MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES

A METAHEURISTIC ALGORITHM TO ELASTICITY BASED SHELF SPACE ALLOCATION PROBLEM

AYŞE HANDE EROL BİNGÜLER

Ph.D. THESIS
Department of Industrial Engineering

Thesis Supervisor
Assoc.Prof. Dr. Serol BULKAN

Thesis CO- Supervisor
Assist. Prof. Dr. Mustafa AĞAOĞLU

ISTANBUL, 2016

MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES

A METAHEURISTIC ALGORITHM TO ELASTICITY BASED SHELF SPACE ALLOCATION PROBLEM

AYŞE HANDE EROL BİNGÜLER

Ph.D. THESIS
Department of Industrial Engineering

Thesis Supervisor
Assoc.Prof. Dr. Serol BULKAN

Thesis CO- Supervisor
Assist. Prof. Dr. Mustafa AĞAOĞLU

ISTANBUL, 2016

THIS PAGE INTENTIONALLY LEFT BLANK

MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES IN PURE AND APPLIED SCIENCES

Ayşe Hande EROL BİNGÜLER, a Doctor of Philosophy student of Marmara University Institute for Graduate Studies in Pure and Applied Sciences, defended her thesis entitled **"A Metaheuristic Algorithm to Elasticity Based Shelf Space Allocation Problem"**, on Aug 31, 2016 and has been found to be satisfactory by the jury members.

Jury Members

Assoc.Prof. Dr. Serol BULKAN (Advisor)
Marmara University .....

Prof.Dr. Selim ZAİM (Jury Member)
Istanbul Technical University .....

Assoc.Prof.Dr. Hakan TOZAN (Jury Member)
Turkish Naval Academy.....

Assist.Prof.Dr. Mustafa Batuhan AYHAN (Jury Member)
Marmara University .....

Assist.Prof.Dr. Adnan ÇORUM (Jury Member)
Bahçeşehir University .....

APPROVAL

Marmara University Institute for Graduate Studies in Pure and Applied Sciences Executive Committee approves that Ayşe Hande EROL BİNGÜLER be granted the degree of Doctor of Philosophy in Department of Industrial Engineering, Industrial Engineering Program on .....02.09.2017 (Resolution no: 2017/01-02...).

Director of the Institute
Prof. Dr. Uğur YALÇI

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENT

I would like to express acknowledgment and appreciation to my dissertation advisory committee members, ***Assoc. Prof. Dr. Serol BULKAN, Assist. Prof. Dr. Mustafa AĞAOĞLU, Assoc. Prof. Dr. Hakan TOZAN***, and ***Prof. Dr. Selim ZAİM,*** for their assistance, direction and dedication throughout my course of study.

I want to thank ***THE SCIENTIFIC AND TECHNOLOGICAL RESEARCH COUNCIL OF TURKEY (TÜBİTAK)*** for supporting science and providing me financial support.

Acknowledgment and appreciation is also due to my special friend, Asım Tokgöz, and his family for their support about programming. I also thank to Murat Toman for his technically contribution. Other thanks to my special friend, Uğur, for his support in this process.

I would like to express my special thanks to my Thesis Supervisor, Assoc. Prof. Dr. Serol BULKAN, for his support, contagious enthusiasm and for the trust in my capabilities (much more than mine for sure). I appreciate his dedication to teaching and emphasize again the deep admiration that I have for him since my OR class.

I owe to give credit to my colleagues who supported me extensively during my course of study.

Acknowledgment and gratitude is expressed to ***my love***, also ***my mother***, ***my father*** and ***my family*** for ***their love***, ***support***, ***sacrifice***, and ***encouragement*** needed to bring this study completion. During this time, I had the pleasure to all of my family and ***my friends who are with me in every condition in my life***.

In my academic career, while walking on the road with putting forward original and useful projects/studies for my country and for the organization that I am working, I offer my thanks and respect to everyone who supported me...

August, 2016 **Ayşe Hande EROL BİNGÜLER**

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

ACKNOWLEDGMENT……i
TABLE OF CONTENTS……..……ii
ÖZET……iii
ABSTRACT……iv
CLAIM FOR ORIGINALITY……..v
SYMBOLS……..…..vi
ABBREVIATIONS……vii
LIST OF FIGURES……viii
LIST OF TABLES……..……ix
1. INTRODUCTION 1
2. BACKGROUND 5
2.1. Knapsack Problem 5
2.2. Shelf Space Allocation Problem 6
3. ALGORITHMS 9
3.1. Genetic Algorithms 9
3.2. Simulated Annealing Algorithm 12
3.3. Variable Neighborhood Search Method 15
4. RESEARCH METHODOLOGY 17
4.1. Test problems 17
4.2. Methods 18
4.2.1. Initial Solution 19
4.2.2. Improvement techniques 19
4.3. Methodology and Problem Definition 20
4.4. Steps of variable neighborhood search method 21
4.5. Validation of the Model 22
5. RESULTS and DISCUSSIONS 25
5.1. Results 25
5.2. Conclusion and Future Researches 26
REFERENCES 29
APPENDIX A 35
ÖZGEÇMİŞ 71
BIOGRAPHY 73

THIS PAGE INTENTIONALLY LEFT BLANK

ÖZET

RAF ALANI YERLEŞTİRME PROBLEMLERİ İÇİN METASEZGİSEL ÇÖZÜM ALGORİTMASI

Sırt çantası problemi, kombinatoryal eniyileme problemlerinden biridir. Her birinin ağırlığı belli olan, n adet parçayı belli bir kapasitesi olan bir sırt çantasına, maksimum kar getirecek şekilde yerleştirilmesi problemidir. Problem, kaynak dağılımında kullanılmaya başlamış, aynı zamanda kombinatorik (combinatorics), bilgisayar bilimi, karmaşıklık teorisi (complexity theory), uygulamalı matematik, eniyileme gibi alanlarda da kullanılmaktadır. Sırt çantası problemi, kombinatoryal eniyileme problemlerinden biridir.

Raf alanı eniyilemesi problemi (shelf space allocation problem) çok kısıtlı sırt çantası problemine benzemektedir. Sırt çantası probleminden farklı olarak kutulama problemi, negatif olmama ve tamsayı kısıtına ek olarak kapasite kısıtı da olan sırt çantası problemidir. Kutulama probleminde bazı kuralları içeren kısıtlar (policy constraints) da mevcuttur. Sırt çantası modeli, bütçe veya insan kaynakları planlaması gibi konular ile ilgili tek baskın kaynak içeren birçok problemde uygulanabilmektedir. Kutulama problemi sınırlı kaynağın dağılımı ile ilgili yönetsel kararlarda uygulanabilmektedir. Karmaşıklığından dolayı raf alanı eniyilemesi problemi, alan yönetim sistemleri için birçok sezgisel yaklaşım geliştirilmiştir.

Kutulama problemi ile ilgili literatür araştırmasını yaptıktan sonra kutulama problemi için elastisite bazlı bir model oluşturulmuştur. Çalışmada raf alanı yerleştirme problemi için kapsamlı eniyileme modeli kullanılacaktır. Perakende sektöründe kullanılabilecek tamsayılı programlama modeli tasarlanmıştır. Alan elastikliği faktörü de amaç fonksiyonuna eklenerek ürün kalemlerini amaç fonksiyonu doğrultusunda uygun raf alanına en uygun yerleştirme yapılmıştır. Önerilen algoritma ile belirlenen kısıtlar ve amaç fonksiyonu doğrultusunda raf alanına veya uzunluğuna bağlı olarak en iyi ürün yerleştirmeleri yapılmıştır. Sezgisel yaklaşım kabul edilebilir bir zaman içinde yaklaşık en iyi sonuç elde edilmiştir.

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

A METAHEURISTIC ALGORITHM TO ELASTICITY BASED SHELF SPACE ALLOCATION PROBLEM

A shelf space allocation problem (SSAP) is sort of knapsack problem with multi constraints regarding to shelf capacity and policy constraints. The knapsack model could include variety of implementations used for resource allocation in order to deal with some problems like human resource management, budget planning and etc. Solving the SSAP could improve retail operations and enables retail management to the use of shelf space efficiently.

SSAP is good for reaching optimal management decision required for dealing with limited resource allocation. In addition, SSAP is related with NP-hard complexity used to describe some decision problems. Thus, in order to solving the commercial space management allocation issue, some heuristics have been proposed and implemented in the retail industry. In addition using simple intuitive rules for improving the guidelines of operational choices relatively, the decision maker in retailing industry can easily use to practically apply decisions about the allocation of shelf space.

The concern for practicability and simplicity for the previous researches, decisions and outcomes in the decisions of space allocation that reach far from the optimum levels of the performance. Yet improvements in the information technology and computer science allowed high computational efficiency and the development of optimization that extended feasible solution to the stage of space management systems to solve SSAP.

This study proposes an elasticity based SSAP model which is an inclusive optimization model for the allocation of shelf space. The model adapted a mathematical model used integer programming to increase its applicability in the marketing and the retail industry. Besides that, it aims to identify and define the allocation space for items precisely. Doing that would allow management to improve retail operations and give helps for an efficient use of the limited shelf space. In other words, the model designed to permit maximize the self space by adding space elasticity factor. In addition that Variable Neighborhood Search (VNS) algorithm, based on neighborhoods, used to locate items into shelf space, subject to restrictions, and the benchmarking problems are generated for this proposed algorithm. VNS outperforms other heuristics for test problems.

THIS PAGE INTENTIONALLY LEFT BLANK

CLAIM FOR ORIGINALITY

A shelf space allocation problem (SSAP) could be related to decision making to get best outcome under some operational constraints in the field of retail industry. The problem is a knapsack problem case as an extension of knapsack problem. Owing to the complexity to obtain its optimal solution, intuitive approaches have largely been used in the commercial space management systems. Along with the commercial systems use relatively simple heuristic rules in order to develop guidelines operationally designed to permit retailers to implement decisions of shelf space allocation easily.

In general, the allocation of shelf space affects could maximize the profitability of the store through both the demand function in view of combining the elasticities of main and cross space. It also could concludes the cost function such as; carrying, procurement, and out of stock costs while previous approaches and heuristics usually emphasis on brands with a limited number and only a few shelves.

It is claimed that the inclusion of particular influencing variables indicates that early works were mostly concern about several circumstances, that shelf space elasticity plays a role in it. However none of them proposed any other relevant circumstances. In this study the elasticity factor is used for the proposed algorithm. For good measure, the VNS model is modified to a mathematical model included integer programming for increasing its applicability in the retail environment. Ultimately the model seeks to determine the best allocation of product items to the available shelf space of a retail store to maximize objective function adding space elasticity factor. An algorithm uses items allocation to the shelves with assumptions and restrictions.

The main goal of this study is to develop the neighborhoods based an efficient metaheuristic algorithm which can be implemented in practice by a retailer. It would create opportunity to observe the sales information system to solve the SSAP and obtain relevant results. Hereby, in brief, this study aims to assist the management of shelf space in any retail outlet and would contribute significantly to the firm's profitability. Last but not least the operations of space allocation and their performances can also change the management strategy after implementing this proposed model. This approach will be initial one since no similar extension of the SSAP is developed for related applications and other researches so far.

August, 2016 Assoc.Prof.Dr. Serol BULKAN Ayşe Hande EROL BİNGÜLER

THIS PAGE INTENTIONALLY LEFT BLANK

SYMBOLS

a_i **:** The length of product *i*

E_{ij} **:** The cross price elasticity if the product *i* is to the right of the product *j* on the shelf

I **:** The number of products

K **:** The number of shelves

L **:** The position of products

L_i **:** The lower bound to allocate product *i*

N_k **:** Finite set of pre-selected neighborhood structures

$N_{k(x)}$ **:** The set of solutions in the kth neighborhood of *x* for $k=1,\ldots,k_{max}$

P_{ik} **:** The profit of the product *i* on shelf *k*

S **:** Solution space

T **:** The length of shelf

T_k **:** The length of shelf *k*

U_i **:** The upper bound to allocate product *i*

X_{ilk} **:** The decision variable to identify if product *i* is in the lth position on shelf *k*

THIS PAGE INTENTIONALLY LEFT BLANK

ABBREVIATIONS

CPU **:** Central Process Unit

GAs **:** Genetic Algorithms

ILP **:** Integer Linear Programming

OR **:** Operations Research

SA **:** Simulated Annealing

SSAP **:** Shelf Space Allocation Problem

TS **:** Tabu Search

VNS **:** Variable Neighborhood Search

1BP **:** One dimensional bin packing problem

2BP **:** Two dimensional bin packing problem

3BP **:** Three dimensional bin packing problem

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 3.1. The structure of the genetic algorithms based on the replacement strategy 10
Figure 3.2. The principle of the simulated annealing algorithm…….…...13
Figure 3.3. Pseudocode of a simulated annealing algorithm….……..….....14
Figure 4.1. Shelf space allocation in initial solution…….……..……...19
Figure 4.2. Shelf space allocation with improvement ..…….……..….19
Figure 4.3. Steps of basic VNS algorithm ...……...…….....21
Figure 4.4. Swap operation within shelves.……..……...…….…….……22
Figure 4.5. Swap operation between shelves.……..……...…….……..……22
Figure 4.6. Adding new product in shelf 1.……..……...…….…….….…....23
Figure 4.7. Initial solution for test problem.……..……...…….……..……...23
Figure 4.8. VNS solution for test problem…...……..……...…….…….……...23
Figure 4.9. The results for problem instances (I (Number of product)=7/K (Number of shelf) =3/ T (Length of shelf) =30) ..……..……...…….…….……...24
Figure A.1. The results for problem instances (I=7/ K=3/ T =30)……...……..35
Figure A.2. The results for problem instances (I=7/ K=3/ T =20)……...……..36
Figure A.3. The results for problem instances (I=7/ K=3/ T =40)…….……....37
Figure A.4. The results for problem instances (I=7/ K=4/ T =30)…. ……...…….…..38
Figure A.5. The results of problem instances (I=7/ K=4/ T =20)…….….…39
Figure A.6. The results of problem instances (I=7/ K=4/ T =40)……...……...40
Figure A.7. The results of problem instances (I=7/ K=5/ T=30)…….. 41
Figure A.8. The results of problem instances (I=7/ K=5/ T =20)……42
Figure A.9. The results of problem instances (I=7/ K=5/ T =40)……..……..…..43
Figure A.10. The results of problem instances (I=8/ K=3/ T =30)…….……..… 44
Figure A.11. The results of problem instances (I=8/ K=3/ T =20)…….……...45
Figure A.12. The results of problem instances (I=8/ K=3/ T =40)……..……..46
Figure A.13. The results of problem instances (I=8/ K=4/ T =30)…….……..….47
Figure A.14. The results of problem instances (I=8/ K=4/ T =20)…….……..….48
Figure A.15. The results of problem instances (I=8/ K=4/ T =40)…….……..….49
Figure A.16. The results of problem instances (I=8/ K=5/ T =30)…….……..….50

Figure A.17. The results of problem instances (I=8/ K=5/ T =20)…….……..….51
Figure A.18. The results of problem instances (I=8/ K=5/ T =40)…….……..….52
Figure A.19. The results of problem instances (I=9/ K=3/ T =30)…….……..….53
Figure A.20. The results of problem instances (I=9/ K=3/ T =20)…….……..….54
Figure A.21. The results of problem instances (I=9/ K=3/ T =40)…….……..….55
Figure A.22. The results of problem instances (I=9/ K=4/ T =30)…….……..…56
Figure A.23. The results of problem instances (I=9/ K=4/ T =20)…….……..…57
Figure A.24. The results of problem instances (I=9/ K=4/ T =40)…….……..…58
Figure A.25. The results of problem instances (I=9/ K=5/ T =30)…….……..…59
Figure A.26. The results of problem instances (I=9/ K=5/ T =20)…….……..…60
Figure A.27. The results of problem instances (I=9/ K=5/ T =40)…….……..…61
Figure A.28. The results of problem instances (I=10/ K=3/ T =30)…….……....62
Figure A.29. The results of problem instances (I=10/ K=3/ T =20)…….……....63
Figure A.30. The results of problem instances (I=10/ K=3/ T =40)…….……....64
Figure A.31. The results of problem instances (I=10/ K=4/ T =30)…….……**...**65
Figure A.32. The results of problem instances (I=10/ K=4/ T =20)…….……....66
Figure A.33. The results of problem instances (I=10/ K=4/ T =40)…….……....67
Figure A.34. The results of problem instances (I=10/ K=5/ T =30)…….……....68
Figure A.35. The results of problem instances (I=10/ K=5/ T =20)…….……....69
Figure A.36. The results of problem instances (I=10/ K=5/ T =40)…….……....70

LIST OF TABLES

Table 3.1. Comparison of metaheuristics and greedy algorithm……...11
Table 4.1. The profit and length of products……...……..…..17
Table 4.2. The cross price elasticities of products……..……....17
Table 4.3. The lower and upper bound to allocate products to the shelves……....17
Table 5.1. Improvement over solutions ……..……..…...25
Table A.1. The results (profit) of solution techniques for problem (I=7/ K=3/ T =30)…...35
Table A.2. The results (profit) of solution techniques for problem (I=7/ K=3/ T =20).……36
Table A.3. The results (profit) of solution techniques for problem (I=7/ K=3/ T =40)..…...37
Table A.4. The results (profit) of solution techniques for problem (I=7/ K=4/ T =30)..…...38
Table A.5. The results (profit) of solution techniques for problem (I=7/ K=4/ T =20)..…...39
Table A.6. The results (profit) of solution techniques for problem (I=7/ K=4/ T =40)..…...40
Table A.7. The results (profit) of solution techniques for problem (I=7/ K=5/ T=30)….....41
Table A.8. The results (profit) of solution techniques for problem (I=7/ K=5/ T =20)….....42
Table A.9. The results (profit) of solution techniques for problem (I=7/ K=5/ T =40)….....43
Table A.10. The results (profit) of solution techniques for problem (I=8/ K=3/ T =30)…...44
Table A.11. The results (profit) of solution techniques for problem (I=8/ K=3/ T =20)…...45
Table A.12. The results (profit) of solution techniques for problem (I=8/ K=3/ T =40)…...46
Table A.13. The results (profit) of solution techniques for problem (I=8/ K=4/ T =30)…...47
Table A.14. The results (profit) of solution techniques for problem (I=8/ K=4/ T =20)…...48
Table A.15. The results (profit) of solution techniques for problem (I=8/ K=4/ T =40)…...49
Table A.16. The results (profit) of solution techniques for problem (I=8/ K=5/ T =30)…...50
Table A.17. The results (profit) of solution techniques for problem (I=8/ K=5/ T =20)…...51
Table A.18. The results (profit) of solution techniques for problem (I=8/ K=5/ T =40)…...52
Table A.19. The results (profit) of solution techniques for problem (I=9/ K=3/ T =30)…...53
Table A.20. The results (profit) of solution techniques for problem (I=9/ K=3/ T =20)…...54
Table A.21. The results (profit) of solution techniques for problem (I=9/ K=3/ T =40)…...55

Table A.22. The results (profit) of solution techniques for problem (I=9/ K=4/ T =30)…...56
Table A.23. The results (profit) of solution techniques for problem (I=9/ K=4/ T =20)…...57
Table A.24. The results (profit) of solution techniques for problem (I=9/ K=4/ T =40)…...58
Table A.25. The results (profit) of solution techniques for problem (I=9/ K=5/ T =30)…...59
Table A.26. The results (profit) of solution techniques for problem (I=9/ K=5/ T =20)…...60
Table A.27. The results (profit) of solution techniques for problem (I=9/ K=5/ T =40)…...61
Table A.28. The results (profit) of solution techniques for problem (I=10/ K=3/T =30)…..62
Table A.29. The results (profit) of solution techniques for problem (I=10/ K=3/T =20)…..63
Table A.30. The results (profit) of solution techniques for problem (I=10/ K=3/T =40)…..64
Table A.31. The results (profit) of solution techniques for problem (I=10/ K=4/T =30)…..65
Table A.32. The results (profit) of solution techniques for problem (I=10/ K=4/T =20)…..66
Table A.33. The results (profit) of solution techniques for problem (I=10/ K=4/T =40)…..67
Table A.34. The results (profit) of solution techniques for problem (I=10/ K=5/T =30)…..68
Table A.35. The results (profit) of solution techniques for problem (I=10/ K=5/T =20)…..69
Table A.36. The results (profit) of solution techniques for problem (I=10/ K=5/T =40)…..70

1. INTRODUCTION

Knapsack problems are kind of cases related with combinatorial optimization. The aim of the problem is to determine numbers of each item to consist of a set-in order to make the total weight is equal to a limit restriction or less than this value and the total value is larger than possible, and a set of items are defined with a mass and value. The problem is implemented to find solution for allocation of resources with constraints like financial, etc. This problem is as well used in the fields of optimization, applied mathematics, combinatorial optimization, computer science and the theory of complexity.

Knapsack problems had been studied in previous researches since 1897. The problem was referred in Tobias Dantzig's early works firstly. This name is come from Dantzig suggestions that can be occurred in traditional stories previous to a problem have been identified in detail mathematically. Reduction algorithms, that are effective, have been proposed to allow one to fix many decision variables objective functions for all knapsack problems (Pisinger, 1995). Multi objective, multidimensional and quadratic knapsack problems, subset sum problem that are variations of knapsack problem.

The multidimensional knapsack problem is further related to bin packing problem. Yet it is different from the bin packing problem. Even though a subset of items can be selected in the bin packing problem, it is necessary to reserve all items into current bins. Overall the concept is based on the idea that the items have multiple dimensions. In them even a minor change can be seen, but it is not equal to adding to the initial knapsack capacity. Therefore inn many allocations, packing and scheduling problems in Operations Research (OR) and the structure of approximation of polynomial time, this variation is implemented.

When it comes to the bin packing problem, the products of different volumes must be filled into a finite number of containers or bins with a capacity in a way that minimizes the usage of bins as less as possible. In the theory of computational complexity, it is a kind of NP-hard problem in combinatorial optimization problems. Two dimensional and linear packing, packing by weight and/or cost, etc. are the variations of this problem. Furthermore, they have many implementations, such as loading containers,

filling up trucks with the constraints of weight capacity. In addition that the cutting stock problem's special case is also related to the bin packing problem. When the number of bins has constraints and each item is identified by both a value and volume, the objective is to maximize items value that can be allocated in the bin is known as the knapsack problem.

Even though the bin packing problem has an NP-hard computational complexity, optimal solutions to large scale problem instances can be created with sophisticated or greedy algorithms. Many heuristic techniques also have been improved like first fit algorithm, heuristic algorithms; Tabu Search (TS) algorithm, genetic algorithm and greedy algorithms, etc.

The objective of the bin packing problem is as well complementary to the minimum makespan scheduling problem. It aims to scheduling jobs with various lengths on a fixed number of machines while minimizing the makespan, or equally to pack the items of various sizes into a fixed number of bins to minimize to the size of the largest bin (Watson, 2007).

Gilmore and Gomory (1965) suggested the first model for the two-dimensional packing problems (2BP), by extending their column generation approach for the one-dimensional packing problems (1BP). Beasley (1985) considered a variant of 2BP (like the cutting stock problem), and formulated an Integer Linear Programming (ILP) model consist of the use of to the classical bin packing model. The model sizes bin capacities both span items allocation that has discrete coordinates. Hadjiconstantinou and Christofides (1995) also proposed a similar model. While Fekete and Schepers (1997) proposed a new model based on the problem with graph theoretical characterization, Lodi et al. (2002) also suggested a model similar to this model, for the special case where the items must be filled ILP models concerning a number of variables and constraints by levels. Multidimensional bin packing problem is an interesting combinatorial optimization problem with applications to machine reassignment, computing of cloud, and virtualized data centers. Quite the opposite of bin packing classical problem, the size of items value and the capacity of bins value span like a vector, it necessitate that feasible assignment respect capacity constraints from corner to corner all dimensions (Michael and Moffitt, 2013).

Apart from all those suggestions, it is widely accepted that one of the most important feature of the shelf space resource management is to appeal more customers by making best optimal decisions in the logistic industry. In addition that the shelf space management can not only reduce inventory level but also improve seller relationship strongly and increase consumer satisfaction substantially (Fancher, 1991). For this reason, shelf space management to figure out how to locate products efficiently in the retail industry is somewhat important for the success (Hwang et al., 2009). Again not to mention the decision of the allocation and management of the shelf space is a vital issue for the retail operations management.

It is widely accepted that a shelf space allocation problem (SSAP) is like a multi constraint knapsack problem with capacity constraints. SSAP also has non-negativity and integer constraints, and it has some restrictions in policies that are the major difference between knapsack problem and SSAP. In general SSAP is used in application to decisions management dealing with limited resources allocation. Besides SSAP is related with NP-hard complexity. The latest improvement in information technology and increase in the usage of computer with high computational efficiency created the optimization development researches to solve SSAP reached feasible solution to the stage of space management systems (Yang, 2001).

SSAP is a decisional issue to be solved for reaching the best objective under some operational constraints in the retail industry. It could be seen as an extension of the knapsack problem. Due to the NP-hard complexity to attain its optimal solution, instinctive researches have mostly been used in space management systems commercially. These systems use reasonably simple rules in heuristic to guide operational decisions considered to let retailers easily to apply shelf space allocation decisions in practice (Yang and Chen, 1999).

THIS PAGE INTENTIONALLY LEFT BLANK

2. BACKGROUND

2.1. Knapsack Problem

The knapsack problems vary from multi objective, multidimensional knapsack problem, non-linear (quadratic) knapsack problems to the subset-sum problems.

The bin packing problem is a kind of classical NP-hard optimization problems (Fekete and Schepers, 2001). It is a basic problem in operational research and the combinatorial optimization field. It consists to find a valid arrangement of all the rectangular objects in items. It is also rectangular called bins, to minimize the number of the boxes used. The solution for this problem is to determine the bins number used to place all the objects in the different bins on well-defined positions and orientations. Even though the traditional problem is defined in one dimension, there are some other alternatives into two or three dimensions.

For instance, the 2BP is one the of dimensional problem generalization (Bansal and Sviridenko, 2007). However, it differs when it comes to the same objective and when all bins and boxes are used with the definition of width and height. As a matter of course this problem is adapted for many industrial implementations. Especially in the optimization of cutting materials like cloth, wood, metal, glass and in the packing industry like transportation and warehousing (Lodi et al., 2002).

When it comes to the three-dimensional bin packing problem (3BP), it could be easy to see the 3BP is less studied than the two-dimensional bin packing problem. In addition, it is very rare to find work on 3D bin packing (Ponce-Pérez et al., 2005). Apart from scarcity, the three-dimensional bin packing is used for filling pallets (Bischoff et al., 1995) and all it is used for truck loading especially for the container loading and the container stowage planning.

By the way there are two kinds of container loading solution. The first one is related to 3BP that basically aims to minimize the container costs used. (Bortfeldt and Mack, 2007) (He and Cha, 2002). The second solution is related to the knapsack problems. This one aims to maximize the stowed volume of the container required (Bortfeldt and Gehring, 2001) (Raidl, 1999).

Carvalho studied the formulation of arc flow including side constraints for 1BP. A “branch and price procedure” that combines overdue the generation of variable and “branch and

bound" is used for the proposed model. OR Library test data sets had been derived for the study. Strong lower bound was used and the relaxation of linearity guides to obedient "branch and bound trees" for the problem instances (Carvalho, 1999).

Lodi et al. (1999) suggested the problem class occurring from all requirement combinations that the items were taken through the edge to edge cuts' order parallel to the bin edges. A combined tabu search and a heuristic algorithm approach were modified to change the neighborhood for a specific problem.

Dual feasible solutions are used by Fekete and Schepers (2001) and a simple generic approach is offered to establish lower bounds to bin packing problems. This approach also provides a general structure for obtaining new bounds.

2.2. Shelf Space Allocation Problem

Shelf space allocation increases retail store profitability when the solution for problem includes the main and cross space elasticities in the demand function, and through the cost function (the cost of carrying, procurement, and out of stock) (Corstjens, and Doyle, 1981). While early works usually focus on a brand number limitation and only a few number of the shelves (Dreze et al., 1994).

Hwang et al. (2009) combined a proposed model for the design of shelf space and item allocation problem mathematically in the way that maximizing the profit of retailer. Concurrently, the researchers used the design of the shelf space and the allocation problem in view of putting the space elasticity and the effect of location on the function of the demand. The elasticities are usually measured with different levels of aggregation, definably at the level of category and/or brand. The variation of the shelf at the brand level includes the switching of brand because the related products are considered substitutes (Eisend, 2014).

A decision maker has to make several resolutions concerning procedures that can manipulate the results. One of the decision in meta analysis refers to the appropriate weighting of the studies that considers the fundamental sample of products or purchase decisions made by customers, because the researchers' decisions in the meta-analysis vary in product number or the decision of purchasing they are implemented on (Eisend, 2014). As regards, the influence of the direction of space variation, Eisend (2014) finds

that the shelf space increases the results greater the estimation of the elasticity than the reduction of the shelf space. This finding highlights the shelf space application variation as a useful marketing tool.

Retailers benefit from the optimum allocation of products into shelves in two ways: first they decrease the shelf replacement and inventory costs, and then enhance sales costs. The sales quantity of products depends on many factors such as location of the product within the shelf, product facings and adjacent products (Dreze et al., 1994). Anderson and Amato (1973) showed that the firms increased the demand for each product by increasing the total area on the shelf.

Yang suggested a greedy algorithm to obtain the solutions as well (Yang, 2001). Lim et al. proposed Yang's approach. This study is the comparison of the original and the improved heuristics which are the integration of three metaheuristic algorithms. The algorithm that incorporates local search found the best results (Lim et al., 2004).

In addition, Corstjens and Doyle (1981) suggested demand model for a product depends on the direct elasticity. Zufryden (1986) proposed a dynamic programming approach to a simplified version for SSAP. Lim et al. (2004) supposed Squeaky Wheel Optimization algorithm for this kind of problem. Lim et al. says that shelf space allocation can provide retailers with a competitive frame makes retailers more effective. Despite the fact that there are a few studies on these problems, yet there is great interest in improving the allocation of product in the retailing sector. This study measures the effectiveness of a practicable linear allocation model for optimizing SSAP. It extends the model to address other requirements such as product groupings and nonlinear profit functions. Further given that a solution of network flow, it can be created by a strategy that mixes a strong local search with a metaheuristic algorithm to space allocation. This strategy improves the flexibility and effectiveness, as it can address not only linear problems of the real life size but also non-linear problems of real life size to achieve optimal or near optimal solutions throughout straightforwardly implemented algorithms in a reasonable time scale. It suggests the opportunities to retailers for more efficient and profitable shelf management, as well as higher quality plan and schedule.

Hwang et al. (2009) has identified an integrated mathematical model on multi-level shelves. This paper points out the retailer problems regarding to selling of various brands

of items in the course of displaying on shelf space. They simply developed a mathematical model with the integration of the shelf space design and item allocation problem with the maximizing the retailer's profit objective function. To resolve the problem, GAs had been projected for the shelf space configurations with two types like slicing composition produced by "guillotine type cuts" and "horizontal cuts" only. The model validity is measured for problem instances.

Chen and Lin (2007) supposed the approach includes data mining and association rule mining for problems. At the same year, Fadıloğlu et al. (2007) proposed a model for two local chains using proprietary data in supermarket. Martinez-de-Albeniz and Roels (2011) measured the price effect of the wholesalers on retailers allocation decisions. Gajjar and Adil (2011) develop three heuristics to solve SSAP. They compared these three heuristics by using empirical study. Since it is a challenge for retailers using large scale number of products or increasing the number of them for efficient allocation of available shelf space to them. Linear profit function is also used for this study.

Eisend (2014) studied meta analysis for SSAP and he proposed a model with elasticities at different aggregation levels. Hübner and Schaal (2016) proposed a shelf space planning problem that includes stochastic demand, the number of facings and their shelf position in retail sector.

3. ALGORITHMS

3.1. Genetic Algorithms

GAs receive their name from the explanation of the manner with their behavior intuitively. Darwin explained it in his natural selection theory. A set of solutions is defined first and then obtain to the replacement of these solutions with better choices based on some fitness criteria, typically the value of the objective function (Burkard and Çela, 1996). Darwin's principle is more likely to encourage the genes to the next generations (Radhakrishnan et al., 2009)

This algorithm is a main algorithm of a powerful and robust approach for developing heuristics for combinatorial optimization problems with large-scale. The motivation underlying GAs can be defined with evolution. The "evolution" term expresses extremely successful in developing complexity and well adapted classes throughout relatively simple evolutionary mechanisms. Evolution theory is using for solving the combinatorial optimization problems. This theory helps solving problems in real life, because of the advantage of evolutionary phenomenon in practice. Holland's research (1975) with using GAs suggestion is the first answer for the usage of this theory (Ahuja et al., 2000).

In evolutionary algorithms, three main algorithmic developments are seen in previous stages. These algorithms are proposed independently and, although these algorithms initially are proposed in the 1960s and 1970s, only in the beginning of the 1990s, the researchers sense the common underlying principles of these approaches (Stützle, 1998).

GAs are a specific case of evolutionary algorithms and randomized search methodology. Evolutionary algorithms based on population, adaptive search algorithms forced to solve the optimization problems (Erol, 2010).

There are three major operators called as "selection", "mutation", and "recombination". Fitter individuals that selected for the next generation refers the "selection" operator. Mutation expresses a "unary operator" that provides random modifications to each individual. Recombination means the combination of the material genetically of two individuals, also called parents, through a crossover operator to generate new individuals (Stützle, 1998).

An array of bits is used for the standard representation of the solution in GAs. For the arrays of other types and structures, the same representation structure is used. The solutions are called chromosomes, when they become together and making a set it is called population (Obitko, 2016).

Figure 3.1 shows the scheme of the genetic algorithms based on the new replacement strategy (Stützle, 1998). GAs reproduce the evolution process on an optimization problem. Each feasible solution of a problem is a treatment like an individual in which includes fitness value, means the value of the objective function (Ahuja et al., 2000).

GAs is also a robustness search techniques and this algorithm can be proved not only theoretically but also empirically (Radhakrishnan et al., 2009). This algorithm has stopping criteria when it has a running time as a pre-defined amount of time, or an optimization constraint is met (Vrcan and Lovrin, 2010). In view of the fact that GAs is one of the most widely used metaheuristic methods (Jurkovic, et al., 2009; Oiso, et al., 2011), it can be used for SSAP (Bilsel et al., 2013).

A function, called objective function, assigns a fitness value to each chromosome in every generation in order to use the best individual through evolutionary process (Radhakrishnan et al., 2009). The fitness function is defined over the representation of genetic and measures the *quality* of the solution representation (Obitko, 2016). In accordance to the objective, the fitness function evaluates the individuals. Each of the chromosomes is measured by using a fitness value and for each chromosome, a fitness value is calculated.

begin

create initial population;

for every generation, repeat

randomly select individuals into the mating pool;

apply genetic operators to generate offspring;

(maybe) apply post-crossover heuristic on offspring;

if not in *generation size* generations

compare every offspring with its similar parent and remove the worse one;

else

compare every offspring with the worst member in the current population and remove the worse one

until some stopping criterion is met

end;

Figure 3.1. The structure of the genetic algorithms based on the replacement strategy (Ji, et al., 2006)

Bilsel et al. (2013) used GA for the shelf space allocation problem instance. They used mathematical model for the SSAP. the population size is fixed at 100, parents were randomly selected at each iteration based on their rank and 50 children are generated. Each individual of the population is subjected to mutation value which is 0.02. A survival criterion is based on fitness and worst solutions were removed from the solution. 300 runs used for the implementation. Mutation operator, random survival, different aspiration criteria, the examination of a random subset of the neighborhood parameters are used in study. And as a result, the outcomes of the study are;

- Tabu Search had better solutions than greedy algorithms on average. These findings contrast with findings of Lim et al. (2004), where Tabu Search failed to outperform the greedy algorithm.
- On average GA has given better results, although it took more time to obtain the solution.
- Variability suggests the sensitivity of TS algorithm to initial solution (Bilsel et al., 2013).

Bilsel et al. (2013) proposed SSAP problem with Simulated Annealing (SA) algorithm. The neighborhood consists of 200 randomly chosen neighbors. A move being a tabu consideration means if it makes a change of an item that the result was recently changed. Size of the tabu list was identified as 10 and a tabu move was allowed only if it obtains a better solution than the best known solution. This process runs 200 times.

Table 3.1. Comparison of metaheuristics and greedy algorithm for SSAP

	Min z	**Mean z**	**Max z**	**Average runtime**
Tabu Search	106.35	155.31	193.90	200.81
Genetic Algorithm	145.95	162.38	176.25	253.13
Greedy algorithm (Yang, 2001)	135.9	135.9	135.9	
Greedy with Improvements algorithm (Ayhan, et al., 2007)	146.1	146.1	146.1	

3.2. Simulated Annealing Algorithm

SA is an algorithm which is used as a local search technique for combinatorial optimization problems and from statistical mechanics problems. Kirkpatrick, Gelatt and Vecchi (1983), and Cern`y (1985) are among the first authors used Metropolis algorithm. They illustrated that the simulation of the behavior of a physical many particle system can be applied as a heuristic for the other problem types. Kirkpatrick et al. proposed SA, which is an iterative local search method combined the "physical solids (crystals) annealing" and combinatorial optimization problems. It is used as an optimization algorithm (Stützle, 1998). They showed how the Metropolis algorithm uses a thermodynamic system behavior in "stability" at a current temperature, according to approximation numerical simulation of the behavior of a many body system at a finite temperature provides a natural tool for bringing the techniques of statistical mechanics to optimize (Kirkpatrick, et al., 1983) (Binguler, et al., 2016).

The structure between a many particle physical system and a combinatorial optimization problem principally withstanding on two types of facts:

- The combinatorial optimization problem's feasible solutions equivalent to the states of the physical system.
- The values of objective function equivalent to the states of the physical system energy.

The *simulated annealing* method reorder the *annealing process* to the optimization problem solution: the objective function, similar to the energy of a material, is then minimized (Dreo, et. al., 2006).

Practically, the technique is exploited by the Metropolis algorithm. In configuration stage, the basic requirements are determined for instance, initial placement of components and initial temperature can be defined. The principle of simulated annealing algorithm is shown in Figure 3.2. The probability must be identified. This probability affects the objective function as an increase. This process is repeated in a manner; while taking the current temperature, the stopping criteria is thermodynamic balance, a "sufficient" iteration of modifications, iteratively. According to the program termination criterion, the temperature can be lowered. In this stage, temperature value is defined according to the law of decrease you identified (Dreo, et. al., 2006).

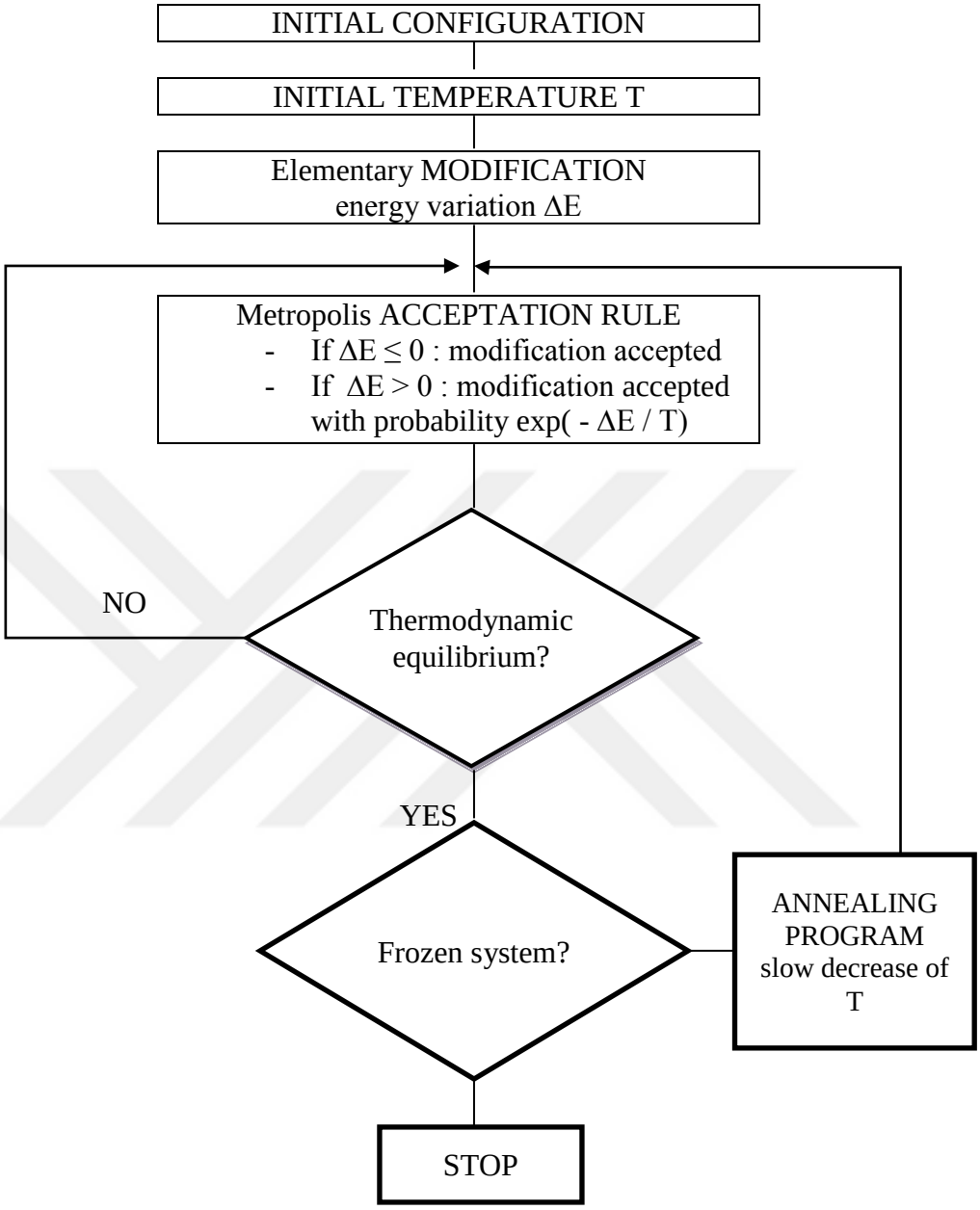

Figure 3.2. The principle of the simulated annealing algorithm (Dreo, et. al., 2006)

In Figure 3.3, a general algorithmic outline is given for SA.

```
procedure Simulated Annealing
generate initial solution k, k_best = k,
initial value for T0, i=0,
while outer-loop criterion not satisfied
do
  while inner-loop criterion not satisfied
do
    k' = Generate_Random_Solution (k)
    k = Acc_Solution (Ti ; k ; k')
    if (f(k) < f(k_best)
      k_best = k
  end
  Ti+1 = Update_Temp ( Ti ), i = i+1
end
return k_best
end Simulated Annealing
```

Figure 3.3. Pseudocode of a simulated annealing algorithm (Stützle, 1998)

In SA algorithm *annealing (cooling) schedule* must be determined. T_0 is initial temperature, and temperature is updating (UpdateTemp) as seen in Figure 3.3. In inner loop criterion, the number of iterations is performed at each temperature and a termination condition is defined in outer loop criterion (Stützle, 1998).

Inhomogeneous ergodic Markov chain is used for SA modeling, and this model has been used for the probabilistic analysis of the convergence of SA algorithms. Under some restrictions in neighborhood structure and non-restrictive conditions on the cooling process slowness, it is obvious to see that asymptotically SA converges to an optimal solution of a considered problem. The analysis of this convergence speed remains an open problem also apparently difficult.

The disadvantages of SA are about the temperature reduction; because the user should have the know-how of "good" adjustments. In addition, the computing time can become very significant, which led to parallel implementations of the method (Dreo, et al., 2006). A multi stage SA based hyper-heuristic algorithm proposed to solve both joint optimization models. These models are then evaluated numerically for different problem sizes (Zhao, et al., 2016).

Binguler et al. (2016) suggested a heuristic algorithm for SSAP using SA structure. Determining the initial temperature and the freezing temperature or the condition of convergence are very vital in increasing the accuracy and speed of the SA algorithm. For this reason, many tests are performed to find effective temperatures. The trial and error

method is used for repetition rate. New generation produced like, the switching operator was used in the inner loop thus, this operator found the best possible result. Proposed model was developed in a Java program (Binguler et al., 2016).

3.3. Variable Neighborhood Search Method

Variable Neighborhood Search (VNS) is based upon a simple principle: systematic change of neighborhood within the search. Several dozen papers already published for VNS. Many extensions have been made and research papers mainly proposed to allow solution of large problem instances. In most of them, the aim is to keep the simplicity of the basic scheme.

VNS is a metaheuristic based on systematically changing of neighborhood set. Usual heuristic searches are based on transformations of solutions that determine one neighborhood structure on the solution space. VNS uses a series of neighborhood structure. The basic idea of the VNS is to change the neighborhood used when the local search is fascinated at a local minimum. During the past decade, this method has been successfully applied to a wide range of rich vehicle routing problems (Wen, et. al., 2011).

VNS is a stochastic local search method that is based on the systematic change of the neighborhood during the search. This technique can be seen as a simple and effective method for solving single-objective optimization problems, including combinatorial optimization problems and scheduling problems (Gorkemli and Karaboga, 2013).

The classical VNS algorithms' concept includes a set of neighborhood search techniques systematically to obtain the optimum or near-optimal solution. The basic solution in VNS algorithm is comparing with the neighboring solutions and updating neighborhoods during the search process. Benchmark optimization problems with single objective are established using constraints with different levels (Arroyo, et. al., 2011).

Almost all methods, in the previous researches, take care of and solve the instances independently and separately. Only one instance was solved on each run which usually deals with the restriction of the search information sharing. However, from the observation on the searching process, it is obvious to obtain that the info overlap among instances with adjacent levels of constraints (Arroyo, et al., 2011) (Hansen and Mladenović, 2001).

This method includes the idea changing of neighborhood systematically, both in descent to local minima and in escape from the valleys which contain them. VNS seriously relies upon three following observations (Hansen and Mladenović, 2001):

Fact 1. A local minimum for one neighborhood structure is not necessarily a local minimum for another neighborhood structure.
Fact 2. A global minimum is a local minimum for all possible neighborhood structures.
Fact 3. Local minima with respect to one or several neighborhoods are relatively close to each other.

This final observation is that a local optima often provides some information about the global optima. In this case, several variables can be with the same value with each other. On the other hand, it is not usually known which have the same value. Therefore, a VNS structure searches different neighborhoods of a local optimum systematically (Hansen and Mladenović, 2010).

4. RESEARCH METHODOLOGY

4.1. Test problems

The test case involves 3 shelves and 7 products. The length of product vary between 2 and 6 units while each shelf's length is 30 units, means that at most fifteen items can be placed on a shelf. For the SSAP, in each solution, its neighborhood includes N*M*(N*M-1) possible swaps of two items plus N*M*K possible alterations of an item. Table 4.1 illustrates the profit of the products on shelves, the length of shelves and the length of products. Table 4.2 and Table 4.3 shows the cross price elasticities in the case of the product *i* is to the right of the product *j* on the shelf, the lower and upper bound to allocate products to the shelves, respectively.

Table 4.1. The profit and length of products (Yang, 2001)

Product/Shelf	0	1	2	Length
Product 0	1.0	2.0	3.0	3 cm
Product 1	1.5	2.0	3.0	5 cm
Product 2	2.0	3.0	4.0	4 cm
Product 3	2.5	3.5	4.0	6 cm
Product 4	1.0	2.0	3.0	2 cm
Product 5	3.0	4.0	5.0	3 cm
Product 6	6.0	1.0	7.0	2 cm
Shelf Length	30 cm	30 cm	30 cm	

Table 4.2. The cross price elasticities of products (Yang, 2001)

Product	0	1	2	3	4	5	6
0	1.0	0.9	-0.5	-0.5	0.1	0.1	-1.0
1	0.9	1.0	-0.5	0.5	0.1	0.1	-1.0
2	-0.5	-0.5	1.0	0.9	0.5	0.5	-0.5
3	-0.5	0.5	0.9	1.0	0.5	0.5	-0.5
4	0.1	0.1	0.5	0.5	1.0	0.9	0.3
5	0.1	0.1	0.5	0.5	0.9	1.0	0.2
6	-1.0	-1.0	-0.5	-0.5	0.3	0.2	1.0

Table 4.3. The lower and upper bound to allocate products to the shelves (Yang, 2001)

Product	LOWER BOUND	UPPER BOUND
0	1	5
1	1	5
2	2	8
3	2	8
4	3	∞
5	3	∞
6	1	∞

The distribution of generated data is Uniform distribution for this implementation is as follows;

- Profit ~U(0.5, 7).
- The length of product ~ U(2, 5).
- Cross price elasticity ~ U(-0.1, 1).
- Lower/upper bound of product that must be allocated to shelves ~ U(1, B). (B is calculated as; the number of shelf multiplies by the length of shelf divided to the minimum length of product).

4.2. Methods

Neighborhood search algorithms (*local search algorithms*) are on the behalf of improvement algorithms class, an improving solution is found by changing the "neighborhood" of the current solution and updating it in each iteration. Choosing the neighborhood structure is an essential issue in a neighborhood search algorithm design, and the definition of the neighborhood (Ahuja et al., 2002) and its constraints. Neighborhood search method is used for allocation. The products' location is changed in a shelf. This is called "Neighborhood Structure within shelves".

The products' location can also be changed between shelves. This is called "Neighborhood Structure between shelves".

The test case involves 3 shelves and 7 products for benchmarking other researches. Product lengths vary between 2 and 6 units while each shelf's length is 30 units, meaning that at most 15 items can be placed on a shelf.

For this study, 180 different problems are generated. In implementation, data set includes by;

- The number of product is chosen as 7, 8, 9 and 10.
- The number of shelves is chosen as 3,4 and 5.
- The length of shelf is chosen as 20, 30 and 40.

There are 4*3*3= 36 alternatives for test data. The model is run on five problems with different values is each problem combination.

4.2.1. Initial Solution

This study starts with Yang's method for SSAP. Yang (2001) used test problem with 3 shelves and 7 products. His method includes the profit of the products, the profit of the products on shelves, the length of shelves and the length of products. In this study, Yang's method used in first step for solution techniques (as seen in Figure 4.1). The same structure is used for proposed VNS algorithm.

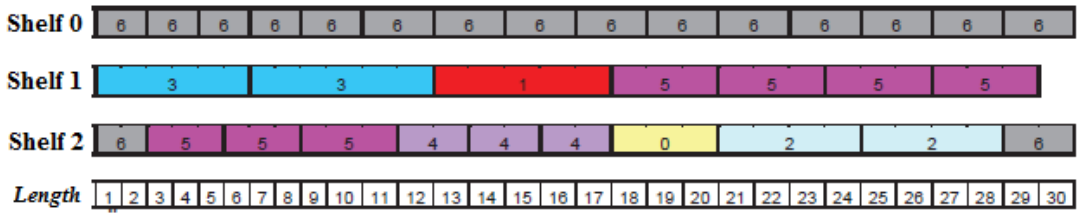

Figure 4.1. Shelf space allocation in initial solution (Ayhan et al., 2007)

4.2.2. Improvement techniques

Ayhan et al. (2007) have used the problem case which was solved with two different metaheuristic algorithms in methods parts. Even for small test case, this amount to 1980 swaps and 315 alterations. Yang's method is improved for SSAP in this research. The objective function is defined as maximizing the profit of the allocation, also the capacity of shelf, the profit of product in each location, cross elasticity factor between products (iin the case of the product *i* is to the right of the product *j* on the shelf), and the lower and upper bound of each product, in each shelf, is defined.

For test problems, second improvement method is Ayhan et al.'s greedy with improvements for solving SSAP. The solution of Ayhan et al.'s study is illustrated in Figure 4.2. In greedy improvement techniques, in this study, "Neighborhood Structure within shelves" and then "Neighborhood Structure between shelves" method was used only once.

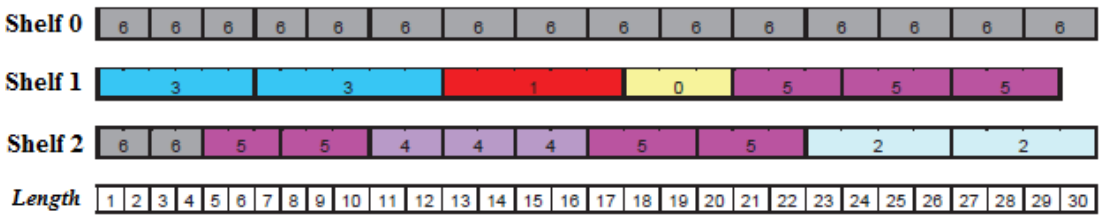

Figure 4.2. Shelf space allocation with improvement (Ayhan et al., 2007)

These models, initial solution (Yang's method), improvement technique and VNS algorithm, are evaluated numerically for different problem sizes.

4.3. Methodology and Problem Definition

P_{ik} is the profit of the product *i* on shelf *k* , X_{ilk} is the decision variable to identify if product *i* is in the lth *position* on shelf *k*, E_{ij} is the cross price elasticity if the product *i* is to the right of the product *j* on the shelf, then the objective function can be formulated as:

$$Max\ P = \sum_{k=1}^{K}\sum_{i=1}^{I} P_{ik} * X_{i1k} + \sum_{k=1}^{K}\sum_{i=1}^{I}\sum_{l=2}^{L}\sum_{j=1}^{I}(P_{ik} * X_{ilk}) * (E_{ij} * X_{j(l-1)k}) \quad (1)$$

Subject to:

$$\sum_{i=1}^{I}\sum_{l=1}^{L} a_i * X_{ilk} \le T_k \qquad k = 1,2,\dots,K\ (shelf\ space\ constraint) \quad (2)$$

$$\sum_{l=1}^{L}\sum_{k=1}^{K} X_{ilk} \ge L_i \qquad i = 1,2,\dots,I\ (lower\ bound\ for\ each\ product) \quad (3)$$

$$\sum_{l=1}^{L}\sum_{k=1}^{K} X_{ilk} \le U_i \qquad i = 1,2,\dots,I\ (upper\ bound\ for\ each\ product) \quad (4)$$

$$X_{ilk} = \begin{cases} 1 \text{ if the product } i \text{ is located to the } l^{th} \text{ position in the } k^{th} \text{ shelf} \\ 0 \text{ otherwise} \end{cases}$$

Where:

k = 1,2, …, K *the number of shelves*
i, j= 1,2, …, I *the number of products*
l = 1,2, …, L *the position of products*
T_k *: the length of shelf k*
a_i *: the length of product i*
L_i *: the lower bound to allocate product i*
U_i *: the upper bound to allocate product i*

In objective function, for the first allocation in shelf, profit value is calculated first with product's profit value in located shelf. For other allocations, if product *i* is located to the lth position in shelf *k*, its own profit value is calculated and its cross price elasticity if the product *i* is to the right of the product *j* on the shelf is calculated. These calculations are completed for each shelf. The sum of these calculations of the shelves gives the total profit (objective function value) of the allocation of products. The aim is to maximize the total profit.

Eq.2 shows that the products are allocated in each shelf, according to the length of the shelf. Each products must be assigned with a limitation, between lower bound and upper

bound. These limitations are illustrated in Eq.3 and Eq.4.

4.4. Steps of variable neighborhood search method

Unlike most local search heuristics that use only a simple neighborhood structure, a VNS structure uses a finite set of pre-selected neighborhood structures denoted by N_k, for $k=1,\ldots,k_{max}$ and the set of solutions in the kth neighborhood of x is denoted by $N_{k(x)}$. Neighborhoods N_k may be obtained by different local search approaches into a solution space *S*. A best solution (hopefully global minimum) x_{best} is a feasible solution where a minimum of the solutions in a neighborhood k is reached. The steps of a basic VNS structure are defined as in the Figure 4.3 (Hansen and Mladenović, 2001).

Initialization:

- Determine the set of N_k, *for* $k=1,\ldots,k_{max}$, *that* will be used in VNS;
- Find an initial solution x by any construction heuristic and its objective function value $f(x)$, set
$x_{best} \leftarrow x$, $f_{best} \leftarrow f(x)$; choose a stopping condition;

Repeat the following operations as long as the stopping conditions are not met:
(1) Set $k \leftarrow 1$;
(2) Repeat the below steps until $k = k_{max}$:
(a) ***Shaking.*** Generate a random solution point *x' and in the* k^{th} neighborhood of x;
(b) ***Local search.*** Find a solution point x'' as the local optimum applying some local search method with *x'* as initial solution;
(c) ***Check for improvements.*** If $f(x')$ is better than $f(x)$, set $f_{best} \leftarrow f(x'')$ and $x_{best} \leftarrow x''$ and $k=1$, otherwise set $k=k+1$
(or if $k = k_{max}$, set $k=1$); go to Step 1.

Figure 4.3. Steps of basic VNS algorithm (Hansen and Mladenović, 2001).

In Step 1, x' is generated at random to avoid cycling (Successive N_k are often nested). In Step 2, if incumbent is changed then started over with N_1, otherwise continue search in N_{k+1} starting with the local optimum of N_k (Hansen and Mladenović, 2001).

Other stopping condition is met (e.g. maximum number of iterations, maximum Central Process Unit (CPU) time allowed, or maximum number of iterations between two improvements) in application. Often successive neighborhoods N_k was nested (Mladenović and Hansen, 1997) (Mladenović and Urošević, 2011).

"Neighborhood Structure within shelves" and "Neighborhood Structure between shelves" method is used together and this is called "VNS algorithm" for test problems. Test problems' value is generated randomly. Proposed VNS method is motivated by an analogy

between the neighborhood structure and SSAP. For algorithm coding, Java, Visual Studio, C# and .Net are used.

4.5. Validation of the Model

The improvement over other solutions in Table 5.1 illustrates VNS performs the best allocation solution. In order to benchmark, the same test problem is used for every methods initially. For each problem sets, 5 problems are generated with random number generator.In this study, the number of neighborhood structures (k_{max}) is chosen as 3. For the SSAP, these neighborhoods are;

a. "Neighborhood Structure within shelves"

Possible swaps of two products are considered within shelves including lower bound of products, upper bound of products, the length of shelves, the length of products, and the profit of the products.

An example of this structure that is used in swap operation for shelf 2 is illustrated in Figure 4.4.

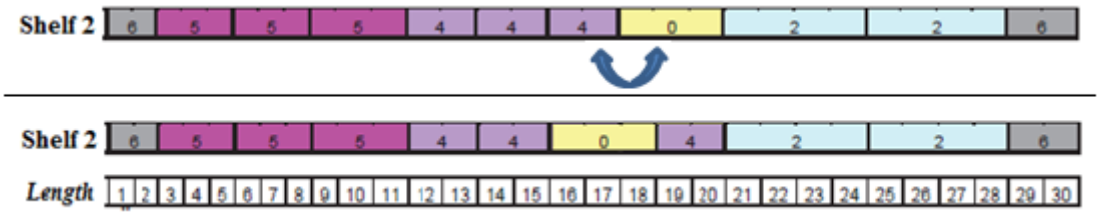

Figure 4.4. Swap operation within shelves

b. "Neighborhood Structure between shelves"

The set of solution includes the neighborhood structure which is the swaps of two products between shelves considered by the length of products, unused shelf space, lower bound of products, upper bound of products, the profit of the products, and the length of shelves.

The neighborhood structure between shelf 1 and shelf 2 that is used in this part is shown in an example in Figure 4.5.

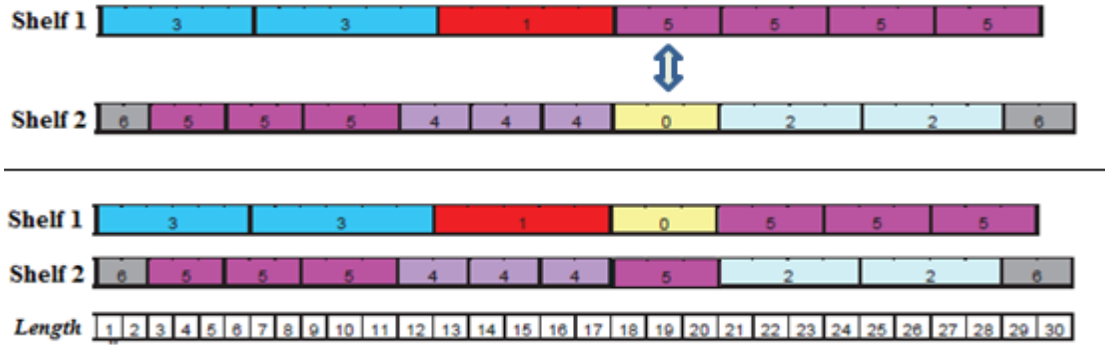

Figure 4.5. Swap operation between shelves

c. Adding new product in shelves

Products are added to shelves with possible replacement and with the constriction such as lower bound of the products, upper bound of the products, the length of products, the length of shelves, the profit of products.

Figure 4.6 illustrates an example of this neighborhood structure in shelf 1.

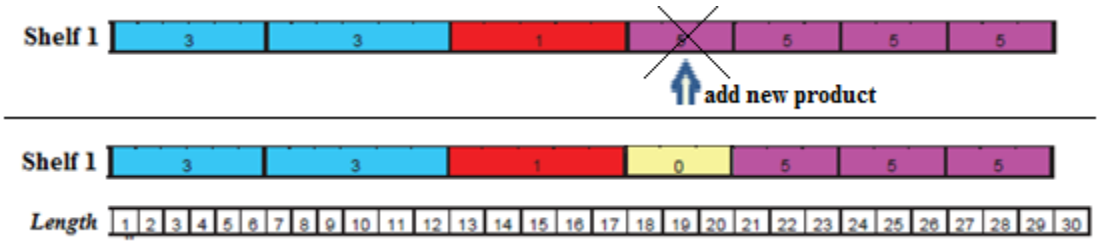

Figure 4.6. Adding new product in shelf 1

Yang's method which includes the profit of products used in first step for test problem. Figure 4.7 illustrates the initial solution of the proposed algorithm (VNS) for test problem.

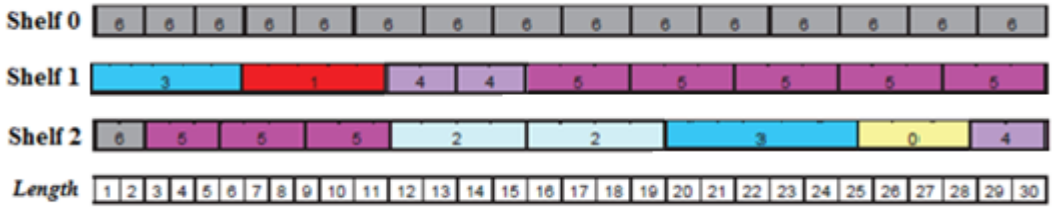

Figure 4.7. Initial solution for test problem

VNS algorithm includes the neighborhood structures such as "Neighborhood Structure within shelves", "Neighborhood Structure between shelves", and adding new product in shelves. Figure 4.8 illustrates the solution of VNS algorithm for test problem.

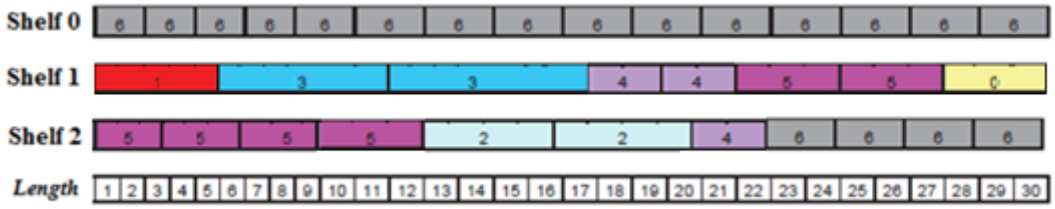

Figure 4.8. VNS solution for test problem

Figure 4.9 illustrates the solutions (total profit) of each algorithm for problem set which consist of the number of product is 7, the number of shelf is 3, and the length of shelf is 30. The graphs are scaled and drawn. In Appendix, figures and tables of results are shown for other problems. The results of 5 different problems with different values for each problem set with different number of product, shelf and the length of shelf. VNS algorithm

gives the same or better results from other solution than any other methods referred in this study for test problems.

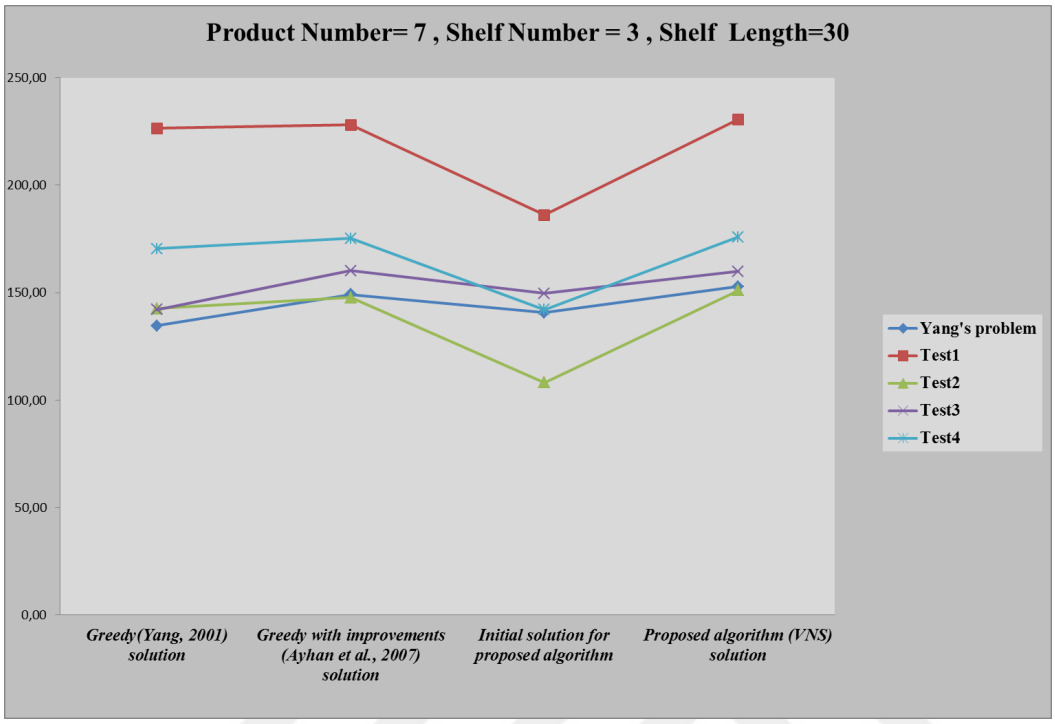

Figure 4.9. The results for problem instances (I (Number of product)=7/K (Number of shelf) =3/ T (Length of shelf) =30)

5. RESULTS and DISCUSSIONS

5.1. Results

The percentage of improvement of VNS algorithm over three solutions of the problem instances is shown in Table 5.1. These problem instances are generated randomly to be solved with different algorithms for comparison.

Table 5.1. Improvement over solutions

The number of product	The number of shelf	The length of shelf	Improvement Over Solutions (%)		
			Yang's Solution	Ayhan et al. solution	Initial Solution
7	3	30	7.41	1.21	20.53
7	3	20	17.25	2.70	23.42
7	3	40	3.32	-0.26	23.25
7	4	30	3.48	0.52	18.80
7	4	20	8.42	1.56	22.93
7	4	40	2.33	0.28	17.16
7	5	30	3.22	0.16	20.56
7	5	20	4.86	1.18	22.23
7	5	40	2.27	0.72	11.92
8	3	30	7.32	1.24	12.26
8	3	20	11.33	0.20	12.20
8	3	40	4.89	0.48	18.37
8	4	30	5.29	0.57	8.37
8	4	20	8.65	1.18	18.12
8	4	40	2.86	0.86	19.10
8	5	30	3.09	0.49	20.66
8	5	20	5.57	1.18	29.52
8	5	40	2.96	0.59	28.87
9	3	30	9.14	1.44	17.88
9	3	20	8.68	-1.26	26.31
9	3	40	5.49	0.58	16.90
9	4	30	4.63	1.55	40.97
9	4	20	6.22	0.33	16.02
9	4	40	4.00	0.92	17.94
9	5	30	3.04	0.29	23.95
9	5	20	6.02	0.40	13.39
9	5	40	2.52	0.23	11.79
10	3	30	9.40	0.82	31.01
10	3	20	17.33	2.50	21.94
10	3	40	6.52	1.39	15.53
10	4	30	5.63	0.59	16.42
10	4	20	7.95	0.99	22.93
10	4	40	3.24	0.33	25.84
10	5	30	4.23	1.46	10.83
10	5	20	7.14	0.15	16.79
10	5	40	2.92	0.19	31.10

Table 5.1 includes; the percentage of improvement of VNS algorithm solution over Yang's solution, the percentage of improvement of VNS algorithm solution over the greedy algorithm of Ayhan et al. and the percentage of improvement of VNS algorithm solution over the initial solution in this study, the number of product, the number of shelves, and the length of shelf. For two problem sets, the greedy algorithm of Ayhan et al. gives better solutions than VNS algorithm because of random generated data. It is measured for different problems; VNS algorithm gives better results for different runs.

5.2. Conclusion and Future Researches

As can be seen from Table 5.1, the proposed VNS algorithm performs better than Yang's method and gives the same or better solutions from Ayhan et al.'s heuristics for different problem sets. For the best (max) solution, VNS outperforms Yang's method and almost most of the problems VNS algorithm outperforms Ayhan et al.'s heuristics.

Elasticity factor or cross elasticity determined for test problems, and data set are generated.

This study can be adapted to other examples or a real life problem instances in retailing industry, considered by using simple, specific rules and other restriction in real life problems. For considering operational choices, this study can be a guideline for retailing industry especially for allocation; the decision maker can easily use to practically apply decisions for shelf space allocation. Compare the model results with other approaches applied in practice can be a roadmap. After using this proposed (VNS) method, the anxiety of reaching optimum levels can obviously be decreased. This method can help firms become more practicable and simple about reaching to the optimum levels of performance. This efficient model can derive rules-of-thumb for decision-maker.

Even though the shelf space variation is a management tool that leads to short term sales response, the results of variations can be a sign of the space variation effects overtime or long term space variation management strategies (e.g. increase followed by reduction of shelf space) is a wide range issue for future research.

Also, for real life problem, according to data structure and new constraints, hybrid heuristic algorithms may be created in practice.

As a future project, cross elasticities may be studied more comprehensively in order to reveal the effect of influencing factors like different product types; products have large scale size and estimations. Findings can provide several implications such as specific market research information, contradictory statements, and action forecasting information for managers and researchers.

The elasticity factor's standard errors that are combined to products cannot be estimated and available in various researches; not only the number of products but also the decisions with these estimation results. In future studies, the estimations and standard error of elasticity factor can be added to the model, the results can be compared with the solutions of VNS algorithm.

THIS PAGE INTENTIONALLY LEFT BLANK

REFERENCES

Ahuja, R.K., Orlin, J.B., Tiwari, A., (2000) A greedy genetic algorithm for the quadratic assignment problem, Computers and Operations Research, 27(10), 917-934.

Ahuja, R.K., Ergun, O., Orlin, J.B., Punnen, A.P. (2002) A survey of very large-scale neighborhood search techniques Discrete Applied Mathematics, 123(1–3), 75-102.

Anderson, E.E. and Amato, H.N. (1973) A Mathematical Model for Simultaneously Determining the Optimal Brand-Collection and Display-Area Allocation, Operations Research, 22, 13-21.

Arroyo, J.E.C., Ottoni, R.S., Santos, A. (2011) Multi-objective Variable Neighborhood Search Algorithms for a Just-in-Time Single Machine Scheduling Problem, 978-1-4577-1676-8/11@ 2011 IEEE.

Ayhan M. B., Bulkan, S., Bilsel, M. and Gulcu, A. (2007) Raf Alanı Optimizasyonu. YA/EM Conference, Izmir, Turkey.

Bansal, N., and Sviridenko, M. (2007) Two-dimensional bin packing with one-dimensional resource augmentation, Discrete Optimization, 4 (2), 143–153.

Beasley, J.E. (1985) An exact two-dimensional non-guillotine cutting tree search procedure, Oper. Res. 33, 49–64.

Binguler, A.H.E., Bulkan, S., Agaoglu, M., (2016) A Heuristic Approach for Shelf Space Allocation Problem, Journal of Military and Information Science, 4 (1), 38-44.

Bilsel, M., Ayhan M. B., Bulkan, S. (2013) Shelf Space Optimization Using Metaheuristic Algorithms, International Journal of Information, Business and Management, 5 (2), 210-217, *ISSN 2076-9202.*

Bischoff, E.E., Janetz, F., and Ratcliff, M.S.W. (1995) Loading pallets with non identical items, European Journal of Operations Research, 84, 681-692.

Bortfeldt, A, and Mack, D. (2007) A heuristic for the three-dimensional strip packing problem. European Journal of Operational Research, 183, 1267–1279.

Bortfeldt, A., and Gehring, H. (2001) A hybrid genetic algorithm for the container loading problem, European Journal of Operational Research, 131(1), 143-161.

Burkard, R.E., Çela, E. (1996) Quadratic and three-dimensional assignments: An annotated bibliography, *Technical report 63*, Discrete Optimisation Group, Technische Universität Graz, Austria.

Carvalho, J.M.V. (1999) Exact solution of bin-packing problems using column generation and branch-and-bound, Annals of Operations Research, 86 (0), 629-659.

Cern`y, V. (1985) Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, Journal of Optimization Theory and Applications. 45, 41-51.

Chen, M.C., and Chia-Ping Lin, C.P. (2007) A data mining approach to product assortment and shelf space allocation, Expert Systems with Applications, 32 (4) , 976–986.

Corstjens, M., Doyle, P. (1981) A Model for Optimizing Retail Space Allocations, Management Science, 27, 822-833.

Dreze, X., Hoch, S.J., Purk M.E. (1994) Shelf Management and Space Elasticity, Journal of Retailing, 70, 301-326.

Dreo, J., Petrowski, A., Siarry, P., Taillard, E. (2006) Metaheuristics for Hard Optimization: methods and case studies, Springer. ISBN: 354023022X.

Eisend, M. (2014) Shelf space elasticity: A meta-analysis, Journal of Retailing, 90 (2), 2014, 168–181.

Erol, A.H. (2010) A heuristic solution algorithm for the quadratic assignment problems. Master thesis, Marmara University, Institute for Graduate Studies in Pure and Applied Sciences, Turkey; Supervisor: Asst. Prof. Dr. Serol Bulkan.

Fadıloğlu, M.M., Karaşan, O.E., and Pınar, M.Ç. (2007). A Model and Case Study for Efficient Shelf Usage and Assortment Analysis. Annals of Operations Research, 180(1), 105-124.

Fancher, L.A. (1991) Computerized space management: A strategic weapon, Discount Merchandiser, 31 (3), 64-65.

Fekete, S.P., Schepers, J. (1997) On more-dimensional packing I: Modeling, Technical Report ZPR97-288, Mathematisches Institut, UniversitYat zu KYoln.

Fekete, S.P. and Schepers, J. (2001) New classes of fast lower bounds for bin packing problems, Mathematical Programming, 91(1), 11-31.

Gajjar, H.K., Adil, G.K. (2011) Heuristics for retail shelf space allocation problem with linear profit function, International Journal of Retail & Distribution Management, Vol. 39 (2), 144-155.

Gilmore, P.C., Gomory, R.E. (1965) Multistage cutting problems of two and more dimensions, Oper. Res. 13, 94–119.

Gorkemli, B., Karaboga, D. (2013). Quick combinatorial Artificial Bee Colony -qCABC- Optimization Algorithm for TSP. ISCIM - International Symposium on Computing in Informatics and Mathematics, 97-101.

Hadjiconstantinou, E., Christofides, N. (1995) An exact algorithm for the orthogonal, 2-D cutting problems using guillotine cuts, Eur. J. Oper. Res. 83, 21–38.

Hansen, P., Mladenović, N. (2001) Variable neighborhood search: Principles and applications. European Journal of Operational Research, 130(3), 449-467.

Hansen, P., Mladenović, N., Pérez, J.A.M. (2010) Variable neighbourhood search: methods and applications. Ann. Oper Res., 175, 367-407.

He, D.Y., and Cha, J.Z. (2002) Research on solution to complex container loading problem based on genetic algorithm, Proceeding of First International Conference on Machine Learning and Cybermatics, Beijing, 4-5, 78-82.

Holland, J.H. (1975) Adaptations in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor.

Hübner, A., and Schaal, K. (2016) A shelf-space optimization model when demand is stochastic and space-elastic, Omega, In Press, Accepted Manuscript, Available online 13 August 2016.

Hwang, H., Choi, B., Lee, G. (2009) A genetic algorithm approach to an integrated problem of shelf space design and item allocation, Computers & Industrial Engineering 56, 809–820.

Ji, P., Wu, Y., Liu, H. (2006) A Solution Method for the Quadratic Assignment Problem (QAP), The Sixth International Symposium on Operations Research and Its Applications (ISORA'06), Xinjiang, China, August 8-12, 106–117.

Jurkovic, Z., Brezocnik, M., Grizelj, B. and Mandic, V. (2009) Optimization of extrusion process by genetic algorithms and conventional techniques, Technical Gazette, 16 (4), 27-33.

Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983) Optimization by simulated annealing. Science (220), 671–680.

Lim, A., Rodrigez, B. and Zhang, X. (2004) Metaheuristics with Local Search Techniques for Retail Shelf-Space Optimization, Management Science, Vol.50, 117-131.

Lodi, A., Martello, S., Vigo, D. (1999) Heuristic and Metaheuristic Approaches for a Class of Two-Dimensional Bin Packing Problems, Informs Journal of Computing, 11(4), 345-357.

Lodi, A., Martello, S., Vigo, D. (2002) Recent advances on two-dimensional bin packing problems, Discrete Applied Mathematics, 123 (1–3), 379–396.

Mladenović, N., Hansen, P. (1997) Variable neighborhood search. Computers & Operations Research, 24 (1), 1097-1100.

Mladenović, N., Urošević, D. (2011) VNS for the TSP and its variants. BALCOR. Thessaloniki, Greece.

Martinez-de-Albeniz, V. and Roels, G. (2011) Competing for Shelf Space, Production and Operations Management, 20(1), 32-46.

Michael, D., Moffitt, M.D. (2013) Multidimensional Bin Packing Revisited, Principles and Practice of Constraint Programming, Lecture Notes in Computer Science 8124, 513-528.

Obitko, M., Genetic Algorithms, [online], http://www.obitko.com/tutorials/genetic-algorithms/ga-basicdescription.php (Accessed date: May 10, 2016).

Oiso, M., Matsumura, Y., Yasuda, T. and Ohkura, K. (2011) Implementing genetic algorithms to CUDA environment using data parallelization, Technical Gazette, 18(4), 511-517.

Pisinger, D. (1995) Algorithms for Knapsack Problems, PhD Thesis, University of Copenhagen, Department of Computer Science, Denmark.

Ponce-Pérez, A., Pérez-Garcia, A., and Ayala-Ramirez, V. (2005) Bin-packing using genetic algorithms, Proceedings of the 15th International Conference on Electronics, Communications and Computers (CONIELECOMP 2005).

Radhakrishnan, P., Prasad, V.M., Gopalan, M.R. (2009) Optimizing Inventory Using Genetic Algorithm for Efficient Supply Chain Management, Journal of Computer Science, 5 (3), 233-241.

Raidl, G.R. (1999) Weight-codings in a genetic algorithm for the multiconstraint knapsack problem, Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on. Vol.1. IEEE, 596-603.

Stützle, T. (1998) Local Search Algorithms for Combinatorial Problems-Analysis, Improvements, and New Applications, *PhD Thesis*, FB Informatik, Technische Universität Darmstadt, Darmstadt, Germany, 28-108.

Vrcan, Z. and Lovrin, N. (2010) Genetic algorithm based optimization of conveyor belt material cross section area, Technical Gazette, 17 (2), 137-143.

Watson, T. (2007) Approximations Algorithms, Scribe: Lecturer: Shuchi Chawla, Topic: Bin Packing and Euclidean TSP, Date: 2/6/2007.

Wen, M., Krapper, E., Larsen, J., Stidsen, T.K. (2011) A multilevel variable neighborhood search heuristic for a practical vehicle routing and driver scheduling problem. Networks. 58:311-322.

Yang, M.H. (2001) An effective algorithm to allocate shelf space, European Journal of Operational Research, 131, 107-118.

Yang, M.H., Chen, W.C. (1999) A study on shelf space allocation and management, International Journal of Production Economics 60-61, 309-317, Elsevier.

Zhao, J., Zhou , Y.W., and Wahab, M.I.M. (2016) Joint optimization models for shelf display and inventory control considering the impact of spatial relationship on demand, European Journal of Operational Research, 255(3), 797–808.

Zufryden, F.S. (1986) A dynamic programming approach for product selection and supermarket shelf-space allocation, Journal of Operations Research Society, 37 (4), 413-422.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A

Table A.1. The results (profit) of solution techniques for problem (I=7/ K=3/ T =30)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Yang's problem	134.60	149.20	140.70	152.82	7	3	30
Test1	226.40	228.10	186.20	230.70	7	3	30
Test2	142.60	147.70	108.20	151.02	7	3	30
Test3	142.20	160.20	149.70	159.96	7	3	30
Test4	170.40	175.20	142.20	175.90	7	3	30

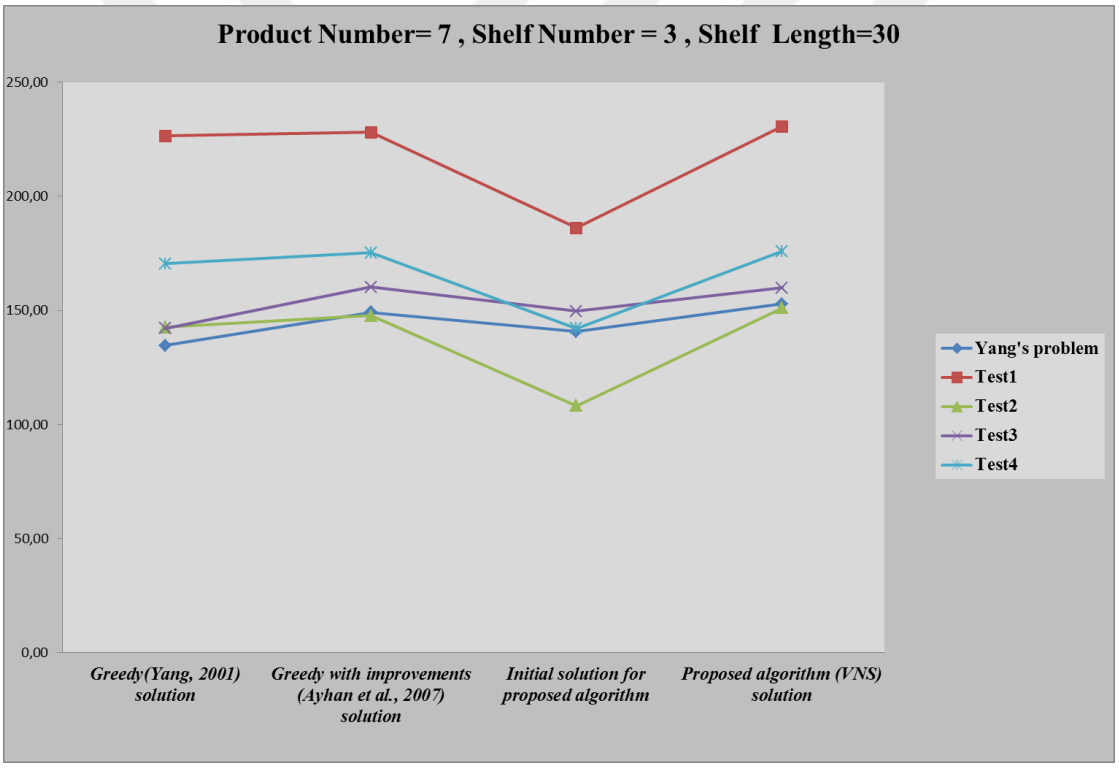

Figure A.1. The results for problem instances (I =7/K =3/ T =30)

Table A.2. The results (profit) of solution techniques for problem (I=7/ K=3/ T =20)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test5	78.30	93.10	74.30	91.26	7	3	20
Test6	82.20	90.10	73.40	95.22	7	3	20
Test7	95.70	114.80	83.50	114.00	7	3	20
Test8	74.30	82.30	86.70	88.80	7	3	20
Test9	75.30	84.60	69.10	86.78	7	3	20

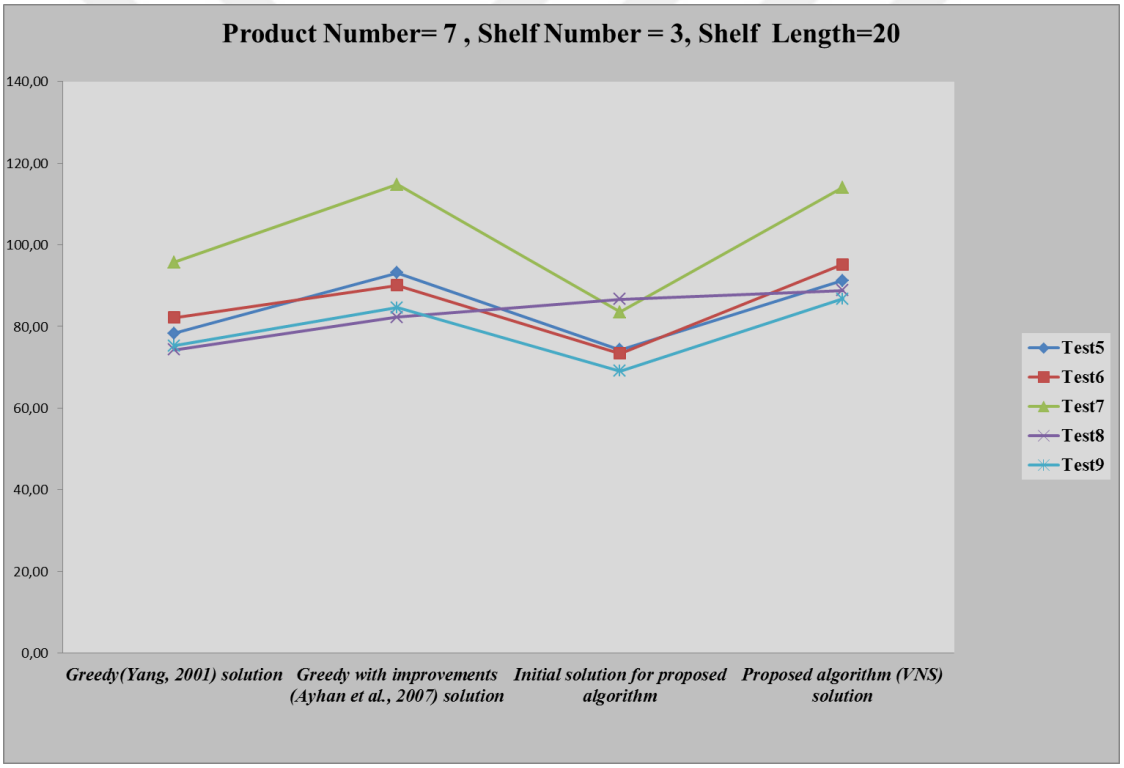

Figure A.2. The results for problem instances (I=7/ K=3/ T =20)

Table A.3. The results (profit) of solution techniques for problem (I=7/ K=3/ T =40)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test10	175.70	186.70	149.11	183.00	7	3	40
Test11	186.00	186.10	151.90	186.10	7	3	40
Test12	259.70	273.50	236.30	275.40	7	3	40
Test13	265.90	272.80	215.00	272.80	7	3	40
Test14	223.90	232.30	182.10	232.30	7	3	40

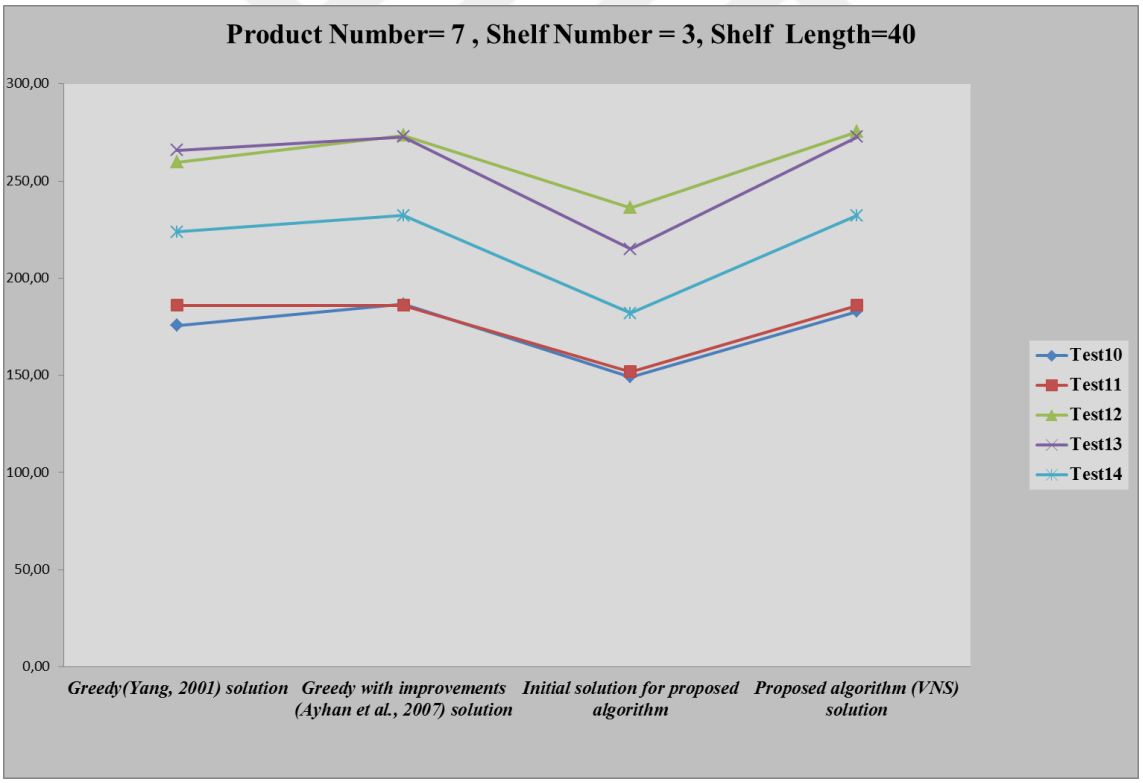

Figure A.3. The results for problem instances (I=7/ K=3/ T =40)

Table A.4. The results (profit) of solution techniques for problem (I=7/ K=4/ T =30)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test15	226.70	234.40	205.00	234.40	7	4	30
Test16	191.30	195.30	155.40	195.70	7	4	30
Test17	169.00	174.50	153.20	174.28	7	4	30
Test18	161.70	166.30	142.90	169.70	7	4	30
Test19	264.70	272.90	226.30	274.26	7	4	30

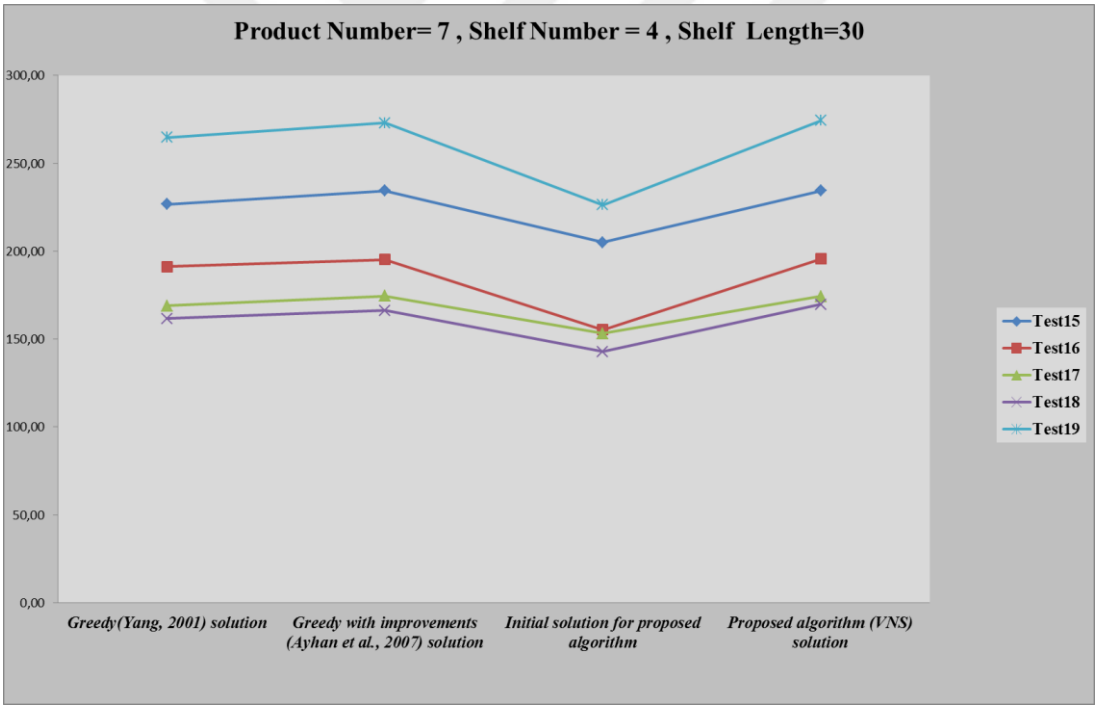

Figure A.4. The results for problem instances (I=7/ K=4/ T =30)

Table A.5. The results (profit) of solution techniques for problem (I=7/ K=4/ T =20)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test20	170.30	176.20	137.70	176.30	7	4	20
Test21	115.30	119.90	107.50	121.00	7	4	20
Test22	82.40	92.20	78.00	95.22	7	4	20
Test23	102.70	112.50	86.30	111.34	7	4	20
Test24	135.90	142.50	121.20	149.04	7	4	20

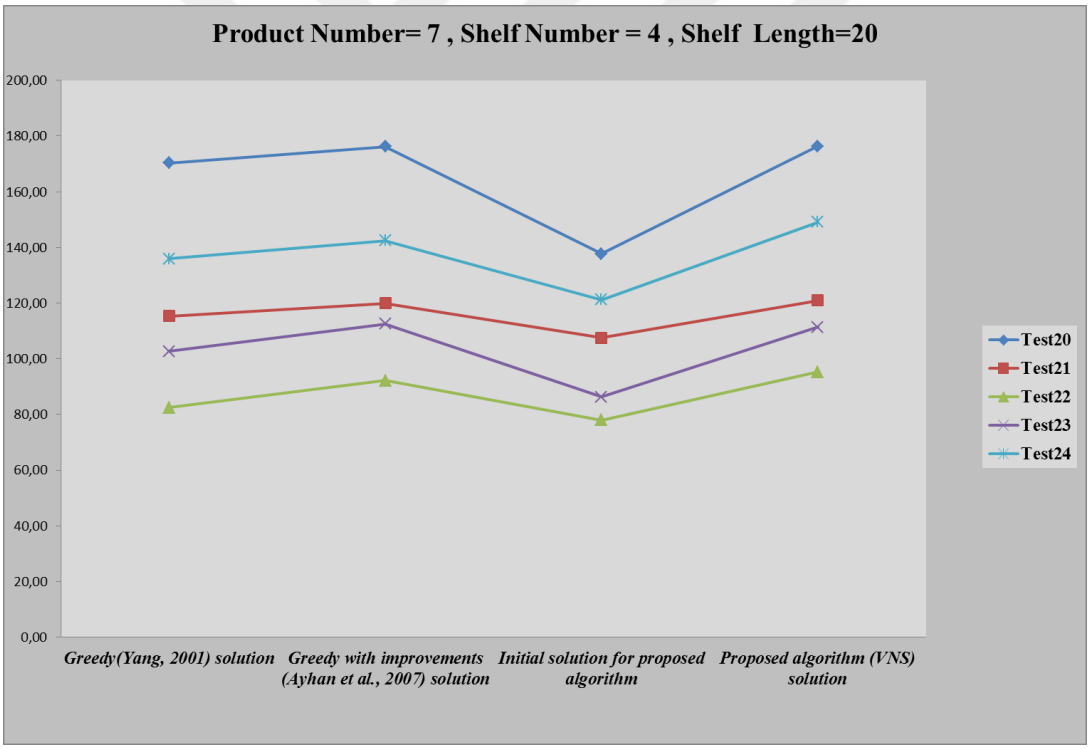

Figure A.5. The results of problem instances (I=7/ K=4/ T =20)

Table A.6. The results (profit) of solution techniques for problem (I=7/ K=4/ T =40)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test25	258.30	265.10	251.00	265.66	7	4	40
Test26	286.60	296.90	213.30	296.90	7	4	40
Test27	246.20	251.10	242.70	253.92	7	4	40
Test28	381.00	385.10	301.90	385.10	7	4	40
Test29	338.70	341.80	315.00	342.00	7	4	40

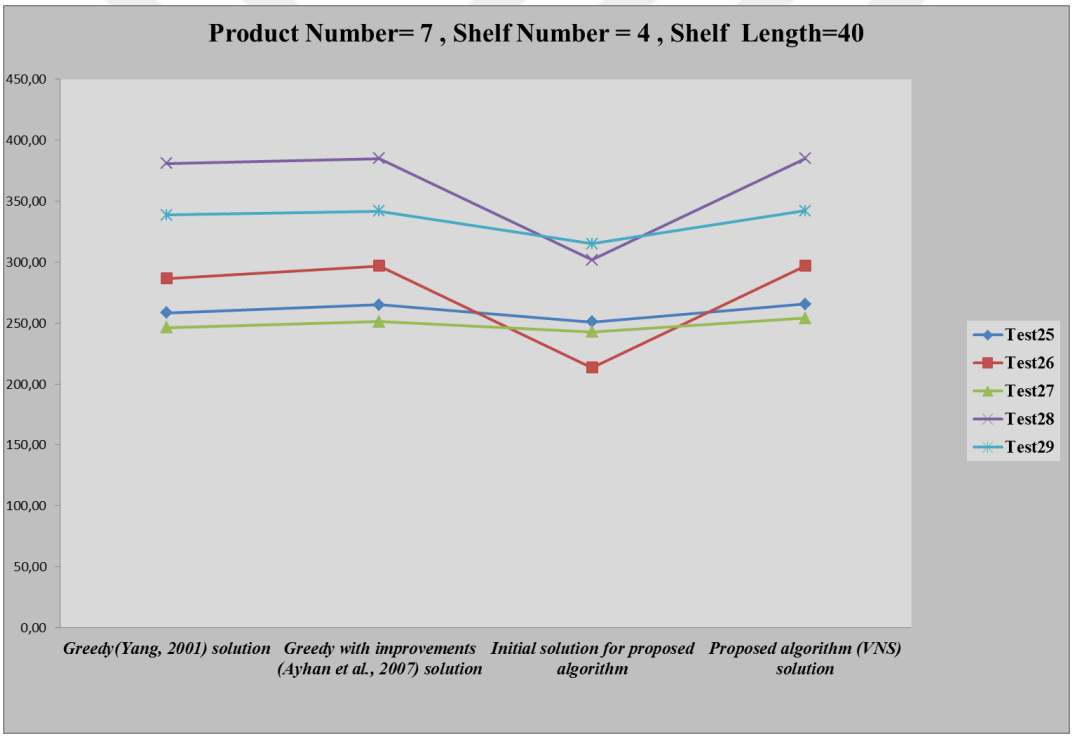

Figure A.6. The results of problem instances (I=7/ K=4/ T =40)

Table A.7. The results (profit) of solution techniques for problem (I=7/ K=5/ T=30)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test30	228.40	234.60	183.40	237.70	7	5	30
Test31	284.50	297.30	231.80	297.52	7	5	30
Test32	211.70	218.50	175.50	216.60	7	5	30
Test33	320.20	323.60	279.80	323.60	7	5	30
Test34	237.70	246.70	233.90	247.34	7	5	30

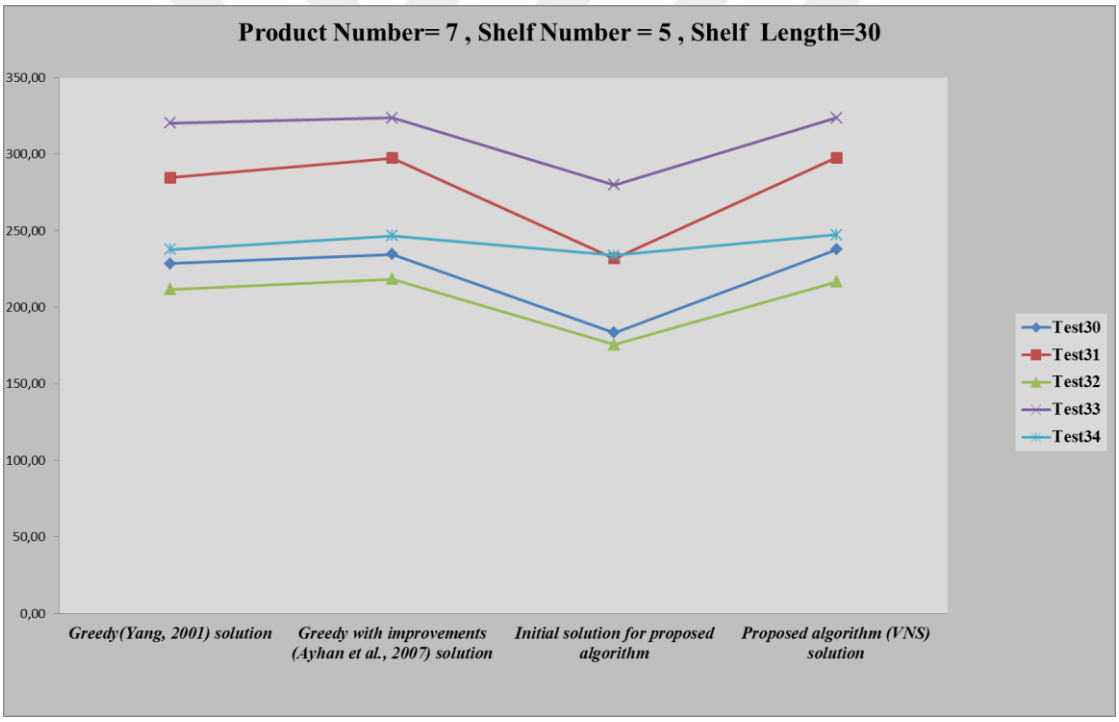

Figure A.7. The results of problem instances (I=7/ K=5/ T=30)

Table A.8. The results (profit) of solution techniques for problem (I=7/ K=5/ T =20)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test35	155.20	159.00	153.80	161.30	7	5	20
Test36	164.60	169.10	142.50	173.22	7	5	20
Test37	176.20	185.00	124.10	187.02	7	5	20
Test38	245.00	253.10	203.30	253.20	7	5	20
Test39	217.30	227.60	209.70	229.58	7	5	20

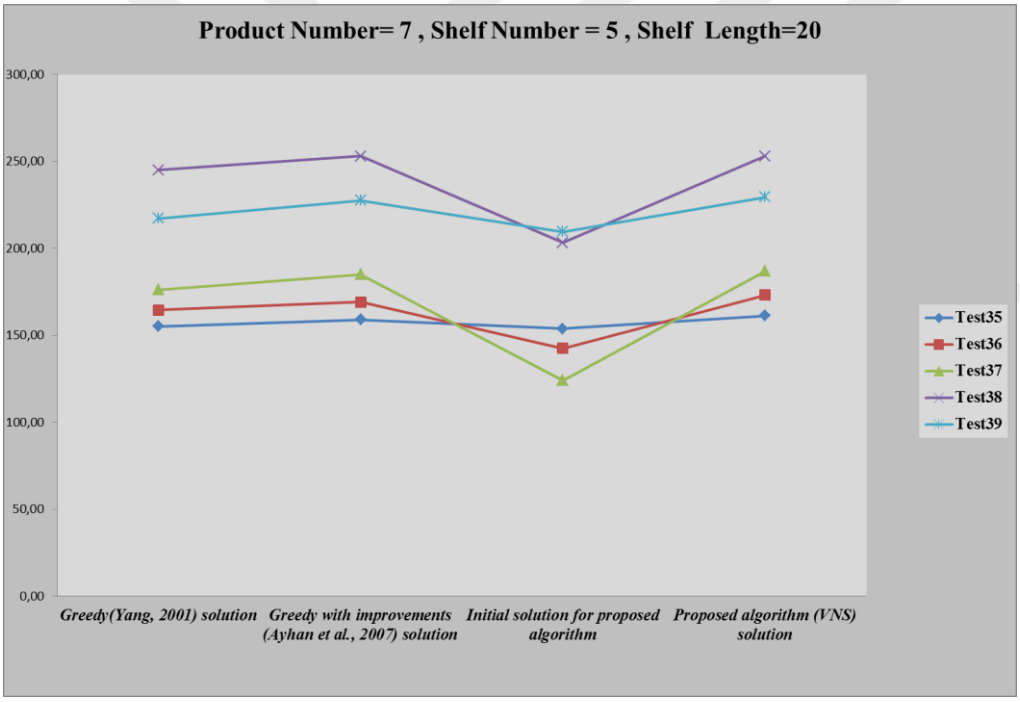

Figure A.8. The results of problem instances (I=7/ K=5/ T =20)

Table A.9. The results (profit) of solution techniques for problem (I=7/ K=5/ T =40)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test40	388.50	392.30	388.50	393.00	7	5	40
Test41	308.20	314.30	283.60	314.56	7	5	40
Test42	285.40	288.60	260.70	288.60	7	5	40
Test43	523.30	536.10	443.10	535.46	7	5	40
Test44	486.00	491.80	438.80	508.84	7	5	40

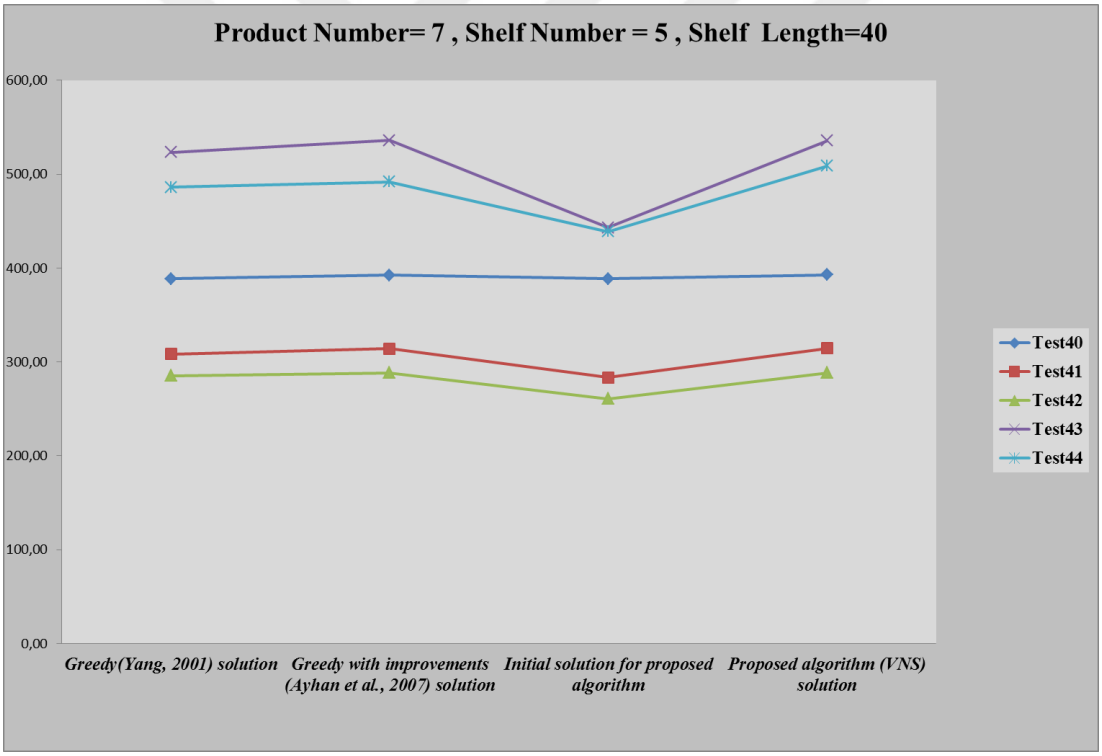

Figure A.9. The results of problem instances (I=7/ K=5/ T =40)

Table A.10. The results (profit) of solution techniques for problem (I=8/ K=3/ T =30)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test45	152.90	162.80	126.10	164.20	8	3	30
Test46	106.00	113.60	111.30	114.60	8	3	30
Test47	191.00	196.90	181.00	198.70	8	3	30
Test48	157.30	164.00	158.50	166.50	8	3	30
Test49	143.00	155.90	140.40	159.06	8	3	30

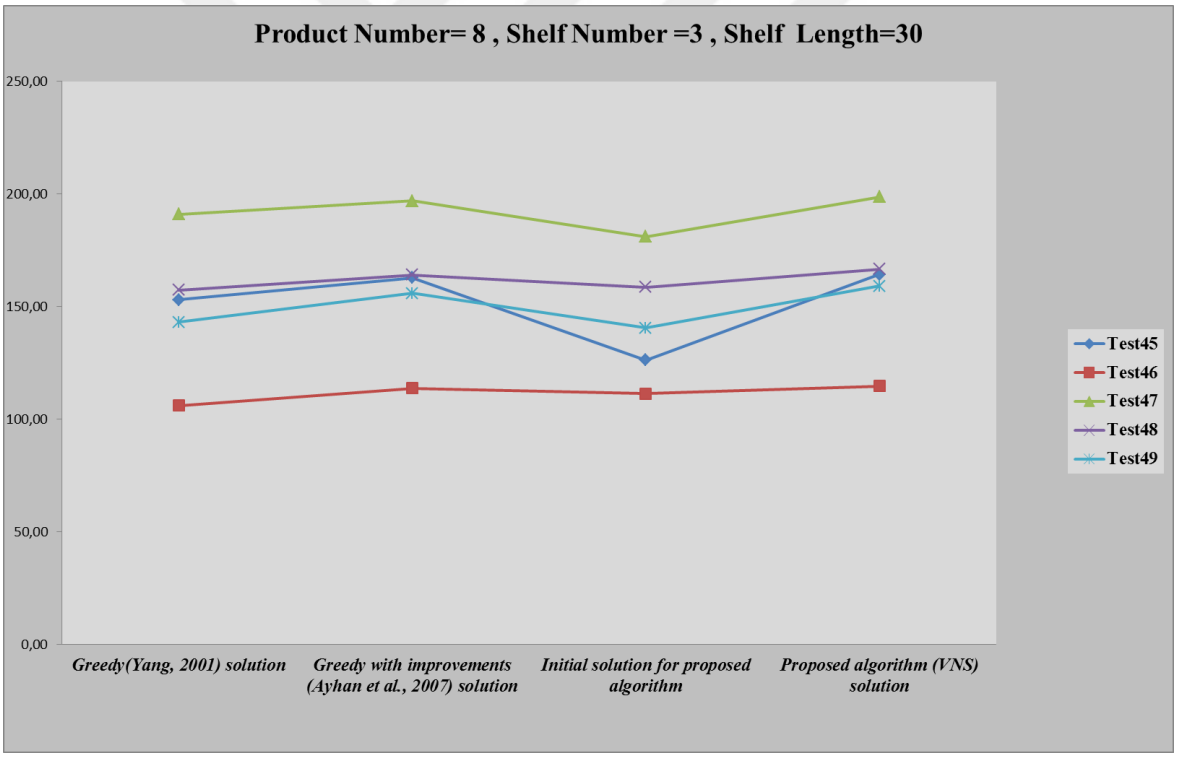

Figure A.10. The results of problem instances (I=8/ K=3/ T =30)

Table A.11. The results (profit) of solution techniques for problem (I=8/ K=3/ T =20)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test50	72.40	79.90	81.30	79.80	8	3	20
Test51	97.00	119.60	86.00	119.80	8	3	20
Test52	94.10	103.40	98.60	104.74	8	3	20
Test53	85.30	92.90	82.60	92.60	8	3	20
Test54	68.60	70.70	67.20	70.70	8	3	20

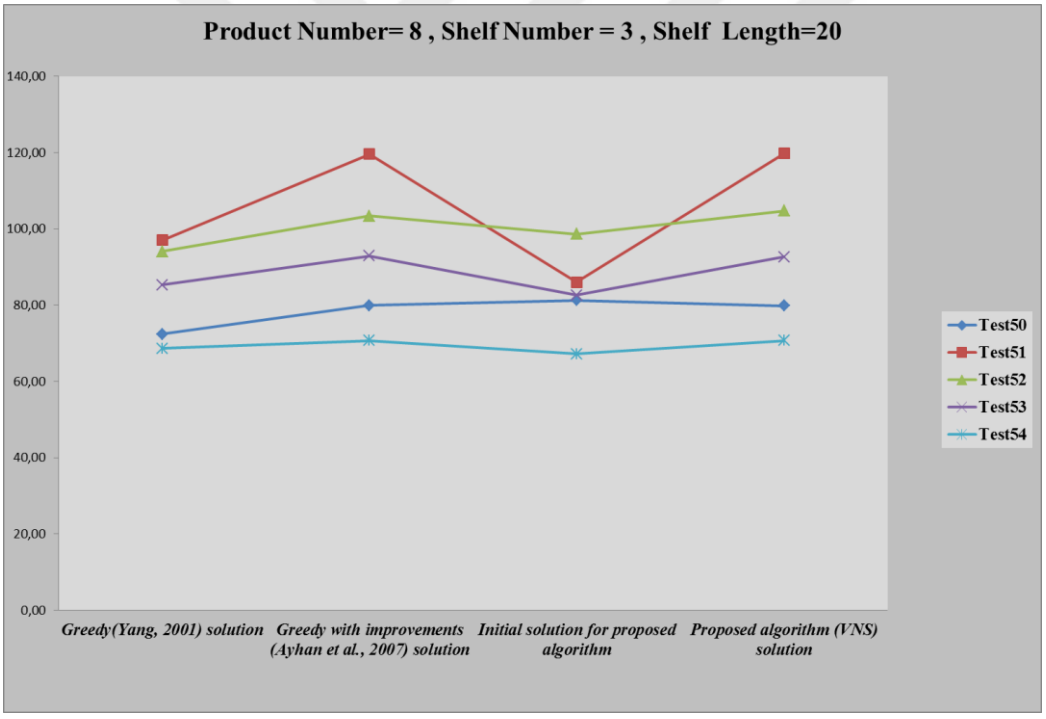

Figure A.11. The results of problem instances (I=8/ K=3/ T =20)

Table A.12. The results (profit) of solution techniques for problem (I=8/ K=3/ T =40)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test55	257.20	263.30	230.20	262.60	8	3	40
Test56	252.80	259.80	222.90	261.50	8	3	40
Test57	253.20	265.70	242.80	269.60	8	3	40
Test58	258.30	275.90	259.40	275.80	8	3	40
Test59	181.60	190.80	134.10	191.90	8	3	40

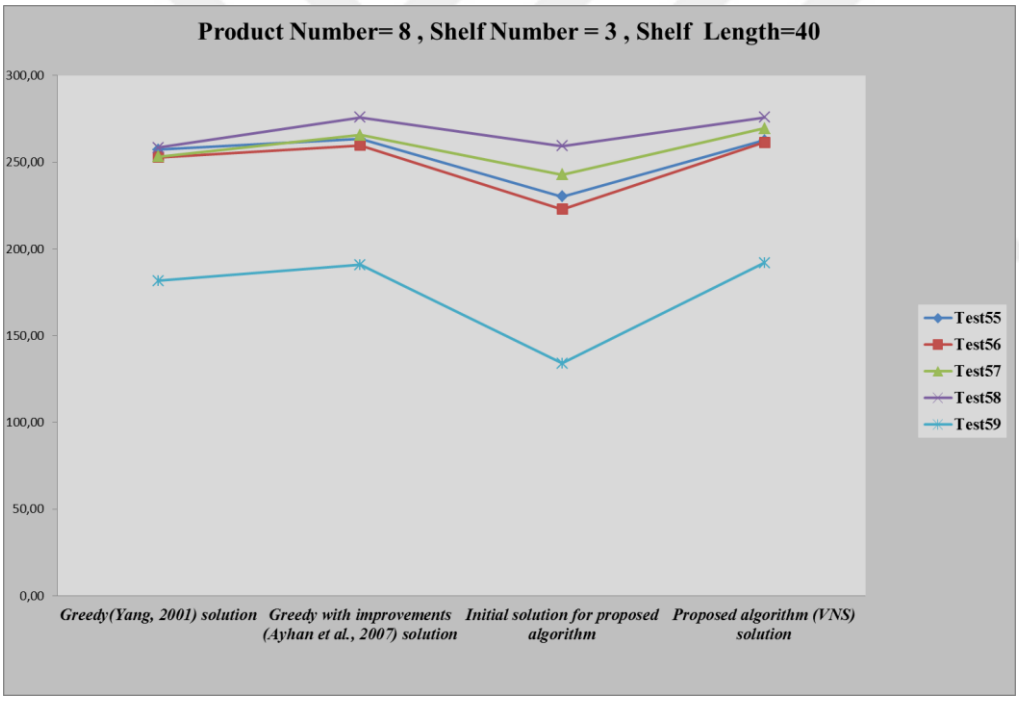

Figure A.12. The results of problem instances (I=8/ K=3/ T =40)

Table A.13. The results (profit) of solution techniques for problem (I=8/ K=4/ T =30)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test60	190.20	193.70	184.30	196.64	8	4	30
Test61	248.20	253.80	232.00	253.80	8	4	30
Test62	171.90	188.30	167.70	189.80	8	4	30
Test63	186.40	197.20	182.30	193.50	8	4	30
Test64	150.00	156.10	150.20	159.90	8	4	30

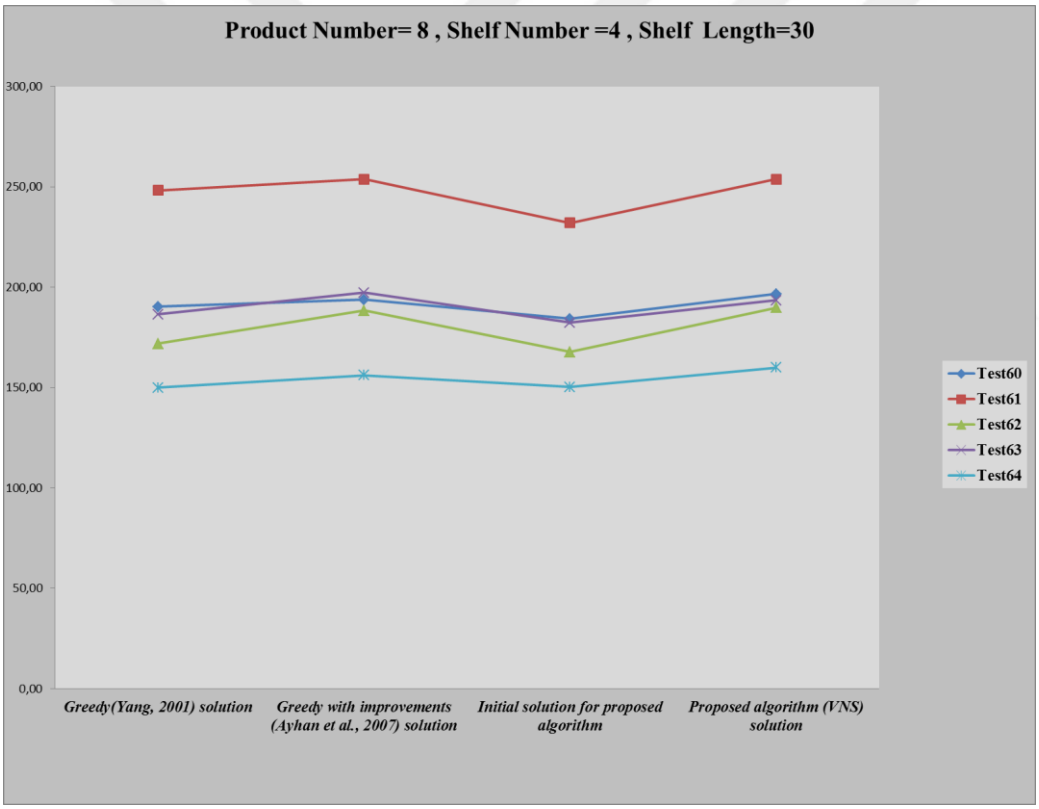

Figure A.13. The results of problem instances (I=8/ K=4/ T =30)

Table A.14. The results (profit) of solution techniques for problem (I=8/ K=4/ T =20)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test65	172.50	182.80	169.60	182.80	8	4	20
Test66	102.00	109.20	82.70	109.00	8	4	20
Test67	105.20	108.40	94.90	109.06	8	4	20
Test68	82.10	95.60	80.50	98.06	8	4	20
Test69	86.90	90.60	81.60	93.24	8	4	20

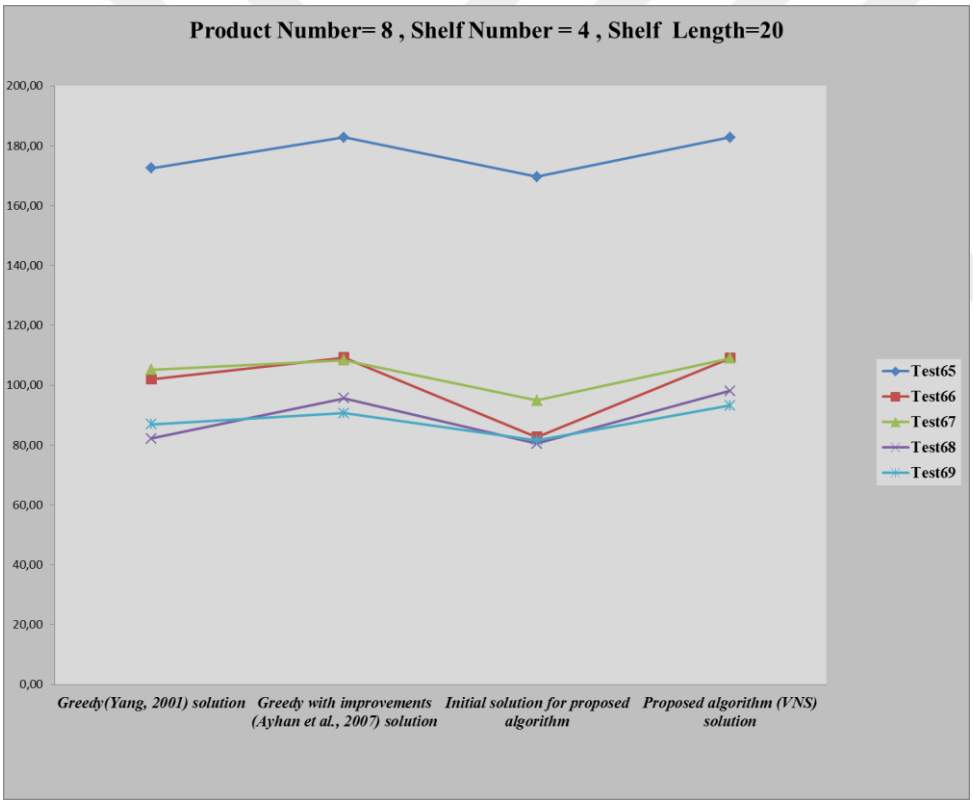

Figure A.14. The results of problem instances (I=8/ K=4/ T =20)

Table A.15. The results (profit) of solution techniques for problem (I=8/ K=4/ T =40)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test70	258.70	266.30	205.00	269.10	8	4	40
Test71	368.70	372.30	347.70	375.70	8	4	40
Test72	346.50	349.00	274.50	350.80	8	4	40
Test73	382.90	390.30	365.70	390.80	8	4	40
Test74	335.00	346.10	289.70	352.02	8	4	40

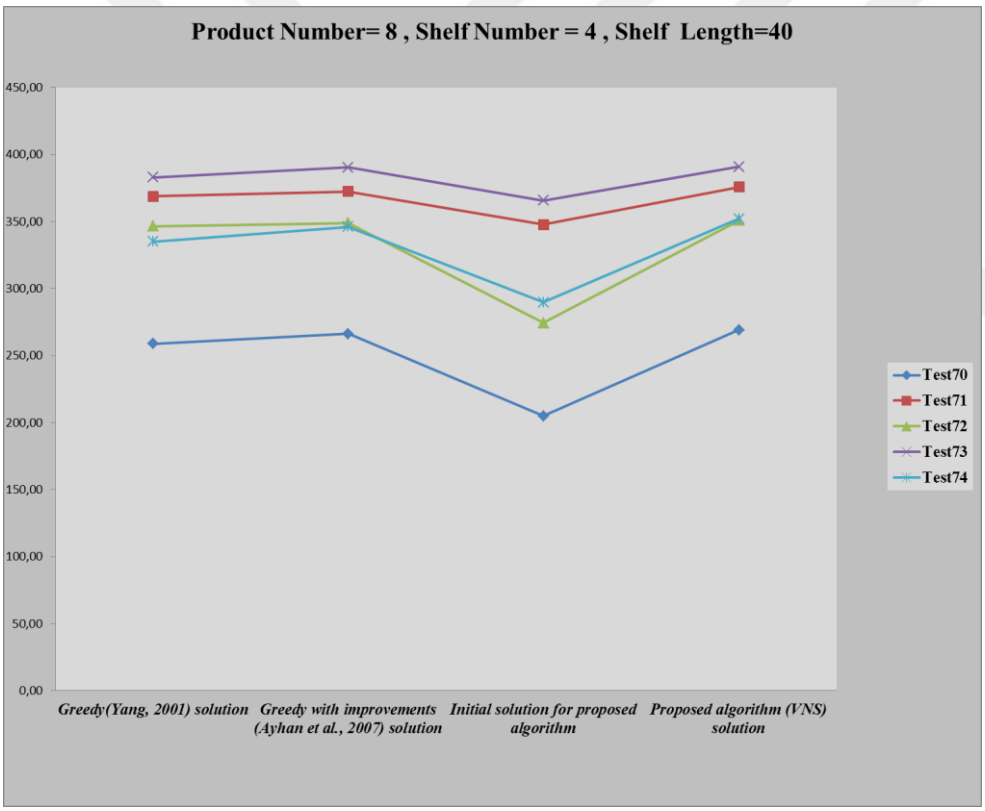

Figure A.15. The results of problem instances (I=8/ K=4/ T =40)

Table A.16. The results (profit) of solution techniques for problem (I=8/ K=5/ T =30)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test75	323.50	329.50	302.90	328.38	8	5	30
Test76	270.60	277.00	249.30	279.70	8	5	30
Test77	321.80	323.00	277.10	323.66	8	5	30
Test78	297.60	314.10	224.30	318.90	8	5	30
Test79	293.20	301.40	243.80	301.60	8	5	30

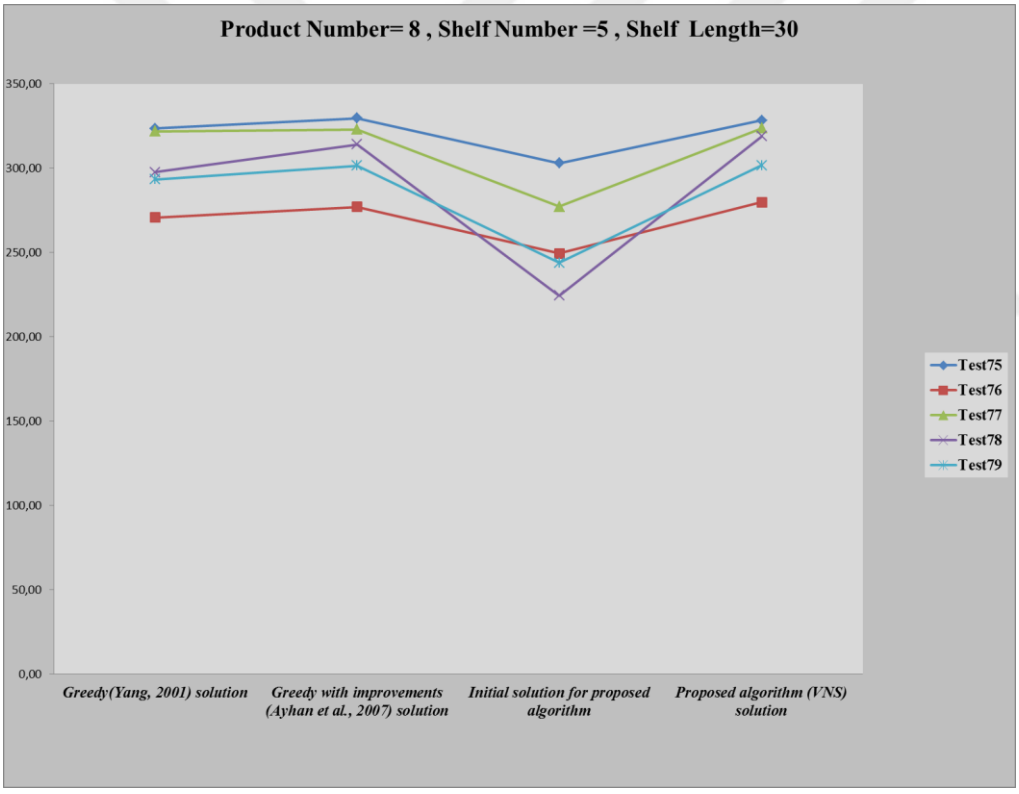

Figure A.16. The results of problem instances (I=8/ K=5/ T =30)

Table A.17. The results (profit) of solution techniques for problem (I=8/ K=5/ T =20)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test80	179.60	184.10	160.90	186.24	8	5	20
Test81	140.00	144.50	129.50	145.50	8	5	20
Test82	181.50	187.70	139.30	188.54	8	5	20
Test83	149.70	159.60	106.90	165.20	8	5	20
Test84	179.00	189.60	146.40	189.76	8	5	20

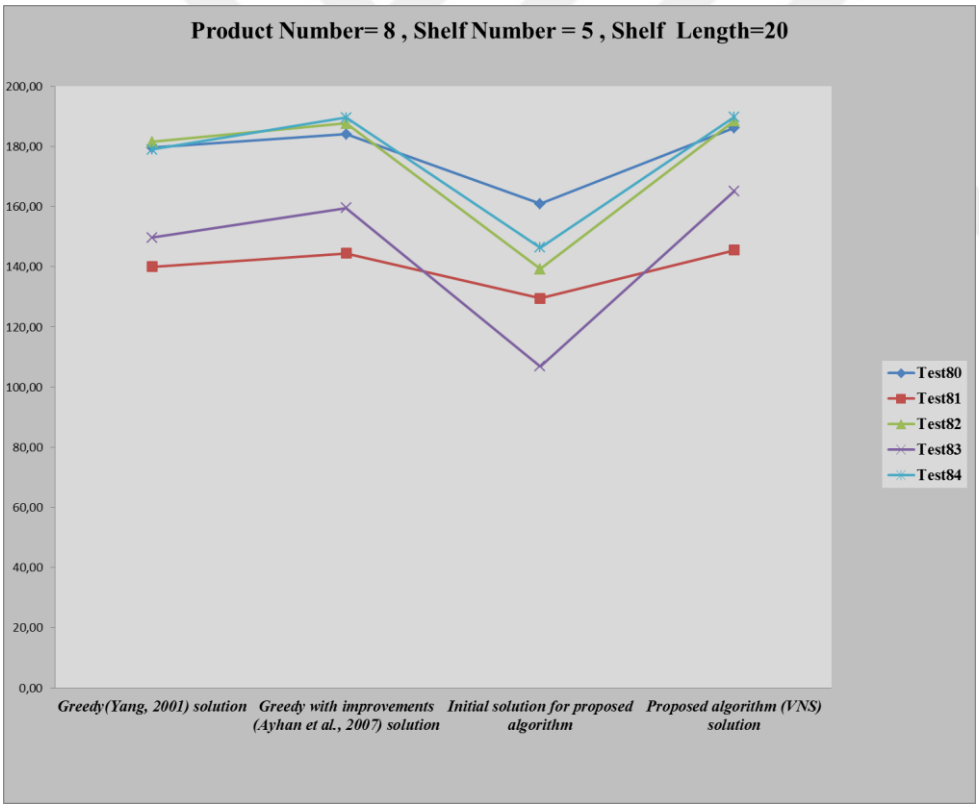

Figure A.17. The results of problem instances (I=8/ K=5/ T =20)

Table A.18. The results (profit) of solution techniques for problem (I=8/ K=5/ T =40)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test85	440.30	442.80	342.00	442.80	8	5	40
Test86	328.40	338.10	233.60	338.90	8	5	40
Test87	274.30	290.00	247.60	295.00	8	5	40
Test88	519.20	524.20	466.40	524.20	8	5	40
Test89	486.60	493.80	360.80	498.80	8	5	40

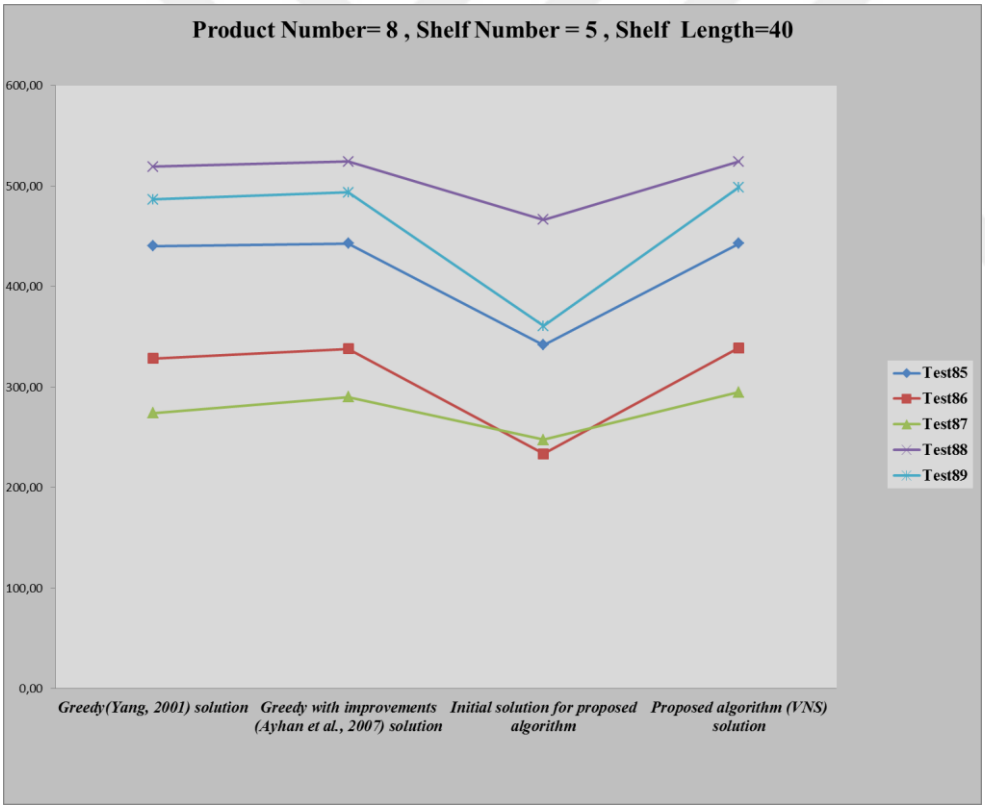

Figure A.18. The results of problem instances (I=8/ K=5/ T =40)

Table A.19. The results (profit) of solution techniques for problem (I=9/ K=3/ T =30)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test90	161.90	170.60	144.00	171.00	9	3	30
Test91	107.90	119.30	107.90	119.30	9	3	30
Test92	118.30	124.50	105.40	127.50	9	3	30
Test93	82.30	90.00	80.60	93.00	9	3	30
Test94	100.70	108.20	88.50	109.50	9	3	30

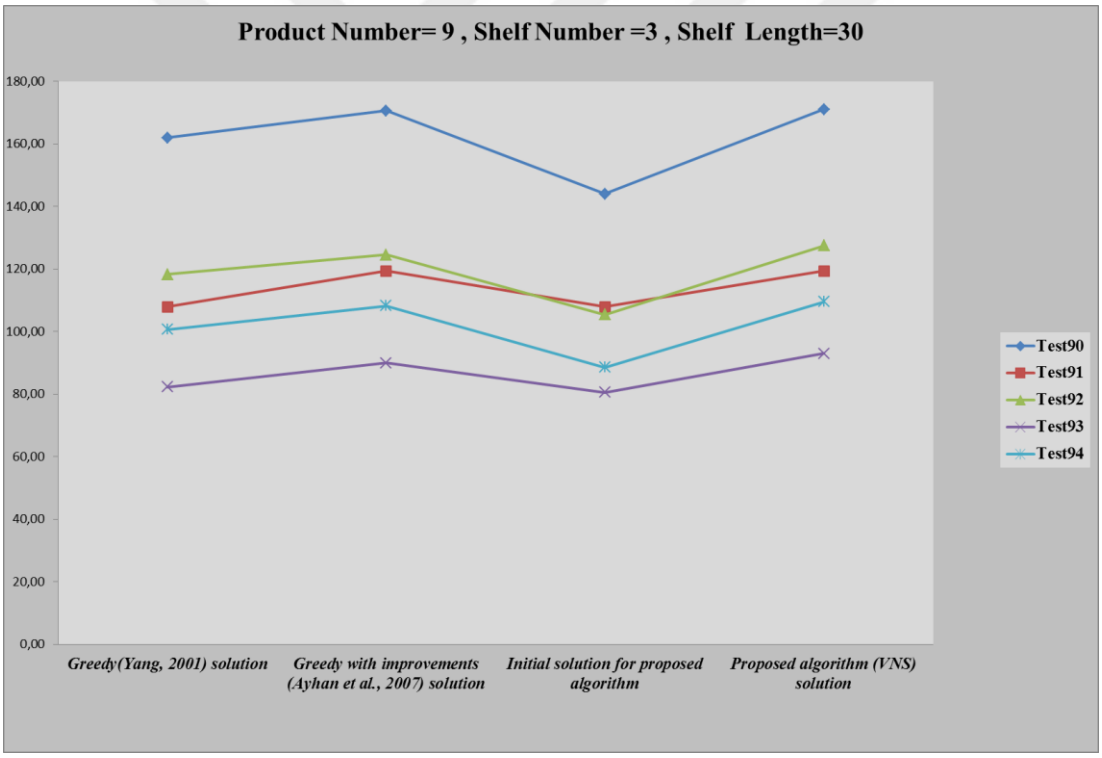

Figure A.19. The results of problem instances (I=9/ K=3/ T =30)

Table A.20. The results (profit) of solution techniques for problem (I=9/ K=3/ T =20)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test95	67.30	75.30	64.00	74.16	9	3	20
Test96	64.50	69.50	51.50	67.54	9	3	20
Test97	70.20	86.40	69.10	84.60	9	3	20
Test98	107.60	115.00	85.20	115.16	9	3	20
Test99	103.60	104.60	82.40	104.60	9	3	20

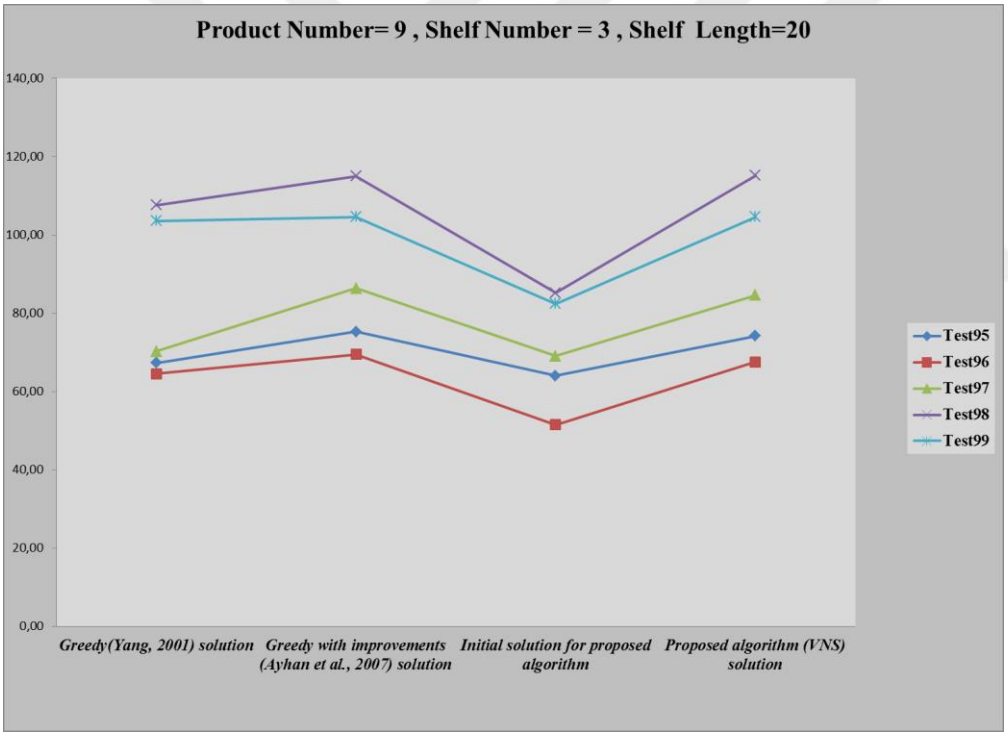

Figure A.20. The results of problem instances (I=9/ K=3/ T =20)

Table A.21. The results (profit) of solution techniques for problem (I=9/ K=3/ T =40)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test100	208.00	218.90	153.00	224.10	9	3	40
Test101	295.20	306.70	285.00	308.80	9	3	40
Test102	341.90	350.50	343.80	353.30	9	3	40
Test103	186.30	195.80	156.30	197.70	9	3	40
Test104	204.60	220.50	215.30	216.20	9	3	40

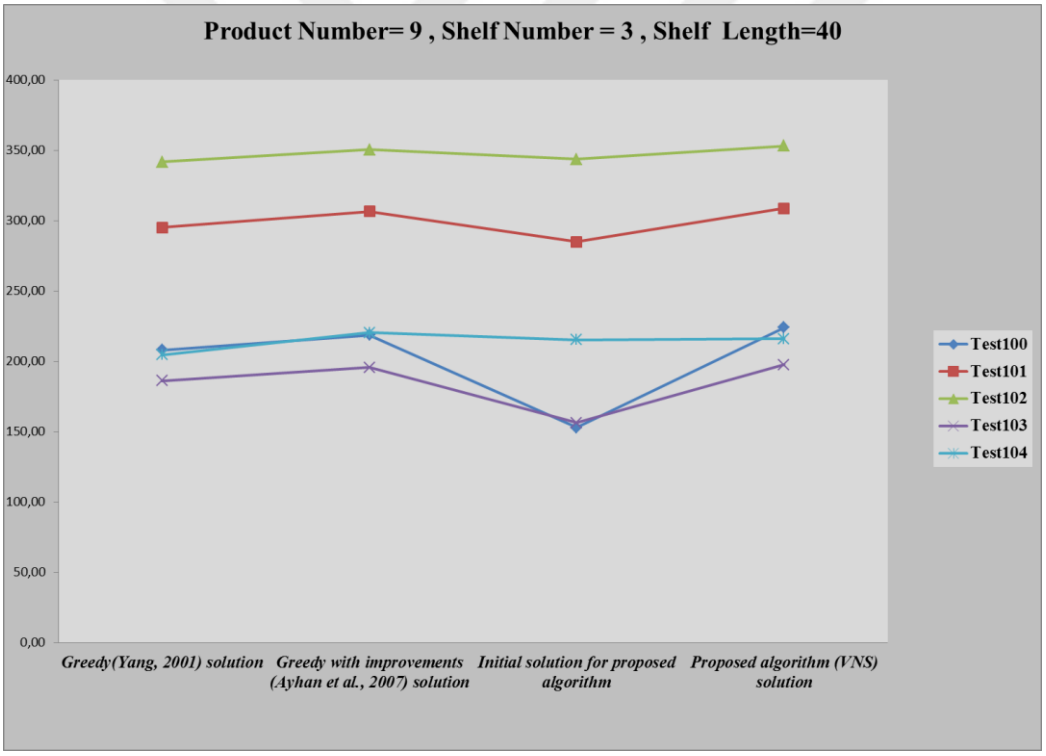

Figure A.21. The results of problem instances (I=9/ K=3/ T =40)

Table A.22. The results (profit) of solution techniques for problem (I=9/ K=4/ T =30)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test105	237.00	242.10	168.10	242.00	9	4	30
Test106	239.80	244.60	195.40	250.60	9	4	30
Test107	225.90	227.70	151.00	227.70	9	4	30
Test108	163.80	170.80	117.20	174.70	9	4	30
Test109	218.70	231.50	179.70	238.62	9	4	30

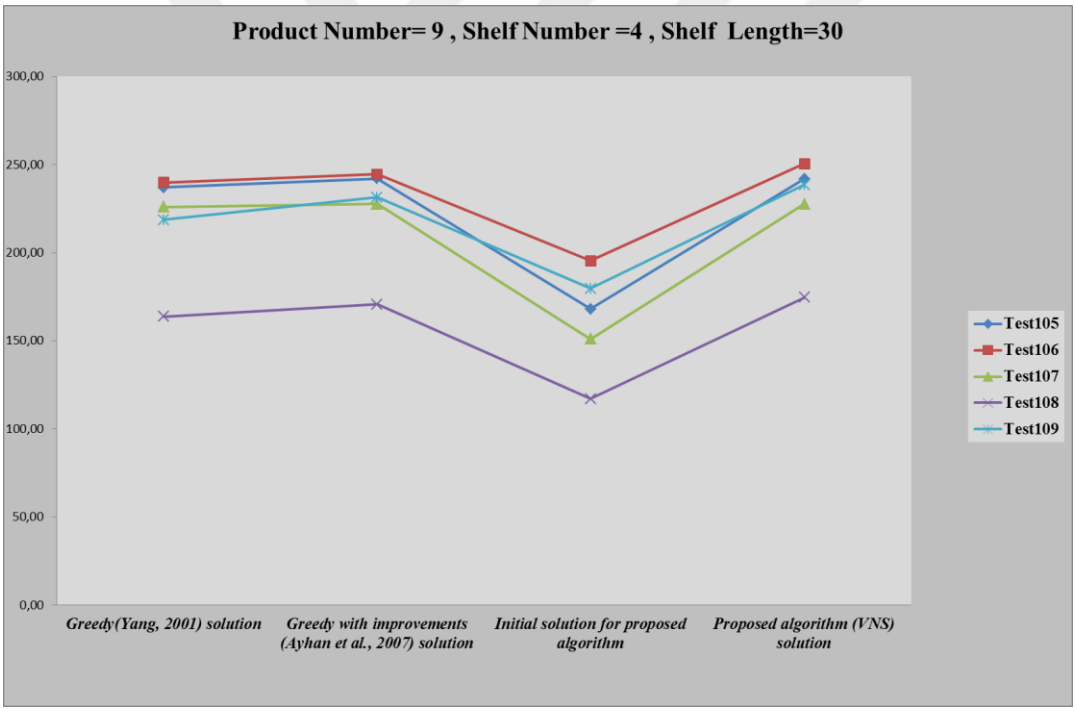

Figure A.22. The results of problem instances (I=9/ K=4/ T =30)

Table A.23. The results (profit) of solution techniques for problem (I=9/ K=4/ T =20)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test110	160.10	169.00	152.90	166.50	9	4	20
Test111	131.30	142.20	115.80	141.04	9	4	20
Test112	134.00	141.00	125.60	145.36	9	4	20
Test113	111.20	119.20	95.70	119.20	9	4	20
Test114	156.50	161.40	149.20	162.80	9	4	20

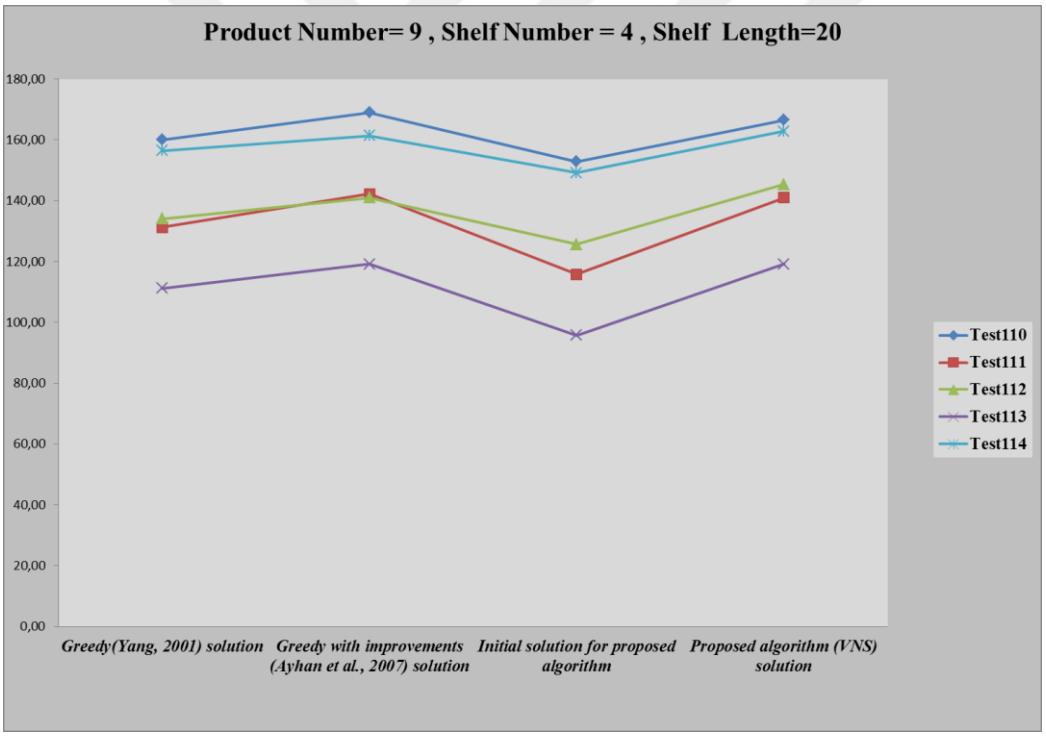

Figure A.23. The results of problem instances (I=9/ K=4/ T =20)

Table A.24. The results (profit) of solution techniques for problem (I=9/ K=4/ T =40)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test115	215.30	227.30	205.90	230.38	9	4	40
Test116	350.00	353.50	290.30	360.78	9	4	40
Test117	277.80	285.80	208.00	284.40	9	4	40
Test118	234.00	237.50	232.60	240.90	9	4	40
Test119	254.00	265.00	234.60	265.70	9	4	40

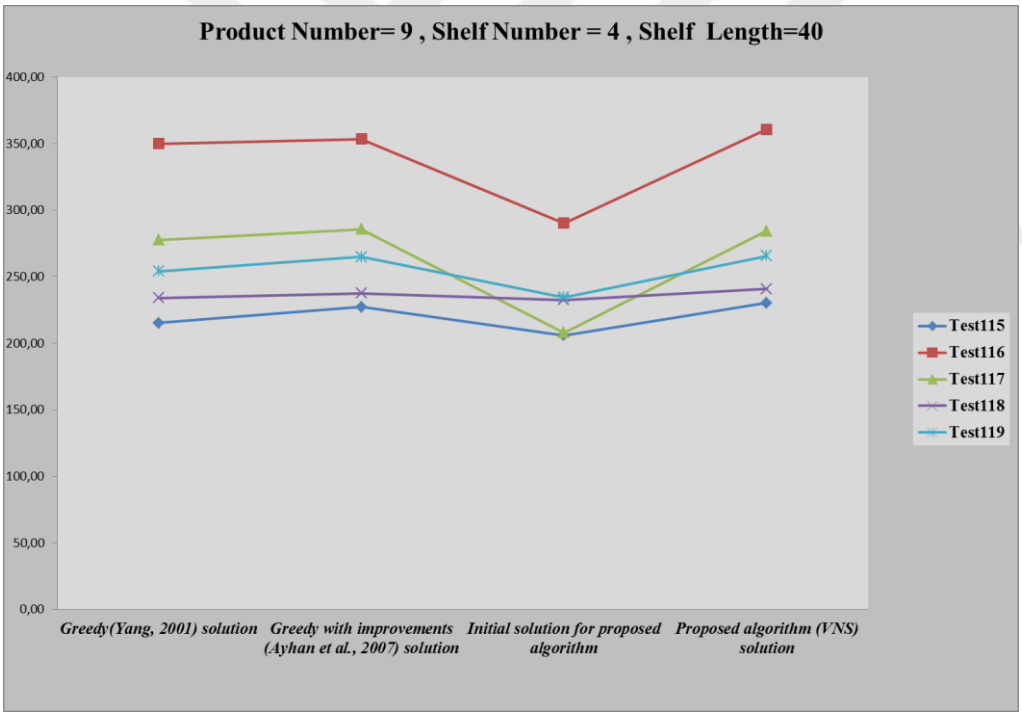

Figure A.24. The results of problem instances (I=9/ K=4/ T =40)

Table A.25. The results (profit) of solution techniques for problem (I=9/ K=5/ T =30)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test120	256.70	262.50	204.90	262.50	9	5	30
Test121	371.40	383.30	324.40	383.30	9	5	30
Test122	330.10	341.80	254.10	343.30	9	5	30
Test123	289.00	296.40	251.80	299.30	9	5	30
Test124	299.20	305.60	255.80	305.70	9	5	30

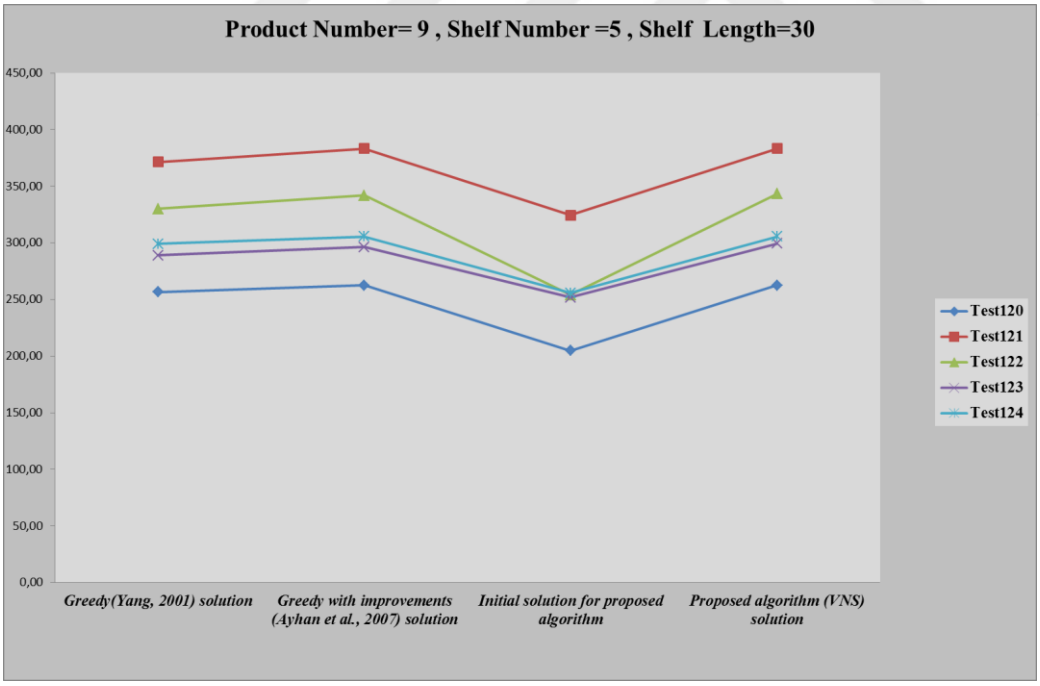

Figure A.25. The results of problem instances (I=9/ K=5/ T =30)

Table A.26. The results (profit) of solution techniques for problem (I=9/ K=5/ T =20)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test125	117.90	121.80	113.80	124.70	9	5	20
Test126	244.70	257.60	208.80	257.60	9	5	20
Test127	256.80	265.50	231.70	261.90	9	5	20
Test128	147.00	163.80	155.20	163.80	9	5	20
Test129	212.80	222.70	194.80	224.82	9	5	20

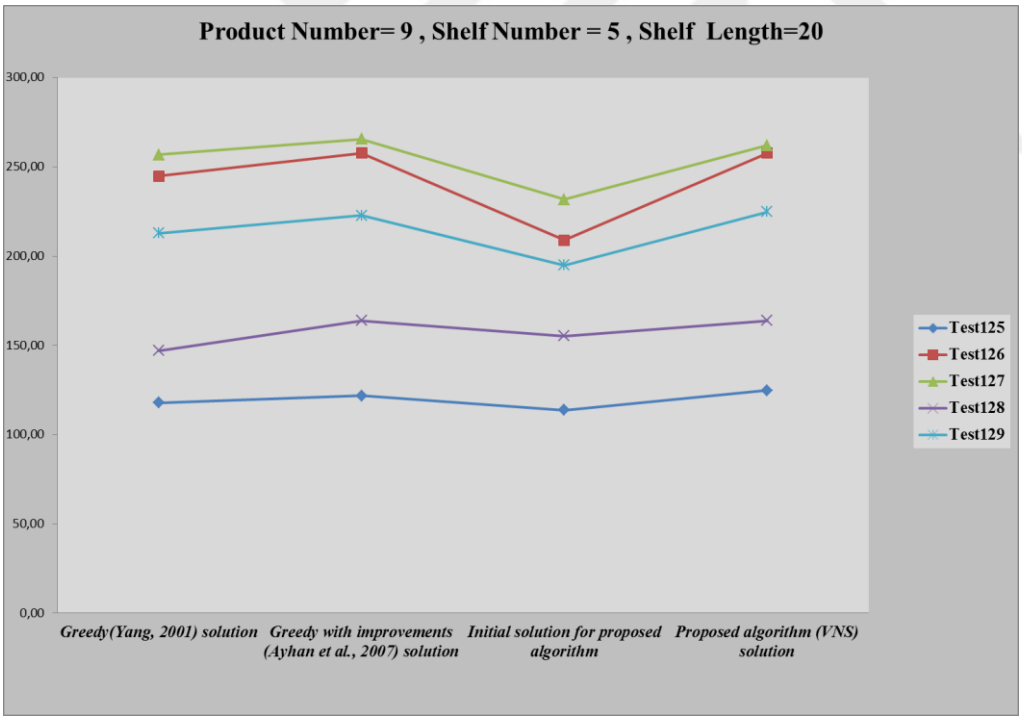

Figure A.26. The results of problem instances (I=9/ K=5/ T =20)

Table A.27. The results (profit) of solution techniques for problem (I=9/ K=5/ T =40)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test130	461.90	471.70	388.80	471.22	9	5	40
Test131	395.10	402.20	351.60	403.00	9	5	40
Test132	472.50	480.60	459.50	480.60	9	5	40
Test133	395.50	408.00	395.50	412.30	9	5	40
Test134	287.70	295.30	258.40	295.30	9	5	40

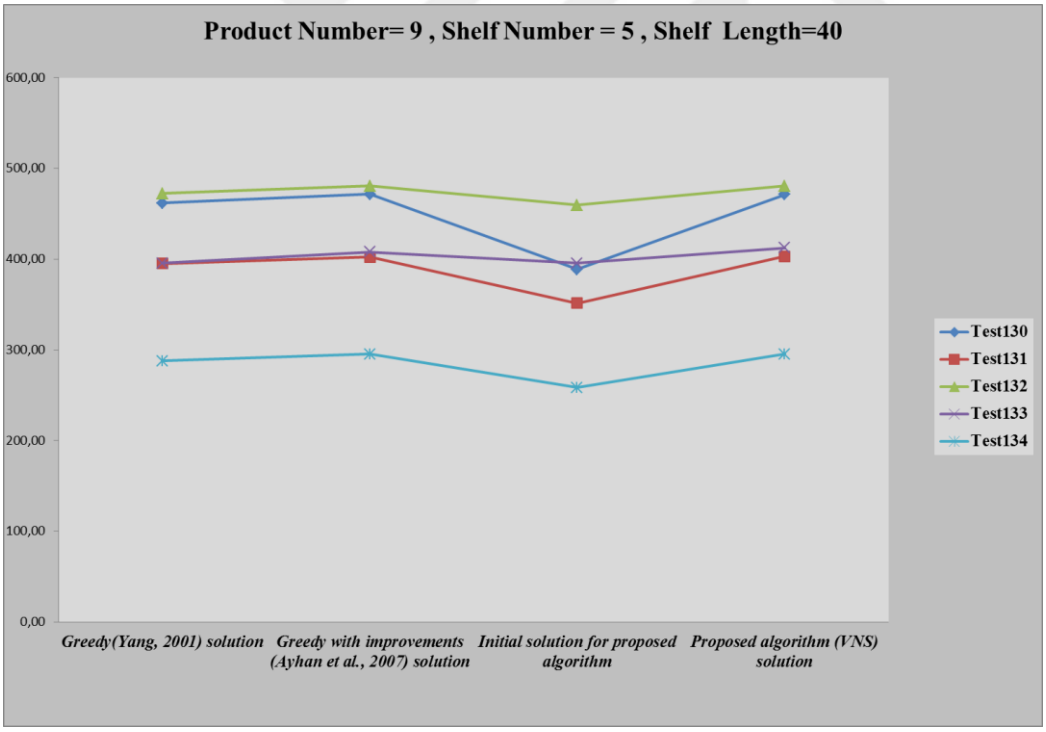

Figure A.27. The results of problem instances (I=9/ K=5/ T =40)

Table A.28. The results (profit) of solution techniques for problem (I=10/ K=3/ T =30)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test135	159.00	167.20	147.50	168.46	10	3	30
Test136	113.00	129.10	94.70	128.00	10	3	30
Test137	118.10	128.10	89.20	129.10	10	3	30
Test138	118.50	129.10	92.60	133.30	10	3	30
Test139	140.40	148.60	127.20	148.82	10	3	30

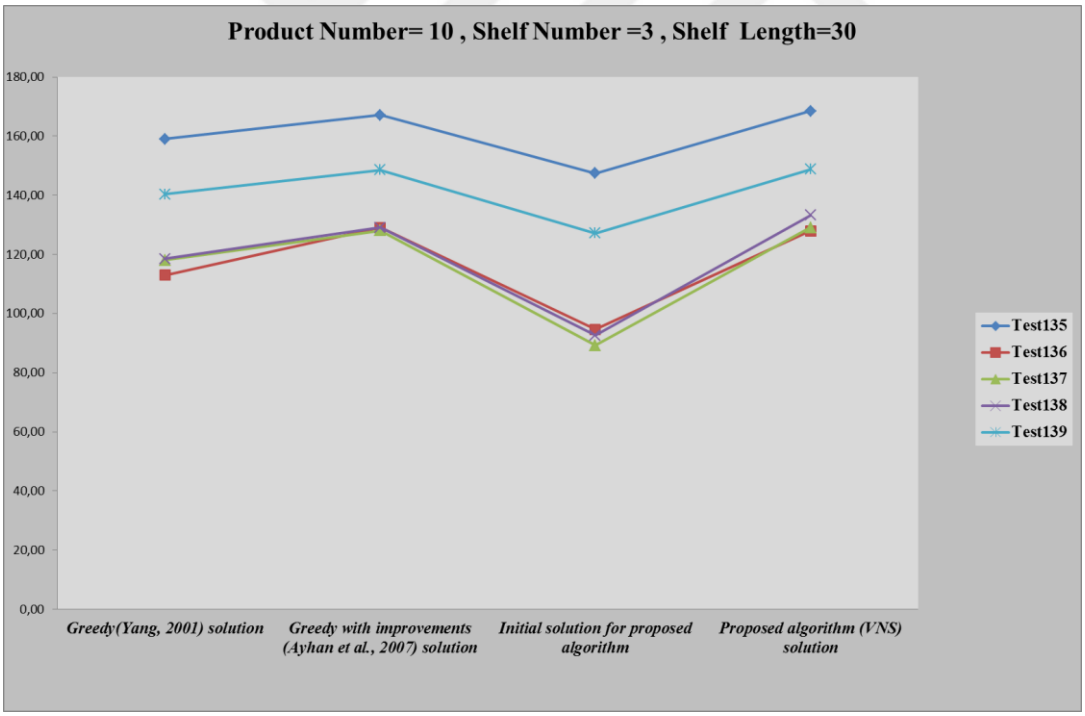

Figure A.28. The results of problem instances (I=10/ K=3/ T =30)

Table A.29. The results (profit) of solution techniques for problem (I=10/ K=3/ T =20)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test140	78.60	82.90	63.40	84.40	10	3	20
Test141	62.90	72.20	64.80	72.66	10	3	20
Test142	98.30	104.90	80.20	105.46	10	3	20
Test143	68.30	79.80	71.10	79.38	10	3	20
Test144	62.70	79.90	72.50	87.94	10	3	20

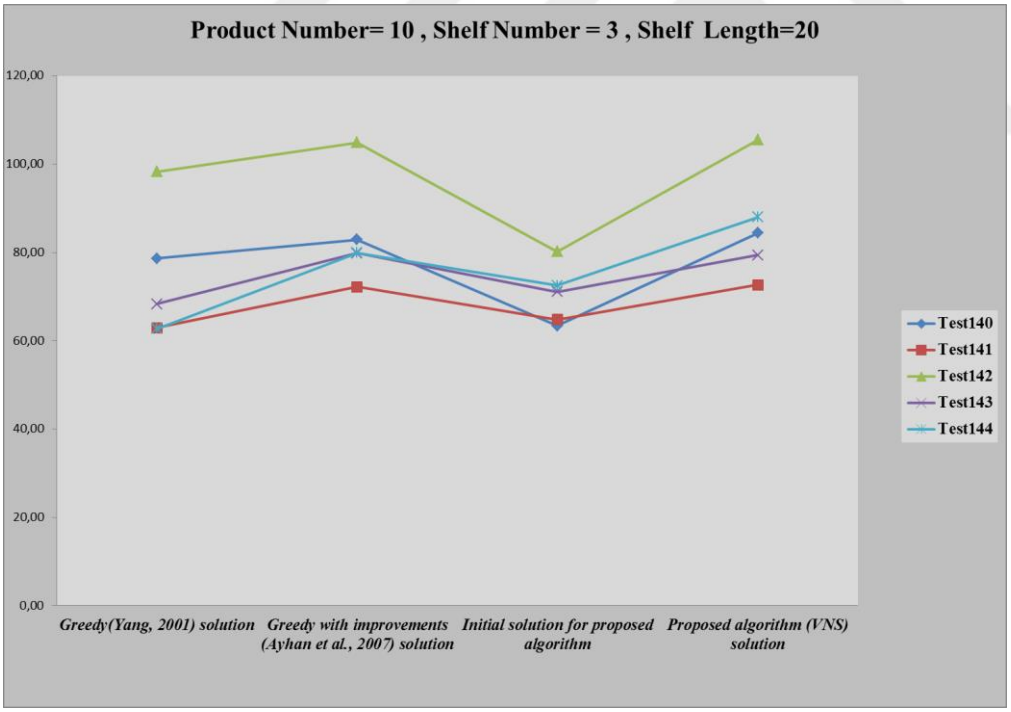

Figure A.29. The results of problem instances (I=10/ K=3/ T =20)

Table A.30. The results (profit) of solution techniques for problem (I=10/ K=3/ T =40)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test145	157.70	165.20	159.60	168.80	10	3	40
Test146	228.70	240.70	216.90	242.50	10	3	40
Test147	182.40	193.90	174.60	198.00	10	3	40
Test148	228.00	241.40	229.20	241.60	10	3	40
Test149	215.90	222.70	160.50	226.74	10	3	40

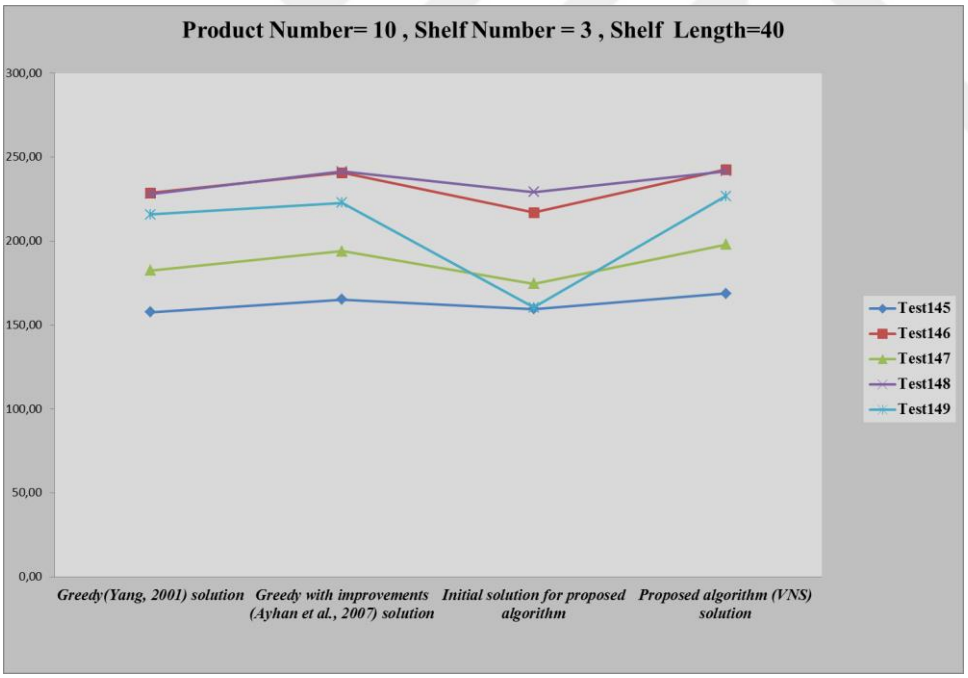

Figure A.30. The results of problem instances (I=10/ K=3/ T =40)

Table A.31. The results (profit) of solution techniques for problem (I=10/ K=4/ T =30)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test150	247.60	256.20	252.60	258.50	10	4	30
Test151	196.90	205.70	175.80	205.48	10	4	30
Test152	265.90	283.10	215.60	283.10	10	4	30
Test153	179.70	190.10	154.40	191.46	10	4	30
Test154	192.10	201.40	190.00	204.34	10	4	30

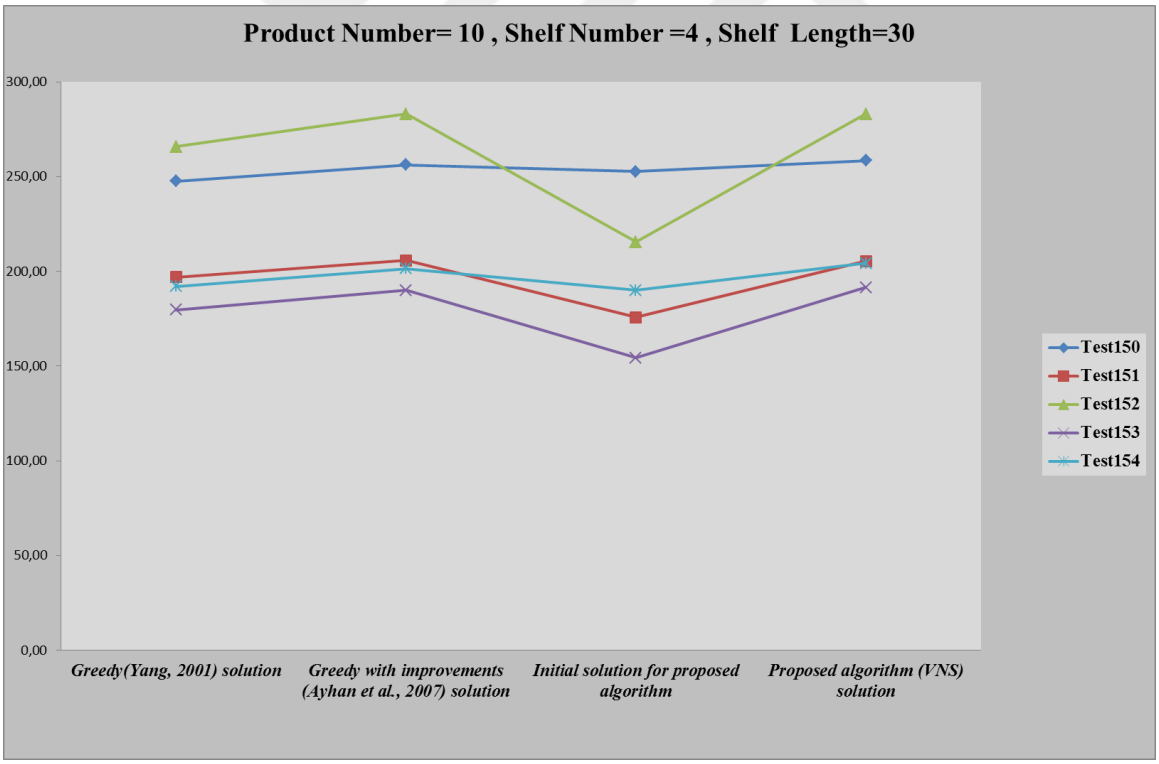

Figure A.31. The results of problem instances (I=10/ K=4/ T =30)

Table A.32. The results (profit) of solution techniques for problem (I=10/ K=4/ T =20)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test155	136.60	138.30	101.00	138.30	10	4	20
Test156	143.00	146.10	114.40	146.30	10	4	20
Test157	141.10	152.80	120.10	152.74	10	4	20
Test158	85.20	98.80	84.00	100.36	10	4	20
Test159	109.40	116.70	116.80	120.50	10	4	20

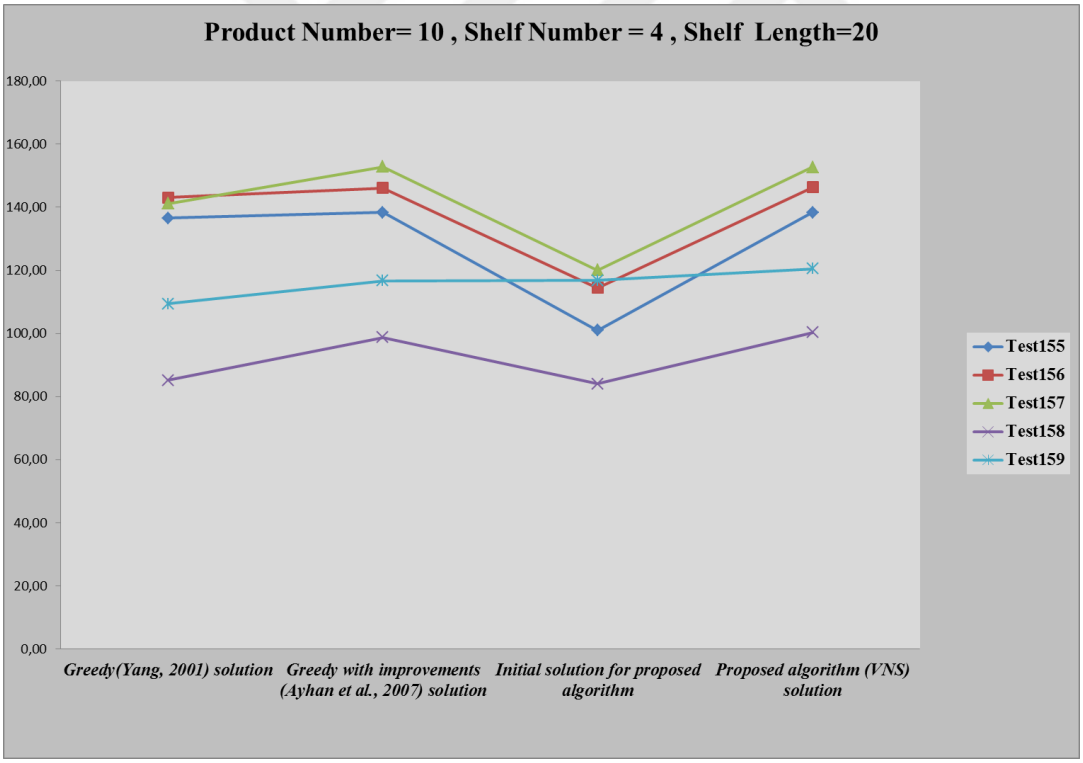

Figure A.32. The results of problem instances (I=10/ K=4/ T =20)

Table A.33. The results (profit) of solution techniques for problem (I=10/ K=4/ T =40)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test160	370.10	379.60	293.00	379.84	10	4	40
Test161	328.30	346.70	223.70	346.70	10	4	40
Test162	297.10	301.90	245.40	306.10	10	4	40
Test163	393.60	406.30	347.90	407.00	10	4	40
Test164	316.00	320.70	311.90	320.78	10	4	40

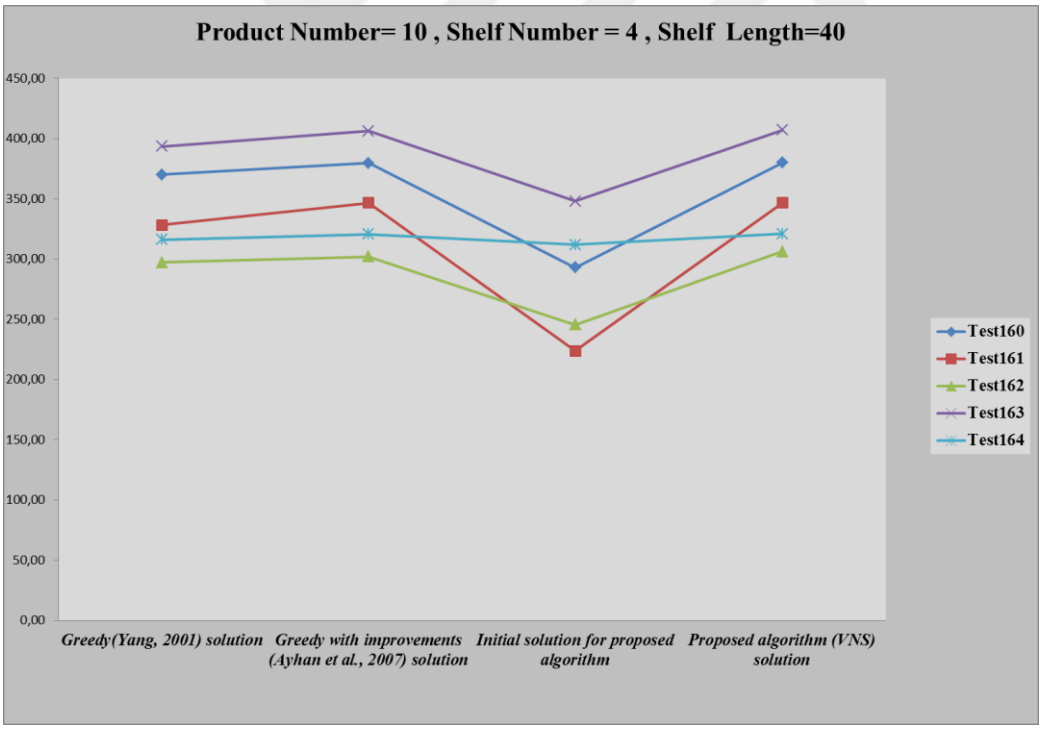

Figure A.33. The results of problem instances (I=10/ K=4/ T =40)

Table A.34. The results (profit) of solution techniques for problem (I=10/ K=5/ T =30)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test165	180.20	184.30	176.90	185.90	10	5	30
Test166	281.80	284.90	244.10	285.20	10	5	30
Test167	202.40	210.00	201.20	219.10	10	5	30
Test168	242.10	250.80	235.50	252.66	10	5	30
Test169	229.70	236.40	206.20	239.32	10	5	30

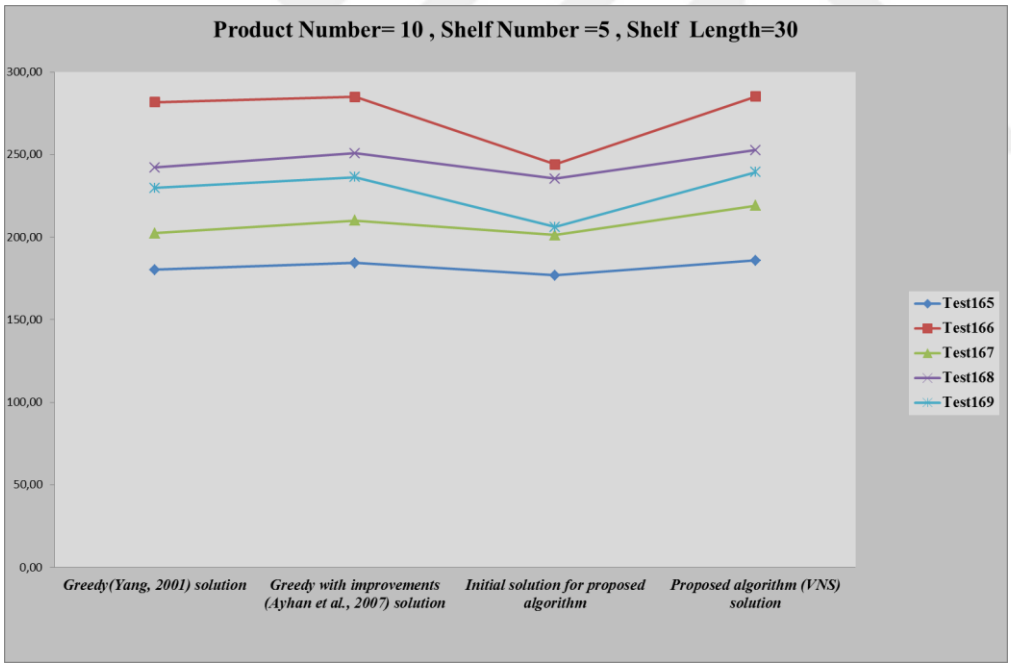

Figure A.34. The results of problem instances (I=10/ K=5/ T =30)

Table A.35. The results (profit) of solution techniques for problem (I=10/ K=5/ T =20)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test170	162.10	188.80	141.10	185.50	10	5	20
Test171	120.00	126.00	117.90	126.20	10	5	20
Test172	206.80	211.70	179.00	214.28	10	5	20
Test173	182.50	186.00	166.80	187.06	10	5	20
Test174	198.20	216.80	191.90	218.00	10	5	20

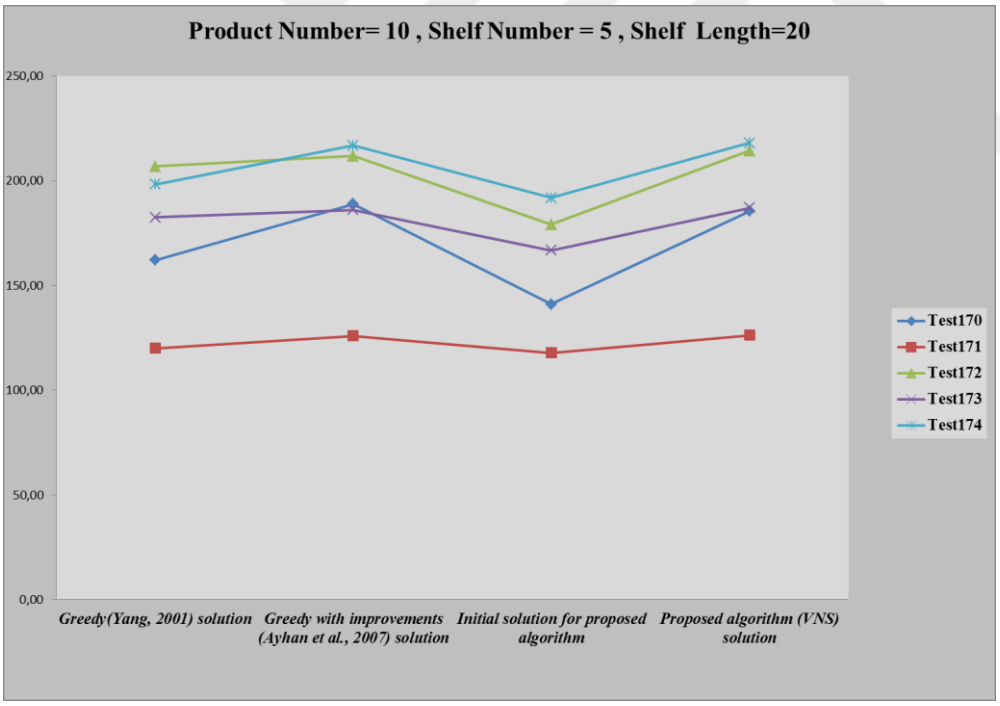

Figure A.35. The results of problem instances (I=10/ K=5/ T =20)

Table A.36. The results (profit) of solution techniques for problem (I=10/ K=5/ T =40)

Test Problems	Greedy solution (Yang, 2001)	Greedy with improvements solution (Ayhan, et al., 2007)	Initial solution	VNS	I	K	T
Test175	448.00	458.80	372.60	460.10	10	5	40
Test176	470.00	484.20	309.80	482.56	10	5	40
Test177	350.10	361.80	289.40	365.50	10	5	40
Test178	394.80	409.70	275.20	409.70	10	5	40
Test179	481.90	487.00	481.90	487.00	10	5	40

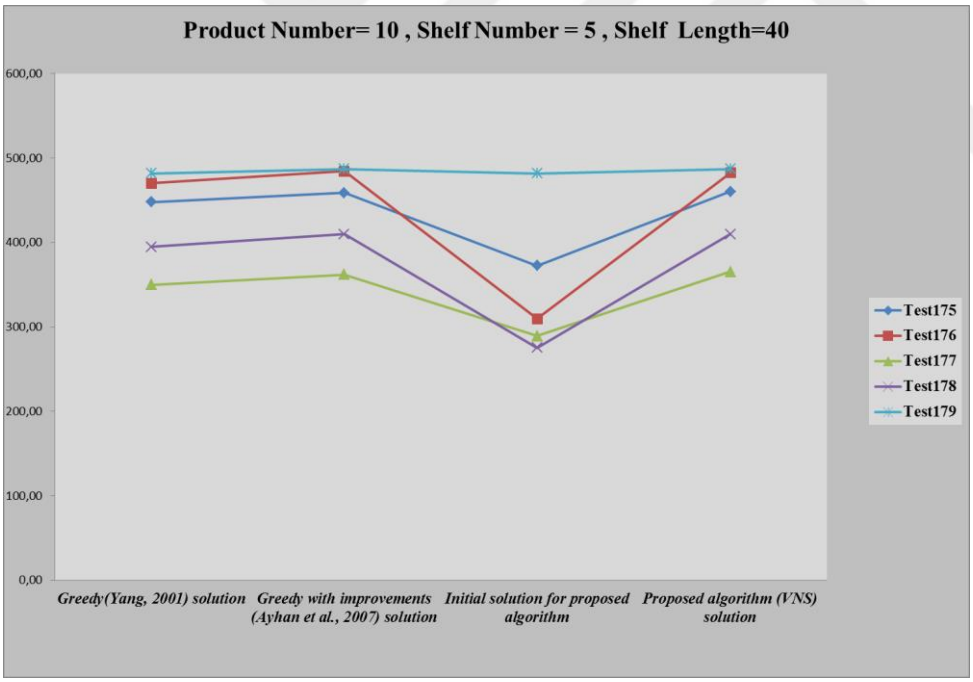

Figure A.36. The results of problem instances (I=10/ K=5/ T =40)

ÖZGEÇMİŞ

Ayşe Hande EROL BİNGÜLER, 1983 yılında doğdu. İstanbul Ticaret Üniversitesi, Fen-Edebiyat Fakültesi İstatistik Bölümü'nü 2006 yılında, Mühendislik ve Tasarım Fakültesi Endüstri Mühendisliği (Çift Anadal) Bölümü'nü 2007 yılında, tam burs olarak, okul birinciliği ile bitirdi. Marmara Üniversitesi, Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü Tezli Yüksek Lisans Programı'nı 2010 yılında bitirdi. Ara vermeksizin başladığı Marmara Üniversitesi, Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü Tezli Doktora Programı'na halen devam etmektedir. Yüksek lisans ve doktora eğitimi boyunca Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TÜBİTAK)'ndan burs almıştır.

ISO 9001 Kalite Yönetim Sistemi, ISO 27001 Bilgi Güvenliği Yönetim Sistemi Baştetkikçi, vb. uzmanlık alanında birçok sertifikaya sahiptir.

2012 yılından itibaren önce Marmara Üniversitesi Rektörlüğü'nde Kalite Koordinatörlüğü'nde göreve başlamış, 2014 yılından itibaren Marmara Üniversitesi Rektörlüğü Bilgi İşlem Daire Başkanlığı'nda akademik personel olarak çalışmaktadır.

Bilimsel çalışmalarını ulusal ve uluslararası platformlarda makaleler, bildiriler ve sunumlar yaparak değerlendirmektedir. Sağlık, eğitim ve mühendislik alanlarında mesleki araştırmalar yapmaktadır. İlgi alanları arasında yöneylem araştırması, matematik, istatistik, kalite kontrolü ve yönetimi, bulanık mantık, bulanık stok yönetimi, sezgisel yaklaşımlar, genetik algoritma, benzetimsel tavlama yaklaşımları, eniyileme, simülasyon, bilgi paylaşım platformları istatistiksel analizleri, tedarik zinciri yönetimi, uygulamalı yöneylem araştırması yer almaktadır. Akademik kariyerinde, şimdiye kadar yapılan bildiri ve makaleler, sezgisel yaklaşımlar geliştirme, üretim ve yönetim alanlarındaki yeni yaklaşımlar üzerinedir.

İleride yapılacak çalışmalar ülkemiz ve kurumumuz için verimli, kaliteli ve etkin çözüm yaklaşımları ile arasında üretim ve yönetim alanında yeni sezgisel yaklaşımlar geliştirme ve bunları istatistiksel analizlerle destekleme yer almaktadır.

THIS PAGE INTENTIONALLY LEFT BLANK

BIOGRAPHY

Ayşe Hande EROL BİNGÜLER was born on Sept 20, 1983 in Istanbul. After graduating from Istanbul Commerce University Statistics Department and Industrial Engineering Department with double major in 2006 and 2007, she received her master degree in Industrial Engineering from Marmara University in 2010. She also joined the same school for PhD degree in Industrial Engineering Department. She has a foundation through her life of education. For her success, she has financial support from The Scientific and Technological Research Council of Turkey (TÜBITAK). She has many certifications about Quality Management, Information Security Management System, etc.

She is working in Marmara University since 2012, and she is still working as a Process Management Expert and she is currently working toward the PhD degree in Industrial Engineering, Marmara University, Turkey. She is also an engineer, a statistician and expert at the Marmara University Information Technology Department. Her current research interests include evolutionary algorithms, operations research, statistical analysis, greedy algorithms, metaheuristics, and applied operations research.

She would like to develop new ideas and projects, in order to contribute for her country and her organization she works.

THIS PAGE INTENTIONALLY LEFT BLANK