, G. (2019), 'Computers & Industrial Engineering Green supplier selection using multi-criterion decision making under fuzzy environment : A case study in automotive industry', *Computers & Industrial Engineering* **136**(140), 663–680.
URL: *https://doi.org/10.1016/j.cie.2019.07.038*

Ho, J. Y., Ooi, J., Wan, Y. K. & Andiappan, V. (2021), 'Synthesis of wastewater treatment process (WWTP) and supplier selection via Fuzzy Analytic Hierarchy Process (FAHP)', *Journal of Cleaner Production* **314**(June), 128104.
URL: *https://doi.org/10.1016/j.jclepro.2021.128104*

Ho, W., Xu, X. & Dey, P. K. (2010), 'Multi-criteria decision making approaches for supplier evaluation and selection: A literature review', *European Journal of Operational Research* **202**(1), 16–24.
URL: *http://dx.doi.org/10.1016/j.ejor.2009.05.009*

Hosseini, S. & Ivanov, D. (2019), 'A new resilience measure for supply networks with the ripple effect considerations: a Bayesian network approach', *Annals of Operations Research* .
URL: *https://doi.org/10.1007/s10479-019-03350-8*

Hu, W., Toriello, A. & Dessouky, M. (2018), 'Integrated inventory routing and freight consolidation for perishable goods', *European Journal of Operational Research* **271**(2), 548–560.
URL: *https://doi.org/10.1016/j.ejor.2018.05.034*

Igarashi, M., De Boer, L. & Fet, A. M. (2013), 'What is required for greener supplier selection? A literature review and conceptual model development', *Journal of Purchasing and Supply Management* **19**(4), 247–263.
URL: *http://dx.doi.org/10.1016/j.pursup.2013.06.001*

Izadikhah, M. & Farzipoor Saen, R. (2019), 'Ranking sustainable suppliers by context-dependent data envelopment analysis', *Annals of Operations Research* (0123456789).
URL: *https://doi.org/10.1007/s10479-019-03370-4*

Jain, V., Kumar, A., Sumit, S. & Nittin, S. (2018), 'Supplier selection using fuzzy AHP and TOPSIS : a case study in the Indian automotive industry', *Neural Computing and Applications* **29**(7), 555–564.

Jensen, F. V. & Nielsen, T. D. (2013), 'Probabilistic decision graphs for optimization under uncertainty', *Annals of Operations Research* **204**(1), 223–248.

Kahraman, C., Cebeci, U. & Ulukan, Z. (2003), 'Multi-criteria supplier selection using fuzzy AHP', *Logistics Information Management* **16**(6), 382–394.
URL: *http://www.emeraldinsight.com/doi/10.1108/09576050310503367*

Kaya, R. & Yet, B. (2019), 'Building Bayesian networks based on DEMATEL for multiple criteria decision problems : A supplier selection case study', *Expert Systems With Applications* **134**, 234–248.
URL: *https://doi.org/10.1016/j.eswa.2019.05.053*

Kilincci, O. & Onal, S. A. (2011), 'Fuzzy AHP approach for supplier selection in a washing machine company', *Expert Systems with Applications* **38**(8), 9656–9664.
URL: *http://dx.doi.org/10.1016/j.eswa.2011.01.159*

Kobryń, A. (2017), 'DEMATEL as a weighting method in multi-criteria decision analysis', *Multiple Criteria Decision Making* **12**(1963), 153–167.

Kuo, Y. & Wang, C. C. (2012), 'A variable neighborhood search for the multi-depot vehicle routing problem with loading cost', *Expert Systems with Applications* **39**(8), 6949–6954.
URL: *http://dx.doi.org/10.1016/j.eswa.2012.01.024*

Lahyani, R., Gouguenheim, A. L. & Coelho, L. C. (2019), 'A hybrid adaptive large neighbourhood search for multi-depot open vehicle routing problems', *International Journal of Production Research* **57**(22), 6963–6976.
URL: *https://doi.org/10.1080/00207543.2019.1572929*

Lei, F., Wei, G., Gao, H., Wu, J. & Wei, C. (2020), 'TOPSIS Method for Developing Supplier Selection with Probabilistic Linguistic Information', *International Journal of Fuzzy Systems* **22**(3), 749–759.
URL: *https://doi.org/10.1007/s40815-019-00797-6*

Li, Y., Diabat, A. & Lu, C. C. (2020), 'Leagile supplier selection in Chinese textile industries: a DEMATEL approach', *Annals of Operations Research* **287**(1), 303–322.

Lin, C. C. & Lee, S. C. (2018), 'Hub network design problem with profit optimization for time-definite LTL freight transportation', *Transportation Research Part E: Logistics and Transportation Review* **114**(April), 104–120.

Liu, F. H. F. & Hai, H. L. (2005), 'The voting analytic hierarchy process method for selecting supplier', *International Journal of Production Economics* **97**(3), 308–317.

Liu, H. C., Quan, M. Y., Li, Z. W. & Wang, Z. L. (2019), 'A new integrated MCDM model for sustainable supplier selection under interval-valued intuitionistic uncertain linguistic environment', *Information Sciences* **486**(2), 254–270.
URL: *https://doi.org/10.1016/j.ins.2019.02.056*

Liu, P. & Li, Y. (2019), 'Multi-attribute decision making method based on generalized maclaurin symmetric mean aggregation operators for probabilistic linguistic information', *Computers and Industrial Engineering* **131**(April), 282–294.
URL: *https://doi.org/10.1016/j.cie.2019.04.004*

Liu, R., Tao, Y. & Xie, X. (2019), 'An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and synchronized visits', *Computers and Operations Research* **101**, 250–262.

Liu, X. & Wan, S. (2019), 'A method to calculate the ranges of criteria weights in ELECTRE I and II methods', *Computers and Industrial Engineering* **137**(March), 106067.
URL: *https://doi.org/10.1016/j.cie.2019.106067*

Liu, Y., Eckert, C., Yannou-Le Bris, G. & Petit, G. (2019), 'A fuzzy decision tool to evaluate the sustainable performance of suppliers in an agrifood value chain', *Computers and Industrial Engineering* **127**(March 2018), 196–212.
URL: *https://doi.org/10.1016/j.cie.2018.12.022*

Lu, E. H. C. & Yang, Y. W. (2019), 'A hybrid route planning approach for logistics with pickup and delivery', *Expert Systems with Applications* **118**, 482–492.
URL: *https://doi.org/10.1016/j.eswa.2018.10.031*

Luan, J., Yao, Z., Zhao, F. & Song, X. (2019), 'A novel method to solve supplier selection problem: Hybrid algorithm of genetic algorithm and ant

colony optimization', *Mathematics and Computers in Simulation* **156**, 294–309.
URL: *https://doi.org/10.1016/j.matcom.2018.08.011*

Memari, A., Dargi, A., Akbari Jokar, M. R., Ahmad, R. & Abdul Rahim, A. R. (2019), 'Sustainable supplier selection: A multi-criteria intuitionistic fuzzy TOPSIS method', *Journal of Manufacturing Systems* **50**(September 2018), 9–24.
URL: *https://doi.org/10.1016/j.jmsy.2018.11.002*

Memari, A., Dargi, A., Reza, M., Jokar, A., Ahmad, R., Rahman, A. & Rahim, A. (2019), 'Sustainable supplier selection : A multi-criteria intuitionistic fuzzy TOPSIS method', *Journal of Manufacturing Systems* **50**(April 2018), 9–24.
URL: *https://doi.org/10.1016/j.jmsy.2018.11.002*

Mesa-Arango, R. & Ukkusuri, S. V. (2013), 'Benefits of in-Vehicle Consolidation in Less than Truckload Freight Transportation Operations', *Procedia - Social and Behavioral Sciences* **80**, 576–590.
URL: *http://dx.doi.org/10.1016/j.sbspro.2013.05.031*

Mladenović, N., & Hansen, P. (1997), 'Variable neighborhood search', *Computers & Operations research* **24**(11), 1097–1100.

Mohammed, A. (2019), 'Towards a sustainable assessment of suppliers: an integrated fuzzy TOPSIS-possibilistic multi-objective approach', *Annals of Operations Research* (0123456789).
URL: *https://doi.org/10.1007/s10479-019-03167-5*

Mohammed, A., Harris, I. & Dukyil, A. (2019), 'A trasilient decision making tool for vendor selection: a hybrid-MCDM algorithm', *Management Decision* **57**(2), 372–395.

Mohammed, A., Harris, I., Soroka, A. & Nujoom, R. (2019), 'Computers & Industrial Engineering A hybrid MCDM-fuzzy multi-objective programming approach for a G-resilient supply chain network design', *Computers & Industrial Engineering* **127**(October 2018), 297–312.
URL: *https://doi.org/10.1016/j.cie.2018.09.052*

Moheb-Alizadeh, H. & Handfield, R. (2019), 'Sustainable supplier selection and order allocation: A novel multi-objective programming model with a hybrid solution approach', *Computers and Industrial Engineering* **129**(November 2018), 192–209.
URL: *https://doi.org/10.1016/j.cie.2019.01.011*

Moon, I., Lee, J. H. & Seong, J. (2012), 'Vehicle routing problem with time windows considering overtime and outsourcing vehicles', *Expert Systems with Applications* **39**(18), 13202–13213.
URL: *http://dx.doi.org/10.1016/j.eswa.2012.05.081*

Nilashi, M., Samad, S., Manaf, A. A., Ahmadi, H., Rashid, T. A., Munshi, A., Almukadi, W., Ibrahim, O. & Hassan Ahmed, O. (2019), 'Factors influencing medical tourism adoption in Malaysia: A DEMATEL-Fuzzy TOPSIS approach', *Computers and Industrial Engineering* **137**(August), 106005.
URL: *https://doi.org/10.1016/j.cie.2019.106005*

Opricovic, S. (2004), 'Compromise solution by MCDM methods : A comparative analysis of VIKOR and TOPSIS', *European Journal of Operational Research* **156**, 445–455.

Ortiz-Barrios, M., Cabarcas-Reyes, J., Ishizaka, A., Barbati, M., Jaramillo-Rueda, N. & de Jesús Carrascal-Zambrano, G. (2020), *A hybrid fuzzy multi-criteria decision making model for selecting a sustainable supplier of forklift filters: a case study from the mining industry*, number 0123456789, Springer US.
URL: *https://doi.org/10.1007/s10479-020-03737-y*

Pishchulov, G., Trautrims, A., Chesney, T., Gold, S., Schwab, L., Manchester, A., West, B. S. & Manchester, M. (2019), 'International Journal of Production Economics The Voting Analytic Hierarchy Process revisited : A revised method with application to sustainable supplier selection', *Intern. Journal of Production Economics* **211**(January), 166–179.
URL: *https://doi.org/10.1016/j.ijpe.2019.01.025*

Pisinger, D. & Ropke, S. (2007), 'A general heuristic for vehicle routing problems', *Computers and Operations Research* **34**(8), 2403–2435.

Qiu, X. & Huang, G. Q. (2013), 'Supply Hub in Industrial Park (SHIP): The value of freight consolidation', *Computers and Industrial Engineering* **65**(1), 16–27.
URL: *http://dx.doi.org/10.1016/j.cie.2012.06.016*

Qu, G., Zhang, Z., Qu, W. & Xu, Z. (2020), 'Green supplier selection based on green practices evaluated using fuzzy approaches of TOPSIS and ELECTRE with a case study in a Chinese internet company', *International Journal of Environmental Research and Public Health* **17**(9).

Rais, A., Alvelos, F. & Carvalho, M. S. (2014), 'New mixed integer-programming model for the pickup-and-delivery problem with transshipment', *European Journal of Operational Research* **235**(3), 530–539.
URL: *http://dx.doi.org/10.1016/j.ejor.2013.10.038*

Ramanathan, R. (2007), 'Supplier selection problem: integrating DEA with the approaches of total cost of ownership and AHP', *Supply Chain Management: An International Journal* **12**(4), 258–261.
URL: *http://www.emeraldinsight.com/doi/10.1108/13598540710759772*

Rodrigues, F., Junior, L., Osiro, L., Cesar, L. & Carpinetti, R. (2014), 'A comparison between Fuzzy AHP and Fuzzy TOPSIS methods to supplier selection', *Applied Soft Computing Journal* **21**, 194–209.
URL: *http://dx.doi.org/10.1016/j.asoc.2014.03.014*

Saaty, T. L. (2008), 'Decision making with the analytic hierarchy process', *International Journal of Services Sciences* **1**(1).

Sacramento, D., Pisinger, D. & Ropke, S. (2019), 'An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones', *Transportation Research Part C: Emerging Technologies* **102**(February), 289–315.
URL: *https://doi.org/10.1016/j.trc.2019.02.018*

Salhi, S. (2017), *Heuristic Search The Emerging Science of Problem Solving*, Springer, Switzerland.

Salhi, S., Gutierrez, B., Wassan, N., Wu, S. & Kaya, R. (2020), 'An effective real time GRASP-based metaheuristic: Application to order consolidation and dynamic selection of transshipment points for time-critical freight logistics', *Expert Systems with Applications* **158**, 113574.
URL: *https://doi.org/10.1016/j.eswa.2020.113574*

Salhi, S. & Thompson, J. (2022*a*), An Overview of Heuristics and Metaheuristics, *in* 'The Palgrave Handbook of Operations Research', Springer Nature, London, chapter 11, pp. 1–26.

Salhi, S. & Thompson, J. (2022*b*), The New Era of Hybridisation and Learning in Heuristic Search Design Hybridisation Search, *in* 'The Palgrave Handbook of Operations Research', Springer Nature, London, chapter 15, pp. 1–19.

Sener, A., Barut, M., Dag, A. & Yildirim, M. B. (2021), 'Impact of commitment, information sharing, and information usage on supplier performance:

a Bayesian belief network approach', *Annals of Operations Research* **303**(1-2), 125–158.
URL: *https://doi.org/10.1007/s10479-019-03504-8*

Shaw, P. (1997), Using constraint programming and local search methods to solve vehicle routing problems, *in* 'Proceedings of the 4th International Conference on Principles and Practice of Constraint Programming', pp. 417–431.

Simeonova, L., Wassan, N., Salhi, S. & Nagy, G. (2018), 'The heterogeneous fleet vehicle routing problem with light loads and overtime: Formulation and population variable neighbourhood search with adaptive memory', *Expert Systems with Applications* **114**, 183–195.
URL: *https://doi.org/10.1016/j.eswa.2018.07.034*

Simoni, M. D., Bujanovic, P., Boyles, S. D. & Kutanoglu, E. (2018), 'Urban consolidation solutions for parcel delivery considering location, fleet and route choice', *Case Studies on Transport Policy* **6**(1), 112–124.
URL: *https://doi.org/10.1016/j.cstp.2017.11.002*

Singh, R. K., Gunasekaran, A. & Kumar, P. (2018), 'Third party logistics (3PL) selection for cold chain management: a fuzzy AHP and fuzzy TOPSIS approach', *Annals of Operations Research* **267**(1-2), 531–553.

Sze, J. F., Salhi, S. & Wassan, N. (2016), 'A hybridisation of adaptive variable neighbourhood search and large neighbourhood search: Application to the vehicle routing problem', *Expert Systems with Applications* **65**, 383–397.
URL: *http://dx.doi.org/10.1016/j.eswa.2016.08.060*

Sze, J. F., Salhi, S. & Wassan, N. (2017), 'The cumulative capacitated vehicle routing problem with min-sum and min-max objectives: An effective hybridisation of adaptive variable neighbourhood search and large neighbourhood search', *Transportation Research Part B: Methodological* **101**, 162–184.

Tong, L. Z., Wang, J. & Pu, Z. (2022), 'Sustainable supplier selection for SMEs based on an extended PROMETHEE approach', *Journal of Cleaner Production* **330**(April 2021), 129830.
URL: *https://doi.org/10.1016/j.jclepro.2021.129830*

Ülkü, M. A. (2012), 'Dare to care: Shipment consolidation reduces not only costs, but also environmental damage', *International Journal of Production Economics* **139**(2), 438–446.

Ülkü, M. A. & Bookbinder, J. H. (2012), 'Optimal quoting of delivery time by a third party logistics provider: The impact of shipment consolidation and temporal pricing schemes', *European Journal of Operational Research* **221**(1), 110–117.

Venkatesh, V. G., Zhang, A., Deakins, E., Luthra, S. & Mangla, S. (2019), 'A fuzzy AHP-TOPSIS approach to supply partner selection in continuous aid humanitarian supply chains', *Annals of Operations Research* **283**(1-2), 1517–1550.
URL: *https://doi.org/10.1007/s10479-018-2981-1*

Wang, J. W., Cheng, C. H. & Huang, K. C. (2009), 'Fuzzy hierarchical TOPSIS for supplier selection', *Applied Soft Computing Journal* **9**(1), 377–386.

Watróbski, J., Jankowski, J., Ziemba, P., Artur, K. & Magdalena, Z. (2019), 'Generalised framework for multi-criteria method selection', *Omega* **86**, 107–124.

Wolfinger, D. (2021), 'A Large Neighborhood Search for the Pickup and Delivery Problem with Time Windows, Split Loads and Transshipments', *Computers and Operations Research* **126**, 105110.
URL: *https://doi.org/10.1016/j.cor.2020.105110*

Wu, C., Lin, Y. & Barnes, D. (2021), 'An integrated decision-making approach for sustainable supplier selection in the chemical industry', *Expert Systems with Applications* **184**(July), 115553.
URL: *https://doi.org/10.1016/j.eswa.2021.115553*

Wu, Q., Zhou, L., Chen, Y. & Chen, H. (2019), 'An integrated approach to green supplier selection based on the interval type-2 fuzzy best-worst and extended VIKOR methods', *Information Sciences* **502**, 394–417.
URL: *https://doi.org/10.1016/j.ins.2019.06.049*

Yu, C., Shao, Y., Wang, K. & Zhang, L. (2019), 'A group decision making sustainable supplier selection approach using extended TOPSIS under interval-valued Pythagorean fuzzy environment', *Expert Systems with Applications* **121**, 1–17.
URL: *https://doi.org/10.1016/j.eswa.2018.12.010*

Zeshui, X. & Zhang, S. (2022), Fuzzy multi-attribute decision-making: Theory, methods and Applications, *in* S. Salhi & J. Boylan, eds, 'The Palgrave Handbook on Operations Research', Springer Nature, pp. 1–44.

Zhang, J., Yang, D., Li, Q., Lev, B. & Ma, Y. (2021), 'Research on sustainable supplier selection based on the rough dematel and fvikor methods', *Sustainability (Switzerland)* **13**(1), 1–24.

Zhang, W., Ju, Y. & Liu, X. (2017), 'Multiple criteria decision analysis based on Shapley fuzzy measures and interval-valued hesitant fuzzy linguistic numbers', *Computers and Industrial Engineering* **105**, 28–38.
URL: *http://dx.doi.org/10.1016/j.cie.2016.12.046*